Linköping Studies in Science and Technology. Dissertations
No. 424

Automatic Inspection of Sawn Wood

Erik Åstrand

Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden

Linköping 1996

Erik Åstrand: **Automatic Inspection of Sawn Wood** Linköping 1996

Linköping Studies in Science and Technology. Dissertations.
No. 424

Automatic Inspection of Sawn Wood

Erik Åstrand

Akademisk avhandling

som för avläggande av teknisk doktorsexamen vid Tekniska Högskolan i Linköping kommer att offentligen försvaras i sal C4, Linköpings Universitet, fredagen den 12 april 1996, kl. 10.15.

Abstract

In the production chain from the tree to the final wood product, the wood material will be subject to various forms of inspection and grading in a number of steps. The subject of this thesis is automatic inspection of sawn wood. The primary target application is inspection for cross cutting in the secondary wood industry, although the results are applicable also to other inspection situations such as sawmill grading.

More specifically, there are two main problems addressed in this thesis. The first is the automatic detection of defects, such as knots, splits, wane edges etc., on sawn wood surfaces. The other is the optimization process when cutting the boards into products based on the defect information resulting from the inspection.

The goals of this thesis are to develop new image processing methods and algorithms for wood defect detection and then to use the defect information to optimize the utilization of the material. The industrial constraints of production speed and cost of the equipment are fully considered, which means that the proposed solutions can directly serve as a basis for developing commercially competitive products.

The thesis contributes to the wood inspection field in the following ways. It is shown that there is a great potential in the smart image sensor concept, i.e. where low-level image processing takes place directly on the sensor chip. An integrated sensor system is presented which measures a number of different properties of the wood surface simultaneously on a single chip. A key contribution is an industrially feasible way to utilize the so called Tracheid effect which is based on the transmittance of light within the wood surface. Another significant part of the integrated sensor system is the dimension measurement. A number of very fast algorithms for sheet-of-light range imaging with range pixel rates up to 10 MHz are presented. This is considerably faster than what has been achieved from other systems.

Another contribution in this thesis is a real-time imaging spectrograph, which is also based on the smart image sensor concept. The system performs a multi-spectral linear pixelwise classification based on 256 color channels and the only data actually output from the sensor chip are classified pixels at a speed of up to 1 MHz.

Finally, it is shown that a fully automatic defect detection gives completely new possibilities to optimize the utilization of the raw material. A new cross-cut optimization algorithm is presented which exploits these new possibilities. The algorithm is based on dynamic programming and is capable of handling arbitrary numbers of quality zones within each product and optimizes not only the placement but also the orientation of each product. Furthermore, the algorithm is capable of handling fixed as well as variable length products.

Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden

Linköping 1996
ISBN 91-7871-693-4 ISSN 0345-7524

Linköping Studies in Science and Technology. Dissertations
No. 424

Automatic Inspection of Sawn Wood

Erik Åstrand

Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden

Linköping 1996

ISBN 91-7871-693-4
ISSN 0345-7524
Printed in Sweden by LJ Foto & Montage/Roland Offset, Linköping 1996

To the memory of my grandfather Sigurd Åstrand

Abstract

In the production chain from the tree to the final wood product, the wood material will be subject to various forms of inspection and grading in a number of steps. The subject of this thesis is automatic inspection of sawn wood. The primary target application is inspection for cross cutting in the secondary wood industry, although the results are applicable also to other inspection situations such as sawmill grading.

More specifically, there are two main problems addressed in this thesis. The first is the automatic detection of defects, such as knots, splits, wane edges etc., on sawn wood surfaces. The other is the optimization process when cutting the boards into products based on the defect information resulting from the inspection.

The goals of this thesis are to develop new image processing methods and algorithms for wood defect detection and then to use the defect information to optimize the utilization of the material. The industrial constraints of production speed and cost of the equipment are fully considered, which means that the proposed solutions can directly serve as a basis for developing commercially competitive products.

The thesis contributes to the wood inspection field in the following ways. It is shown that there is a great potential in the smart image sensor concept, i.e. where low-level image processing takes place directly on the sensor chip. An integrated sensor system is presented which measures a number of different properties of the wood surface simultaneously on a single chip. A key contribution is an industrially feasible way to utilize the so called Tracheid effect which is based on the transmittance of light within the wood surface. Another significant part of the integrated sensor system is the dimension measurement. A number of very fast algorithms for sheet-of-light range imaging with range pixel rates up to 10 MHz are presented. This is considerably faster than what has been achieved from other systems.

Another contribution in this thesis is a real-time imaging spectrograph, which is also based on the smart image sensor concept. The system performs a multi-spectral linear pixelwise classification based on 256 color channels and the only data actually output from the sensor chip are classified pixels at a speed of up to 1 MHz.

Finally, it is shown that a fully automatic defect detection gives completely new possibilities to optimize the utilization of the raw material. A new cross-cut optimization algorithm is presented which exploits these new possibilities. The algorithm is based on dynamic programming and is capable of handling arbitrary numbers of quality zones within each product and optimizes not only the placement but also the orientation of each product. Furthermore, the algorithm is capable of handling fixed as well as variable length products.

Acknowledgements

There are many persons without whom this thesis would not have been possible. First I would like to thank my supervisor Professor Per-Erik Danielsson for letting me run my project under his wings, for his initial efforts to raise funding for the project and for many valuable comments on both the material and the presentation. I also want to thank all other friends and colleagues in the Image Processing Laboratory for providing the creative atmosphere. Special thanks go to Anders Åström for valuable contributions to nearly all parts of my project. Thanks also to Mattias Johannesson for valuable discussions on the sheet-of-light algorithms, to Mats Österberg and Olle Hagman for their contributions to the spectrograph, to Mikael Rönnqvist for providing the optimization expertise and to Bryan Lyles for comments on the language as well as the presentation.

There are also a number of organizations to which I am very grateful. First to Innovativ Vision for leading me into the field of automatic wood inspection in the first place, and especially to Per Rowa for his support in the initial phase of the project. Thanks also to SP-Snickerier, Edsbyn, for supporting the project and providing test data for the optimization experiments.

The financial support from the Swedish National Board for Industrial and Technical Development, NUTEK, grant no. 9206171-4, is gratefully acknowledged.

Finally, a dozen roses to Christina first for supporting my decision to take on this project and then for enduring the consequences with love and patience. Thanks also to Mikaela for reminding me that there are more important things in life than a Ph.D. thesis. Her father sitting by the computer is unfortunately now the number one item on her list of boring things.

Erik Åstrand
Linköping, February 1996

Preface

An automatic wood inspection system can be divided into a number of sub-systems, and the subdivision is reflected in the structure of this thesis.

Chapter 1 gives an introduction to the production from logs to final products and to the different applications where automatic inspection is applicable. The different defects on which the inspection is based are also characterized, with emphasis on their appearance in digital images. Chapter 1 is a major revision of the introductory chapters in the previous report [7].

Chapter 2 contains a literature study describing a number of reported systems and algorithms for wood inspection. The chapter is previously published in [7], but has been updated with new material reported in the literature since then.

Our own contributions to the field start in Chapter 3 with the description of the integrated laser-based sensor system, with the tracheid effect measurement being one of the key features. This material was previously published in [10, 11, 12].

The subsequent Chapter 4 describes the dimension measurement which is another of the sensor principles in the integrated system. A number of sheet-of-light range imaging algorithms are presented. This material has been published in parts in [13, 14, 15].

The PGP-MAPP real-time imaging spectrograph is the subject of Chapter 5. This system was originally presented in [18], but has since then undergone major improvements. The new version of the system as described in this chapter has previously been published in a shorter version in [19].

Chapter 6 combines the different sensors into a complete compact defect detection system. This part of the work is previously unpublished and will also probably be subject to substantial improvements in the future.

In Chapter 7, a cross cutting optimization algorithm is presented where the new possibilities of an automatic system is fully exploited. The fixed length version of the system was reported in [8, 9], whereas the variable length version is previously unpublished. The variable length version has also been implemented in a commercial system.

Chapter 8 concludes the thesis and various aspects on the presented systems and algorithms are discussed.

Contents

Chapter 1

Introduction

1.1 The forest industry in Sweden and worldwide

The forest industry constitutes a fundamental part of the Swedish industrial life. Around 1/10 of the Swedish gross national product is directly or indirectly related to the forest industry. Two thirds of this volume come from paper and pulp while the remaining third comes from timber and wood products. One fifth of the total Swedish export income is generated by the forest industry. However, since the relative net export is large compared to other industrial sectors, e.g. the car industry where a lot of components are imported, almost 50% of the net export comes from the forest industry.

In 1992, the total world production of sawn softwood was approximately 320 million m^3 [149], while the Swedish production was around 12 million m^3. This puts Sweden in the fifth place worldwide, see Table 1.1. In terms of export of sawn softwood, Canada is in top with Sweden in the second place.

1	USA	81
2	Canada	56
3	Former Soviet Union	52
4	Japan	24
5	Sweden	12
6	China	11

Table 1.1 World production of sawn softwood 1992 (million m^3) [149].

1	Canada	44.6
2	Sweden	8.1
3	USA	6.5
4	Finland	4.9
5	Former Soviet Union	3.2

Table 1.2 World export of sawn softwood 1992 (million m^3) [149].

The total number of people directly or indirectly employed within the forest industry in Sweden is around 200 000. One third of the railway transport and one fourth of the road transport is related to this sector. The forest industry is also very important from a regional perspective due to its localization, often in rural areas.

1.2 Industrial inspection of wood

1.2.1 The situation today

The Swedish wood industry is characterized by a high degree of automation in all steps of the process, from the growing tree to the final wooden product. Work intensive operations are now automated and only have a few operators monitoring and controlling the process. The major exception to this extensive automation are inspection and grading tasks which are still largely performed manually. Wood is a natural material and two pieces of wood never look the same, and the inspection task has therefore been very difficult to automate.

To find the optimal use of wood involves several inspection and classification steps in the production chain from the tree to the final product, as illustrated in Figure 1.1. In the present production system, logs are first graded into standard quality classes as well as sorted according to dimension. The logs are then sawn in the sawmills and the resulting boards are graded again into standard quality classes. Finally, in the end-use grading, the boards from the sawmills are subject to inspection again and the optimum use in different final products is determined. The grading of both logs and boards are normally performed in the sawmills, while the end-use grading normally is performed in the *secondary wood industry*, i.e. the production facilities buying boards from the sawmills and manufacturing the end products, such as construction components, windows, doors, floors, furniture etc.

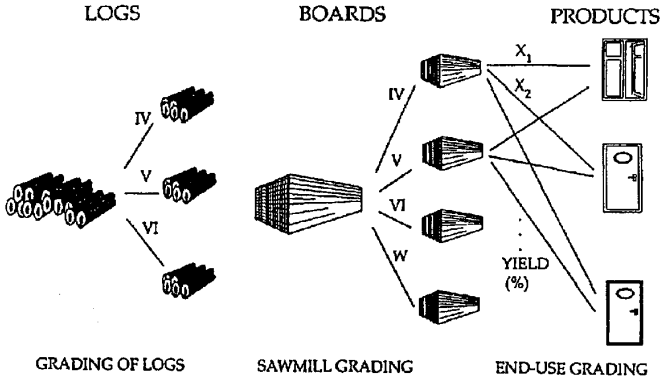


Figure 1.1 *The chain of classification steps in the present production system. From Grönlund [69].*

The subject of this thesis is the inspection of the sawn wood, primarily in the secondary wood industry, but many of the results should be valid also in sawmill grading. The first stage, measurement and grading of logs, is somewhat different and will only be discussed briefly. Normally, inspection of sawn wood means the process of detecting a number of *defects* on an otherwise clear wood surface. Each such defect will in one way or another contribute to a degrading of the wood, i.e. it will exclude one or more possible ways to use that particular piece of wood. This is true for the majority of inspection situations, but there are exceptions to this, e.g. the sorting of parquet floor pieces into different classes according to the darkness of the piece or the direction of the grain without necessarily considering those properties as defects.

Which defects to detect and which properties of the defects to measure depend on the situation. The required accuracy may also vary. In some cases, high speed may be the most important issue, even at the cost of a higher error rate. In other cases the inspection may take as long time as necessary, but it must be 100% correct. This time factor is very important to take into account when considering the introduction of an automatic inspection system as it may have a great impact on the feasibility of the system. A human inspector is incredibly good at adapting the ambition level of his work depending on the amount of time given for the task. A machine will probably never be that flexible.

It should be noted that in the present production structure, it is only in the secondary processing stage where the primary purpose of the inspection is to optimize the use of the material. In the sawmill grading, the purpose is merely to put price tags on the boards. Hence, the actual use of the raw material is only indirectly optimized in this stage.

1.2.2 Automatic inspection of wood

Today, automatic inspection and grading systems are being introduced in different stages of the production chain. Presently, most of them replace existing manual inspection tasks and perform the inspection according to the quality rules previously used by the manual inspector. Such systems may increase production efficiency and give a more consistent inspection, but as will be shown, the real potential in the automatic systems lie in the ability to implement much more complex and flexible classification rules than what is possible in the manual environment.

The motivation for automatic inspection in the wood industry has been well documented [81, 109, 110, 135]. A summary of the most frequent arguments are given below.

- **Expensive labor**. One obvious reason is to replace labor, which is getting relatively more and more expensive, while the price of computers is continuously decreasing.
- **Tedious work**. The manual inspection of wood requires high skill and long experience in order to do a good job. On the other hand, the work is very tedious and tiresome and the personnel turnover is high. It can be assumed that it will be more difficult in the future to get and keep skilled personnel.
- **Speed**. Manual inspection is a bottleneck in today's production. If the inspection can be performed faster, the overall throughput can be increased.
- **Accuracy**. Depending on the performance of the automated system, a more accurate inspection may be achieved, leading to a better yield and a higher and more even quality of the produced material.
- **Flexibility and complexity**. In an automated environment, much more complex decision situations can be handled. Information about every single defect, its type, position and size, can be taken into account. It is also possible to handle a much greater variety of options simultaneously than what would be possible for a manual inspector. This means that we are in a better position to adapt the quality rules to the requirements of the different end products.

The last reason is very important and is often overlooked when investigating possible automation of an existing manual inspection task. An automatic system may not only replace the manual inspector, but may also give new possibilities for optimization of the material. On the other hand, to really make use of the new possibilities, substantial modifications of the whole production line may be required.

An investigation of the quality variations in the chain of the present production system has been performed by Grönlund [70]. 1000 logs harvested from 16 randomly selected stands in Sweden were followed through the chain and the log grading, the sawmill grading and the final use of the material at two window and two door manufacturers were recorded. In this investigation, it was found that the correspondence between the log grading and the result of the sawmill grading was quite low. It was also found that the predictability between the sawmill grading of the boards and the yield in the end-use stage was very low. This means that neither the log grading nor the sawmill grading is actually performed according to the demands of the end products. The conclusion is that to really make use of the raw material in an optimum way, the grading in all stages must be made much more flexible and much more adapted to the real quality requirements of the end products. This is not possible within the present structure of the production chain. The standard sawmill grading constitutes a deeply rooted procedure which is not easily changed.

Another observation from the investigation is the low correspondence between different graders grading the same material. In the sawmill grading, only 60% of the boards were assigned the same grade (out of four) by two different graders, i.e. the grading is quite subjective. Automatic grading would probably give a more consistent result.

1.2.3 The future production system

The key to improving the utilization of the raw material is *information*. Fully automatic wood inspection gives a lot more information about the material than what could be obtained from a manual inspector. Furthermore, the more information we have about the end product in a given inspection situation, the better is our possibility to make use of the material in an optimum way. As will be shown, the exact position of a small knot, which is totally insignificant in the current sawmill grading rules, can greatly affect the output yield in a secondary processing stage. That is, given detailed knowledge about the production processes at a number of different customers, the sawmills could instead of sorting into standard grades directly calculate the value of each board for each customer and thus distribute the material in the most profitable way.

The trend today is actually to use more and more customer specific versions of the standard sawmill grades. Such customer specific grading is however *statistic* in the sense that one knows from experience that the given modification to the grading rules gives a certain statistic increase in output yield. However the future customer specific rules will be *deterministic*, meaning that the actual demands on the end product is directly used in the grading of each individual board.

Since manual inspection today is beginning to be replaced by automatic systems, we may end up in a situation where a board is automatically sorted into a standard garde as the last step in a sawmill, shipped to a customer in the secondary wood industry and then inspected again, possibly using an identical system, but now using completely different inspection rules. If the complete defect information was supplied together with the board, e.g. on a diskette, there were no need to inspect the board again. Furthermore, the customer could provide the sawmill a program which for each board calculates the exact yield and thereby the exact amount of money the customer is willing to pay for the board. The result could be an electronic bidding from several potential customers on each individual board.

1.2.4 Internal wood scanning

In the future production system, the inspection of the entire log will be an integrated part of the process of finding the optimum use of the raw material. Today the geometry of the logs are measured in order to find the optimum way of sawing them. There is, however, a lot of research in progress in the field of *internal* wood scanning, e.g. using computer tomography, CT.

If a full three-dimensional density volume of the log is available, we may optimize the sawing of the log, not only based on dimension but also based on defect contents. Such an optimization procedure is outlined by Chang and Guddanti [35] and is illustrated in Figure 1.2. The log is electronically cut according to different patterns and the resulting electronic boards are evaluated in order to find the best sawing combination. A similar project is run by Davis et al. [47]

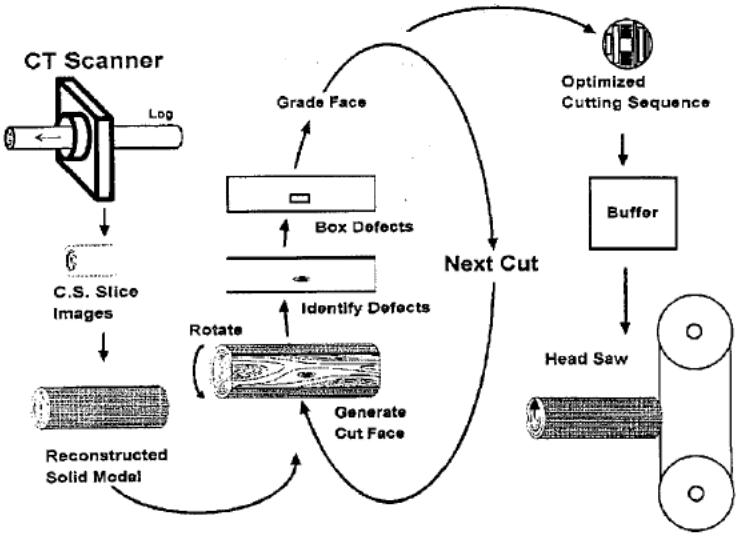


Figure 1.2 Future sawing optimization system based on internal wood scanning. From Chang and Guddanti [35].

An internal scanning system has to cope with many problems, some of which are the following. First, we must be able to acquire the full data volume at production speed. Second, we must have 3D-methods to detect and classify the different defects within the volume. Third, we need models to represent the three-dimensional geometrical structure of the defects from which we can estimate the two-dimensional defect contents of a board cut from within the log. The latter requirement stems from the fact that the 3D-system has to be interfaced to other hardware and software which specify their requirements in 2D-terms.

Lindgren [106] and Davis and Wells [46] discuss the acquisition of CT data from logs using medical CT scanners. It is found that the medical CT scanners of today are far from meeting the demands of the industrial environment, especially regarding speed. An alternative, at least in the near future, is scanners based on a limited number of projections. An example of such a system is TINA (Hagman [75]), a two-projection γ-ray scanner shown in Figure 1.3.

Similar X-ray systems are presented by Grundberg and Grönlund [72] and by Aune [23]. A system based on a limited number of projections can not be used for full three-dimensional reconstruction and optimization, but can be used to aid in the classification of the logs in the first stage of the present production process as outlined in Figure 1.1.

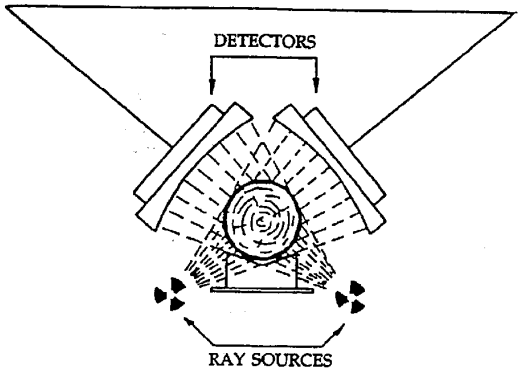


Figure 1.3 TINA, a two-projection γ-ray scanner (Hagman [75]).

The appearance of different wood defects in CT density volumes is discussed by Svalbe et al. [155] and Schmoldt et al. [145]. It is shown that most defects show significant density variations but that the appearance is very different for different wood species.

A model to describe the internal knot geometry in a log is proposed by Grundberg and Grönlund [71]. The purpose of this kind of models is to reduce the enormous amount of data resulting from a CT scanner and to represent the significant defect information in a more compact way. A knot can, somewhat simplified, be modelled by a cone-shaped structure originating from the pith and growing in the radial direction of the log as shown in Figure 1.4.

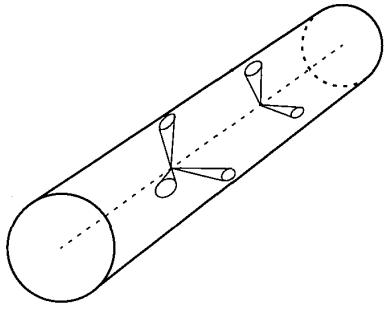


Figure 1.4 Internal modelling of knots using cone-shaped structures.

The aim of the optimization scheme outlined in Figure 1.2 is to find the optimum way to saw the log into boards, i.e. to maximize the value in terms of standard grades of the boards. However, the future production system where we make use of full information about the end products should of course be extended to the log scanning stage as illustrated in Figure 1.5. Preferably, we should examine the trees already as they grow in the forest to find the best use

of each given tree. This is actually not a new idea. In the past, material for ship building used to be gathered that way with people strolling around in the oak forests trying to find trees suitable for given sections of the ship, as illustrated in Figure 1.6.

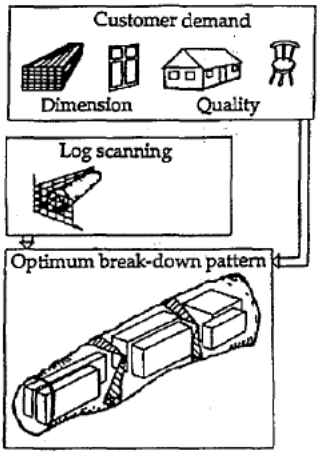


Figure 1.5 *The future production system where the demands of the end products are directly matched to the logs. From Grönlund [69].*

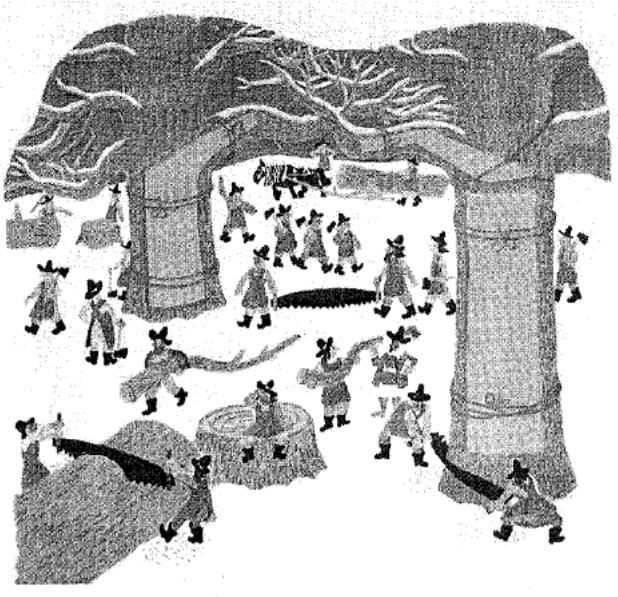


Figure 1.6 *Optimizing the use of wood directly in the forest. From Almqvist [4].*

1.3 Applications for automatic wood inspection

1.3.1 Quality standards

A fundamental part of all kinds of inspection is the quality rules on which the inspection is based. In most Swedish sawmills, boards are still sorted according to the traditional standard grades defined in *Gröna Boken* (the Green Book) [68]. The rules in the Green Book are written for manual inspectors, and they are sometimes very vague in their formulations. A major revision of the grading rules has resulted in (the blue book) *Nordiskt Trä* [121]. The latter is a result of the demands for more stringent and well defined rules, necessitated by the increasing use of automatic sorting equipments. However, as pointed out in Section 1.2.2, even if more stringent standard grading rules are a step in the right direction, they do not take full advantage of the new optimization possibilities in a fully automatic environment.

In the secondary processing, there exist less standards due to the great variety of applications. In some particular application fields there exist standards, one example being [156] which defines quality rules for glued joint boards. However, this standard, like many others, only defines the quality criteria for the final product, the glued joint board itself, which are not always directly applicable to the sorting of the components making up the product. Experience has shown that in practice, the quality rules actually in use are often defined by those performing the grading, and there can be a great discrepancy between the rules in use and the rules formulated in a quality standard like [156]. The conclusion is that a great deal of work must be put into the quality rules for any installation of automatic inspection equipment.

1.3.2 Sawmill grading

The purpose of the traditional sawmill grading is to classify each board into one of the quality classes defined by the standard classification rules. In practice, the manual grading is performed during cross transport with the grader sitting in one end as shown in Figure 1.7. Typical feeding speed is 30 boards per minute and the length of the boards is up 6 m. Under these circumstances, it is obvious that the grader can not make decisions based on exact size and position of all defects. It is merely a decision based on the general appearance of the board.

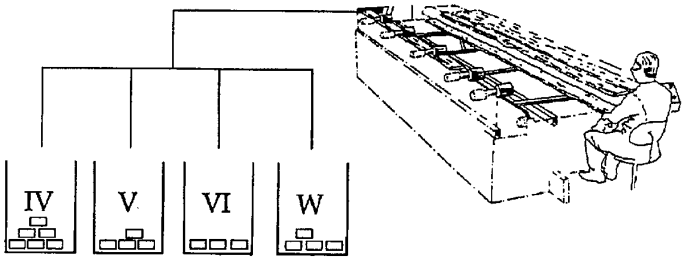


Figure 1.7 Manual sawmill grading. From Grönlund [69].

The grader also has the option of cutting the board from one end. If a severe defect can be removed, the board may be assigned a higher quality class and thereby a higher value even if it is shorter.

The grading is normally performed as the final stage of the processing in the sawmill. However, the boards may also be sorted at an earlier stage before drying, so called *green sorting*. The reason for this is the drying process which may introduce splits. By preliminary sorting the material, the higher qualities may be dried more carefully, i.e. using lower temperature and longer time, and thereby reduce the risk of splits. Actually, the first installations of automatic sorting equipments in sawmills have been in green sorting applications since the accuracy demands are lower than in the final grading. A missed defect will only lead to the board being unnecessarily carefully dried, but the defect will be captured in the final grading anyway.

Another application for automatic defect detection in sawmills is to improve edging operations, i.e. the process to find the optimum edge cuts for side boards as shown in Figure 1.8. Traditionally, edging is performed based on the geometrical shape, but since the amount of allowed wane is dependent on the grade, the edging lines can be adjusted outwards if we know that the board is of low quality because of other defects. This process can also be integrated with the green sorting.

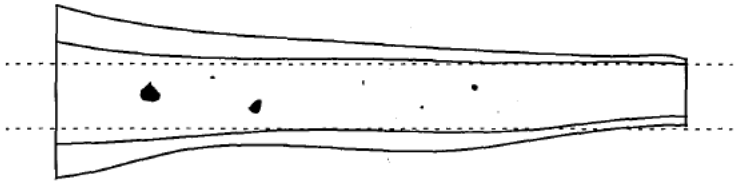


Figure 1.8 Optimum edging of boards may depend on defect contents.

1.3.3 Secondary processing

In the secondary processing we can make a distinction between the two applications sorting and cutting. In the sorting application, pieces of a given size are sorted in different piles according to some rules. A typical example of this is the production of parquet floors where the tiles making up the floor are sorted according to various criteria. Here the exact positions of the defects may be less important as the decision applies to the entire piece. In some cases there may be an option to cut the piece to a smaller dimension in order to remove a defect. This could be considered as a special case of the more general cutting problem.

In the cutting application long boards are cut into a number of pieces depending on the products manufactured. The cutting can be either one-dimensional, where we just cross cut, or two-dimensional where the cross cutting is combined with rip sawing in any order. The resulting pieces can be of either fixed or variable length.

The traditional cross cut saw is completely manual. The operator tries to fit different pieces between defects which are removed, see Figure 1.9. This kind of manual cutting is still widely used in the wood industry.

In a more modern cross cut saw, the operator is using fluorescent crayon and marks the major defects on the board (Figure 1.10). Areas containing minor defects may be given a lower quality by marking the area according to a defined pattern, e.g. a double or triple line. The marked board is fed into the saw where a special camera detects the positions of the crayon marks. Given the positions, a computer calculates an optimal cutting strategy and the board is cut accordingly. The optimization algorithm may use different criteria, but typically, a certain length of a certain quality will be assigned a predefined value and waste sections will have zero

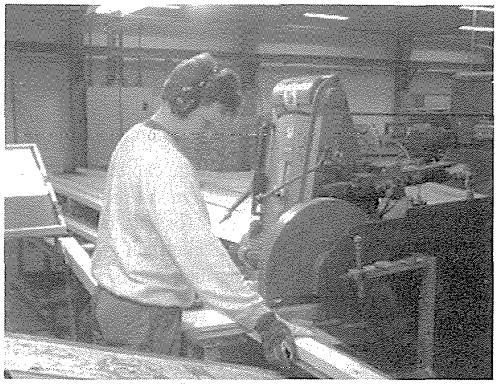


Figure 1.9 Manual cross cutting of boards (SP-Snickerier, Edsbyn Sweden).

Figure 1.10 Cross cutting of boards using crayon-marking (SP-Snickerier, Edsbyn Sweden).

value. The total value of the cutting combination is then to be maximized. The values may be adaptively modified according to the amount produced of that product so far. When introducing automatic inspection (Figure 1.11), one approach is to just replace the crayon mark camera with a vision system detecting the positions of the defects directly and sending positions to the optimizing computer as if they came from the crayon mark camera. However, as will be shown in the next section, it turns out that this approach will limit the possibilities for the subsequent optimization.

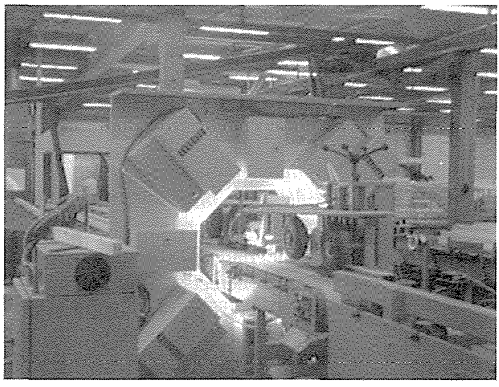


Figure 1.11 An automatic system for cross cutting (A Woodeye system [142], SP-Snickerier, Edsbyn Sweden).

1.3.4 Some examples from the secondary processing

A rather simple cutting example is cutting for finger-jointing. Major defects are removed and the resulting pieces are glued together as shown in Figure 1.12. No fixed lengths are used, but the pieces between the defects must have a minimum length. An automatic system will in this case only replace the manual inspector and no additional functionality will be gained.

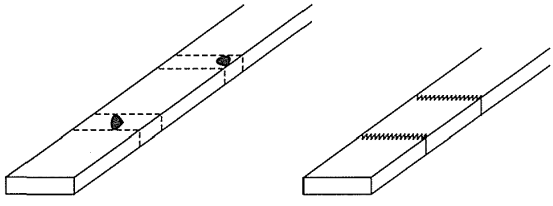


Figure 1.12 Finger-jointing.

A more complex cutting example from the manufacturing of windows will clearly show the new possibilities with an automatic system. The one below is taken from a Swedish window manufacturer (SP-Snickerier, Edsbyn). Several different dimensions are cut, one example being the 35×45 mm dimension, which is used to produce three different window casement profiles as shown in Figure 1.13. The windows are produced in different sizes, in some cases even according to customer specific orders. This means that the casement pieces are to be cut in a number of different lengths.

Two of the profiles in Figure 1.13 are used as inner and outer casement, respectively, in a double glass window, while the third profile is the outer casement in a triple glass window. Depending on the position within the window, a certain side will either be visible (marked B) or hidden (marked C). On the hidden sides, more defects will be allowed than on the visible

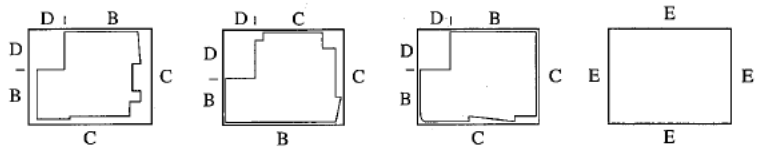


Figure 1.13 Window casement profiles from the dimension 35×45 mm.

sides. Furthermore, the corner in which the fold for the glazing is routed, marked D in Figure 1.13, may contain virtually any kind of defects except large splits. In addition to the window casements, the 35×45 mm dimension may also be used as the interior of a door frame. In this case, it will be completely hidden and its sides, marked E, may contain any defects. Hence, five different quality areas are defined, and an example of possible quality rules for each area is given in Table 1.3.

Defect	A	B	C	D	E
Black knots	-	< 10 mm	< 20 mm	OK	OK
Sound knots	-	< 30 mm	OK	OK	OK
Splits	-	-	-	< 0.5 mm	OK
Pith	-	-	-	OK	OK
Pitch Pockets	-	-	-	OK	OK
Wane	-	-	-	OK	OK

Table 1.3 Sample quality rules for different areas.

Furthermore, there may also be different quality requirements in the lengthwise direction of a fixed length. A typical case is when the casement parts are being joint together as shown in Figure 1.14. Defects in the ends of a piece may cause the tenon and mortice to break. Therefore, the window manufacturer referred to above requires that the cut pieces are completely free of defects on all sides in a 50 mm zone at each end. This corresponds to the area A in Table 1.3 and Figure 1.14.

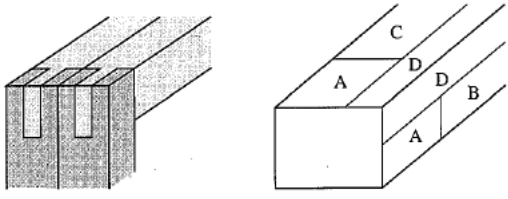


Figure 1.14 The joining of casement parts may require defect free ends.

Depending on the symmetry of the profiles, it may be possible to match the quality requirements of the pieces to the defects on the board using different orientations. In the case of a square cross section, we may have up to eight possible orientations. If we have more than a few different lengths or profiles, it is clear that the problem to find the best mix of lengths at different positions and orientations becomes rather complex.

This kind of complicated quality rules can not be handled in a crayon marking system. There is no way the operator can supply the optimization with enough information about the defects. He can only mark larger defects and assign general qualities to the areas in between. Assume he is cutting boards according to the quality rules given above. If he finds a small defect near a corner he may disregard it, assuming that it will be possible to turn the piece so that the glazing fold ends up in that particular corner. But what if he sees two such defects in opposite corners some distance apart? If the board is cut between the two defects everything is OK, but there is no way the operator can force this to happen. The same is true for defects that end up too close to a sawcut, i.e. within a defect free zone at an end. The only way to handle this kind of complex quality rules is to use an automatic defect detection. One major conclusion from this is that a fully automated system, besides replacing the operator and doing his job a little bit faster, gives completely new possibilities to optimize the use of the raw material. The solutions to the new optimization problems arising is the subject of Chapter 7.

Another example, this time from the sorting application, is manufacturing of glued joint boards. Fixed lengths of a certain dimension are rip-sawed into stripes which are sorted. A number of such stripes are glued together edge by edge as shown in Figure 1.15. This kind of board is typically used as a furniture component and may end up in a table or a bookshelf.

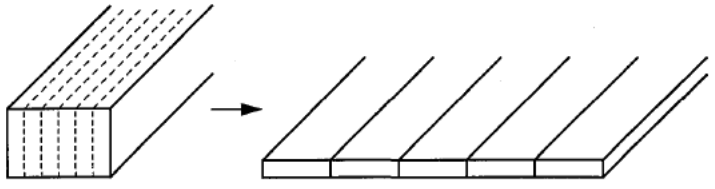


Figure 1.15 Glued joint boards.

Such a board can be produced according to different specifications. [156] defines five qualities 1-5, the highest being 1. The final product may be of for instance quality 2/4, which means that the front side must correspond to quality 2 and the back side to quality 4 (at least). The key here is to organize the production in such a way that we always produce a mixture of different qualities on order to use the material in an optimal way. With an automatic system, we would be able to sort into much more qualities simultaneously than what could be handled by a manual inspector.

1.4 Outline of an automated system

1.4.1 General

In this section we will take a closer look at the different components making up an automatic wood inspection system. Most existing systems have, with some variations, the same basic structure shown in Figure 1.16. The system typically consists of a *conveyor belt*, which trans-

ports the boards in front of a number of *sensors*. To acquire data using the sensors we also normally need some kind of *illumination*. Data from the sensors is sent to the *image processing* unit in which the actual detection of the different defects takes place. Information about the defects is sent to the *optimization* which compares the defect information to predefined *quality rules* and finally decides what to do with this particular piece of wood. The decision is sent to some kind of *actuator*, such as a sorting device, a cross cut saw etc. In order to monitor the process and to change parameters etc. we also need some kind of *user interface*.

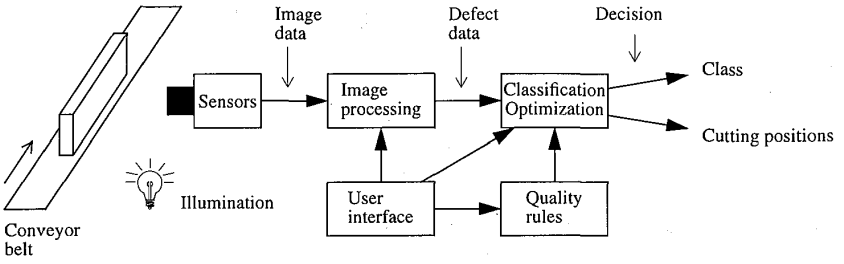


Figure 1.16 Outline of a wood inspection system.

1.4.2 Sensors

It is obvious that the result of whatever image processing methods we apply is very dependent on the quality of the images we receive from the image sensors. By the term *sensor* we normally mean some kind of camera, but any type of sensor could be applicable. A wood inspection system should include a number of different types of sensors in order to capture all significant features of the board.

Most defects can be detected from grayscale images. However, for defects such as blue stain and decay, which do not have any typical shape nor contrast in a grayscale image, color information will be needed. Color information can be acquired using either standard RGB cameras, special optic filters, or full spectrometry. This will be further discussed in Chapter 5.

What resolution do we need? The type of defect requiring the highest resolution is small splits, which in some applications have to be detected down to a width of 0.1 mm. As the splits normally run in the lengthwise direction of the board, the high resolution is mainly needed across the board. By using rectangular pixels we can reduce the amount of data and still have enough resolution to be able to detect the small splits. Rectangular pixels will also give the image a more handy format. A typical resolution is 0.1-0.2 mm in the cross direction and 1-2 mm in the length direction.

A natural solution for input of this kind of images is the use of line scan cameras and transporting the boards in front of the cameras in the lengthwise direction. One camera is required for each of the four sides of the board. Depending on the application, it may be sufficient with only one or two cameras. A line scan camera also makes it easy to vary the resolution along both axis independently. A typical feeding speed to be in parity with saws and other machinery is about 1 m/s. Assume that we are scanning a width of 10 cm with a cross resolution of 0.2 mm and a length resolution of 2 mm. We then need to be able to input 500 lines of 500 pixels

per second, and we have to do it from four cameras. Thus, in a real time system we have to be able to process 2 Mpixel/s in this case. It is obvious that powerful computers will be needed to handle this high data rate.

1.4.3 Length vs. cross transport

There are arguments against length transport, especially in the sawmills where the high volumes would give very high feeding speeds. As the manual grading in today's sawmills is usually performed using cross transport, installation of an automatic grading system using length transport in an existing sawmill will require substantial mechanical modifications. A cross transport system will not naturally give the desirable rectangular pixels and it will be difficult to inspect all four sides. Nevertheless, in order to develop an industrial system for the existing sawmills, it would probably be necessary to use cross transport. A cross transport situation can of course be converted to length transport by moving the sensor along the board while the board is moving crosswise. This will however give a mechanically very complex system and we will still have problems scanning all four sides of the board.

1.4.4 Illumination

An important part of the sensor system is the illumination of the board. A diffuse light source is normally preferable but a direct light at a given angle may improve the detection of some defects. Here it must be noted that wood surfaces can look very different depending on the direction and the angle of the illumination. This will be further discussed in Chapter 3. If we use color cameras it is also important that the illumination is spectrally stable.

Regardless of the type of illumination, it is often very difficult to achieve a uniform light distribution which requires the use of some kind of shading correction. Controlling the background illumination may help in detecting the edges of the board as well as to detect holes through the board. If we are scanning profiled boards with a non-rectangular cross section, the illumination problem requires even more attention.

1.4.5 Dimension measurements/range cameras

An important part of the inspection is to verify that the dimensions of the wood piece are within certain limits. Given the geometry of the camera arrangement, we can easily measure the width and height of the wood piece within the images. In some cases it is not the absolute measure that is important but rather the width variation within one single board.

A little bit more difficult is the detection of wane edges. Even if some material is missing at one edge, the width measured perpendicular to the surface may still indicate full width or only a very small deviation from the correct value. This depends on the focal length of the lens and the position of the edge relative to the optical axis as shown in Figure 1.17. If we are scanning several dimensions simultaneously, it is usually difficult to always keep the optical axis in the middle of the board. Wane edges are therefore difficult to detect this way. This is a situation where the human grader has a great advantage since he can view the board from several directions.

The wane edge is often dark and in those cases it can be detected directly in the image. Also, an arrangement with cameras placed at a certain angle from the board may be used to measure its diagonals and thereby to detect the wane edge. One problem here is that the board must be fed absolutely flat. To avoid this we may instead use the sum of the diagonals, which makes it easy to detect that wane is present but makes it difficult to measure its size and position. This is OK if we just should reject a piece containing wane but can not be used in the window example in section 1.3.4.

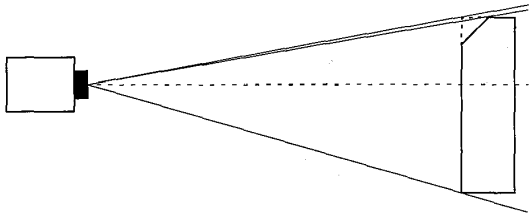


Figure 1.17 The width measured perpendicular from the surface at a wane edge depends on the focal length of the lens and the position relative to the optical axis.

There are several laser based distance measurement devices on the market which may be used to detect various dimension faults. Unfortunately, the majority only measure one or a few single points. Ideally, we should capture a complete range image of the surface, i.e. the full three-dimensional profile. One type of range camera uses a laser line projector and triangulation methods as shown in Figure 1.18. If an ordinary video camera is used as sensor, the line frequency of the system will be limited by the frame rate of the camera. Given a transport speed of 1 m/s, this gives a length resolution of 2 cm using a 50 Hz camera. This may be OK just to detect wane, but with a higher resolution we could do more, e.g. detect splits and small holes. A number of high speed range camera solutions will be presented in Chapter 4.

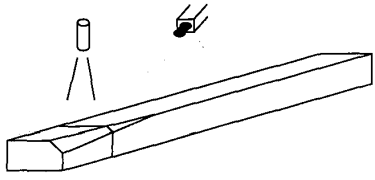


Figure 1.18 Line-laser based dimension measurement.

1.4.6 Image processing

The next step is to derive useful information from the images acquired from whatever sensors used. The goal is to get a list of all detected defects, correctly classified and localized on the board. In Chapter 2 we will give an overview of different systems, methods and algorithms proposed so far in the literature. The methods vary in complexity and computation demands, but even the simplest algorithms require a substantial amount of computing power to be able to handle a data flow in the order of Megapixels per second. In practice, the image processing will be partitioned into several levels and a real time system will probably handle the different levels using different types of hardware in a pipelined fashion. Typically, the image processing

can be divided into *segmentation* which partitions the image into possible defects (objects), *feature extraction* which calculates a number of features for each object and *classification* which based on the features determines which defect (if any) the object represents.

1.4.7 Optimization

By the term optimization we mean the process of taking a decision about what to do with a piece of wood given the list of defects. This may span a wide range of applications from standard sawmill sorting to the kind of complex cutting optimization described in Section 1.3.4. Depending on the complexity, this step may or may not be implemented in the same hardware as the defect detection.

It is desirable that the defect detection is independent from the optimization. A knot is always a knot and should be classified as a knot in the image processing stage no matter what the optimization stage decides to do with it. In reality however, experience has shown that the stages are more intertwined. Since we will never achieve 100% correct classification, we have to take into account the consequences of a classification error. The economical consequence of a false positive classification compared to a false negative one can be very different depending on the quality rules used. A false positive defect would lead to an unnecessary degradation of the material value, while a false negative, i.e. a missed defect, may lead to a reject further down the line, so that a complete finished door or window may have to be discarded in the final inspection, which could be very expensive. All classification is statistical in the sense that we may be more or less confident that a potential defect is of a certain type, and which confidence level to use in the classification depends heavily on both the type of defect and on the products we manufacture.

This dependency is most undesirable. Better and more accurate defect detection will of course reduce the problem, but it is impossible to eliminate it completely. One possibility is that the defect detection part also supplies a confidence measure which can be used in the optimization.

Another situation where the border between the defect detection and the optimization is a little bit fuzzy is when the actions caused by a defect is dependent on other defects. For example, it is rather common that the quality rules state that a certain number of knots of a given size are allowed per meter of the board. If a number of such knots are located too close to each other, this can either be regarded as a task for the optimization to figure out or the entire area containing the knots can be defined as one single high level defect. In the case with knots per meter, the first approach may seem natural, but if the problem is to detect a cluster of say small worm holes, the second approach could be more relevant.

1.4.8 User interface

An important part of any computer system is the user interface. When the system is used in actual production, no user interaction should be required. To start and stop the machine and to load different sets of predefined quality rules could also be made quite simple. What will require more efforts is an environment to define and test the quality rules. It must be possible to define the rules in terms familiar to the operator. One problem is that the different applications have very different structure in their quality descriptions.

It is also necessary that all levels of the scanning process can be monitored in order to verify the correctness of both classification and optimization. The definition of a set of classification and optimization rules will probably include the adjustment of a number of different parameters, and it must be possible to verify the effect of the different settings. The system may for instance include various simulation modes, e.g. the possibility to physically scan a

board once and then run several tests on the stored image using different quality rules. It would also be an advantage to be able to run the optimization part repeatedly from a given defect list using different values etc.

A discussion on wood scanning systems from a user's point of view is given by Bates [25].

1.4.9 Mechanical aspects

There are a number of mechanical problems that must be solved in an industrial system. During the scanning the board must be fed without vibrations and often at a constant speed. In a cross transport situation where the sides are scanned one by one we also need some turning device in between.

The environment in a wood working factory with sawdust everywhere also creates problems. Careful housing of sensitive electronic parts is required and the cameras and lamps must be kept free from dust. This is especially difficult if we use cameras or lamps from below to scan the bottom side of the board. Other problems arise from electrical noise from starting and stopping motors etc. Finally, a very important aspect is the potential hazard of fire. The regulations are naturally very strict in the wood industry.

1.4.10 Manual verification and backup

We will probably never have a wood scanning system which correctly detects and classifies 100% of the defects. The system will make errors, but a good system should minimize those errors. And what counts is not the number of errors but rather the total economical consequence of them. A missed defect at an early stage may lead to lower quality, in a later stage of the production to an expensive reject or even customer complaints on the final product. An incorrectly detected false defect will lead to an unnecessary degradation of the material and thereby lost profit.

Depending on the economical consequences of errors, some kind of manual post-inspection may be required. To verify that the pieces in a sorting application are correctly sorted require much less manual effort than to perform the entire sorting manually. Sometimes an alternative could be to automatically classify the major part of the material, but to assign a separate class for the uncertain cases, which are manually sorted at a subsequent inspection station.

In a cutting application, the situation is somewhat more complicated. It is always possible to verify that the products produced meet the quality rules but it is more difficult to manually correct false positive errors, i.e. when false defects are cut away and disappear on a waste conveyor belt.

The need for manual backup is something that the suppliers of automatic inspection equipment naturally is unwilling to discuss when selling the systems. They will rather discuss what the system can do than what it can't. Experience has shown however [108] that it is *very* important to study the consequences of classification errors already when planning for the installations so appropriate mechanical arrangements for manual backup can be planned for. In many cases the system can still be profitable despite the errors, and if it is not, it should not have been installed in the first place.

1.5 Characterization of defects

1.5.1 General

In this section we will take a closer look at the different defects necessary to detect in order to classify the wood. As pointed out before, the term defect is taken in a general sense as some detectable property of the wood. In most cases, clear wood is considered to be of the highest quality. Hence, defects are all deviations from clear wood.

The defects will be described in terms of their visual appearance, especially their appearance given the image from an ordinary grayscale or color camera. For more detailed information about the physical properties of different wood defects, we refer to [121].

So far nothing has been mentioned about different species of wood. Dark wood like oak presents a much lower contrast than for instance pine and can therefore be more difficult to handle. However, as pine and spruce are so totally dominating in the Scandinavian wood industry, we will concentrate on those species which themselves have a rather similar appearance. Even so, most of the reasoning is valid for other species as well.

Most of the images shown in this section to illustrate the various defects were copied (with permission) from [162], while a few were scanned by a Woodeye system [142] at Ala sawmill, Sweden. The resolution of the first images is around 0.5×0.5 mm and for the Woodeye images around 0.2×1.5 mm. The rectangular shape of the pixels in the latter case can be clearly seen in the images. Some of the images from [162] are also shown in color in Appendix A.

1.5.2 Knots

The most common defect in almost all kinds of wood is the knot. There are different types of knots which more or less represent the different stages in the knot's life cycle. A *sound knot* (Figure 1.19) contains fresh growing wood. The sound knot is usually darker than the surrounding wood and can thus be detected by a simple thresholding operation. It happens however, especially in spruce, that the knot is not dark at all compared to the surrounding wood and that the only sign of the knot is the structure in the grain direction. The knot may also be only partly dark. In that case the knot can be detected using thresholding but the size of the knot can not be accurately measured. Whether this is important or not depends on the application.

The position and orientation of the knot relative to the surface of the board is also important. If the knot is cut perpendicular to its growing direction we will get a *round knot* (Figure 1.19a). If it ends up close to the edge of the board it will look like Figure 1.19b. Depending on the angle between the knot and the saw plane, the knot may also be more or less oval. In the extreme case where the knot is cut in its length direction it may look like Figure 1.19c. This knot is called a *splay knot* if it reaches the edge of the board or a *spike knot* (Figure 1.19d) if it does not. These knots must also be classified as knots even though their visual appearances are completely different from normal knots.

The *dry knot* (Figure 1.20) consists of dead wood and is usually (at least partially) dark, and thus easy to detect using simple thresholding techniques. A *black knot* is a dry knot which is very dark. If the knot is very small it is usually called a *pin knot*. The different orientations of the knot described above for the sound knot are applicable also for the dry knot.

A special case of knots is when the knot itself is fresh but surrounded by encased bark. This is called a *barkringed knot* or an *encased knot* (Figure 1.21). According to [121], the knot should be classified as a sound knot if the bark covers less than 1/4 of the knot. Otherwise it should be classified as a dry knot.

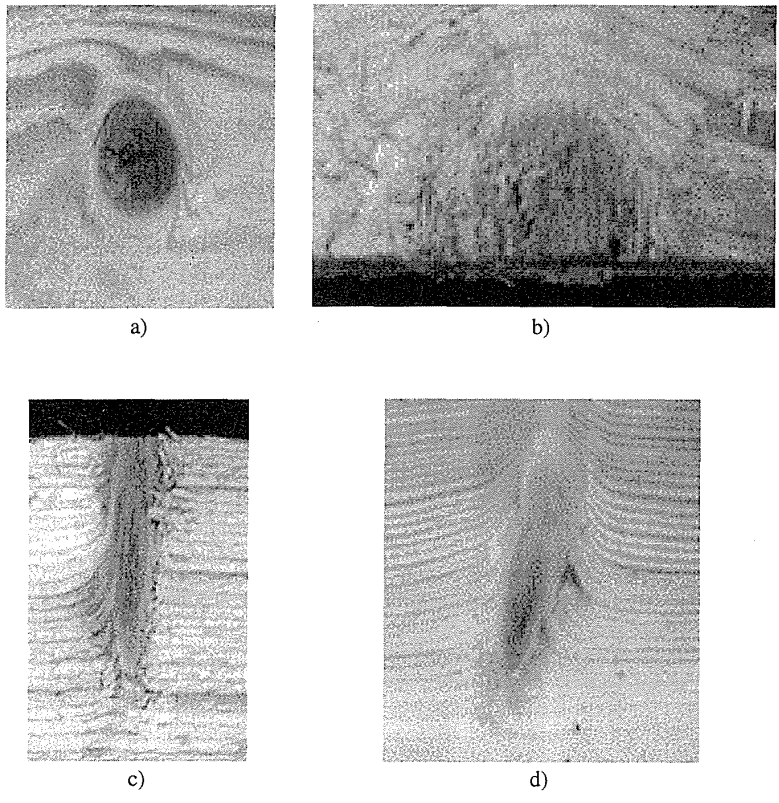


Figure 1.19 Sound knots a) Round knot b) Edge knot c) Splay knot d) Spike knot.

The *decayed knot* is a dry knot which has started to rot (Figure 1.22). A knot which is often considered to be a severe defect is a *sound knot with a split*. Such knots can be very difficult to detect, especially if we use rectangular pixels with low resolution in the length direction, as suggested in Section 1.4.2. The reason for using rectangular pixels is to make it easier to detect small splits which almost always run in the lengthwise direction of the wood. One of the cases, however, when the split may run in any direction is when it is located within a knot. A final type of "knot" is the *knot hole* where the knot has fallen out (Figure 1.23). This can be difficult to distinguish from a black knot in an image, unless using range imaging or illumination from the back side of the board.

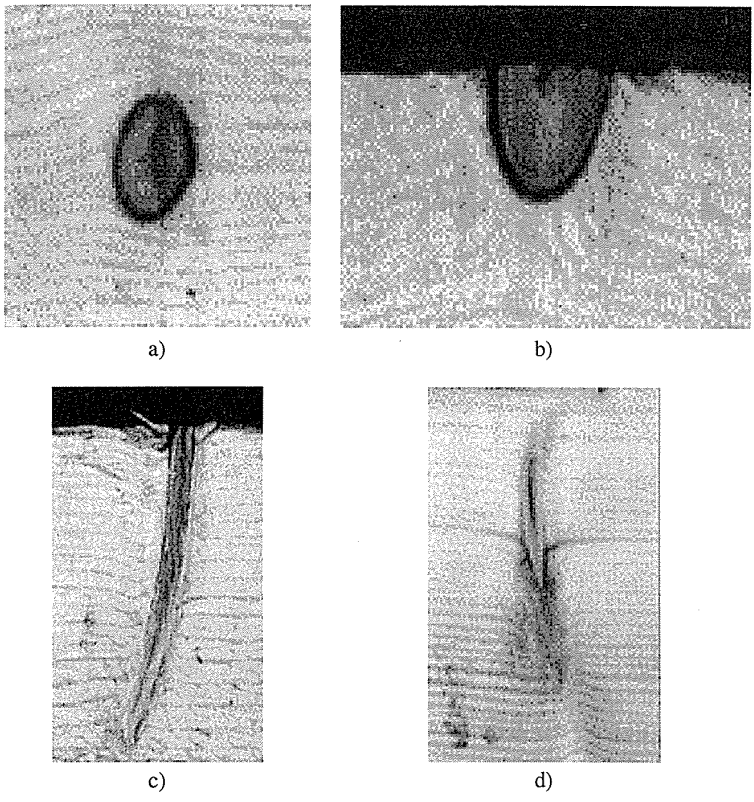
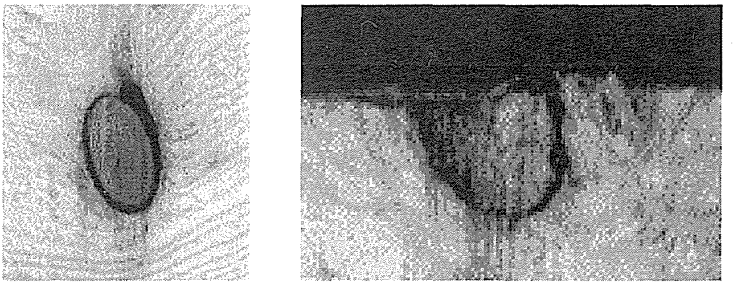


Figure 1.20 Dry knots a) Round knot b) Edge knot c) Splay knot d) Spike knot.

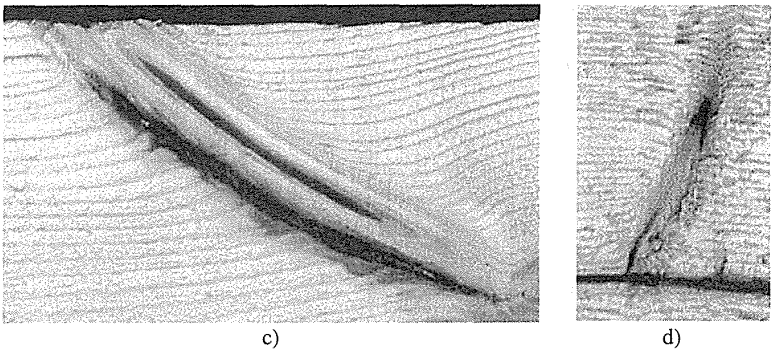
1.5.3 Splits

There are different types of *splits* (sometimes called *checks* or *shakes*) depending on the physical origin of the split. Most common are seasoning splits which originate from tensions in the wood during the drying process, but from the defect detection point of view we may handle all splits as one single type of defect. The width of the split can be anything from a tenth of a millimeter up to several centimeters and especially the narrow splits can be very difficult to detect.

Normally, the split is darker than the surrounding wood and can therefore be detected by thresholding. The long and narrow shape is typical (Figure 1.24). A fact that may cause problems is that a longer split often consists of a series of smaller ones. As quality rules sometimes are expressed in terms of split length, we must be able to merge the small splits together to one defect. The contrast of the split is also highly affected by the illumination. If we are using



a) b)



c) d)

Figure 1.21 Barkringed knots a) Round knot b) Edge knot c) Splay knot d) Spike knot.

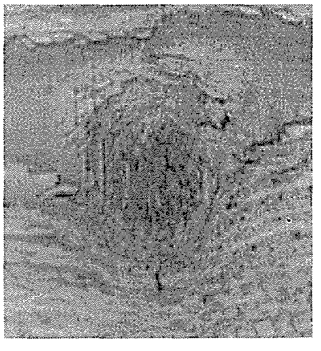


Figure 1.22 Decayed knot.

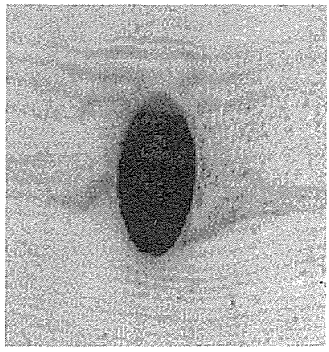


Figure 1.23 Knot hole.

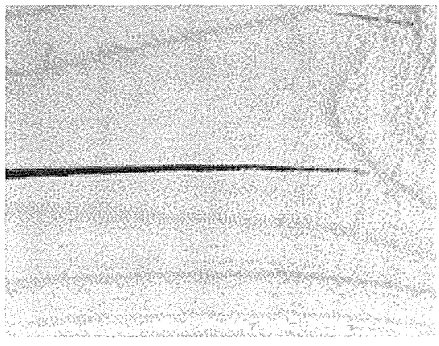
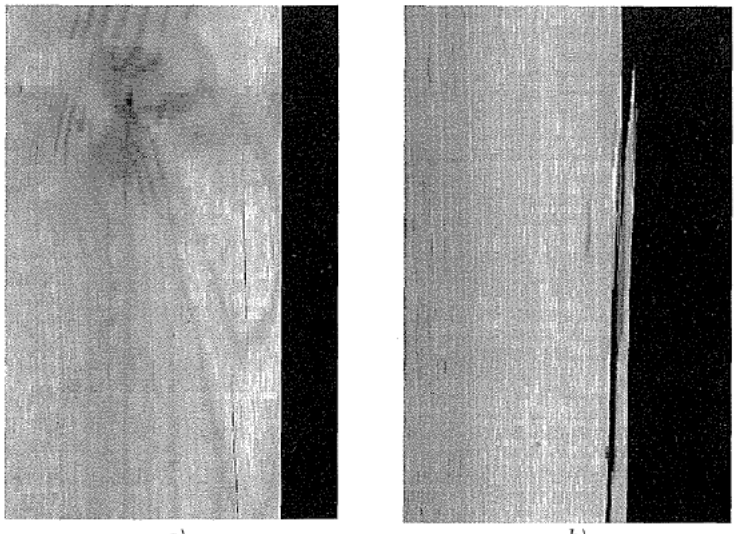


Figure 1.24 Split.

direct light from a number of lamps and if the angle of a lamp coincides with the angle of the split, the contrast may be very low. Generally, diffuse illumination arrangements work better in this case.

A problem with the split detection is the fact that there are other things on the wood surface that are long and dark. If we are scanning rough wood, the roughness of the surface in combination with direct illumination may generate a lot of small narrow shadows which can be misinterpreted as splits. Also the grain of the wood could sometimes very well be as dark as a split. In Figure 1.25a we can clearly see the splits with our eyes but the splits are actually not darker than the grain in the top of the image.

Another "split problem" is shown in Figure 1.25b, where a small part of an edge is broken, giving a *false edge split*. In the image this appears as a split, but in many situations this one may be harmless, depending on the size and what we are going to do with the wood piece. On the other hand, a real split along the edge could look approximately the same on the surface but continue deeper and in another direction inside the wood and thereby be a serious defect. This



a) b)

Figure 1.25 a) Small splits b) False edge split.

problem can be even more difficult when scanning rough wood as there naturally will be splinters along the edges. As splits are rather severe defects, a false negative detection, i.e. where something is incorrectly classified as a split, can be of substantial economical consequence.

1.5.4 Pitch pockets

Pitch pockets or *resin pockets* (Figure 1.26) are long cavities filled with resin. They are most common in spruce but appear sparsely also in pine. They are normally not as dark as splits and are also wider. Some pitch pockets have a typical shape with one straight and one slightly curved edge which is more or less emphasized depending on the angle between the surface of the board and the year ring direction. If the surface is perpendicular to the pitch pocket it will appear more narrow and it will probably also be deeper which makes it darker in the image. In this case it can be hard to distinguish it from a split.

1.5.5 Pith stripes

Pith (or *core*) appears in long stripes as shown in Figure 1.27. Significant for pith stripes is the relative length and that they normally are wider and not as dark as splits and longer than pitch pockets. However, as both pitch pockets and pith can have very different sizes it is usually impossible to distinguish them based only on shape and darkness. Still, a human inspector will have no problems telling the difference and hence, it should be possible to extract discriminating features from the image.

1.5.6 Bark pockets

Bark pockets, or *inbark*, shown in Figure 1.28, are normally black and in general there is no problem to detect that some kind of defect is present. The problem is that the shape and the structure of the bark pocket is unpredictable, and it can therefore easily be misinterpreted as

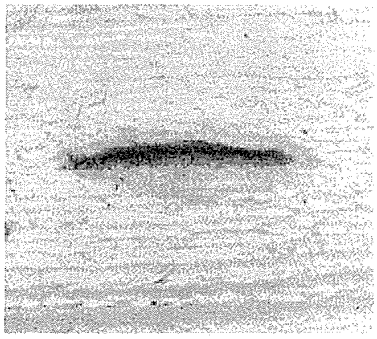


Figure 1.26 Pitch (resin) pocket.

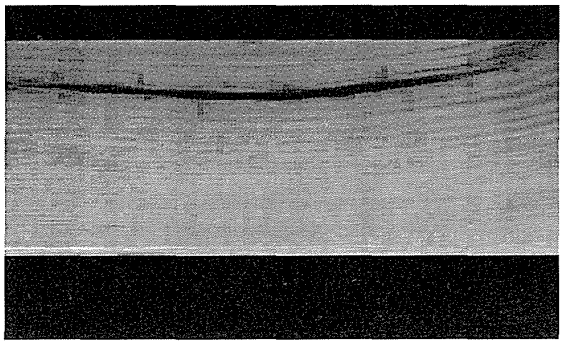


Figure 1.27 Pith (core) stripe.

another defect, especially if the bark pocket is small. Bigger bark pockets are usually much bigger than most other defects so one approach could be to use a general "big ugly" defect class for anything that is dark and big. Sections of wood with these defects should probably go to waste anyway so it really does not matter if the cause was a very big black knot or a bark pocket.

1.5.7 Blue stain and other color defects

Blue stain shown in Figure 1.29 (see also the color images in Appendix A) is a blue discoloration of the wood caused by certain types of fungi. It appears mainly in pine. It is virtually impossible to detect in a grayscale image as other parts of the clear wood can be well as dark and the blue stain itself does not present any typical shape or structure. On the other hand, given the right color information it is usually possible to detect.

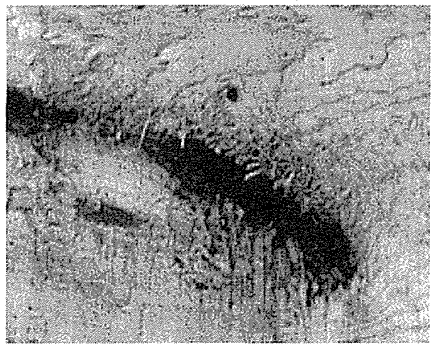


Figure 1.28 Bark pocket.

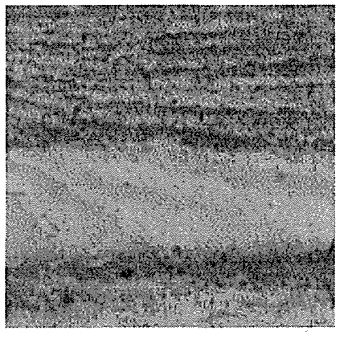


Figure 1.29 Blue stain. See also Appendix A.

Other color defects include chemical discoloration caused by various environmental exposure during drying and storing of the wood, e.g. *brown stain* (Figure 1.30). Important here is how deep into the surface the wood is affected. If the boards are to be planed after the scanning, the discoloration may be acceptable. However, they could still be a problem as the detection of other defects may be more difficult in the affected areas.

1.5.8 Decay

Decay can appear in many different ways and is therefore hard to detect, sometimes also for a human grader. One example is shown in Figure 1.31. Probably some kind of textural or color properties must be used as neither shape nor contrast is characteristic.

1.5.9 Compression wood

Compression wood is irregularities in the year rings caused by mechanical stress while the tree is growing, for instance due to hard winds. The wood on the leeward side will thereby become hard and brittle and the wood will easily bend after sawing. Compression wood is often, but

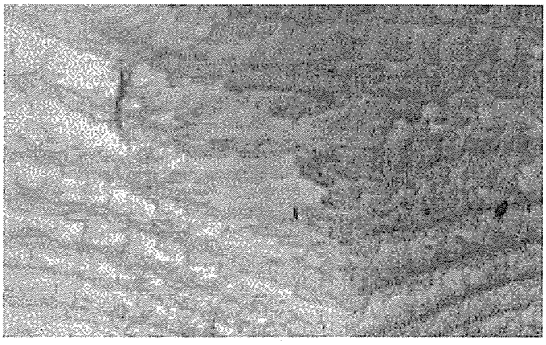


Figure 1.30 Brown stain. See also Appendix A.

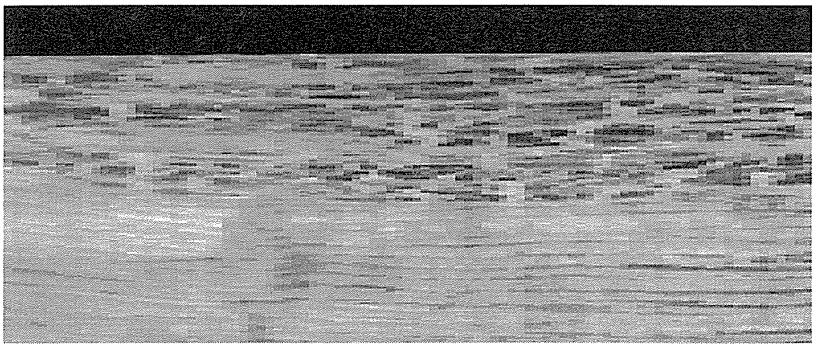


Figure 1.31 Decay.

not always, characterized by bigger and fatter grains as shown in Figure 1.32. Unfortunately the grain may look the same also in other cases. A board containing compression wood is often bent and that may be easier to measure, see Section 1.5.12. The boards which are straight but still contain compression wood are of course missed, but on the other hand the compression wood may be less harmful in this case. The board may bend, however, in a later stage, for instance after rip sawing. A study performed by Hagman [77] shows that compression wood has some significant spectral properties. Using color information could then perhaps solve the problem.

1.5.10 Dimension Faults

One of the basic tasks for an automatic inspection device is to verify that the material meets stipulated minimum and maximum dimensions. Here we should have a significant advantage compared to a manual grader which often is limited to random tests with callipers. Different

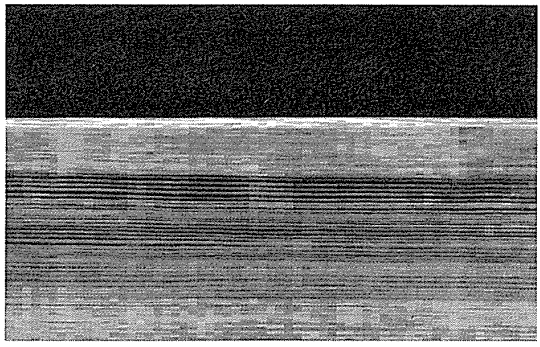


Figure 1.32 Compression wood.

camera arrangements to handle the dimension checking problem was discussed in Section 1.4.2. Here we just note that the most common dimension fault is the wane edge shown in Figure 1.33.

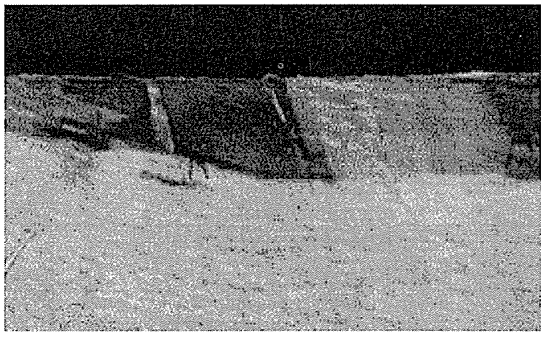


Figure 1.33 Wane.

If the wane edge is covered with bark, i.e. is dark, it is normally no problem to detect it directly in the image as it is always located adjacent to an edge and normally is longer than other defects. If we are scanning all four sides of the board we can also correlate the image from two cameras and thereby measure the size of the wane edge, assuming that it is convex. However, if the wane edge is light we need other methods. Typically, various laser arrangements are used, one was shown in Figure 1.18 on page 16.

If a high resolution *range camera* is available, a lot of other types of defects where dimension is involved could be detected. Examples could be splits and holes which could be detected with better accuracy than what is possible in the ordinary grayscale images.

Another dimension fault which requires very high resolution to detect is shown in Figure 1.34. Here the wood piece has been disturbed when moved through a planer and the result is a small wave-shaped defect which can be nearly invisible to the eye but still give a bad result if the surface is to be glued. Manually it can be detected by moving a finger along the edge but this is a rather complicated procedure in a real sorting environment.

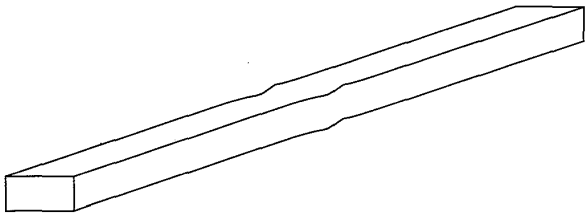


Figure 1.34 One type of dimension fault.

1.5.11 Roughness

When gluing surfaces, it is very important that the surfaces are well planed, which means we have to be able detect roughness as a defect. Roughness is easy to see in reality but could be difficult to detect in an image. Usually, roughness is caused by under-dimension and could then be detected using dimension measurements, but there could be other reasons as well, such as badly adjusted or worn out tools.

1.5.12 Bow and twist

The dimension faults discussed in Section 1.5.10 were all of local nature, i.e. we could use information from line scan cameras, either normal cameras or range cameras. More global geometrical faults such as bow and twist can not be measured in this way. Furthermore, if we are using length transport we will probably have some pressure wheels holding the wood piece during the scanning and this will effectively straighten out this kind of defects. The result may be an image which is absolutely straight even though the board itself looks like a banana. Once again, this is a property which is easily detected manually when the board is viewed from a distance.

Very bent wood should normally go to waste. If it is slightly bent it might be used but with restrictions, e.g. in a cutting system it might be allowed to cut short pieces. This means that we have to detect the bow as well as measure how bent it is. In order to do this we also need some geometrical model to describe the shape of the board. A model and a laboratory bow measurement procedure is proposed by Milota [119].

1.5.13 False defects

Compared to a human inspector, automatic inspection is very easy to fool by false defects. Dirt, grease, oil, paint from conveyor belts, foot prints, pen markings, burn marks from saw blades etc. can easily be misinterpreted as serious defects. Normally this kind of defects only affect the surface of the wood and will probably be removed in a later stage of the production,

e.g. if the boards are planed. The false defects present no problem for a manual inspector but the unpredictable shape and darkness of those areas makes them difficult for the machine to ignore.

The problem with dirt etc. is actually one of the major reasons why scanning rough wood is much more difficult than if the wood is planed. If it is sawn within the same plant and we are in control of the handling of the boards from the saw to the inspection station, we can make sure that the false defects are not added to the wood in the first place. Experience has shown, however, that it is a tough missionary work to make the workforce in a factory understand the importance of keeping the boards clean, especially as this does not seem to be a problem when the boards were inspected manually.

1.5.14 Summary

From the above survey, we can note that there are many different defects to detect. There are also large variations in appearance within each type of defect. There will always be special cases of the defects which for one reason or another do not look like they are supposed to do. The aim is of course to find methods to detect as many defects as possible, but the cost of a method to detect a certain defect should be traded for how often it occurs and the cost of the damage if it is left undetected. A 100% accurate system will never be achieved. On the other hand, the manual inspection we are replacing is not close to 100% accuracy either.

A more realistic target is to get the right balance between false positive and false negative classifications, even if this balance can be very different for different types of defects and different inspection situations depending on the economical consequence of an error.

Some of the defects can be rather easy to detect but hard to distinguish from each other, e.g. pith and pitch pockets or different types of knots. In some cases the distinction does not matter, but in other cases there can be a significant difference.

Figure 1.35 shows some of the common defects characterized by the fundamental properties shape and darkness. Not all defects can be described in this way, nor are the regions as perfectly non-overlapping as shown in Figure 1.35. Defects like blue stain, decay and compression wood do not have any characteristic shape or darkness, at least not in a grayscale image. To detect them we need textural methods or additional spectral information.

The relative frequency of various defects in different species is also of great significance, both when designing classification procedures and when evaluating the performance of a system. A study of the relative frequency of knots and splits of different sizes in Swedish pine is presented by Granstöm and Hilbers [66].

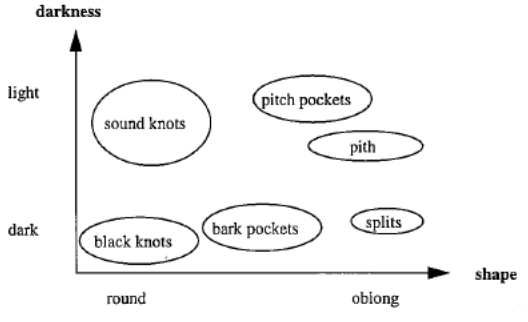


Figure 1.35 Characterization of some defects.

Chapter 2

Previous work

2.1 Introduction

A number of research groups around the world have been active in the field of automatic wood inspection. In this chapter, various image processing algorithms for surface wood scanning emanating from this research and found in the literature will be reviewed. Systems and procedures for geometrical measurement will be covered in Chapter 4. The overview in this chapter is based on publicly available articles, reports and conference papers. Very few commercial systems will be covered as the information available often is very limited and without any technical details about algorithms etc.

The aim of the image processing algorithms described in this chapter is to both detect and correctly classify various defects on the wood surface. It is important to note the difference between detection and classification. In some applications it may be necessary to detect a number of different defects, but not to determine the type. In practice, it is normally easier to tell that something abnormal is present than to exactly tell what it is. The problems arise when we have abnormal things that are not defects, e.g. dirt, pencil marks etc. It may also be the case that we can correctly classify the majority of the defects, but that two particular classes of defects are hard to distinguish. An algorithm may for example be good at detecting knots based on their shape, but unable to differ between sound and unsound knots. This may or may not be a problem. Whether an algorithm is a good or bad one depends on the situation and the classification rules of the problem.

The computational complexity of an algorithm is also very important. In Section 1.4.2 we noted that in order to scan a typical board in real time we have to be able to process data in the order of Mbytes per second. The speed requirements may vary, but nevertheless, the computational requirements of a certain algorithm is of extreme importance. This is also a reason why more general and widely used image processing techniques such as Fourier analysis and different general operator sets have not found their way into the field of wood scanning. We have also noted that the performance of a human grader is a direct function of the time given for the grading task. This is normally not the case for a wood inspection system based on a fixed algorithm. If the available time (or the available computational resources) drastically increase, completely different algorithms should probably be employed.

It would be interesting to compare the defect detection performance of the different algorithms proposed. Most references include some kind of test for a certain number of defects leading to some conclusion about the detection accuracy, e.g. by giving a percentage figure in the range 50-99%. In many cases a confusion matrix is given showing how different defects are classified. Experience has shown, however, that due to the varying conditions in different scanning situations, it is very difficult to measure the performance of a given system as well as presenting the result. Detection of false defects, incorrect size measurement (e.g. only a small fraction of a knot is detected) and cases where two different defects are incorrectly merged together can not be presented in a confusion matrix. Therefore, in this overview the focus will be on the algorithms themselves, not the reported performance figures.

Despite the difficulties, a first attempt to formulate a general test procedure to measure the performance of different wood scanning systems has been initiated by Trätek, the Swedish Institute of Wood Technology Research [107,108], even though this procedure focus more on the overall performance of a given system than its ability to detect individual defects. Work towards a further refinement of the procedure is currently in progress.

Finally, image processing techniques can be used for other purposes related to wood. In [90], Jonsson describes a system to analyze annual ring development from an image of a cross section of a board and in [126], Penman et al. describes a system for defect detection in wood fibre boards and particle board. Although being a wood inspection application, the visual appearances of the boards and the type of defects to detect are entirely different from traditional wood inspection.

2.2 Special sensors

Most systems reviewed in this chapter are based on images from traditional line scan or matrix cameras. Some of them are, however, based on special sensors where the sensor itself is an integrated part of the process. This is also the case for our own sensor system described in Chapter 3. This system is partly based on the work by Soest et al. [153] who present various sensor systems based on the interaction between the light and the wood surface. This will be further discussed in Chapter 3.

The use of X-rays for log scanning was described in Section 1.2.4, but X-rays can also be used for grading of sawn wood. Such a system is described by Portala and Ciccotelli [133] and is shown in Figure 2.1. The resolution of the system is claimed to be 0.9×0.45 mm at a feeding speed of 1 m/s. Portala and Ciccotelli also describe a microwave system, shown in Figure 2.2, based on 9.4 GHz microwaves giving a resolution of 8×8 mm at 1 m/s. A similar system is also described by Choffel et al. [38, 39]. These kinds of systems are mainly intended for grading of structural wood.

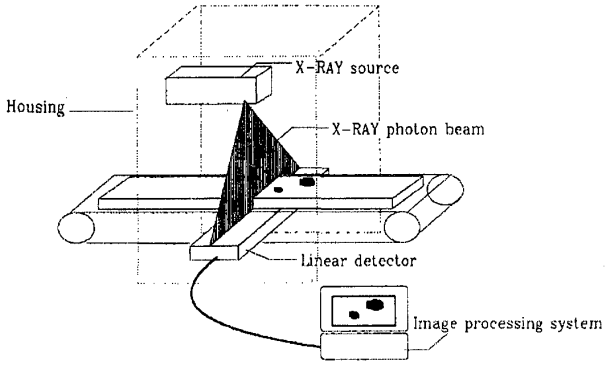


Figure 2.1 Grading using X-rays. From Portala and Ciccotelli [133].

Another interesting sensor described by Steele et al. [154] detects slope-of-grain direction by measuring the dielectric constant. A sensor is placed on a rotating 20 mm disc. As the dielectric constant is different along and across the grain, the grain angle is given by the phase

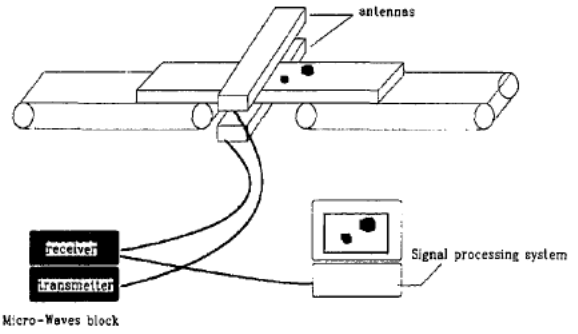


Figure 2.2 Grading using microwaves. From Portala and Ciccotelli [133].

shift of the output from the sensor. The accuracy is claimed to be ±1°. A stationary similar dielectric sensor system to detect knots and voids is described by Rice et al. [136]. An advantage of these non-optical methods is that they work perfectly well with rough and dirty lumber.

2.3 Pixelwise classification

A widely used technique in many image processing applications, e.g. remote sensing, is to directly classify each individual pixel in the image based on the output from a number of different sensors. The sensors could typically be sensitive to different wavelength bands both inside and outside the visible spectrum. The resulting feature vector for the individual pixel is then used for classification, e.g. by using standard statistical tools such as Bayes classification.

An attempt to use pixelwise classification to detect wood defects has been done by Hagman [74, 76]. By using a number of different monochromatic filters within the visible and infrared spectrum, an image stack is created. A multivariate image analysis approach is used for the classification. Basically, a number of orthogonal principal component score images are calculated as linear combinations of the input vectors. Those images are scatter plotted against each other two by two in order to find a combination with clear separation of different areas. The areas in the scatter plot are manually marked and the result is projected back on the original image in order to verify the classification. Finally, new optimal filter functions are calculated based on the manual classification.

The conclusion of the paper is that the wood surface is too monochromatic for pixelwise classification based only on the visible spectrum. Adding infrared information gives a stronger separation, but more wavelength bands should be investigated.

Experiments with pixelwise classification on Douglas-Fir veneer using different color space transformations of RGB-images and quadratic discriminant analysis has also been performed by Maristany et al. [113], but it was found that the performance of the method was not good enough to be used alone. A further evaluations of different color spaces for wood defect detection is given by Adel et al. [2], Brunner et al. [31, 32] and Maristany et al. [114] where also the physical properties of the wood surface is taken into account. A *Dichromatic Reflection Model* (DRM) is adopted where the reflected intensity is assumed to be the sum of the specular and the diffuse reflection.

The conclusion of Brunner et al. [32] is that RGB does not provide sufficient information and that an extended-dimension color system should be considered. The use of an extended number of color channels has been addressed by Vaarala et al. [163]. They use full-fledged *imaging spectroscopy* techniques to classify the pixels of the wood surface. This will be further discussed in Chapter 5.

Although some systems use pixelwise classification, additional features may be calculated for each pixel position based on the neighborhood. Ersbøll and Conradsen describe a system [54, 55, 56] where parquet floor beechwood slabs are sorted. The three features *lightness, speckle* and *dark deviation* are calculated for each pixel and its neighborhood, based on local mean, local standard deviation and local skewness respectively. From these features the pixel is classified as either clear wood or as belonging to one of 11 different defect classes. Based on the relative distribution of different defect pixels, the entire slab is classified as belonging to one of five different quality classes, from prime quality to rejects.

Note that this system classifies the entire slabs directly based on the classified pixels without attempting to segment the image and isolate the different defect areas. As pointed out before, such systems can only be used for sorting. In order to use the system for cutting, the exact positions of the defects must be known. In any case, a pure pixelwise classification system can in most cases not be regarded "stand-alone", but merely as a first data reduction and segmentation step followed by further processing.

2.4 Fixed area classification

Many of the early references found in the literature propose methods where the image is subdivided into a number of non-overlapping rectangular areas which are classified separately as either clear wood or as containing or being part of some particular defect. One advantage of this method is that it naturally gives possibilities for parallel implementations. A disadvantage is of course that the image cannot be classified at a greater resolution than the size of the rectangles. Subdividing into smaller rectangles gives better resolution but at the cost of higher computational requirements. The size of the areas must also to some extent be matched to the size of the defects to be detected. Problems arise when the rectangle contains a number of different defects simultaneously, or when a rectangle only contains a limited section of a defect.

Sobey and Semple [150] propose a classification of fixed rectangles using only tonal measures, i.e. measures from the gray level distribution. Different sizes of the rectangles are investigated and 64×64 pixels (of size 1×1 mm) is found to be satisfactory. For each such rectangle, the following four tonal measures are calculated:

1. mean

$$\mu = \sum_{l=0}^{L-1} lP(l) \tag{2.1}$$

2. variance

$$\sigma^2 = \sum_{l=0}^{L-1} (l-\mu)^2 P(l) \tag{2.2}$$

3. skewness

$$s = \sum_{l=0}^{L-1} (l-\mu)^3 P(l) / \sigma^{3/2} \tag{2.3}$$

4. kurtosis (flatness)

$$k = \sum_{l=0}^{L-1} (l-\mu)^4 P(l) / \sigma^2 \tag{2.4}$$

where $P(l)$ is the estimated probability of gray level l and L is the total number of gray levels in the image, in this case 256. The feature vector containing these four values is used in the classification step. The classes are separated by hyperplanes in the four-dimensional feature space and the article includes an iterative algorithm to find optimum hyperplanes based on a training set. The use of the four measures and their ability to discriminate the defects is discussed in the article and it is concluded that the skewness adds very little. The detection accuracy is claimed to be around 95% but it is not clear from the article exactly which defects are detected. It should be noted that the procedure only *detects* defect areas, no classification is performed. In later work [151], Sobey exploits the method with both an evaluation of the textural methods described below and proposes a method to determine the size and type of the defects present in the detected areas. This will be further described in Section 2.5.

In a widely referenced article [40] and with a summary in [41], Conners et al. propose a method similar to Sobey's but with the addition of some texture related measures using *cooccurrence matrices*. The same measures, but followed by a slightly different classification, is used by Koivo and Kim [97]. Sobey also evaluates these measures in his Ph.D. thesis [151]. Basically, a cooccurrence matrix $s(i,j,\delta,T)$ is a matrix with estimated probabilities of going from gray level i to gray level j given the displacement vector δ within the region T. Typically, the matrix is calculated for a fixed number of displacements representing different angles and distances from the pixel. Koivo uses five distances, 1, 2, 4, 8 and 16, and four angles, 0°, 45°, 90° and 135°. The result is 20 matrices $s(i,j)$ of size N×N, where N is the number of available gray levels. This is a lot of data to handle, and therefore both Conners and Koivo reduce the number of gray levels to 8, after the tonal measures are calculated, using an equal probability quantizer (EPQ) algorithm. This algorithm assigns new gray values to the image such that the probability $P(i)$ for a gray scale value i is $1/L$, where L is the number of gray levels in the reduced image. A second advantage with this procedure is that the matrices become independent from the first-order tonal measures.

Based on the calculated cooccurrence matrices, six statistical measures are calculated. Both authors use the same measures, but call some of them by different names (Koivo's in parenthesis).

1. inertia (contrast)

$$I = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 s(i,j) \tag{2.5}$$

2. cluster shade

$$A = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i+j-\mu_x-\mu_y)^3 s(i,j) \tag{2.6}$$

where

$$\mu_x = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} s(i,j) \qquad \mu_y = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} j s(i,j) \tag{2.7}$$

3. cluster prominence

$$B = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i + j - \mu_x - \mu_y)^4 s(i,j) \tag{2.8}$$

4. local homogeneity

$$L = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{1}{1 + (i-j)^2} s(i,j) \tag{2.9}$$

5. energy (angular second moment)

$$E = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} [s(i,j)]^2 \tag{2.10}$$

6. entropy

$$H = -\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} s(i,j) \log [s(i,j)] \tag{2.11}$$

Finally, Koivo also calculates correlation as

$$C = \frac{\left[\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i \cdot j) s(i,j) - \mu_x \mu_y \right]}{\sigma_x \sigma_y} \tag{2.12}$$

where

$$\sigma_x^2 = \frac{\sum_{i=0}^{L-1} \left(\sum_{j=0}^{L-1} s(i,j) - \mu_x \right)^2}{L-1} \qquad \sigma_y^2 = \frac{\sum_{j=0}^{L-1} \left(\sum_{i=0}^{L-1} s(i,j) - \mu_y \right)^2}{L-1} \tag{2.13}$$

The measures are averaged over each matrix set giving together with the tonal measures a feature vector. The size of the vector is in Conners' case ten elements and in Koivo's case nine as only the two first tonal measures are used.

In the classification stage, Conners uses a pairwise Bayes classifier. The pairwise classifier compares all combinations of classes two by two. In each class-pair decision, the most discriminating feature is used and after all pairs are considered, the sample is said to belong to the class in which it was placed most times. To select which feature to use in each test, a forward sequential search algorithm is used.

Koivo on the other hand tries to reduce the computational load by using a smaller number of features calculated as linear combinations of the original features. The new features were selected using the *scatter matrix* method. The result is a N-1 element feature vector, where N is the number of classes. Three different classifiers, a pairwise Bayes classifier, a pairwise Minimum Distance classifier and finally a tree classifier were evaluated. It is shown that the tree classifier is both more reliable and computationally less complex than the other two. The structure of the tree classifier is shown in Figure 2.3. At each node, a Bayes or MD classifier is used to select the next branch, and when a terminal node is reached, the sample is classified accordingly. Note that a single defect type may appear at several terminal nodes.

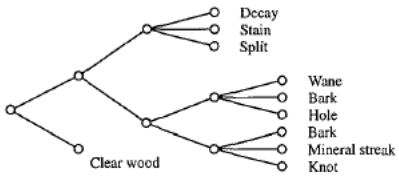


Figure 2.3 Tree classifier (Koivo and Kim [97]).

In subsequent papers [98, 99, 100], Koivo and Kim propose an alternative to the cooccurrence matrix method for texture analysis based on *Casual Autoregresive* (CAR) models. It is assumed that the image $y(s)$ obeys the following model:

$$y(s) = \theta^T z(s) + \sigma\omega(s) \tag{2.14}$$

where θ is a vector of unknown parameters and $z(s)$ is a vector specifying a number of neighbors of $y(s)$. $\omega(s)$ is a noise term and σ is an unknown parameter. This means that each pixel is assumed to be a weighted sum of a number of its neighbors and a noise term. The purpose of the method is to find estimates for the parameters $\theta_1,\theta_2,...,\theta_N$ and σ and use them as features in the classification. Two different algorithms for estimating the parameters are given in the papers.

In the practical experiments, the image is partitioned into a number of disjoint rectangles of size 64×64 pixels with resolution 0.4×0.4 mm. Three neighbors of the pixel $y(i,j)$ are used, $y(i-1,j)$, $y(i,j-1)$ and $y(i-1,j-1)$, giving four parameters to estimate for each 64×64 pixel sample. These four parameters and the sample mean form the feature vector. The scatter matrix method is used to calculate a new feature vector and the samples are classified using a tree classifier. The tree classifier shows a similar structure to the one shown in Figure 2.3, but the classification at each node is somewhat simpler. In each decision, one of the features of the transformed feature vector is compared to a threshold and one of two possible branches is chosen.

A general problem with the described rectangle classification schemes is that the pattern related measures are computationally very complex. Conners shows that if we only want to separate clear wood from defects, without classifying them, a simpler classification based only on the tonal measures may be used. Given the fact that the major part of a typical board is clear wood, a two stage classifier which first separates clear wood and then calculates the pattern related measures for the remaining samples will need much less computations. The classification used in the first stage is a standard chi-square test. Assume that the clear wood pixels are $N(\bar{\mu},\Sigma)$. If a measurement vector x is satisfying

$$(\bar{x} - \bar{\mu})^t \Sigma^{-1} (\bar{x} - \bar{\mu}) \leq T \tag{2.15}$$

where T is a threshold, the sample is considered to be clear wood.

An alternative approach to texture based classification is described by Ojala et al. [122]. It is noted that cooccurrence matrices of second-order gray level statistics are computationally too complex for practical use. Instead edge-based methods are proposed. The image of the board, in this case veneer, is partitioned into 64×64 pixel regions. A gradient operator is applied and by thresholding and thinning, a binary edge pixel image is established. A number of features are then calculated from the gradient and the edge images. The features include simple measures such as the total number of edge pixels within a region and curvature measures calculated from 3×3 regions as shown in Figure 2.4. The measure denoted C8 detects 45°

and 90° angles and the extracted feature is simply the average of the local angles within the 64×64 region. C9 is a similar measure but where each angle is weighted by the value of the mid pixel gradient magnitude G.

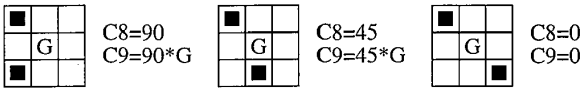


Figure 2.4 Edge-based features according to Ojala et al. [122].

In addition to the simple measures described above, a number of slightly more complex features are calculated based on *edge pairs*. Edge pairs are identified as edge points where 1) the euclidean distance is less than 10 pixels and 2) the relative orientations of the edges differ less than 45°. From each detected edge pair, the following measures are calculated:

- Euclidean distance between the edge points.
- Average gray level on the line segment between the edge points.
- Standard deviation of gray level on the line segment.
- Absolute difference between the slopes of the edges.
- Absolute difference between the contrasts of the edges.

The final features are then calculated as the mean of the measures for the entire region. These features combined with the simple features described above are then used for subsequent classification of the regions. A number of different standard classifiers such as minimum distance, Mahalanobis distance, quadratic and kNN classifiers are evaluated.

Forrer et al. [62, 63] concentrate on the problem of classifying fixed areas of the board as either clear wood or as a potential defect. The algorithms are given the name *sweep-and-mark* algorithms, meaning that they first sweep through an image gathering information and then use this information to mark defect areas. The marked areas, which only should be a small fraction of the total number of areas, can then be processed by other algorithms to determine the type of defect. Note that the second stage may determine that no defect was present and reclassify the area as clear wood. The consequence of this is that a missed defect in the first step will lead to a classification error, whereas an incorrectly marked area only will lead to a higher computational burden.

Three algorithms for defect detection in Douglas-fir veneer are proposed and evaluated. Image resolution is approximately 1×1 mm and RGB cameras are used. Different color space transformations are discussed and it is found that the best combination is a two-dimensional scheme using the intensity image (R+G+B)/3 and a color image (R-B)/2. All three algorithms work by classifying disjoint rectangular areas, 8×8 pixels in the first and third case and 16×16 in the second.

The first algorithm, called *SISM* (Statistical Image Sweep-and-Mark), calculates the usual statistical features mean, variance, skewness and kurtosis, as defined in Equation (2.1)-(2.4), for each rectangle in both the intensity and the color image. In order to classify the samples, histograms over the entire image are constructed for each feature. The histograms are subdi-

vided into a number of intervals. Let $P(k,I_k)$ be the estimated probability that feature k falls in the interval I_k. If the features are assumed to be independent, the joint probability P_{joint} of features 1,...,m falling in intervals I_1,...,I_m is

$$P_{joint} = \prod_{k=1}^{m} P(k, I_k) \tag{2.16}$$

The actual measured feature vector for each rectangle is inserted in equation (2.16) and if the resulting product is lower than a given threshold T, the area is marked as a potential defect. The underlying assumption is that the major part of the board contains clear wood, and that the areas representing low-probability events therefore contains something abnormal, e.g. a defect.

The second algorithm, called *MISM* (Morphological Image Sweep-and-Mark) uses a morphological approach [147]. The main operation is *Dilation* of the grayscale image with a binary *structuring element*. Without going into morphological details, the result is that a part of an image containing a feature with a shape similar to the shape of the structuring element will turn darker while other parts of the image will be virtually unchanged. The dilation operation is performed on each rectangular subimage using a number of line-shaped structuring elements with nine different lengths and four different orientations. The gray levels of each dilated image are summed and the different sums form a feature vector. The subsequent classification procedure is identical to the SISM algorithm.

The third algorithm, *CISM* (Color clustering Image Sweep-and-Mark), is based on cluster analysis. The mean and variance for each subimage in both the intensity and the color image are calculated. If the sum of variances from the two images for a certain area is lower than a given threshold, which means it is most certain a clear-wood area, mark it as a *core-candidate*. Associate with each area a vector of the two mean values. Within the set of core-candidate areas, find clusters with similar mean vectors. Calculate a new vector, a *core vector*, for each such cluster. We now have a set of representative mean vectors originating from typical clear-wood areas. Finally, each area is marked as potential defect if the euclidean distance between its mean vector and the nearest core vector exceeds a threshold calculated as the mean of the shortest and the largest distance between any two core vectors.

An evaluation of the algorithms is performed and it is shown that SISM generally produces the best result with acceptable detection of all defects but pitch streaks and pitch pockets. It is also shown that using SISM the number of features can be reduced to only mean and variance for the two images. In a subsequent article by Butler et al. [33], the SISM procedure is improved by using a two phase classifier. The estimated probability is compared to a threshold just like SISM, but if the area is marked, its neighbors are compared to a higher threshold and marked accordingly. This procedure works better when the areas are located at the boundary of a defect. Other improvements to the algorithm include reducing the number of intervals in the histogram, a slight low-pass filtering of the histogram and finally to use the minimum instead of the product in equation (2.16). The problem with the product is that it assumes that the features are independent, which is not always the case.

Another system which is based on fixed area classification for the initial sound wood elimination step is described by Kauppinen and Silvén [91, 92, 148]. The system uses RGB cameras. Areas of approximately 2.5×2.5 cm are classified based on features from the histograms of the three color channels using a kNN classifier. An additional line detector is used to alarm about the presence of splits, as they do not have any significant color properties. In this first step, about 90% of the regions are eliminated. Apart from separating clear and defective areas, wane edges are also detected in this fixed area classification step by using information about the position of the rectangle relative to the edge of the board. To classify the remaining defective areas, a segmentation procedure is adopted. This will be further described in Section 2.5.

A system which utilizes areas of fixed size but where the positions of the areas are established using a coarse thresholding technique is described by Lampinen and Smolander [104] and compared to the system by Kauppinen and Silvén [103]. From the 64×64 pixel subimage of the defect, responses from a number of Gabor filters are calculated. Gabor filters are orientation and frequency sensitive band pass filters given by:

$$\psi(f, \theta, x, y) = \exp\left(i\,(f_x x + f_y y) - \frac{f^2 (x^2 + y^2)}{2\sigma^2} \right) \tag{2.17}$$

where

$$f_x = f\cos\theta, \quad f_y = f\sin\theta \tag{2.18}$$

f is the central frequency of the pass band, θ is the spatial orientation and σ determines the bandwidth, and thereby the size, of the filter.

In the system by Lampinen and Smolander, eight orientations were used for each frequency giving eight feature values for each pixel in the image at each resolution level. Even though the Gabor filters give both amplitude and phase, only the amplitude response was stored. Clearly, this procedure gives an enormous amount of information. To reduce the data, an unsupervised clustering procedure based on self-organizing maps, SOM, was adopted and the final classification was performed using a multilayer perceptron network, MLP. Some alternative approaches, e.g. including color information, were also discussed.

2.5 Image segmentation based on thresholding

Most wood defect detection methods found in the literature follow the scheme of first segmenting the image into a number of regions and then classifying the regions as either a particular defect or as clear wood. Typically the regions of potential defects are disjoint separated by the large region of clear wood. The segmentation can thus be viewed as locating and classifying a number of objects on a clear wood background. The major problem of segmenting images of wood is that the clear wood background normally is anything but clear, containing grain with varying darkness and direction. A special case of the image segmentation problem is the separation of the board itself from the background of the image. As the darkness and/or the color of the background can be controlled, this can in most cases be done using simple thresholding.

It is a fact that most defects are darker than the surrounding wood, implying that thresholding can be used to segment the image. The problem is merely to find the correct threshold to use. Klinkhachorn et al. [96] describe a system where two thresholds are used, one to separate the board from the background and one to mark potential defect areas. The threshold levels are determined using a Neural Network, but it is not clear from the paper what the actual input to the network is. Once the image is segmented, various shape and location attributes are calculated for each object using another Neural Network, and those attributes are then used for classification.

In the system by Kauppinen and Silvén [91, 92, 148] described in Section 2.4, the fixed rectangle areas remaining from the sound wood elimination step are segmented using thresholding. The threshold value is selected based on the histograms of the neighboring sound wood regions. How this is done is not described in the papers. A connected component analysis is applied to isolate the blobs constituting potential defects and from the blobs, shape related as well as color related features are extracted for the subsequent classification. [91] also includes some other aspects on the system such as camera calibration and quality grading following the defect classification.

In a paper by Lee et al. [105], two algorithms for the detection of knots are described. Actually, the problem addressed in the paper is to find surface knots on entire logs with a background containing both clear wood, which is lighter than the knots, and bark which is darker. The first algorithm is based on thresholding and it is assumed that the intensities of bark, knots and clear wood are overlapping and that the gray level histograms do not show any clear valleys. The two thresholds t_l and t_h are given by the following formulas:

$$t_l = \langle x_i | min E_1(x_i), I_{kl} < x_i \leq I_{bh} \rangle \tag{2.19}$$

$$t_h = \langle x_j | min E_2(x_j), I_{wl} < x_i \leq I_{kh} \rangle \tag{2.20}$$

where

$$E_1(x) = \sum_{i=x}^{I_{bh}} P_b(i) + \sum_{j=I_{kl}}^{x} P_k(j) \tag{2.21}$$

$$E_2(x) = \sum_{i=x}^{I_{kh}} P_k(i) + \sum_{j=I_{wl}}^{x} P_w(j) \tag{2.22}$$

and I_{bh} is the highest intensity of bark, I_{wl} the lowest intensity of wood, I_{kl} and I_{kh} the lowest and highest intensities of knots and $P_b(i)$, $P_k(i)$ and $P_w(i)$ the probabilities of bark, knot and wood respectively when the intensity is i.

The binary image, obtained by the thresholding, is then subject to a region labelling process. Then, for each connected region, the following second order moments are calculated:

$$M_{20} = \frac{\Sigma_i (x_i - \bar{x})}{Area} \qquad M_{02} = \frac{\Sigma_i (y_i - \bar{y})}{Area} \tag{2.23}$$

If the area and the shape, calculated as M_{20}/M_{02}, are within given limits, the area is classified as a knot. Finally an ellipse growing technique is used to match the classified region with an ellipse and thereby calculating a more precise size and position of the knot.

In [151], Sobey extends the fixed area classification described in Section 2.4 with a procedure to segment the image within an area marked as defective. The purpose is also to merge adjacent areas belonging to the same defect. Sobey proposes two methods, of which one is based on an adaptive threshold. The threshold is adaptive in the sense that it is based on the histogram of the region in question. Basically, the histogram is low-pass filtered and then searched for peaks and valleys. A rather complicated flow diagram procedure is applied in order to find a suitable threshold depending on the structure of the histogram. The procedure is shown in Figure 2.5.

In [93, 94], Kim and Koivo add a threshold based segmentation to their fixed area classification scheme described in Section 2.4. First, in a preprocessing stage, the image is thresholded to separate background and dark defects such as knots, splits etc. from clear wood and light defects such as decay and stain. The threshold level is chosen based on the histogram of the image using discriminant analysis. Next, dark areas along the edge of the board (wane) is separated from the background by an iterative procedure based on comparison of measured variance, assuming that the variance of the background is smaller than the variance of the wane.

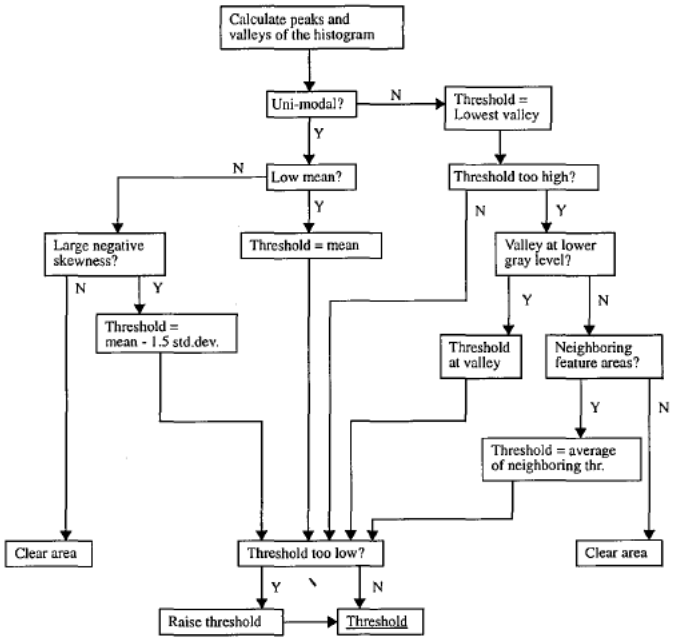


Figure 2.5 Finding threshold according to Sobey [151].

In the next step, dark areas within the board are classified as belonging to either subset *Circle* or *Line*. This is done by describing the boundary of the dark regions using a chain code. A slope histogram of the chain code is created. Assuming eight-connectivity, eight different slopes are possible with frequencies f_0-f_7. The frequencies are arranged such that $f_0 \geq f_1 \geq ... \geq f_7$. A variable Q_f is calculated as

$$Q_f = (f_0 + f_1) / \sum_{i=0}^{7} f_i \tag{2.24}$$

If $Q_f < 0.65$ the region is assigned to subset Circle, otherwise to subset Line. Within each subset, the defects are classified as knot, bark, hole, split and mineral streak respectively using CAR-models and the pairwise Bayes classifier as described in Section 2.4. To classify the light parts of the board (Subset *Texture*) the image is subdivided into non-overlapping subimages of size 64×64 pixels. These areas are classified using CAR models as described in Section 2.4, except that already classified dark defect areas within the subimages are excluded from the calculations.

In later work by Conners et al. [42, 43] the use of threshold based segmentation is adopted. In this case the segmentation is based on the two-dimensional histogram using both the red and the blue channel from an RGB camera. First the pixels are separated from the background pix-

els by a simple table lookup in a 256×256 table. Since the color of the background can be controlled, this can easily be done pixel by pixel. It is found that the best color to use for the background is blue.

During the board/background separation, the pixels within the board are collected into a two-dimensional histogram. After the entire board is processed, a thresholding is performed on the histogram to remove points having a relatively low frequency of occurrence. The resulting regions in the histogram are slightly grown to fill in small holes etc. and each region is given a label. The histogram is searched for the largest region and that region is assigned the label clear wood. All pixels not belonging to a region is assigned the label unknown. As the clear wood cluster often is connected to one or more defect regions, yet another one-dimensional histogram is created for the pixels in the clear wood cluster based on the sum of the R and B channels. If this histogram contains one or more valleys, new threshold(s) are added and the image is segmented accordingly. Next, a thinning procedure is applied to the image to remove spurious points and to label single pixels having the label unknown. Finally a connected component algorithm is used to merge labeled pixels into objects.

The different objects are classified as one of a number of defects using a simple rule-based box classification scheme. The parameters used are width, height, height/width ratio, area and average gray value. Additionally, a flag indicating whether a defect is located at the boundary of the board is used.

The algorithms are further refined in [36, 37] with an overview of the entire system in [44]. Especially is the problem of grading rough lumber addressed. The initial separation of board from background is performed exactly as above. Note that this is the only place in the system where the color information is used, the rest of the processing is entirely based on grayscale pixels derived from the color information. It is, however, the intention of the authors to make use of the color information in the future to be able to detect more defects than just the four mentioned in these papers, split, hole, wane and knot.

The board is segmented using multiple thresholds. First, a threshold is assigned for each valley in the gray level histogram. Next, the interval within the histogram containing the largest number of pixels is located. This interval corresponds to the clear wood area. After a gaussian smoothing, this interval is examined further and additional thresholds are assigned depending on the width of the interval and the behavior of the first and second derivatives. Once the thresholds are determined, they are applied to the image and all pixels having gray level values between two consecutive thresholds are marked. A connected component labeling algorithm is applied, and all regions are given a unique label. A small region elimination procedure is used to remove noise and reduce the number of regions. Especially will a rough board surface give rise to a large number of small regions due to small shadows etc.

Further region merging is performed based on a statistical T-test to find regions with similar average gray levels. Let x be the sample mean, σ^2 the sample variance and n the number of pixels in a region. If for adjacent regions

$$t_{obs} = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{2.25}$$

where

$$\sigma_p^2 = \frac{(n_1 - 1)\,\sigma_1^2 + (n_2 - 1)\,\sigma_2^2}{n_1 + n_2 - 2} \tag{2.26}$$

is less than a given threshold, the regions are merged. In the next step, a feature vector is calculated for each region. The vector contains the following parameters:

1. Area (number of pixels).
2. Average gray level within the region.
3. Center of mass

$$(\bar{r}, \bar{c}) = \left(\frac{1}{N} \sum_{(r,c) \in R} r, \frac{1}{N} \sum_{(r,c) \in R} c \right) \tag{2.27}$$

where $(r,c) \in R$ represents the coordinates of a pixel in region R and N is the number of pixels in the region.

4. Minimum bounding rectangle

$$(r_1, c_1) = (min(r), min(c)), (r, c) \in R \tag{2.28}$$

$$(r_2, c_2) = (max(r), max(c)), (r, c) \in R \tag{2.29}$$

5. Elongatedness

$$ELONG = e_1 / e_2 \tag{2.30}$$

where e_1 and e_2 are eigenvalues of the covariance matrix of the distribution of $(r-\bar{r},c-\bar{c})$ and $e_1 > e_2$.

6. Perimeter, the length of the boundary of a region.
7. Compactness

$$C = \frac{(perimeter)^2}{4\pi (area)} \tag{2.31}$$

8. A boundary flag indicating whether a region is touching board boundary or not.

The feature vector is used to calculate a confidence measure for each type of defect, i.e. a high confidence value means that the particular defect has a high probability of being of a certain type. The idea is to first find and classify "exemplary" examples of different defects. The classification is done for each region independently. In the next step, more ambiguous regions are considered using contextual dependency and similarity of region properties. To label these regions, several defect detection procedures are applied, one for each type of defect. The procedures used so far are split/check, hole, wane and knot detection. Typically, the procedures check spatial dependence, e.g. the split/check detection procedure considers regions located colinear with regions already classified as splits. If the elongatedness of the two regions merged together are higher than of each separate region, the new region is also labeled as a split/check. The defect detection procedures also try to merge regions of the same type together. Otherwise, a large region such as a large knot will typically be partitioned into a number of different regions.

Another threshold based segmentation method is described by Tatari et al. [158]. [125] contains a more detailed description of the special image processing hardware used, the so called VISTA system. Their system uses cross transport and CCD line scan cameras to scan the boards at a 1×1 mm pixel resolution. By using an arrangement with lamps placed behind the scene, open splits and holes are easily detected using a threshold as they appear much brighter than the surrounding wood. Other defects are detected using a combination of fixed and adaptive thresholds as the image is scanned on a column by column basis. As the image was acquired using cross transport, this corresponds to lines across the board direction. Very dark

defects, such as black knots, bark pockets etc. are detected by a fixed threshold. Medium dark defects such as sound knots, pitch pockets etc. are detected using a threshold which is adapting to the average gray level of the board (in the particular column) minus a certain offset. The adaptation is performed using a first order low pass filter. A sudden change in darkness giving pixel values below the threshold will give rise to a defect. Within the low intensity defect area, the updating of the threshold is temporarily inhibited.

There are two additional modules to detect some other defects. Roughness defects are detected using a separate line scan camera with an illumination placed at a certain angle from the wood surface. If the surface is rough, a lot of small shadows will appear giving a high frequency component which is detected and the area is marked accordingly. The last module detects stria caused by small dents in the cutting blade. Stria appears as narrow lines across the board. Even though the contrast is very low, the line covers the whole width of the board and is always parallel to the image columns. It can thus easily be detected by just summing the difference between the pixels along a column and its neighbors.

A connected component labeling algorithm is applied to the detected regions, and for each region the following properties are calculated:

1. Minimum and maximum gray value.
2. Mean gray value.
3. Gray value variance.
4. Width and length of the bounding rectangle.
5. Elongation (length/width ratio of bounding rectangle).

A region clustering is performed to merge regions close to each other. This is simply done using the city block distance of the bounding rectangles. The features are then updated for the merged regions. Finally a box classification scheme is used to determine the type of the defects. To reduce computation time, it is implemented as a hierarchical decision tree.

A further discussion by the same author about the textural properties of the wooden surface is given in [159] and [160]. In addition to the system described above, the following procedure is proposed to enhance the ability of the adaptive threshold to detect light defects, such as pitch pockets, without the false detection of other similar properties such as dark grain etc:

1. Scan the image linewise perpendicular to the board (grain) direction. Detect local gray value minima. If the gray level of a minimum is significantly different from the levels of the neighboring minima, mark the corresponding pixel.
2. Scan the image linewise with the adaptive threshold as described above. Mark the pixels below the threshold.
3. Perform a connected component labeling of the marked pixels from step 2.
4. Combine the labeled regions logically with the marked pixels from step 1. Regions that do not contain a marked minimum are deleted.

2.6 Other segmentation techniques

The segmentation schemes discussed so far have all been based on thresholding. Thresholding is sometimes a very drastic way to reduce information and if it is done in an improper way, significant information may be lost. An alternative procedure proposed by Westman and Alapuranen [165, 3] is entirely based on color differences, in this case euclidean distances between RGB vectors of color images. The algorithm, denoted Hierarchical Vector Connected Component (HVCC) analysis, works in two steps. In the first step, connected components are

formed containing pixels with low vector difference. If any two adjacent pixels have a difference lower than a given threshold δ, they will be connected. In the next step the resulting regions are analyzed. If the average difference along the boundary of two adjacent regions *and* the average difference between the interiors of the regions are lower than another threshold ε, the regions are merged. Typically δ is a function of ε, e.g. $\delta = 0.5\varepsilon$.

Ten geometrical, color and structural features are calculated for the regions. The features are not fully specified in the papers, but they include area, elongatedness and dimensions of minimum bounding rectangle. A combined binary tree and a kNN-classifier is used for the classification.

A slightly more complicated iterative segmentation scheme based on a pyramidal representation is proposed by Brazakovic et al. [28]. The original image I_0 is of size $2^n \times 2^n$. A pyramid is created where each level 1...*n* has half the dimension of the level below and the level *n* is a single pixel. An element at level *l*, $I_l(x,y)$, is obtained as a weighted sum of the elements of level *l*-1 in a 4×4 neighborhood. Let the node on the higher level be denoted a *father* node and the nodes on the lower level *son* nodes. The son nodes for each father node are overlapping and the purpose of the algorithm is to establish pointers from each son node to the one of four possible father nodes where it most likely belongs. Let $t_l(i,j)$ be a local image property and $s_l(i,j)$ the strength value of a link between a father node and its son nodes. The iterations proceed as follows:

1. At level l=0, set $s_0(i,j)$=1 and $t_0(i,j)=I_0(i,j)$.
2. For each level l to n-1, set

$$s_l(i,j) = \sum_{i'j'} s_{l-1}(i',j')\,\phi_{i,j,i',j'} \tag{2.32}$$

$$t_l(i,j) = \sum_{i'j'} t_{l-1}(i',j')\,\phi_{i,j,i',j'} \tag{2.33}$$

   where $\phi_{i,j,i',j'}$ is some measure of the similarity of the node values $I_l(i,j)$ and $I_{l-1}(i',j')$ in the range 0-1.
3. For each node at level l for $0 \le l < n$-2, assign a pointer $p_l(i,j)$ to the father node at level l+1 that has the highest link strength among the four candidate father nodes.
4. Recompute every node except at level 0 as

$$I_l(i,j) = \frac{t_l(i,j)}{s_l(i,j)} \qquad \text{for } s_l(i,j) > 0 \tag{2.34}$$

5. If any link was reassigned during the current iteration, go back to step 2.
6. Partition the image starting at level n-1. If the link strength between a son node and the father node pointed at by $p_l(i,j)$ exceeds a threshold τ, replace the son node by the father node.

After a merging operation of neighboring regions, the regions are classified using a three level hierarchical decision tree. Level one uses compactness (Perimeter^2/Area) to find exemplary cracks, streak knots and holes using a Bayes classifier. More ambiguous cases are classified by thresholding in the next level using area and width and the most difficult defects are classified in the last level using a measure of the variation of intensities within the defect region. The assumption is that defects like knots and streak contain fine texture, while open defects like splits and holes have an even distributed intensity. Hopefully, it will only be necessary to calculate this measure for a small fraction of the regions.

A somewhat different approach to segmentation of images of wooden boards is described by Pölzleitner [127]. The image is first subdivided into small 4×4 pixel regions. Each such region is classified as one of 16 *primitives*, which are listed in Table 2.1 together with the symbol used by the author for each primitive. The classification uses second order statistics of the gray level distribution and a linear classification scheme. The next step is based on Hough transform ideas. Each primitive in the symbol image "votes" in an accumulator array for the presence of some kind of defect. The size of the accumulator array is the same as the symbol image. Each symbol will cause a value to be added in the accumulator array in some elements within a neighborhood of the symbol's position. The accumulator elements selected within the neighborhood depend on the type of symbol, with some examples shown in Figure 2.6. The value added is dependent on the confidence of the classification of the symbol.

Symbol	Segmentation primitive
0	North-west object border
2	North-east object border
8	South-west object border
4	South-east object border
+	Interior of an object
x	Intensity peak
d	Left border
D	Right border
e	Top border
E	Bottom border
<	Vertical valley of intensity
>	Vertical peak of intensity
v	Horizontal valley of intensity
^	Horizontal peak of intensity
?	No classification
	Clear wood

Table 2.1 Segmentation primitives (Pölzleitner [127]).

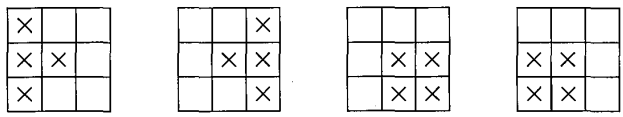


Figure 2.6 Examples of accumulation neighborhoods (Pölzleitner [127]).

In the final step, the values in the accumulator array are thresholded. If the value is above the threshold, the corresponding symbol in the symbol image is part of an object, otherwise it is classified as clear wood. The next step from here is the usual connected component labeling, calculation of features and defect classification. These steps are not covered in the paper.

In later papers by Pölzleitner and Schwingshakl [128, 129, 131], the Hough transformation segmentation is abandoned, but the 4×4 pixel symbol classification scheme is still used. Four new primitives are added and the classification of pixel areas into symbols is slightly more complicated. In the first step, each *pixel* is assigned a four dimensional vector of *association factors* using one-dimensional edge and line filters in the vertical and horizontal directions. Cooccurrence matrices, as described in Section 2.4, based on the calculated factors are used to classify 4×4 regions as symbols. Additionally, a relaxation-like procedure is used to check if the association factors within the region indicate presence of cracks. If so, the region is classified as one of the new potential crack symbols added to symbols shown in Table 2.1.

Next, a syntactic segmentation is performed using connectivity rules defined for all possible arrangements of symbols in a 2×2 neighborhood. This can be done very efficiently using lookup tables and logical AND-operations. A connected component labeling procedure is incorporated and the output is a number of distinct elliptical objects. An additional procedure to refine the boundary of an elliptical object using dynamic programming is given in [132].

The subsequent classification phase follows the usual scheme by calculating a number of geometrical properties and then using a hierarchical decision network. The properties include second order moments, elliptical approximations, orientation, elongation as well as ordinary statistics of gray scale and association factor distribution. Some properties are also calculated from the symbol image. The decision network itself is described in [130] and will be further discussed in Chapter 6.

2.7 Other approaches

Even though most algorithms described in the literature follow the scheme of object segmentation followed by classification, there are other approaches. Labeda [101] proposes a method based on neural nets with basically entire images as input. The problem focused on is the sawmill situation where entire boards are graded into one of a few classes. No attempts are made in the system to classify individual defects. Instead the overall appearance is used, which in a way can be said to mimic the work of a human grader. First, the entire image of the board is compressed to 1-8 kbyte, but the compression scheme used is not described in the paper. All pixels in the compressed image are input to a large neural network with three outputs, one for each grade. The training of the network is based on back propagation.

In a later paper [102], Labeda adopts the traditional procedure of isolating objects in the image and calculating geometrical and darkness properties. The objects themselves are not classified though. Instead, the object properties together with three image histograms (from the three color channels) are input to the neural net classifier which outputs one of three board grades.

Another approach to classification of entire boards is given by Yläkoski and Visa [168]. The image of the board is divided into non-overlapping windows (approx. 10×10 cm). For each window, a cooccurrence matrix $f_d(i,j)$ is established based on a single distance d. From the cooccurrence matrix, the features *inertia* and *entropy* are calculated as follows:

$$Inertia = \sum_{ij} [\,(i-\mu_x) - (j-\mu_y)\,]^2 f_d(i,j) \tag{2.35}$$

$$Entropy = -\sum_{ij} f_d(i,j) \log [f_d(i,j)] \tag{2.36}$$

The terms μ_x and μ_y are not explicitly defined in the paper, but can be assumed to be similar to the μ_x and μ_y used by Conners et al. and Koivo and Kim as defined in (2.7). Using these features, the window is classified (unsupervised using a self-organizing feature map) into a *texture symbol*. The symbols from the board are connected into a *symbol string* from which the entire board is classified using a supervised syntactic classifier. In the classifier, the distances to a number of *prototypes* are calculated and the board is assigned the grade of the nearest prototype. The complete procedure is shown in Figure 2.7. Note that no individual defects are classified at any stage.

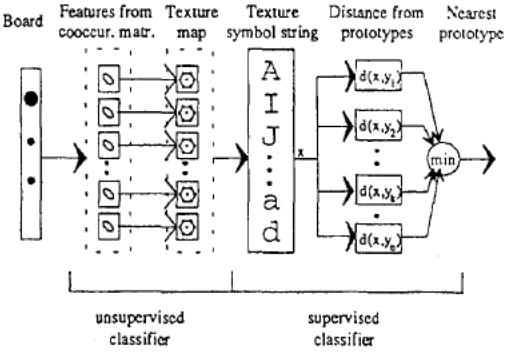


Figure 2.7 Two-stage board classifier according to Yläkoski and Visa [168].

Finally, the paper by Lee et al. [105] describes a second algorithm for knot detection (the first was described in Section 2.5). The algorithm uses a pattern recognition scheme to find knots of elliptical or circular shape. Assume that the image of the board is oriented with the grain and board edges horizontally. The steps of the algorithm are as follows:

1. Find edges in the image using 3×3 directional masks and a threshold.
2. Apply a skeleton thinning algorithm.
3. Remove all horizontal line segments. This step will delete most edges originating from grain and board edges.
4. Remove isolated lines and points. The degree of isolation is determined by counting the number of pixels within a 15×15 neighborhood.
5. Search for pixels located in slanted edge lines like '\' and '/' and curved edges like '(' and ')'. Single pixel gaps in the lines are allowed.

6. Form a one-dimensional histogram in the horizontal direction of the edge line pixels from step 5.
7. Find intervals in the histogram with high local frequency of edge pixels. Within such an interval, form a similar histogram in the vertical direction and find corresponding high frequency intervals. If the intervals are large enough, add the corresponding board area to a possible knot region list.
8. Search the possible knot region list for a number of different characteristic shapes like a pair of curved edges, three of four slanted edges in the corners etc.
9. Check the average intensity of the region to discriminate it from bark areas with similar shape.

Chapter 3

A sensor system for wood inspection

3.1 Introduction

In the following chapters, the results from the wood inspection project at the department of Electrical Engineering at Linköping University will be reported. The project was started in 1992 and has since then resulted in a number of methods and algorithms, some of which now are actually used in full production at a number of installations around the world. The project has also resulted in two patens [16, 17].

In many of the systems reviewed in Chapter 2, it is assumed that a digital image of the board, grayscale or color, is available without considering how the actual acquisition takes place. Very few authors address the fact that wood shows some very special optical properties, and that different types of illumination can give very different images from the same piece of wood.

The requirements of production environments has led this research to focus on the use of integrated sensors and processing to achieve high-speed, high-throughput, image processing. These integrated devices are called smart image sensors (or smart cameras) and are a single VLSI chip which includes both the actual sensor elements and a processing unit. Smart image sensors have been the subject of ongoing research at Linköping University since the early 1980s. For a comprehensive report on this project, see Åström [21].

The smart sensor project has also led to the foundation of the spin off company Integrated Vision Products, IVP, which markets the LAPP (Linear Array Picture Processor) and MAPP (Matrix Array Picture Processor) sensors and cameras. As will be evident from the following chapters, the MAPP2200 smart sensor is extensively used throughout our wood inspection project.

3.2 The MAPP2200 smart image sensor

3.2.1 General

The MAPP2200 chip, shown in Figure 3.1, consists of a two-dimensional 256×256 pixel sensor array, a line parallel A/D-converter for 256 pixels and 256 bit-serial processor elements (PE). The PEs work in SIMD mode, i.e. one instruction is executed by all processors simultaneously. The chip does not contain any instruction memory or any instruction sequencing logic. Instead, the 16-bit instruction words are input to the chip one by one via an instruction bus which is connected either to special hardware or directly to an I/O-port of a PC.

The processor array itself consists of a shift register S0-S7, 256×96 bits of memory R0-R95, and an ALU. The ALU consists of a Global Logic Unit (GLU) which performs global operations on the processor array, a Neighborhood Logic Unit (NLU) which performs neighborhood operations between each PE and its two nearest neighbors, and a Point Logic Unit (PLU) which performs Boolean operations within each PE. More detailed descriptions of MAPP2200 can be found in [111, 59].

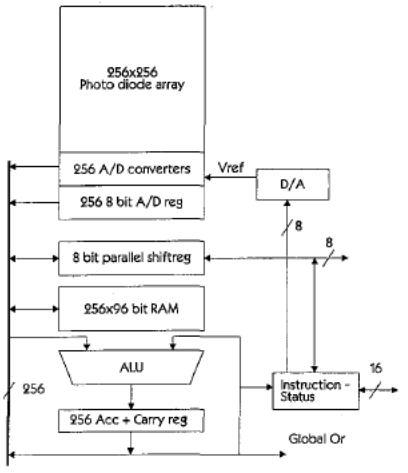


Figure 3.1 Block diagram of the MAPP2200 smart image sensor chip.

MAPP2200 is designed for an instruction clock frequency of 4 MHz. However, our experiments have shown that clock frequencies up to 8 MHz is possible on selected chips [86]. The maximum A/D converter clock frequency is 16 MHz whereas the maximum shift register clock frequency is 10 MHz.

The normal way to use MAPP for image processing applications is to read the lines of the image in a cyclic fashion one by one to the A/D-converters, convert the entire line in parallel to digital values and then let the processor array operate on the image data one line at a time. Intermediate values can be stored in the registers R0-R95 and two-dimensional operations can thus easily be implemented. In this case, the sensor array only fulfills the role of the camera in a traditional system. However, the structure of the readout logic and the A/D-converters also permits some "near-sensor" operations which can not be performed in a traditional system. These near-sensor features are extensively used in the wood inspection project.

3.2.2 Readout logic and A/D-conversion

Figure 3.2 shows the structure of MAPP's photodiode columns and readout circuit. The logic outside the dashed lines is common to all columns while the readout logic itself is repeated for all 256 columns.

The sensor is controlled by manipulating the switches, S_1, S_2 and S_3. Each sensor element in the column has its own S_1 switch and the active one is selected from the row select register, which is an eight bit register set by the SETR instruction. Since we can write arbitrary values into this register, the sensor rows can be addressed in any order and we can also choose to operate only on a limited set of sensor rows.

An exposure cycle begins by temporarily closing the selected S_1 and the two S_2 switches. The readout capacitor C and the capacitor at the selected sensor row will thus be precharged to the PD-bias level. Illuminating the photodiode, with S_1 and S_2 open, will discharge the sensor capacitor and by closing the S_1 and S_3 switches, the loss of charge in the sensor capacitor will

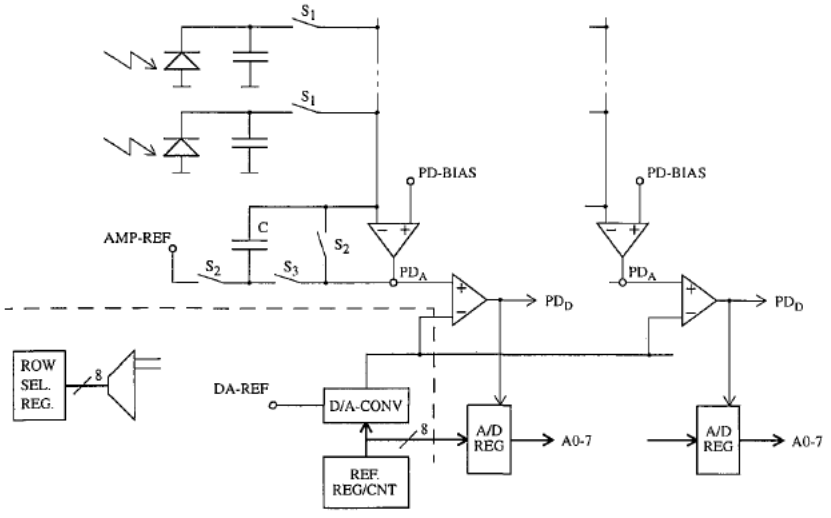


Figure 3.2 Two photodiode columns with readout circuits. The logic outside the dashed line is common to all columns.

be transferred to the capacitor C while the sensor capacitor will be precharged again. The readout is thus destructive. The result from the readout will then be available in the (analogue) register PD_A.

The conversion to a digital value can be performed in two ways. The readout logic contains a comparator to compare the acquired intensity with a global reference voltage. The reference voltage is generated by a D/A-converter from the contents of an 8-bit reference register which is set by the SETV instruction. The result from the comparison is available to the processor array in the digital one-bit register PD_D. We can thus compare the pixel intensity with a software selected threshold in parallel for all 256 processor elements. The second way to convert to a digital value is A/D-conversion. Here we let the 8-bit reference register act as a counter which will generate a ramp at the comparator input. By conditionally storing the contents of the counter into the 8-bit register A0-7 in each cycle as long as the comparator output is 1, the final result in the A-register will be the 8-bit A/D-converted value. Each column has its own 8-bit register A0-7 so that A/D-conversion is performed in parallel for all columns.

Two interesting features can be noted from Figure 3.2. First, since the readout resets the sensor capacitor, the exposure time for a given sensor row will simply be the time between two successive readouts. Since we can address the senor rows in any order and read the data at any given time, we may *control the exposure time for the different sensor rows individually.* Second, when we close the S_3 switch, the readout amplifier will effectively become an integrating circuit and by closing a number of subsequent S_1 switches one by one, i.e. by changing the contents of the row select register while keeping S_3 closed, the result will be an integration of charges from a number of sensor rows. We can thus perform a *columnwise analogue summation of intensities from any set of selected sensor rows.* These two features, individual exposure time and analogue summation, are the basis of a number of algorithms proposed in this chapter as well as in Chapter 4 on dimension measurement and Chapter 5 on color analysis.

3.2.3 The MAPP ALU

The ALU in MAPP consists of a 256-bit accumulator register and three different logic units. The Point Logic Unit, PLU, performs traditional bitwise logical operations such as AND, OR, XOR etc. Using the additional 256 bit carry register, the PLU may also perform bit serial arithmetic using instructions such as ADD, ADC etc. Other useful instructions include a MUX operation which based on the contents of the carry register selects either a bit from the accumulator or a bit from the bus.

The Neighborhood Logic Unit, NLU, performs operations on a pixel together with its two nearest neighbors. The NLU returns a one in those positions where a mask of the form (xxx) is matched where x is either 0, 1 or don't care. The NLU can do many useful things such as detecting left edges (01-), right edges (-10), isolated ones (010), shift left (--1), shift right (1--), shrink from left and right (111) etc.

The Global Logic Unit, GLU, operates on entire 256-bit vectors. The basic operation is MARK which keeps objects which are columnwise connected with a mask register, see Figure 3.3. Other instructions can be derived from this operation, such as the LFILL operation which sets all bits to the right of the leftmost 1-bit. Another important feature of the GLU is the possibility to count the number of 1-bits in the accumulator. This value is available as the COUNT field of the status register.

Figure 3.3 Some useful MAPP instructions using the NLU and GLU.

3.3 Experimental setup

In our wood inspection sensor system, six different measuring principles are integrated into one single MAPP2200 sensor chip. The setup is shown in Figure 3.4 and includes one MAPP camera and three line laser projectors. Perpendicular to the board, there are two lasers, A and B, mounted in the same plane as the camera. The lasers are aligned to illuminate two adjacent sensor rows on the wood surface. The third laser, C, is mounted at a 45° angle in the lengthwise direction of the board. An additional illumination source for color defect detection as will be described in Section 3.9 is not shown in the figure.

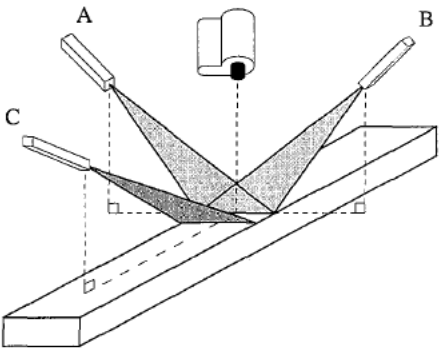


Figure 3.4 *Experimental setup containing a MAPP2200 camera and three line laser projectors. The board is fed lengthwise on a conveyor belt.*

The boards are fed lengthwise on a conveyor belt and the scanning takes place during constant movement of the board. Traditionally, one-dimensional linear array sensors are used to scan moving objects. The MAPP2200 is a two-dimensional sensor which at a first glance seems unsuitable, or at least unnecessary, for scanning the moving boards. However, in this project we do not use MAPP as a two-dimensional camera. Instead we use different sensor lines for different measurements. We thus employ the two-dimensional sensor as a multiple linear array.

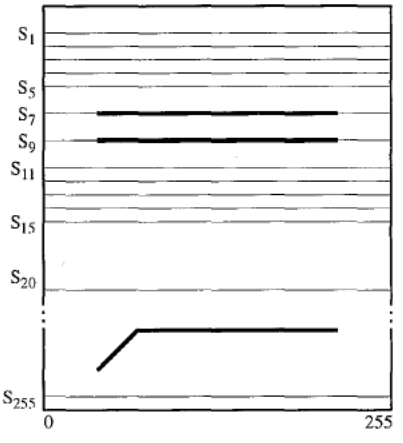


Figure 3.5 *Snapshot of the view from the MAPP camera. The lasers A and B are adjusted to illuminate sensor lines S_7 and S_9. The illumination from the third laser is illustrated by the line between sensor lines S_{20} and S_{255}.*

A snapshot of the view from the camera is shown in Figure 3.5. The two side-mounted lasers, A and B, are adjusted to illuminate sensor lines S_7 and S_9 and the illumination from the third laser, C, is illustrated by the line in the area between sensor lines S_{20} and S_{255}. In our sensor system the sensor lines are combined in the following way:

- Grayscale = $S_7 + S_9$
- Surface orientation = $S_7 - S_9$
- Tracheid effect = $\sum S_i + \sum S_j$ where $i = 1, \ldots, 5$ and $j = 11, \ldots, 15$.
- 3D-profile = Impact position within S_{20} - S_{255} for laser C.
- Grain deviation = Reflected intensity from laser C.

The different measurements will be described in more detail in the following sections. Performance figures and cycle counts for each individual measurement will be given. Total performance of the integrated system is dependent on subsequent processing and on how different operations can be interleaved. This will be further discussed in Chapter 6.

3.4 Grayscale

The sum of the two directly illuminated sensor rows, $S_7 + S_9$, gives a a normal grayscale image of the wood surface. The wavelength of the lasers is in our experimental setup 670 nm which gives a monochromatic "redscale" image. However, it has been shown [76] that normal wood surfaces give a high contrast in the red wavelength band.

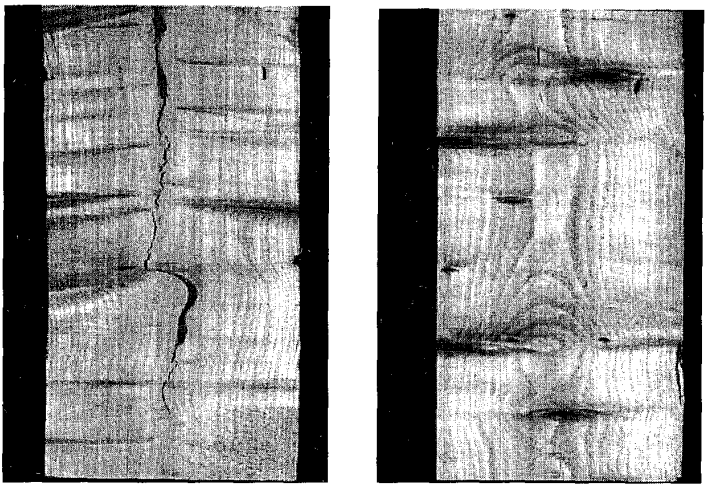
The majority of the wood defects are darker than the surrounding wood and can thus be detected by simply thresholding the image. This is the way most existing wood inspection systems work and the MAPP architecture is well suited for thresholding operations. A number of schemes to select threshold levels for wood surfaces were described in Chapter 2. By using multiple thresholds, additional darkness information can be obtained for the interesting areas while still keeping the amount of data at a low level.

Figure 3.6 shows grayscale images of two boards. The size of the images is 390×256 pixels which corresponds to a pixel resolution of 2.6×0.5 mm. In the images we can see that the black knots, the pitch pockets, the pith stripes and the splits all are darker than the surrounding wood. Some of the splits are smaller than the crosswise pixel size giving a very low contrast and will thus be difficult to detect only by thresholding. Figure 3.6a also contains some very light sound knots which are difficult to distinguish from the clear wood. Small splits and light sound knots are known to be difficult to detect in thresholding systems. Furthermore, pitch pockets and pith stripes may have very similar appearance, shape as well as darkness, and can be difficult to distinguish even though they both can be detected.

To obtain the grayscale line requires 8 instruction cycles to select and read the two sensor lines plus two A/D-conversions which takes $2 \cdot T_{AD} = 30$ µs. We also need a bit serial addition which takes $3b$ instruction cycles where b is the number of bits, normally 8. Using MAPP instruction frequency $f_P = 4$ MHz gives 8 µs for the readout and addition, but note that the addition can be interleaved with the A/D-conversion.

3.5 Wane edges/Surface roughness

Two other defects that can be difficult to detect in a grayscale image are wane edges and surface roughness. If the wane edge is covered by bark it can be detected using thresholding but if it is not, as in the upper end of Figure 3.6a, other methods must be used. In our system we take the difference between the two lines illuminated by the side lasers A and B, i.e. S_7-S_9. Depend-



a b

Figure 3.6 Grayscale images of two boards acquired as the sum of the two sidemounted lasers.

ing on the angle of the surface relative to the horizontal plane, the illumination from one of the lasers will be dominating, see Figure 3.7. Figure 3.8 shows the absolute value of the difference images and here we can clearly detect the wane edges using thresholding.

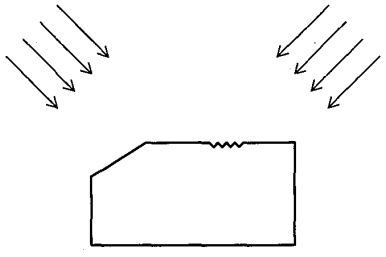
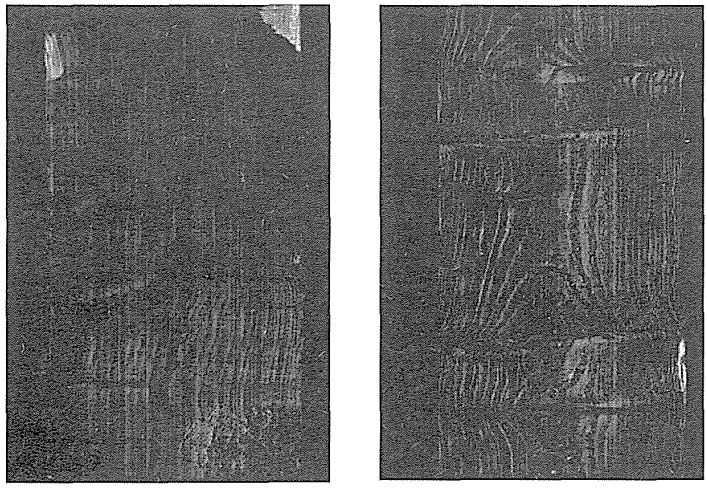


Figure 3.7 Detection of wane edges and surface roughness.

Rough (unplaned) surfaces are important to detect in applications where the wood pieces are to be glued together. Such a surface will consist of rapid variations in surface orientation and will thus give high frequency variations in the difference image. Very rough surfaces will also give a lot of small shadows which are different from the two lasers. The lower right edge of Figure 3.8a contains a rough area, although it may be difficult to detect by just studying the absolute value. Let us instead compute the gradients in the cross direction by convolving the image with a simple [1 -1] filter and then threshold the absolute value of the response. The result is shown in Figure 3.9 and here we can clearly see the rough area in the lower right edge.



a b

Figure 3.8 *Difference images of the two boards acquired as the absolute value of the difference of the two sidemounted lasers.*

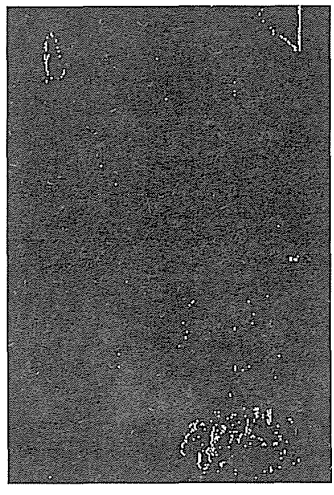


Figure 3.9 *The result from thresholding the response from a crosswise gradient filter applied on the difference image. The rough area in the lower right edge is clearly visible.*

Since the lines were A/D-converted in the grayscale measurement, we only have to add $2b$ cycles to temporary store the data, $3b$+3 cycles to perform the subtraction and $4b$-2 cycles for the absolute value. If we include the filter and the absolute value for the roughness detection, this takes $3b$+3 and $4b$-2 cycles respectively. With 8 bits, this gives 73 cycles without and 130 cycles with the roughness detection.

3.6 Tracheid effect

Light striking a wood surface will be scattered *within* the wood giving a bright region around the point of influence. This is known as the Tracheid effect (Figure 3.10) which was originally described in a patent by Matthews and Beech [117] and further developed by Soest et al. [153]. The amount of scattering is dependent on the density which is different for clear and defective wood. Hence, by measuring the amount of scattering, defective wood can be detected. A knot for instance will scatter much less light than the surrounding clear wood due to its higher density. If we can detect the amount of scattering we can thus detect knots, even if they are very bright and visually without any significant contrast to the clear wood.

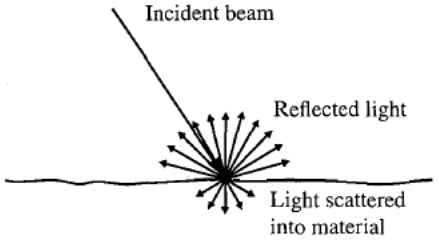


Figure 3.10 The Tracheid effect. Light striking a wood surface will be scattered within the wood. Adapted from Soest [153].

The original method required the measuring device to be in physical contact with the board or located very near the board as shown in Figure 3.11a. A similar arrangement to measure not only the amount of scattering but also the direction using a number of detectors located around the light source has been proposed by Seltman [146]. A non-contact version, shown in Figure 3.11b, is also described by Matthews and Beech [117]. The wood piece is illuminated by a laser beam through a hole in a mirror. The scattered light will be reflected by the mirror into a set of sensors located at a given offset from the optical axis.

The previous implementations of the tracheid effect have all been based on a point light source and will thus only measure the light scattering at a single point. By using a number of such sensor elements or by scanning the laser beam in the non-contact version, larger areas can be scanned but this will in both cases be expensive and mechanically complicated solutions. If we instead use a line shaped light source and restrict the measurement to scattering in the lengthwise direction using a two-dimensional sensor, the result will be a line scanning system. In our system we use the two sidemounted line laser projectors A and B as light sources and the view of the camera will therefore be as illustrated in Figure 3.12.

The amount of scattering is difficult to measure using a traditional camera for several reasons. First, from a traditional camera we will get entire images. We do not have the possibility to address only the interesting sensor lines. Second, due to the very different intensities from the scattered and reflected light, it is difficult to measure both grayscale and tracheid effect

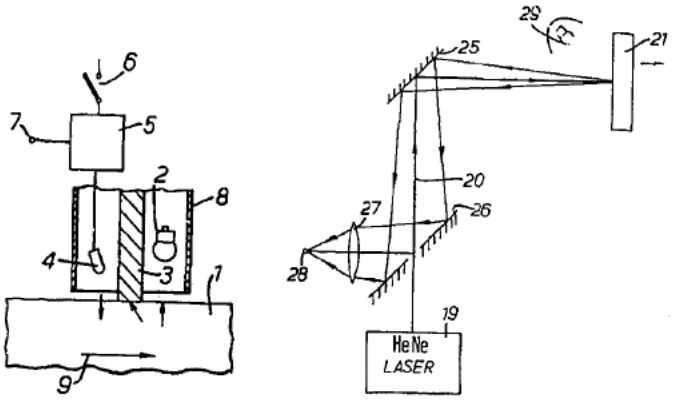


Figure 3.11 Two methods to measure the tracheid effect. From Matthews and Beech [117].

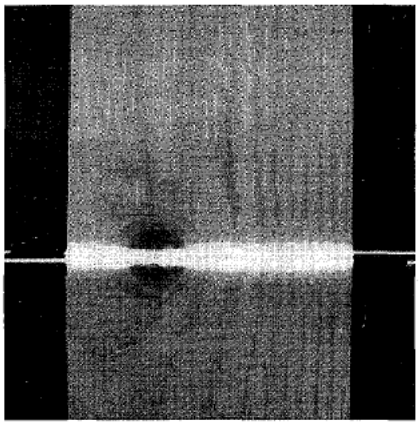
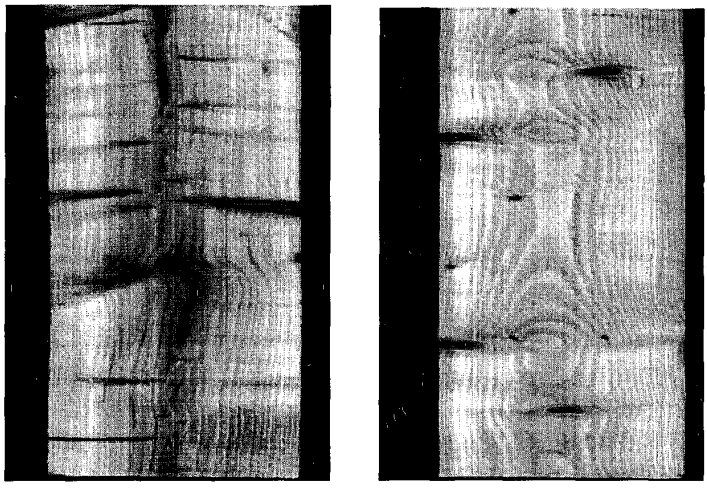


Figure 3.12 View from the MAPP camera illustrating the tracheid effect. The knot will scatter less light than the surrounding clear wood.

simultaneously. In the MAPP sensor, however, the weak scattered light can be measured by summing a number of, say five, sensor rows columnwise on each side of, but not including, the central laser lines. Here we take advantage of the possibility in the MAPP sensor to perform an analogue summation of a number of sensor lines before A/D-conversion. By summing ten lines of the weak scattered light we get an intensity in the same range as one of the directly illuminated lines. If this is not enough, we can also use longer integration time for the scattered

light, e.g. we may acquire two grayscale lines for each "tracheid line". The result will be a "tracheid image" with half the resolution in the lengthwise direction. This principle of measuring the tracheid effect using a smart sensor has resulted in a patent [16].



a b

Figure 3.13 Tracheid images of the two boards acquired as an analogue sum of five lines of each side of the central laser lines.

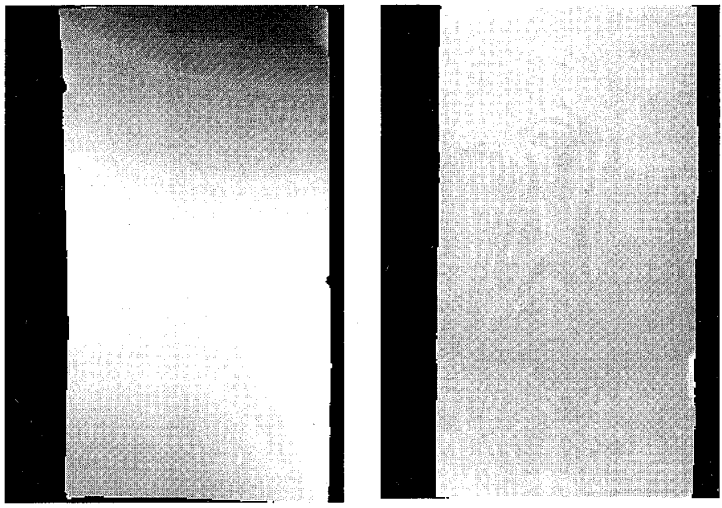
Figure 3.13 shows the results from the tracheid measurement of our two boards. The images have some very interesting properties. First, the light sound knots in Figure 3.6a appear as much darker regions in Figure 3.13a because the density is higher in the knots and they will thus scatter less light. Second, the pith stripes are dark in Figure 3.13a indicating a low amount of scattering while the pitch pockets in both Figure 3.6a and b have virtually disappeared in Figure 3.13a and b. This information can thus be used to distinguish pitch pockets from small pith stripes. Third, the splits in the grayscale images have also disappeared in the tracheid images which could simplify detection even though some of the splits will be difficult to detect because of the low contrast.

The cycle count for the tracheid measurement is n+3, where n is the number of sensor lines involved, e.g. 10. We also need an A/D-conversion requiring 15 µs.

3.7 3D-Profile

The third laser is mounted at an angle, e.g. 45°, in the lengthwise direction of the board. By detecting the position of the laser line in each column, a three dimensional profile of the board can be obtained using triangulation. This is a widely used technique known as sheet-of-light range imaging even though the ordinary video cameras normally used restrict the measuring frequency to the frame rate of the camera. When using the MAPP2200 sensor much higher speeds can be achieved. There are a number of interesting algorithms for the laser line detection and some of them will also give grayscale information simultaneously. This will be the subject of Chapter 4 and here we just use one of the algorithms described, namely the *Total*

Integration Method, TIM. This algorithm gives reasonable fast and accurate dimension information together with a measure of the reflected grayscale. Note that all 256 sensor lines can not be used for the dimension measurement since the grayscale and tracheid measurements require some 10-20 lines. In our experimental system 160 sensor lines were used.



a b

Figure 3.14 Range (dimension) images of the two boards. A brighter pixel means that the surface is located farther away from the sensor.

The range data for our two boards are shown in Figure 3.14. Brighter regions indicate that the surface is farther away from the camera. The wane edges can be seen but also the fact that board a is slightly twisted and that the surface of board b is slightly convex. The range measurements can be used to detect all kinds of dimension faults and other geometrical properties even though the difference image works just as well for detecting wane edges.

3.8 Deviating grain direction

The fifth measurement utilizes the grayscale images from both the side-mounted lasers A and B and the length-mounted laser C. It is well known [153] that light striking a wood surface is reflected differently in different directions depending on the orientations of the wood cells. This is illustrated in Figure 3.15. If we regard the wood cells as a number of silvered cylinders, it is obvious that light striking the surface from above will be reflected in many different directions crosswise due to the shape of the cylinders. By comparing the reflection in the cross- and lengthwise direction we can detect if the orientation of the cells differs from the normal lengthwise direction.

The phenomenon is further illustrated in Figure 3.16. Two pieces of wood are illuminated from a position above the camera. Note that board A is darker than board B in Figure 3.16a. By turning both pieces 90° while keeping the position of both the illumination and the camera, board A turns brighter and board B turns darker as shown in Figure 3.16b.

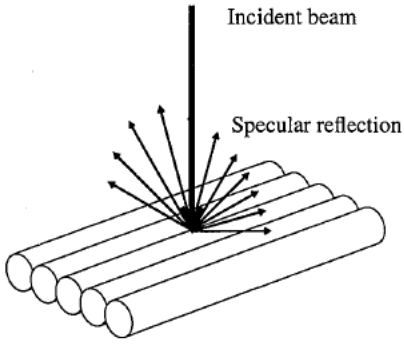


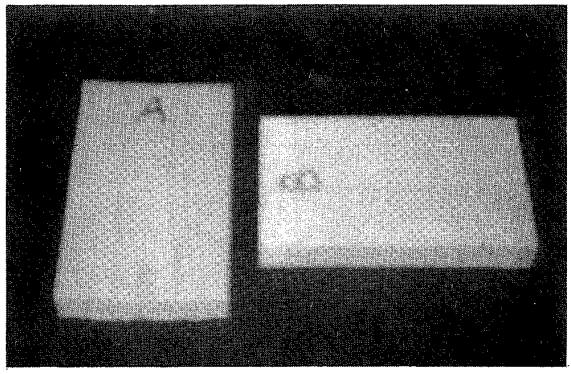
Figure 3.15 Illustration of the reflection of light on the wood surface. Adapted from Soest [153].

In our experimental setup in Figure 3.4 we place the camera on the top and compare images illuminated from the cross- and the lengthwise direction. By comparing the intensity from the grayscale measurement, i.e. from laser A and B in Figure 3.4, and the intensity information from the profile measurement using laser C, grain direction deviating from the lengthwise direction of the board can be detected. This measurement will thus be a by-product of the other measurements since no extra lasers etc. need no be added. There is a problem, however, of matching the pixels from the two measurements. In the cross direction, one line of the board maps onto one sensor line, but in the length direction the actual distance position of the illuminated pixel will be depending on the height of the surface. We thus need to use the range values in order to reorder the pixels. On the other hand, areas with deviating grain direction will typically cover fairly large sections of the board and a perfect pixel by pixel match will therefore probably not be needed.

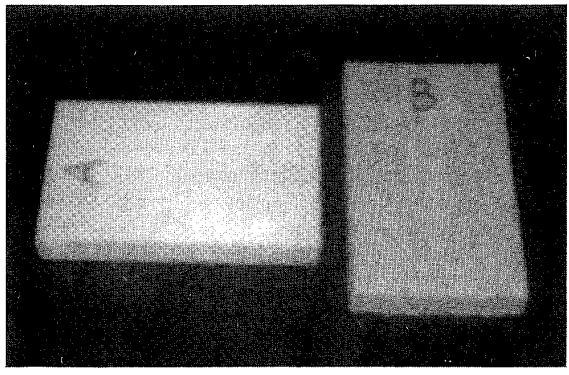
Typically, clear wood areas around knots have a deviating grain direction and this can be seen when comparing the knots in Figure 3.6 and Figure 3.17. Whether this is important to detect depends on the application but it can for example be used to indicate the presence of an "invisible" knot below the surface. Some of the dark areas in Figure 3.6a actually correspond to knots on the opposite side of the board. Alternatively, the measurement can be used to avoid dark areas around knots in the ordinary grayscale image to be included in the defect, which otherwise easily happens in a thresholding system. In Figure 3.17b we can see that the wood surrounding the knots is brighter than the clear wood while it is rather dark in Figure 3.6b.

3.9 Color defects

Some defects, e.g. blue stain, are very difficult to detect without using some kind of color information. A general discussion on the use of color for wood inspection is given in Chapter 5, and there we also propose new methods for general real time spectral classification. In this chapter we merely regard the detection of color defects as an addition to the integrated sensor system. In our setup, color can be added by reserving a few additional sensor lines for that purpose and by using colored illumination, e.g. a green laser, or white illumination and colored filters. Since both the tracheid effect and the 3D measurements are quite sensitive to



a

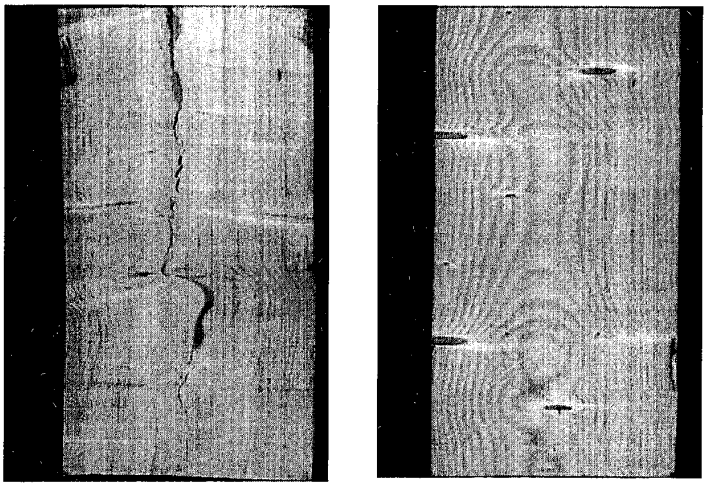


b

Figure 3.16 Illuminating from different directions gives different reflections.

background illumination, a color filter would probably be needed anyway. Without the addition of color we can use an ordinary filter in front of the lens. If we add color, a strip-type filter directly on the MAPP chip is needed where the main part of the filter is tuned to the wavelength of the main lasers while the remaining strips of the filter are tuned to the spectral properties of the defects we want to detect.

Regardless of the type of illumination, the purpose is to acquire different images corresponding to different spectral components. One component is given by the ordinary grayscale measurement so in the simplest case we only need to add one, for instance blue, component. By comparing the intensities we can detect blue color defects such as blue stain. Here we can use both a longer integration time and analogue summation of neighboring lines to increase the sensitivity of the color measurement. We can thus compensate for the poor sensitivity of the sensor in the blue wavelength band. Color defects usually cover rather large areas so the decrease in resolution for the color measurement is normally not a problem. However, due to the difficulties with the filters mentioned above, the color measurement has so far not been implemented and tested as a part of the integrated sensor system.



a b

Figure 3.17 Grayscale images acquired from the lengthwise mounted laser.

3.10 Other applications

Although primarily being intended for wood inspection applications, some of the sensor principles described in this chapter may be used in other fields. It is obvious that the profile measurement can be used in all kinds of applications where range information is needed. This will be further discussed in Chapter 4. More surprising is the fact that the tracheid measurement can be used in other fields. One example where it has been used is the inspection of adhesive tape on diapers.

The task is to detect the presence and measure the positions of the two adhesive tape pieces used for fitting the diaper on the baby. The diaper production line runs at a speed of 2-6 m/s and the positions of the tape pieces must be measured at a precision of 1 mm, which means that the real time requirements of the system are hard to meet.

The main problem is that the tape is transparent and shows little contrast to the surrounding material (Figure 3.18a). However, the internal structure of the plastic tape gives a scattering of the light when illuminated by a laser and a corresponding "tracheid" image will therefore look like Figure 3.18b. Subtracting the two images gives Figure 3.18c and here the tape area can be robustly segmented from the background using a simple thresholding. The GLU unit in MAPP is then well suited for measuring the size and position of the object.

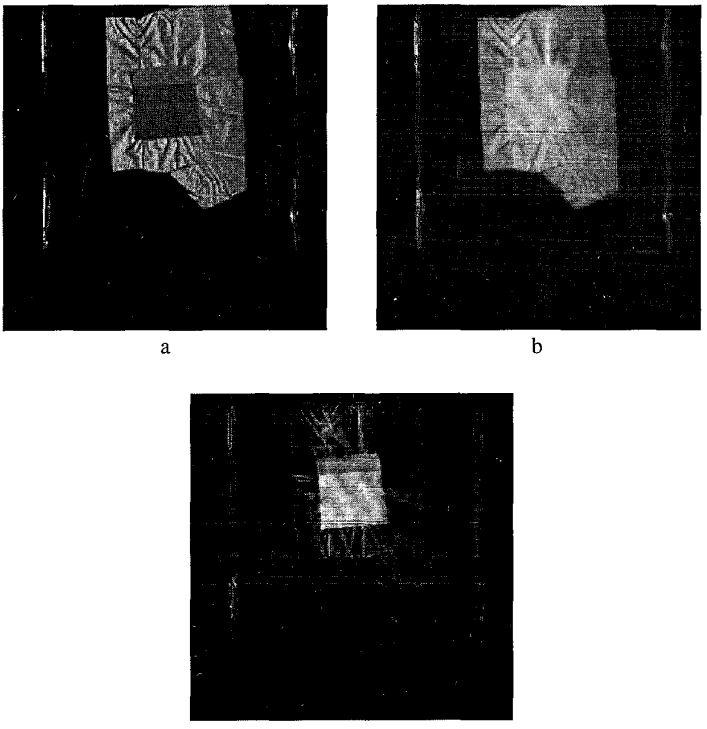


Figure 3.18 Inspection of adhesive tape on diapers based on scattering light. a) shows the grayscale image and b) the "tracheid" image of a tape piece on the diaper background. c) is the difference of the images, where the tape can be robustly segmented by thresholding.

Chapter 4

Dimension measurement

4.1 Introduction

An important task for a wood inspection system is to measure the three-dimensional shape of the wood piece. If we can acquire the full profile information of a board we can detect a number of dimensional defects such as wane edges, holes, larger splits and mechanical damage. We can also use the profile information to measure more general properties such as thickness of the board as well as bow, twist and other shape deviations as shown in Figure 4.1.

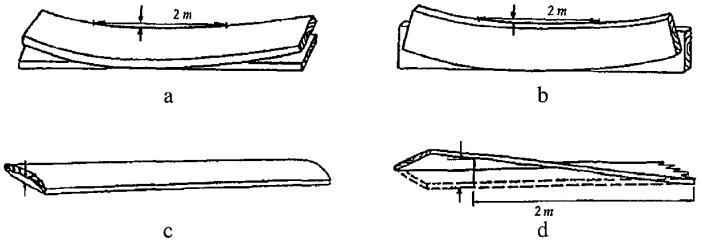


Figure 4.1 Different shape deviations [121]. a) Bow b) Edge bend c) Cup d) Twist.

We will describe methods for geometrical measurements of sawn boards. However, the same basic techniques can be also be in other applications, such as log and cant scanning.

4.2 Previous work

4.2.1 Shadow methods

One of the simplest ways to measure the dimensions of an object is to use light curtains of parallel rays, and a detector array to detect the outer dimension of an object. By combining a number of such arrays in a number of directions, the cross section dimension in several orientations may be measured. This technique is widely used in the sawmills to measure the dimension of the logs. A two-axis system described by Green [67] giving oval approximations of the log profile is shown in Figure 4.2.

Another shadow based system to measure wane edges for edging and trimming operations is the Edgar system [53] manufactured by Söderhamns Verkstäder, Sweden. The boards are fed crosswise, as shown in Figure 4.3, in front of two arrays of light emitters and detectors. The movement of the board is monitored by position encoders and by correlating the positions with the shadow detectors, both outer dimension and board thickness can be measured.

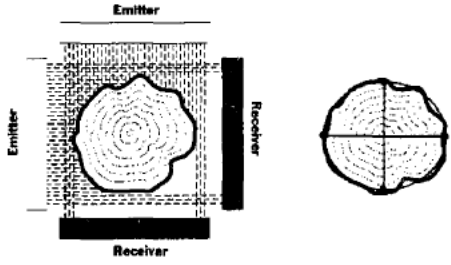


Figure 4.2 *A two-axis shadow measuring system to obtain oval approximations of the log profile. From Green [67].*

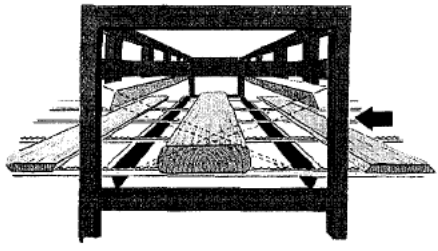


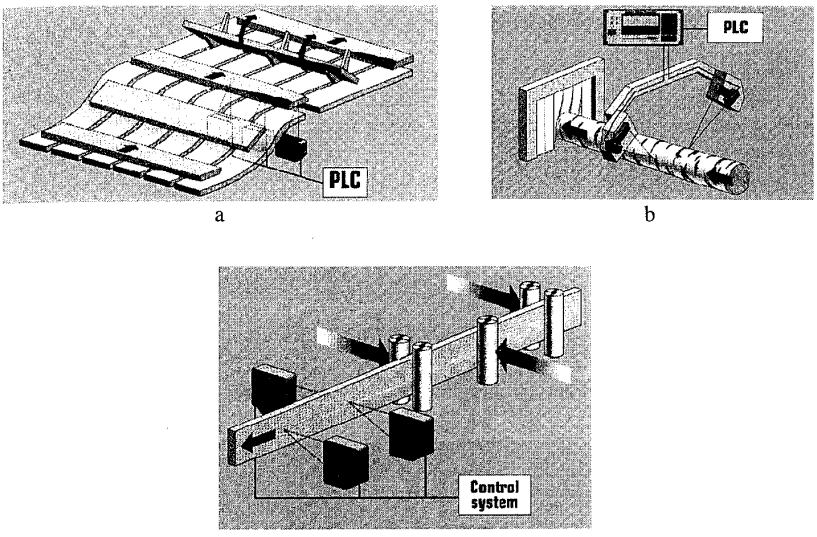
Figure 4.3 *The EDGAR system for wane edge measurements [53].*

4.2.2 Point triangulation methods

There are a number of devices on the market which measure the distance to an object in a single point. Most of them are based on triangulation where the reflected light from a laser beam is detected in a one-dimensional detector array located at a given offset from the laser. Such a device is manufactured by Selcom AB in Sweden [118] and has been used in a number of applications within the wood industry as shown in Figure 4.4.

To measure full 3D-profile we may use a number of point measuring devices configured in an array. An example of such a system is the Opcon 400 DynaVision [45] for log profile measurements shown in Figure 4.5. In this system, each scan head consists of 16 point measuring devices and by using three heads 120° apart, a full three-dimensional profile of the log can be obtained. The scanning rate is 250 scans per second which corresponds to a length resolution of 1 cm when moving the log at 2.5 m/s.

Another possibility is to use a scanning laser point in combination with triangulation detection. This principle is used in Autopos [24], another edging and trimming system by Söderhamns Verkstäder, Sweden. The system, shown in Figure 4.6, uses a scanning laser and two detectors. A parabolic mirror is used to generate a parallel scanning beam.



a b c

Figure 4.4 Three applications of the Selcom distance measuring device [118]. a) Turning wane upwards. b) Positioning logs before sawing. c) Measuring thickness and curvature for glulam beams.

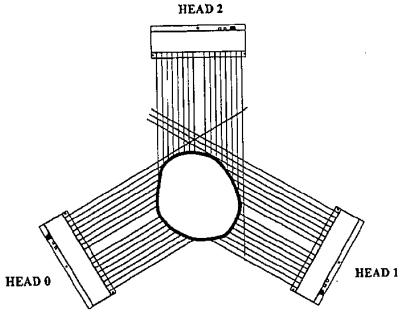


Figure 4.5 The DynaVision log scanner [45].

4.2.3 Reflection-based methods

The wane edge detector in our sensor system described in Chapter 3 utilizes the fact that the reflection from a surface is dependent on the angle between the surface and the light source. We may call this a reflection based method. A similar principle is used to detect wane edges in the BoardMaster system [138] by Finscan, Finland. The BoardMaster is a complete cross

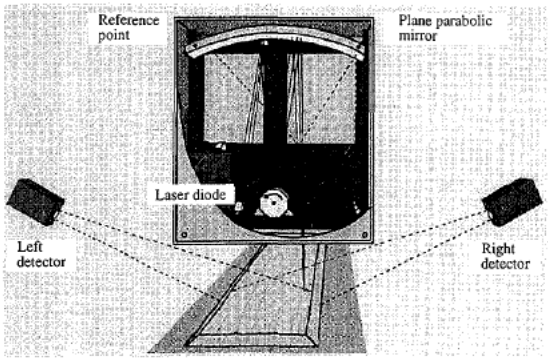


Figure 4.6 The Autopos [24] edging and trimming system.

transport wood inspection system for sawmill applications and the wane edge detector is an integrated part of the image acquisition system. The system consists of a number of line scan cameras oriented along the board which is fed in the crosswise direction. Two ramps of lamps illuminate the board from two directions as shown in Figure 4.7. The lamps are turned on and off synchronized with the exposure of the line scan cameras so that every second line is illuminated from the front light and every second line from the back light. The result will be an ordinary image of the main part of the board while the wane edges will give a line pattern which can be detected.

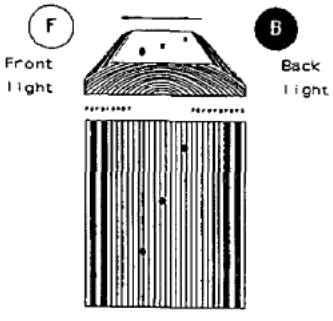


Figure 4.7 The blinking light wane detector used in the BoardMaster system [138].

4.2.4 Sheet-of-light methods

A common way to obtain a three-dimensional profile of an object is triangulation based on so called sheet-of-light (Figure 4.8). The scene is illuminated by a line laser projector and a two-dimensional image is acquired at a certain angle from the laser. By detecting the position of the

line within each column of the image, the three-dimensional profile of the object can be obtained. This technique is widely used in different applications within the wood industry for measuring logs as well as boards.

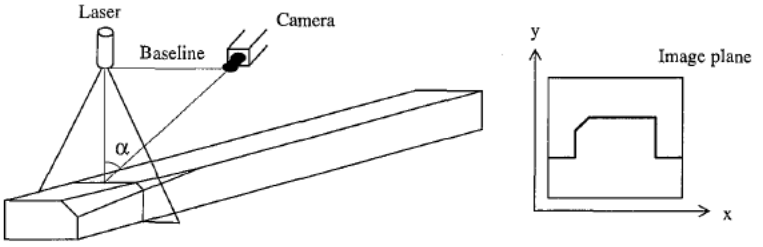


Figure 4.8 *Dimension measurement based on sheet-of-light. The scene is illuminated by a line laser projector and an image is acquired. The height profile is obtained by first detecting the y-position of the line within each column x of the image and then converting this to depth using triangulation.*

The main advantage of the method is that we get a high resolution in the crosswise direction compared to measuring isolated points. On the other hand, it is difficult to get high resolution in the lengthwise direction if we use a traditional video camera to acquire the image. We can at most calculate one line of profile data from each frame from the camera. This will restrict the scanning rate of the profile measurement to around 50 lines/s, giving a 2 cm length resolution if the board is fed at a speed of 1 m/s. However, as will be evident from the methods presented in this chapter, much higher scanning rates can be achieved using smart sensors.

There are a lot of aspects on sheet-of -light range imaging concerning calibration etc. that must be considered in an industrial system. A comprehensive report on this subject is given by Johannesson [87], and these issues will therefore not be discussed further here. The main focus in this chapter will be algorithms to detect the position of the laser line on the sensor.

There are a number of algorithms to extract the position of the line within an image, such as the position of the maximum intensity (Johannesson and Åström [86]), the position of the zero-crossing of the derivative of the intensity (Blais and Rioux [26]) or the position of the center-of-gravity (Araki at al. [6]). It is also possible to threshold the data and compute the centroid of the binary data (Johannesson and Åström [86]). The thresholding algorithm can be iterated with several different thresholds (Garcia et al. [64]) giving sub pixel measurements similar to the center-of-gravity.

A special case of a sheet-of-light range sensor is the BIRIS system described by Blais et al. [27]. The illumination is still a line laser projector but is located in the same plane as the camera as shown in Figure 4.9a. The key to the system is the special lens equipped with a double iris giving two projections onto the sensor from each point of the scene, as shown in Figure 4.9c. The distance between the two projections on the sensor, i.e. the distance between the two projected lines, can thus be used to calculate one line of range information. An application of the BIRIS sensor for log profile measurements is shown in Figure 4.9b. The limitations of one measurement per frame when using a standard video camera still remain, but variants of the smart sensor algorithms described in this chapter could probably be used if the BIRIS lens

were combined with a MAPP camera. However, the by nature very limited aperture of the BIRIS lens would probably necessitate very high laser power if the fastest MAPP algorithms are to be used.

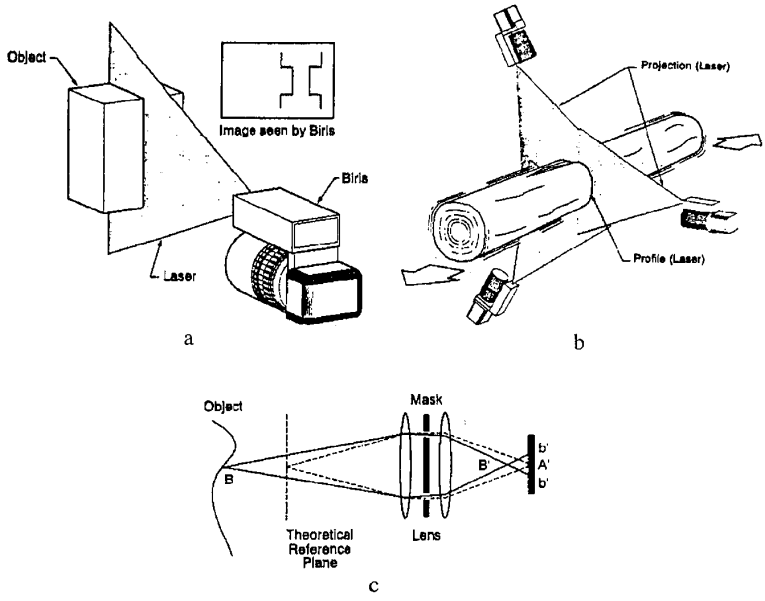


Figure 4.9 *a) The BIRIS range sensor [27]. b) An application of the BIRIS sensor for log profile measurements. c) The BIRIS lens with a double iris.*

4.3 Five near-sensor sheet-of-light algorithms using MAPP2200

4.3.1 General

There are a number of algorithms able to find the position of the laser line using the MAPP sensor. The algorithms have different performance concerning speed and range resolution. Some of them give grayscale information, concurrent and in perfect registration with the range data, while others may include different error detection capabilities.

A number of algorithms for sheet-of-light range imaging using MAPP2200 are described by Johannesson [87]. A basic algorithm is thresholding where the position is found as the centeroid of the binary data. This algorithm can be implemented both horizontally, i.e. with the laser line parallel to the processor array, and vertically with the line perpendicular to the processor array. See Figure 4.10. In the first case, each processor is responsible for finding the position within its own column, and this process is performed in parallel. In the vertical case, the sensor lines are scanned one by one and the position of the laser line is found using MAPP's powerful GLU instructions. Another vertical algorithm finds the position of the maximum value based on successive approximation using MAPP's Global Or-flag.

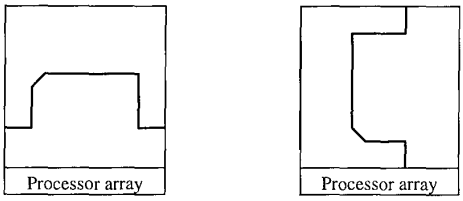


Figure 4.10 Two ways to implement sheet-of-light range imaging on MAPP. a) Horizontally b) Vertically.

The above described algorithms are conventional in the sense that the sensor lines are scanned and thresholded in a cyclic fashion and the sensor elements are subject to a constant exposure time. Johannesson [87] also describes a vertical near-sensor algorithm where the dynamic exposure process is monitored to detect the position of the maximum intensity. MAPP's Global Or-flag is used to detect when any pixel along a given sensor line reaches a given threshold level and the pixel where this occurs is the position of the maximum within that line.

In the following sections, five new MAPP2200 range finding algorithms will be described, all variants of the basic principles described above. They are all of the horizontal type where the laser line is projected on the sensor area parallel to the processor array and where each processing element finds the laser line impact position within its own column. As the number of processors is 256, this will also be the spatial width resolution of the range camera. The novelty of the new methods lies mainly in the somewhat unconventional implementations where the special MAPP features of individual exposure times for different rows and columnwise analogue summation of pixel data are extensively used. The result is a number of algorithms with drastically increased performance compared to other implementations. The algorithms described are the *Total Integration Method* (TIM), the *Odd-Even Method* (OEM), the *Forward-Backward Method* (FBM), the *Unique Sequence Method* (USM) and finally the *Fool-the-A/D-converter Method* (FAM). The algorithms are described using the syntax of MAPPC [112], which is a preprocessor enabling full interleaving of C and MAPP code and where C variables and expressions can be directly used within MAPP instructions.

4.3.2 The Total Integration Method (TIM)

This method is based on the possibility to perform analogue summation of several sensor rows and it gives both range and intensity information. After resetting the readout circuits, the sensor rows are added one by one to the readout logic. When all sensor rows have been added, an A/D-conversion is performed. The result is, as indicated by the name, a total columnwise integration of the sensor area.

The result of the A/D-conversion is the intensity of the object as illuminated by the laser line. The range information, i.e. the position of the line, is determined by continuously thresholding the sum and keeping track of at which position the integral reaches a given threshold. This process can be very efficiently implemented on MAPP2200 using a Gray code counter to keep track of the position:

```
SETV (threshold)                    /* set threshold voltage */
SETR 0                              /* start at row 0 */
RESET                               /* reset readout logic */
for (i=0; i<n; i++) {
    INTEG,+                         /* add next line */
    LD PD                           /* thr. sum, 1=still below */
    XOR R(gray(i))                  /* increment Gray counter */
    ST A,R(gray(i))
}
HOLD
SETV 0
INITAD                              /* A/D-conversion */
```

The algorithm requires only four instructions (clock cycles) in the innermost loop. The Gray code counter works by inverting a given bit position as long as the sum is below the threshold. The `gray(i)` operation gives the Gray code bit position to invert for each sensor row, i.e. from the sequence 0,1,0,2,0,1,0,3,0,1,... After the A/D-conversion, the intensity data is contained in register A0-A7 and the position (Gray coded) in register R0-R7.

Some extra instructions must be added to convert from Gray code and to move the results to the shift register. These instructions can, however, be executed concurrently with the A/D-conversion and will not cost any extra time. The output of data from the camera, i.e. via the shift register, is also transparent. Note also that even though the algorithm is described using a for-loop, an actual implementation will use straight code with the Gray bit positions hard-coded so that no speed is lost in the control structure. All in all, the execution time to extract one line of range and intensity data is

$$T_{TIM} = \frac{14+4n}{f_P} + \frac{256}{f_{AD}} \tag{4.1}$$

where n is the range resolution, i.e. the number of sensor rows used in the integration. If all 256 rows are used and assuming instruction frequency f_P=8 MHz and A/D-converter clock frequency f_{AD}=10 MHz, the total time is 155 μs corresponding to a line frequency of 6.4 kHz, or a range pixel frequency of 1.65 MHz.

Although being quite fast, TIM has an obvious drawback. The range information is intensity dependent as the width of the laser line is not taken into account. As the intensity value is available, it could be possible to make some correction to the range value afterwards but this has not been tested. TIM is best suited for applications where speed and good intensity data are essential, while exact range data is of less importance.

The range and grayscale images in the integrated sensor system described in Chapter 3 were acquired using TIM. If we compare Figure 3.14 and Figure 3.17 we can clearly see that the range values are affected by the gray levels of the board surfaces.

4.3.3 The Odd-Even Method (OEM)

The Odd-Even method is an extension to TIM where we by integrating odd and even lines separately try to overcome the drawback of intensity dependance and to extract some sub pixel range information. The algorithm works as TIM but the summation is performed in two passes, one for odd and one for even lines. In each pass, the position where the integral reaches the threshold is recorded and the integrated intensity is A/D-converted. The intensity pixel value is then simply the sum of the intensities from the two passes.

By integrating odd lines in one direction on the sensor and even lines in the other, the position of the laser line is detected from both directions, although only at half resolution. The actual range value is then given as the mid position. Note also that this procedure is capable of detecting errors, i.e. multiple peaks, if the distance between the two positions is too large. The implementation on MAPP2200 is given as follows:

```
SETV (threshold)                /* set threshold voltage */
SETR 0                          /* start at first row */
RESET                           /* reset readout logic */
INTEG                           /* start integration */
for (i=1; i<n/2; i++) {
    SETR (2*i)                  /* add next line */
    LD PD                       /* thr. sum, 1=still below */
    XOR  R(gray_even(i))        /* add to gray code counter */
    ST  A,R(gray_even(i))
}
HOLD
SETV 0
INITAD                          /* A/D-conversion */

SETV (threshold)                /* set threshold voltage */
SETR (n-1)                      /* start at last row */
RESET                           /* reset readout logic */
INTEG                           /* start integration */
for (i=n/2-2; i>=0; i--) {
    SETR (2*i+1)                /* add next line */
    LD PD                       /* thr. sum, 1=still below */
    XOR R(gray_odd(i))          /* add to gray code counter */
    ST A,R(gray_odd(i))
}
HOLD
SETV 0
INITAD                          /* A/D-conversion */
```

Some extra instructions will be needed for Gray code conversion, error detection, addition of intensity data, moving data to shift registers etc., although most of this may be performed during the A/D-conversion. The total time for one line of data will be:

$$T_{OEM} = \frac{30 + 4n}{f_P} + \frac{2 \cdot 256}{f_{AD}} \tag{4.2}$$

giving 183 µs corresponding to 5.5 kHz line frequency and 1.4 MHz pixel frequency. Note that equation (4.2) includes two full 8-bit A/D-conversions giving a 9-bit intensity result. If lower precision is accepted, the A/D-conversion can be made faster.

Sub pixel range information can be obtained by comparing the two intensity values. Given that the laser line is reasonably narrow, it is clear that one of the two sums will dominate if the center of the laser line is located in the middle of a pixel while the sums will be equal if it is located in the middle between two pixels. The odd and even sums appear as two sinusoidals with different phase as the laser line is moved on the sensor area. By detecting the phase, the position of the line can be calculated with sub pixel accuracy. In reality when implementing the algorithm on MAPP2200, line width together with several aspects of the sensor must be considered. First, the sensor aperture is not 100%, that is, it is not photo-sensitive on the whole

pixel area, and second in MAPP2200 the pixel spacing is not constant since the sensor lines are grouped two by two with buses in-between [83]. The sub pixel version of OEM has been further investigated by Johannesson [89], including a number of experiments.

4.3.4 The Forward-Backward Method (FBM)

The Forward-Backward Method is related to the sub pixel OEM in the sense that it is based on relations between exposures. First, perform a total summation of the entire sensor area in one direction, A/D-convert giving the sum S_1, and then do the same thing in the other direction giving the sum S_2. In the previous algorithms, all pixels have been subject to a constant integration time. Even though OEM used both forward and backward scans, these were performed on different pixels and the time between two successive readouts for a given pixel was always the same. Here the entire sensor will be read during both scans and only the rows in the middle of the sensor will have a constant integration time. The rows in the outer parts of the sensor area will have a short integration time during the forward scan and a long integration time during the backward scan and vice versa. As the integration time for each row is proportional to the row number, the result from the forward scan will be the sum:

$$S_1 = \sum xf(x) \tag{4.3}$$

and from the backward scan:

$$S_2 = \sum (N-x)f(x) \tag{4.4}$$

where x is the position, $f(x)$ is the true grayscale of the object at position x, and N the total number of rows. The total accumulated intensity S under the laser line is given by:

$$S = S_1 + S_2 = \sum xf(x) + \sum (N-x)f(x) = N\sum f(x) \tag{4.5}$$

The laser line position on the sensor can then be calculated as the center-of-gravity of the peak:

$$P = \frac{\sum xf(x)}{\sum f(x)} = N \cdot \frac{S_1}{S_1 + S_2} \tag{4.6}$$

which gives the position as a row number between 0 and N. FBM can be implemented on MAPP2200 in the following way:

```
RESET                          /* reset readout logic */
INTEG                          /* start integration */
for (i=0; i<n; i++) {
    SETR (i)                   /* add row i */
}
HOLD
INITAD                         /* A/D-conversion */

RESET                          /* reset readout logic */
INTEG                          /* start integration */
for (i=n-1; i>=0; i--) {
    SETR (i)                   /* add row i */
}
HOLD
INITAD                         /* A/D-conversion */
```

If the positions are to be calculated in the MAPP processor array, a bit serial division is required which takes approximately $6b^2$ cycles where b is the number of bits. To avoid this cumbersome operation, the two intensity values can be shifted out directly and the range positions can be calculated in the host computer, for example using a lookup table.

FBM has a potential for sub pixel peak detection but the limiting factor is the resolution of the A/D-converter. Since the A/D-conversion is limited to 8 bits we can only hope to achieve a range resolution in the order of 8 bits. Matlab simulations by Johannesson [89] indicate that 9 bits can be attained. However, this resolution is independent of the number of sensor rows actually used. Experiments have shown that around 8 bits of range resolution can be obtained using only 8 sensor rows. A nice consequence of this is the fact that the active sensor area can be made very short in the lengthwise direction which makes it possible to use small angles between the laser and the camera. This will be evident from the experiments in Section 4.3.5.

The FBM execution time for obtaining one range and intensity line, including additional data movements etc. is:

$$T_{FBM} = Max\left(\left(\frac{2 \cdot 2^b}{f_{AD}} + \frac{18 + 2n}{f_P}\right), \left(\frac{2 \cdot 256}{f_{SH}}\right)\right) + \frac{2b}{f_P} \tag{4.7}$$

where n is the number of rows used and b the number of bits in the A/D-conversion, normally 8. If $n<68$ then the shift register speed will be the limiting factor giving 53 µs corresponding to 18.8 kHz line frequency and 4.8 MHz pixel frequency. It is possible though to calculate a new set of range and intensity values after each scan, forward *or* backward. This will double the speed giving 37 kHz line and 9.5 MHz pixel frequency respectively, although the resulting data will be slightly low pass filtered since the two last scans are used in each calculation.

Besides the high speed, FBM is especially advantageous in situations like the integrated sensor system described in Chapter 3 where other parts of the sensor area are used for other purposes. Note that the MAPP processor array is only occupied during a small portion of the time given in equation 4.7 since the A/D-conversion and shifting can be performed transparent to other processing.

The possibility to select an active subset of the sensor area to be used in the range imaging process can be advantageous in other ways than just increasing speed. In many wood applications we have a situation where we must be able to measure within a large range interval due to the varying dimensions of the boards, but where we at a given production situation know which dimension we are scanning for the moment. Using conventional range imaging methods, we have to make a mechanical readjustment and recalibrate the sensor when changing dimension. Using MAPP2200 we can by software select to use another part of the sensor as the active area and thereby change the active range interval without any mechanical adjustments.

This procedure can be exploited even further. Consider a situation as shown in Figure 4.11 where our task is to measure the thickness of a board at high precision using two range cameras. The board may be curved, and also slightly twisted, but we know that we at any given line have range values within a limited interval.

Here we can use FBM on a limited interval, and after a given number of scans we move the active interval in the direction where the majority of the range values appear. The active interval can thus follow the contour of the board. Using this adaptive FBM procedure we can effectively get around 13 bits of range information at a pixel frequency of nearly 10 MHz, provided that all range pixels within one given line lie within a limited interval and that the range variation from line to line is limited. We will however lose one scan every time the interval is changed.

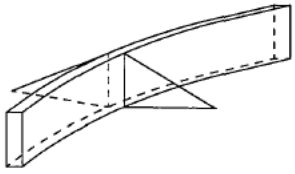


Figure 4.11 Thickness measurement of a curved board.

Finally it should be noted that to really obtain full resolution at maximum speed using FBM on only a few sensor lines, it must be possible to turn off the laser during A/D-conversion. Otherwise, there will only be a small intensity difference as all sensor rows will have integration times approximately equal to the A/D-conversion time. To turn the laser on and off is not difficult but high laser power is required since the active integration times will be extremely short. On the other hand, if the processor array is occupied by other tasks not interfering with the sensor array, the operations can be interleaved by just issuing SETR instructions at predefined intervals.

There is another way to avoid the problem if we allow ourselves a small modification to the MAPP chip as shown in Figure 5.12 on page 104. Introducing a sample and hold stage will enable us to perform readout operations concurrently with A/D-conversion. For details refer to Section 5.4.9.

4.3.5 Experiments with FBM

An experiment with FBM was performed using a piece of a window casement profile as shown in Figure 4.12. An intensity and a range image were acquired using FBM with 16 active

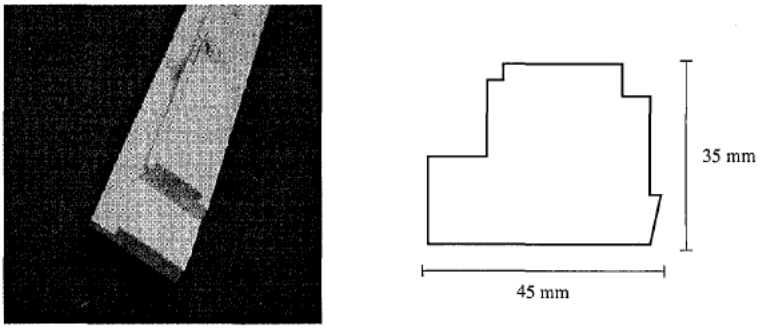
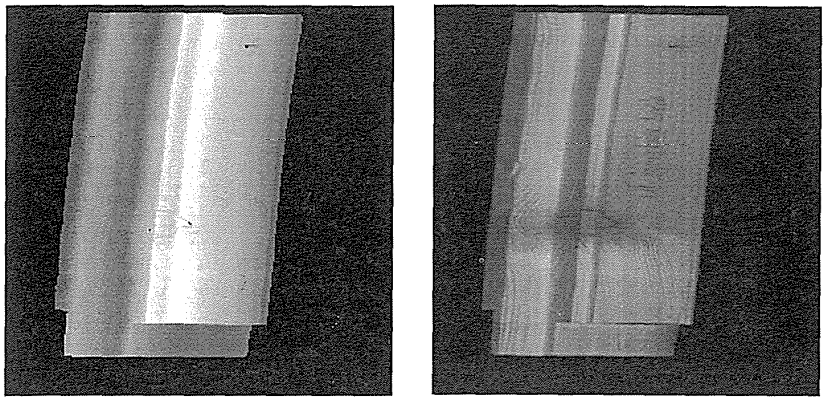


Figure 4.12 A profiled board from a window frame.

sensor rows and an 11° angle between the laser and the camera. The range image is shown in Figure 4.12a and the intensity image in Figure 4.12b. In this case the range image occupies approximately 100 range levels of the available 256 levels. Figure 4.14 shows a reconstructed view of the board where the intensity image is superimposed on the range data.



a b

Figure 4.13 a) Range image acquired using FBM. b) Intensity image from FBM. The images are based on 16 active sensor rows and 11° laser angle.

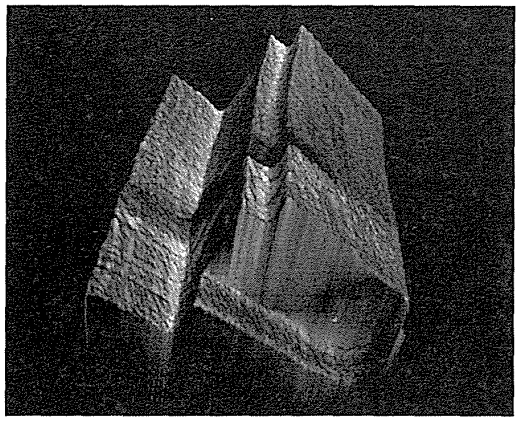


Figure 4.14 Reconstructed view with intensity image superimposed on range data.

4.3.6 The Unique Sequence Method (USM)

The Unique Sequence Method, USM, is entirely different from TIM, OEM and FBM. Functionally, it is equivalent to the basic method of thresholding each line and computing the centroid of the binary data, and where sub pixel range information can be obtained by using multiple thresholds and averaging a number of centroid positions. We can thus sacrifice speed and in return get sub pixel information. The novelty of USM is the rather unconventional implementation giving drastically increased speed performance.

To understand USM, regard the following sequence of 182 numbers, all between 0 and 15:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 0  1  3  4  6  7  9 10 12 13 15  0  2  3  5  6
 8  9 11 12 14 15  1  2  4  5  7  8 10 11 13 14
 0  1  4  5  9 12 15  2  3  6  7 10 11 14  0  4
 5  8  9 12 13  1  2  6  7 11 14 15  3  4  8  9
13  0  1  5  6 10 11 15  2  4  7  8 12 13  0  3
 5  9 10 14 15  2  6  8 11 12  0  1  7  9 13 14
 2  3  8 10 12 15  0  4  6  9 11 13  1  3  5  7
10 12 14  0  2  4  6  8 13 15  1  3  7  9 11 14
 2  4  8 10 13 15  3  5 11 12  1  2  7  8 13 14
 3  4  9 10 15  0  5  6 11 12  2  3  7  8 14 15
 4  5 10 11  0  1
```

The sequence has the following property: If we at any position within the sequence select a subsequence of 3, 4, 5, 6, 7, 8 or 9 consecutive numbers, then the combination of numbers will always be unique. That is, given a set of 3-9 numbers, this set, if it exists, will define a unique subsequence at a given position and with a given length within the sequence. Note that the *combinations* are unique, not only the *permutations*, so the set does not have to be ordered. Note also that the same number will never appear twice within a subsequence.

How can this sequence be used as a range camera? Assume that we let the numbers define bit positions in a 16-bit word. Assume also that the laser line projected on the sensor area has a width, at a given threshold level, of no less than 3 and no more than 9 pixels. The sensor area is scanned in a sequential fashion where each sensor row is thresholded giving a one in the positions where the laser line appears. By OR'ing these bits into bit positions in the 16-bit word where the bit position for each row is given by the sequence, the result will be a 16-bit code uniquely defining both the position and the width of the laser line. The procedure is illustrated in Figure 4.15.

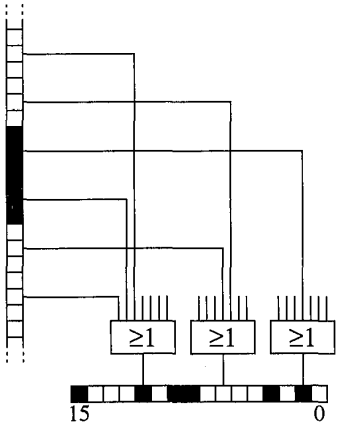


Figure 4.15 *Range finding using USM. The figure shows a sensor column and its corresponding 16-bit code word. Each bit in the code word is calculated as a logical OR of a unique set of sensor lines.*

The implementation of USM can be described as

```
int tab[182]  =  {0, 1, 2,..., 11, 0, 1};

SETR 255
for (i=0; i<182; i++) {
    RESET,+                        /* read pixel */
    INTEG
    LDI PD                         /* threshold */
    OR R(tab[i])                   /* OR into bit given by tab */
    ST R(tab[i])
}
```

After scanning the sensor area, the register stack R0-R15 contains the 16-bit codeword. The conversion to a range value is not trivial to do in the MAPP processor array, but can easily be performed using a table lookup provided that the data is shifted out from the camera. The table lookup typically gives the center position of the sequence directly with one bit of sub pixel information as the width of the line can be an odd or even number of pixels.

However, the algorithm can be more efficiently implemented. Each bit position is computed as a logical OR between a number of predefined sensor rows. As we can address ourselves freely on the MAPP sensor, there is no reason why we can not calculate one bit position at a time. We therefore convert the sequence of numbers to the following table where each line gives the positions within the sequence where each number appear, i.e. the sensor rows to be OR'ed together for each bit position:

```
 0:  0, 16, 27, 48, 62, 81, 94,106,118,131,165,180
 1:  1, 17, 38, 49, 69, 82,107,124,138,154,181
 2:  2, 28, 39, 55, 70, 88,101,112,132,144,155,170
 3:  3, 18, 29, 56, 76, 95,113,125,139,150,160,171
 4:  4, 19, 40, 50, 63, 77, 89,119,133,145,161,176
 5:  5, 30, 41, 51, 64, 83, 96,126,151,166,177
 6:  6, 20, 31, 57, 71, 84,102,120,134,167
 7:  7, 21, 42, 58, 72, 90,108,127,140,156,172
 8:  8, 32, 43, 65, 78, 91,103,114,135,146,157,173
 9:  9, 22, 33, 52, 66, 79, 97,109,121,141,162
10: 10, 23, 44, 59, 85, 98,115,128,147,163,178
11: 11, 34, 45, 60, 73, 86,104,122,142,152,168,179
12: 12, 24, 35, 53, 67, 92,105,116,129,153,169
13: 13, 25, 46, 68, 80, 93,110,123,136,148,158
14: 14, 36, 47, 61, 74, 99,111,130,143,159,174
15: 15, 26, 37, 54, 75, 87,100,117,137,149,164,175
```

The calculation of each bit position according to the table can then be implemented by performing the logical OR operation between different sensor rows as an *analogue addition* using MAPP's analogue summation feature. Recall that if the laser line fulfills the width requirement, then there can only be one sensor row giving contribution to the sum. The sum is thresholded using the original threshold and the result will be the accumulated logical OR operation. The implementation is given as:

```
int tab[16][12] = {  {0, 16, 27,..., 180},
                     {1, 17, 38,..., 181},
                                ...
                     {15, 26, 37,..., 175}};

int no_rows[16] =    {12, 11, 12,..., 12};

for (i=0; i<16; i++) {
    RESET                          /* reset readout logic */
    SETR (tab[i][0])               /* select first row */
    INTEG                          /* start integration */
    for (j=1; j<no_rows[i]; j++) {
        SETR (tab[i][j])           /* sel. rows from tab */
    }
    ST PD,R(i)                     /* thresh. and store bit i */
}
```

In this implementation we use only one instruction in the innermost loop. Note that the entire double loop can be written as straight code in a real implementation. The acquisition time for one line of range information is approximately:

$$T_{USM} = \frac{n + 16 \cdot 4}{f_P} \tag{4.8}$$

which with n=182 gives 31 µs corresponding to a line frequency of 33 kHz or a pixel frequency of 8.3 MHz. However, as we have to shift out 16 bits of data for each position, there is no way to get 8.3 Mpixels per second out from the camera. The limiting factor will thus be the bandwidth of the shift register and the line acquisition time will therefore be:

$$T_{USM} = \frac{16}{f_P} + \frac{2 \cdot 256}{f_{SH}} \tag{4.9}$$

where the first term represents the move instructions that can not be hidden by the shift operation. Using f_{SH}=10 MHz, the resulting time will be 53 µs corresponding to 18.8 kHz and 4.8 MHz line and pixel frequency respectively.

4.3.7 Sequence generation in USM

The key to the algorithm is the sequence. Many different sequences having different length, number of bit positions and minimum and maximum line widths are possible. A long sequence gives better range resolution but the sequence length is limited by both the number of bits in the codeword and the width of the laser line. It is typically the minimum width of the line that sets the limit as there are fewer possible bit combinations. The number of bits in the codeword is also important as it affects the speed performance of the method. More bits in the codeword means more data to shift out via the shift register bottleneck.

In the example above, 16 bits was a natural selection as the shift register is 8 bits wide. 16 bits can also easily be converted using a lookup table. Selecting the minimum width 3 pixels then gives the maximum length 182 and maximum width 9. The following procedure can be used to generate a sequence s_0-s_{n-1} of length n with given number of bits b and line width between d_{min} and d_{max}:

```
for i = 0 to d_max-1
    s_i=i
endfor
i = d_max
while i < n
    s_i = 0
    while not all sequences d_min - d_max in s_0 - s_i are unique
        s_i = s_i + 1
        if s_i = b then
            quit, sequence not possible
        endif
    endwhile
    i = i + 1
endwhile
```

There is no guarantee that the procedure generates optimal sequences although we believe that no significantly longer sequences will be possible for a given set of input data. Using the procedure, a number of alternative sequences were generated. The length, number of bits and minimum and maximum widths are shown in Table 4.1. The first sequence is the one used in our example. The second sequence is an attempt to use the entire sensor area. As a consequence, the minimum possible width is 5 pixels. On the other hand, the maximum width is now 11 instead of 9. The last two sequences are based on 24 bits instead of 16. Here we can use widths down to 2 pixels.

Length	Bits	Min width	Max width
182	16	3	9
256	16	5	11
256	24	2	9
232	24	2	11

Table 4.1 Four possible USM-sequences.

Using 24 bits instead of 16 will decrease performance as three bytes per pixel must be shifted out from the camera. However, the actual information contents is far from 24 bits. In the third case in Table 4.1, 256 positions and 7 different widths give 1792 position-width combinations, minus a few as not all widths are meaningful at all positions. If we could encode the 24 bit word using fewer bits we could retain the high speed. Theoretically, 11 and 12 bits respectively are needed to encode all possible position-width combinations in the two last cases in Table 4.1, but as only 512 range values are possible after the final table lookup, the actual information contents is only 9 bits. However, for all practical purposes, if we could find a 16 bit code we would be satisfied. The problem is that the conversion must be performed in the MAPP processor array using a limited number of instructions to be meaningful. We have so far not managed to find a solution to this channel coding problem.

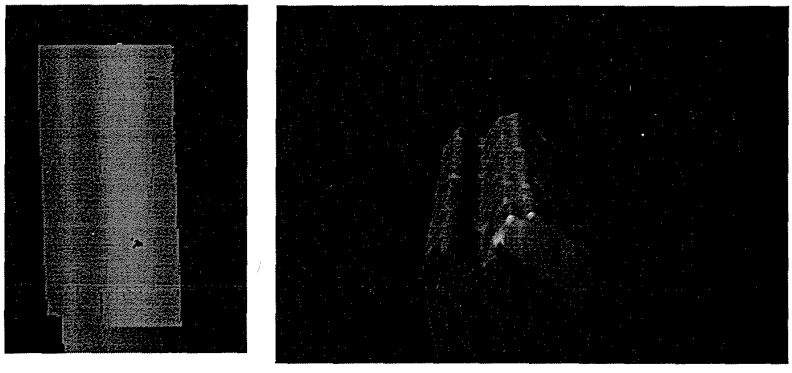
Another approach is to use some kind of hash procedure. If we shift the 24 bits out from the camera, this would be the natural choice, at least if we can not accept a full table lookup which would require 16 million entries in the table. However, if we want to reduce the amount of data to shift out, the hash function must be calculated within the MAPP processor array. A hash

function based on XOR operations on different bit planes can easily be implemented, but there is no rational way of handling collisions because of the SIMD structure. If any column gives a collision, then additional information must be obtained and shifted out from all 256 columns. If the collisions do not occur too frequently, this could still give better average performance than shifting out 24 bits every time. Another idea would be to use neighboring pixels and/or previous lines to resolve collisions given that the position-width combinations that give collisions are reasonably different. By switching between different hash functions, we can probably make the collisions appear at different places from line to line. This is just speculations though and requires further research.

Finally it should be noted that USM has a limited error detection capability. If multiple peaks appear on the sensor, e.g. if the laser line illuminates an edge in the scene, there is a rather high probability that the algorithm generates an invalid codeword which can be detected using a suitable entry in the lookup table. There is no guarantee that an error actually gives such a codeword, and if not, the result may give a more or less random appearance. However, errors where the line is too narrow, e.g. if the object is dark, will always be detected as the number of one bits in the codeword always is equal to the width of the line, and the resulting codewords can therefore never correspond to valid position-width combinations.

4.3.8 Experiments with USM

The same window profile as in Section 4.3.5 was used to acquire an image using USM. The 182-number sequence was used and the angle between the laser and the camera was approximately 45°. With this geometry, the interesting part of the board only occupied 30 levels out of the available 182. To reduce table size, the sub pixel bit was not used. Figure 4.16a shows the range image and Figure 4.16b a reconstructed view. Note that there are some missing data in the range image at dark positions. This is due to the fact that the width of the laser line at the given threshold level will be less than the three pixels required. Increasing the aperture or exposure time or lowering the threshold will make the line too wide at the brightest sections. The conclusion is that USM requires a narrow laser line or can only be used in situations where the intensity variations of the objects are limited.



a b

Figure 4.16 a) Range image acquired using USM. b) Reconstructed view. The images are based on 182 active sensor rows and 45° laser angle.

4.3.9 The Fool-the-A/D-converter Method (FAM)

The fifth method, the Fool-the-A/D-converter method, is perhaps of less theoretical interest but it is interesting from an engineering point of view as it utilizes some unintended features of the MAPP chip. As indicated by the strange name, the idea is to fool the internal A/D-converter to think it is performing an A/D-conversion while it actually is detecting the position of the laser line. The method requires some hardware modifications to the MAPP2200 camera and has therefore not been tested.

To understand the method, we must first take a closer look at the internal A/D-converter in MAPP2200 shown in Figure 4.17. The reference voltage to the comparator is generated by an internal D/A-converter from an 8-bit register. This register can either be set directly using the SETV instruction or it can act as a counter in which case a ramp will be generated at the comparator input. The output of the comparator can be read directly via the PD-register, but it also controls an 8-bit register A0-A7. During an A/D-conversion, initiated by the INITAD instruction, the contents of the counter is at each clock cycle stored in register A0-A7 as long as the input voltage is higher than the ramp. When the ramp reaches the input value, the storage will be inhibited and the last value will remain in the register. Thus, after 256 clock cycles A0-A7 contain the A/D-converted value. Note that the A/D-converter is controlled by a separate clock asynchronously from the instruction clock. For more details, refer to [59].

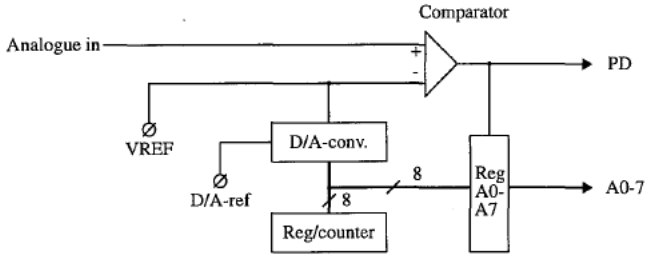


Figure 4.17 The A/D-converter in MAPP2200.

When the MAPP chip was designed, it was assumed that the precision of the internal D/A-converter might not be good enough for all applications. Therefore, an input pin, VREF, was added, and the chip is thus designed so that the external voltage can override the internal D/A-converter [84]. Now, let us connect a stabilized external voltage to the VREF pin. Furthermore, let the A/D clock f_{AD} be synchronized with the instruction clock f_P and with $f_{AD} = f_P/2$, i.e. with one A/D clock cycle for every second instruction written to MAPP2200. Then execute the following MAPP program:

```
INITAD
for (i=0; i<256; i++) {
    RESET,+                    /* reset readout logic */
    INTEG                      /* read value*/
}
```

First an A/D-conversion is initiated and then the sensor rows are read one by one and are thresholded using the external voltage. If the sensor intensity is larger than the external voltage, i.e. if the laser line illuminates the pixel, the A/D-converter thinks that the pixel value is

higher than the "ramp value" and the counter is stored in register A0-A7. The result is that after the "A/D-conversion" the last position where the laser line is above the threshold level is stored in A0-A7.

Adding 8 instructions to move register A0-A7 to S0-S7 gives the total time for one line:

$$T_{FAM} = \frac{2n+9}{f_P} \tag{4.10}$$

which using 256 lines gives 65 µs corresponding to 15 kHz line frequency and 3.9 MHz pixel frequency.

Since the width of the line is not taken into account, FAM is intensity dependent in the same way as TIM. It would however be possible to use the OEM procedure to scan even lines forward and odd lines backward but this is impossible to do with only two instructions in the innermost loop. The main merit of FAM compared to the other methods is that it is the fastest method which gives the range values directly in the camera without further processing, such as a table lookup, outside the MAPP chip.

Finally, note that if the enable bit to the A0-7 registers could be inverted, i.e. so that the counter is stored if the analogue signal is *lower* than the reference voltage, then we could implement TIM directly in hardware using the FAM method with only one instruction in the innermost loop. After range data acquisition we could then turn off the external reference voltage and perform a normal A/D-conversion to obtain intensity data.

4.3.10 Thresholding range data

If the purpose of the range imaging is to perform a range thresholding, e.g. in an inspection situation where we just want to detect dimensional defects on a flat surface, then simplified versions of the methods can be used. Range thresholding in TIM can for example be implemented as:

```
SETV thr
SETR  0
RESET
for (i=0; i<T; i++) {          /* integrate line 0 to T */
    INTEG,+
}
LD PD                          /* threshold integral */
for (i=T+1; i<n; i++) {        /* integrate line T+1 to n */
    INTEG,+
}
HOLD
SETV 0
INITAD                         /* initiate A/D-conversion */
```

which is much faster than the original TIM. The range threshold is given by the line T and the LD PD instruction thresholds the integral to detect which columns have reached the laser line before sensor line T. A similar simplification of OEM is also possible.

Range thresholding in FBM can also be simplified as we only need to compare the relation between the two intensities with a fixed threshold level, and can thus use a multiplication instead of a division:

$$\frac{I_1}{I_2} < T \Leftrightarrow I_1 < I_2 T \tag{4.11}$$

Note that we do not normalize the range values in this case. If the threshold can be selected exactly at a sensor row, then we can simply use FBM symmetrically around that row and just compare the two intensities. If the range threshold is to be located between two sensor rows, we would still use a sensor area approximately symmetrical around the threshold row and the threshold T (i.e. the multiplication factor) will thereby be close to 1.

Another possibility is to insert a short delay (with the laser turned on) in one end between the forward and backward scanning. We can thus let the backward integration time be equal to the forward integration time plus an offset T_O. The idea is to move the position where the two integration times are equal so that the thresholding can be performed by just comparing I_1 and I_2, despite the fact that the threshold level is not located exactly at a sensor row. Unfortunately, due to the physical width of the laser line the exact threshold position will be intensity dependent so the method can only be used if the total intensity of the object is fairly constant. The procedure is illustrated in Figure 4.18. The lines indicate when the different sensor rows are read. Without a delay, the position where the integration times are equal (solid lines) is located at sensor row $N/2$. If a delay is inserted (dashed lines), this position will be moved towards a lower sensor row number.

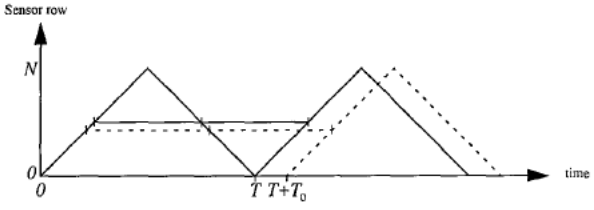


Figure 4.18 Moving the FBM threshold position (i.e. where the integration times are equal) by inserting a delay T_0 between the forward and backward scan.

4.4 Conclusions

We have presented five novel high speed sheet-of-light range imaging implementations on the smart image sensor MAPP2200. The achieved range image speeds with pixel frequencies 1-10 MHz are very good compared to other smart sensor designs presented by Araki et. al. [6], Verbeek et.al [164], Gruss et. al. [73] and the more conventional MAPP2200 implementations described in [87, 88]. The performance of the different methods are summarized in Table 4.2.

From Table 4.2 it seems that FBM is outstanding in all aspects except error detection. FBM also has the additional advantages of only occupying a limited sensor area and not occupying the MAPP processor array. This makes it perfect to combine with other measurement principles as in the integrated sensor system described in Chapter 3. Another advantage is the fact that FBM performs well using limited laser angles. If we want the range resolution to be much higher than the spatial cross resolution, high laser angles must be used in most other sheet-of-light methods due to the geometry of the sensor, and high laser angles will give other problems with occlusion etc. If we compare the USM experiments in Section 4.3.8 to the FBM experiments in Section 4.3.5, we can clearly see how difficult it is to really make use of the full range resolution using other methods than FBM. The conclusion is that FBM seems to be the method of choice for wood inspection applications, at least if intensity data is desired. If we do not need intensity data, traditional thresholding methods or USM with a narrow laser have the advantage of being less sensitive to overexposure of the sensor and may therefore more easily handle darkness variations of the boards.

Method	Line width (pixels)	Pixel freq.[1] (MHz)	Line freq.[1] (kHz)	Range res. (bits)	Intensity data	Error detect.
Araki PSD 128	128	0.5	4	12[2]	Yes	No
Verbeek PSD96	96	2.0	20	8[2]	Yes	No
Gruss 28*32[3]	32	0.9	28	8	No[4]	No
MAPP vert. succ. appr.	1-256[5]	0.16	≥0.62[5]	9	Yes	Yes
MAPP hor. threshold	256	1.6	6.2	7[9]	Yes	Yes
TIM	256	1.6	6.4	8[6]	Yes	No
OEM	256	1.4	5.5	8-11[7]	Yes	Yes
FBM	256	9.5	37	8-13[8]	Yes	No
USM	256	4.8	19	8-9[9]	No	Yes
FAM	256	3.9	15	8[6]	No	No

Table 4.2 Summary of the performance of the different methods.

Notes: 1) Including intensity data where applicable.
2) A/D-converter precision. Reported accuracy is 0.3% and 1% respectively.
3) 2D range image on chip.
4) Intensity readout in presented design but not in prototype.
5) Programmable width.
6) Intensity dependent
7) Estimated value
8) Sub pixel resolution only in limited intervals.
9) Sub pixel resolution possible at reduced speed using multiple thresholds.

Since the performance of different methods are set by different limits, i.e. some are limited by the instruction frequency while others are limited by A/D-converter speed or shift register bandwidth, other methods than FBM may turn out to be advantageous in newer versions of MAPP. The proposed methods could also be an inspiration for range image implementations on other sensor designs.

There are also a number of aspects of the proposed methods that require further investigation. Given below are a number of possible future research directions.

- Compensation of the intensity dependence in TIM using the acquired intensity value.
- Range image acquisition of moving objects. So far we have only considered stationary or slowly moving objects. The effect of intensity variations during a scan should be investigated further. In FBM it is clear that intensity gradients will give range errors if the object move between the forward and backward scan. In USM the sensor lines are addressed in a rather complicated order and the errors due to object movement may therefore be difficult to predict.
- Noise sensitivity in all algorithms.
- Procedures to find optimal (maximum length) sequences in USM.
- Encoding of 24-bit codewords in USM.
- Hash procedures in USM, both inside and outside MAPP.
- Evaluate the error detection capability in USM.

Chapter 5

Color analysis

5.1 Introduction

Many of the systems reviewed in Chapter 2 utilized color information in various forms as an aid in the detection of the defects. Some of them relied entirely on color information while others, including our own sensor system in Chapter 3, merely used color to detect some specific defects, such as blue stain.

The conclusion from Chapter 2 was that RGB cameras are insufficient to capture all significant spectral features of the wood surface and that an extended number of color channels should be considered. This means that a spectrographic approach should be adopted, and this is also the conclusion of Marszalec and Pietikainen [115] and Hagman [77]. However, the use of a spectrograph for on-line wood inspection has so far failed to meet industrial requirements, both concerning the cost of the equipment and the speed. A multi-color sensor system in combination with a reasonable spatial resolution will result in an enormous amount of data and massive processing resources seem to be needed to classify the pixels. A possible solution to this problem is the PGP-MAPP spectrograph presented in this chapter. By using an imaging spectrograph in combination with a smart image sensor, real time pixelwise spectral classification is obtained at a rather moderate cost.

The system presented in this chapter is a tool in which any linearly weighted spectral filter function can be implemented in real time. The focus is on the tool itself, whereas less efforts are spent on optimizing the filter functions for the wood inspection problem. The latter subject is a major research field of its own and the interested reader is referred to Hagman [77], Marszalec [115] and Vaarala [163]. Some limited wood defect classification experiments are included in this chapter but we are eagerly awaiting the results from the above mentioned researchers in order to really verify the potential of the PGP-MAPP spectrograph for wood defect detection.

As a general spectral classification tool, the PGP-MAPP spectrograph could be employed for other applications as well. One example from the medical field is given at the end of this chapter.

5.2 Spectral properties of wood defects

An investigation of the spectral properties of different defects in pine wood has been performed by Marszalec and Pietikainen [115]. Some examples of reflectance spectra are shown in Figure 5.1 for a) clear wood, b) sound knots, c) resin pockets and d) blue stain. The two curves represent dark and light areas within each sample.

In [115], the non-homogenous nature of many of the defects as well as the clear wood areas is observed. This means that a given type of defect may show very different spectral characteristics depending on where it is measured. The same observation is made by Hagman [77] indicating that the underlying physical properties of the wood surface should be studied further. It is possible that a number of sub-classes should be defined for different wood "elements" and that a given defect should be considered as built up from a combination of the different ele-

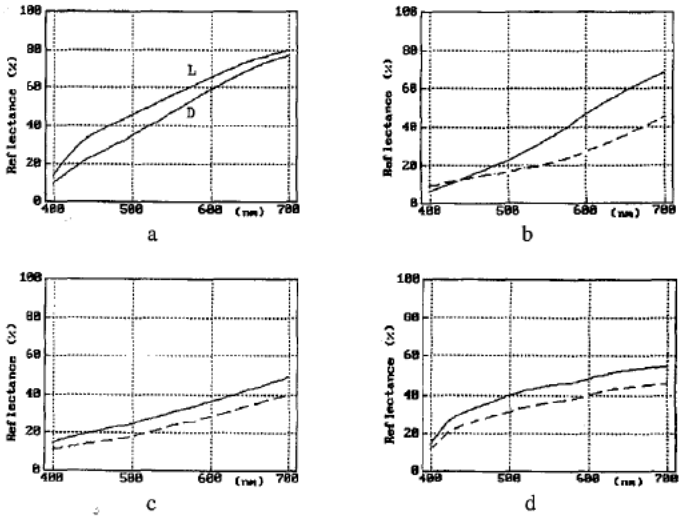


Figure 5.1 *Reflectance spectra of pine wood for a) clear wood, b) sound knots, c) resin pockets and d) blue stain. From Marszalec et al. [115].*

ments. The inspection system would then classify the pixels into the different sub-classes and the classified pixels would then be combined, with or without consideration of their spatial relation, into objects for subsequence classification as various defects.

Although a lot of work remains to be done, a preliminary model for compression wood has been presented by Hagman [77]. It is shown that compression wood is spectrally rather homogenous compared to other defects, indicating that compression wood can indeed be detected using pixelwise spectral classification. This is an important result since no other methods to detect compression wood has been reported.

5.3 The PGP imaging spectrograph

The PGP-MAPP spectrograph described in this chapter is based on the PGP (Prism-Grating-Prism) imaging spectrograph developed by VTT, Finland [79] and marketed by Specim Ltd [82]. The system, shown in Figure 5.2, consists of a lens which projects the image on a slit screen which extracts one single line of the image. The line is passed through a holographic grating which partitions each pixel of the line into its spectral components and the result is projected onto a two-dimensional sensor. Each row of the sensor will thus be exposed by a given spectral component of the originating line and each column on the sensor will give a complete spectrum of a single pixel along the line.

The PGP is available in different spectral ranges. In our experiments we have used the VIS version covering 400-710 nm. According to [82], the spectral resolution is 5 nm but this is dependent on the slit width, which in the standard version is 100 µm. A smaller slit gives better spectral resolution to the cost of higher attenuation. The spectral linearity is ±3 nm over the

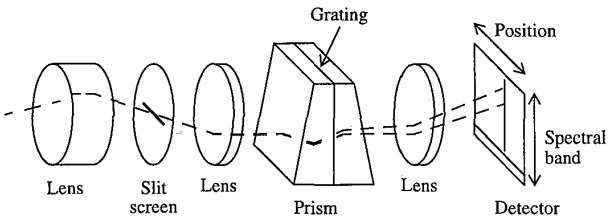


Figure 5.2 The PGP imaging spectrograph [79].

entire spectral range and ±1 nm in the central 450-650 nm region. The PGP unit does not include the first lens nor the sensor. Instead, it is equipped with standard C-mounts in both ends so that any standard lens and any standard matrix camera can be used.

Some initial wood inspection experiments using the PGP have been performed by Vaarala et al. [163]. They used an ordinary 768×576 CCD camera oriented so that the number of spatial pixels were 768 and the theoretical number of spectral channels were 576. In practice, the PGP does not give this spectral resolution and the sensor lines were therefore averaged ten by ten giving 57 spectral channels, which still is a considerable improvement over the three channels of a standard RGB camera.

In the experiments seven classes should be distinguished: sound wood, knot, blue stain, bark, split, resin pocket and core stripe. After acquiring training data, a principal component analysis approach combined with a minimum distance classifier was used to classify the boards pixelwise. In this case five principal components were used. Although the results look promising, it was also concluded in this work that the selection of training data required more consideration. The high input data rate was also a matter of concern.

As could be expected, it was also concluded that the illumination constitutes a problem. In this case a 500 W AC halogen lamp was used. The rather low fraction of blue in the light from a halogen lamp in combination with poor sensitivity of a CCD sensor in the blue wavelength band resulted in a loss of performance since many of the wood defects show significant differences only in the blue end of the spectrum. A xenon lamp will give more blue light but has a rather unstable spectrum. This instability can be accepted in principal since we can measure the spectrum simultaneously with the image acquisition, but still more calculations are then required in the classification stage.

5.4 The PGP-MAPP real time classification system

5.4.1 General

It is clear that the PGP is a powerful tool compared to an ordinary RGB camera, but combined with an ordinary sensor it produces an enormous amount of data. Ideally we would like to reduce the amount of data as close to the sensor as possible, preferably at the sensor itself. This is possible if we combine the PGP with a MAPP camera as shown in Figure 5.3.

The MAPP sensor, which was described in Chapter 3, contains a 256×256 photodiode array which combined with the PGP gives a 256 pixel line scan camera with 256 spectral channels. The sensor is oriented with the processor array parallel to the spatial axis of the PGP, i.e.

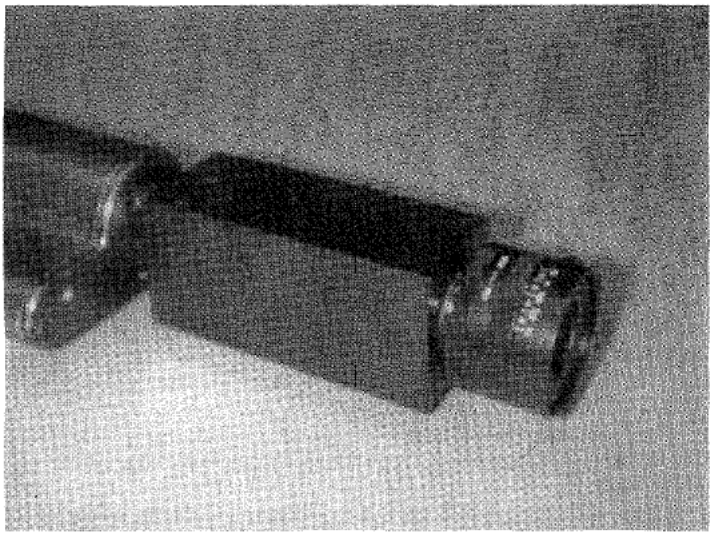


Figure 5.3 The PGP spectrograph combined with the MAPP2200 camera.

each processor element in MAPP operates on the complete spectrum of a single spatial pixel along the line. Each processor element in MAPP is thus responsible for the classification of its own pixel based on the 256 color channels in the column.

Of special interest in this particular application is the spectral characteristics of the photodiode elements in the MAPP sensor array. This has been evaluated by Andersson [5] giving the result shown in Figure 5.4.

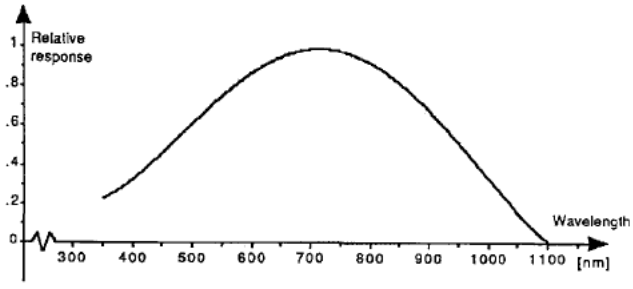


Figure 5.4 Spectral characteristics of the MAPP sensor (from Andersson [5]).

5.4.2 Implementation of linear discriminant functions in MAPP

There are a number of classification models [52] that lead to linear discriminant functions of the form:

$$g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x} + w_{i0} \qquad i = 1, 2, \ldots c \tag{5.1}$$

where $\mathbf{x}$ is the feature vector and c the number of classes. The actual run time classification then comes down to the calculation of a number of weighted sums of the spectral components, one sum for each class. A sample is then assigned to the class giving the largest sum g_i.

The weight coefficients $\mathbf{w}_i$ must be established during some kind of training procedure, supervised or unsupervised. In this project we assume that this is done off line based on the complete spectral information acquired using MAPP as an ordinary camera. Thus, the overall spectral characteristics of the system, comprising the lens, the illumination and the spectral sensitivity of the MAPP sensor will thus be embedded in the training data. The calculation of the weight coefficients will be further discussed in Section 5.4.5.

The processor array in MAPP can be used to calculate the sums in equation 5.1 directly. Since all processors calculate the same sums, the SIMD structure is very well suited for this operation, although the time required for the bit-serial multiplications will limit the performance of the system. To circumvent this bottleneck we have chosen an even more near-sensor approach where we exploit the unique features of the MAPP readout logic. This is illustrated in Figure 5.5 which is identical to the previous Figure 3.2.

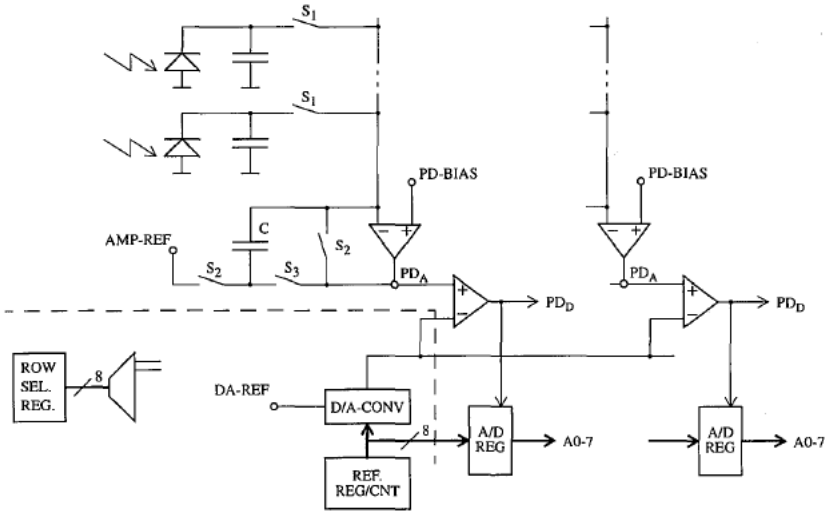


Figure 5.5 Two photodiode columns with readout circuits. The logic outside the dashed line is common to all columns.

As the integration times can be individually controlled for different sensor rows, the weighting of different wavelength bands can be performed by adjusting the individual integration times accordingly. The summing is then performed as an analogue summation of the sensor rows making up a given sum. The accumulated intensity is A/D-converted and stored for later processing.

The implementation of a given set of coefficients therefore comes down to the generation of a suitable timing scheme defining when the exposure of different sensor rows should be initiated and when the summation and A/D-conversion should be performed. A typical timing scheme is shown in Figure 5.6 where the horizontal axis represents the time and the vertical axis different sensor rows. The solid lines indicate exposure that is used in the sum and the dashed lines exposure that is thrown away at reset time. This particular timing scheme implements the sum:

$$S = 3x_1 + x_2 + 2x_3 \tag{5.2}$$

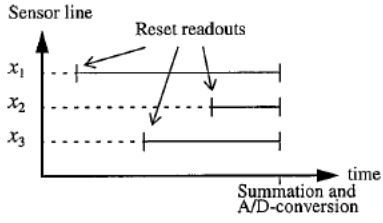


Figure 5.6 Timing scheme for a weighted sum.

General discriminant functions have positive as well as negative coefficients. Since it is not possible to have negative exposure times, one positive and one negative sum is calculated for each discriminant function g_i. The subtraction of the two sums as well as the addition of the constant term w_{i0} in equation 5.1 can then be performed digitally in the MAPP processor array. The result is compared to the sums from the other discriminant functions on a line by line basis and the class corresponding to the largest sum is stored. The only data we then output from the two-dimensional MAPP sensor is a single line of classified pixels.

5.4.3 Problems and limitations

The method has some inherent problems. One obvious problem is that the scene is moving, and different exposure times mean that the different contributions to a given sum come from slightly different areas of the object. One pixel in the sensor is actually mapping the image convolved with a truncated triangle as shown in Figure 5.7. There is also a shift in time from one sum to another as we calculate and A/D-convert them one at a time. The only way to overcome this problem is to use some kind of post-processing of the classified image where we have to accept the fact that steep gradients in the image, i.e. near edges between different regions, may cause false classifications. How serious this limitation is will depend on the application.

The different sums can either be exposed one at a time using a sequence of different timing schemes or be exposed simultaneously, as illustrated in Figure 5.8. Exposing simultaneously will both increase speed and reduce the movement problem, but we have a limitation arising from the fact that one spectral component is coupled to one single sensor element which cannot be used simultaneously with two different exposure times. The consequence is that the

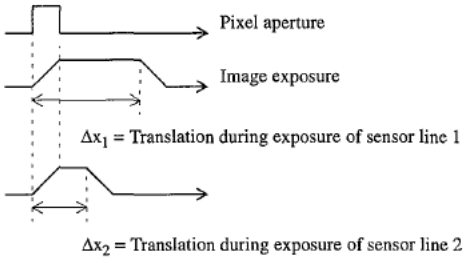


Figure 5.7 *Exposure during movement results in a convolution by a truncated triangle.*

measure for a given wavelength may only appear in one sum. However, since the effective resolution of the PGP is lower than the sensor resolution, even a sharp spectral peak will be spread out on a number of pixels in the spectral direction. The different color channels will thus probably contain a lot of redundant information, and the channel exclusion constraint does not therefore need to be a serious limitation.

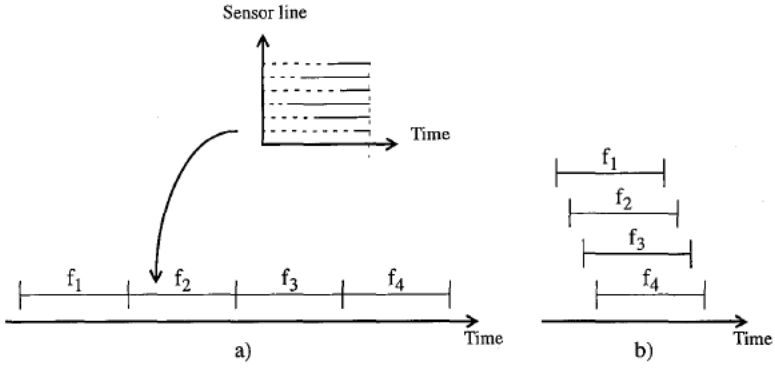


Figure 5.8 *The different discriminant functions can either be exposed a) one at a time or b) simultaneously, in which case a given color channel only can be used in one sum.*

Thus, we may assume that groups of neighboring sensor lines correspond to approximately the same wavelength. For example, if we have four classes we may group the sensor lines four by four and calculate our coefficients based on 64 color channels. A given sum will then use only one of the four sensor lines in each group.

The constraints are also possible to deal with in a more formal way when establishing the weight coefficients. This will be discussed in Section 5.4.5.

As will be evident from the experiments in Section 5.4.10 and 5.4.11, the overall performance of the system will in most practical situations be limited by the illumination. We therefore want to make use of the available amount of light as efficiently as possible. All exposure corresponding to the dashed lines in the timing scheme in Figure 5.6 indicates wasted light. If a sum contains a single coefficient which is much larger than all others, we could end up in a situation where only a small fraction of the total exposure is actually used. Besides, the above mentioned problems arising from the movement of the scene will increase if the coefficients differ a lot.

Again, these problems can be alleviated by using the neighbors of a given wavelength band. If the optimum classifier for example requires the expression:

$$S = 3x_i + x_j \tag{5.3}$$

shown in Figure 5.9a, then in most cases the expression:

$$S = x_{i-1} + x_i + x_{i+1} + x_j \tag{5.4}$$

shown in Figure 5.9b will perform just as well from a classification point of view, but will only need one third of the cycle time to give equivalent exposure. This technique can also be handled more formally as will be discussed in Section 5.4.8.

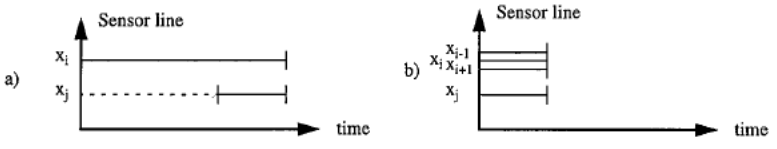


Figure 5.9 Approximating using neighboring sensor lines.

5.4.4 Scaling the coefficients

The coefficients are scaled so that the largest coefficient is assigned the value 1.0 meaning that the corresponding sensor row is exposed 100% of the cycle time. The total exposure, i.e. the combination of cycle time and lens aperture, should then be adjusted so that the largest positive or negative partial sum among all discriminant functions uses the full range of the A/D-converter, i.e. gives a value close to 255. Since all coefficients are implemented as fractions of the total cycle time, they will all be scaled accordingly when the exposure is changed.

However, the constant term w_{i0} in equation 5.1 is added digitally and is therefore not automatically scaled when the exposure is changed. Since the training data acquisition and the run time classification run under completely different exposure conditions, the actual run time value of the term w_{i0} is difficult to calculate. In practice, the easiest way to establish the term is to first ignore w_{i0} when generating the timing scheme from the coefficients and then experimentally find out suitable offset values when the optimum cycle time/aperture level has been set.

So far we have assumed that the illumination is constant along the line. In practice, we normally have a shading resulting in different scaling of the sums for different pixels along the line. This means that the offset terms must be scaled accordingly for the different pixels.

On the other hand, if we omit the term w_{i0} entirely when establishing the weight coefficients, the system will be insensitive to the absolute exposure condition as well as shading. Reducing the amount of light will just decrease the classification distance but the relation

between the discriminant function responses will remain the same. This is equivalent to having all discriminant functions passing through the origin. Omitting the term results in a loss of one degree of freedom in the classification, but a lot of practical problems are avoided.

5.4.5 Establishing the weight coefficients

There are a number of ways to calculate weight coefficients for linear classifiers using different classification models. Unfortunately, these models do not lend themselves to include the method-imposed constraint that a given color channel only may be used once. As will be evident from the following, the limited dynamic range of the MAPP A/D-converter will also create problems that are not handled in traditional models.

A general approach to linear pattern classification is described by Tian et al. [161]. The problem to establish the weight coefficients $W \in \Re^{C:N}$ is regarded as an unconstrained optimization problem:

$$\min \quad \|W \cdot D - T\|^2 \tag{5.5}$$

where $D \in \Re^{N:CM}$ is the training data base:

$$D = \{p_1^1, \ldots, p_M^1, \ldots, p_1^C, \ldots, p_M^C\} \tag{5.6}$$

and $T \in \Re^{C:CM}$ is a target response:

$$T = \begin{bmatrix} \{1\}_M & 0 & \ldots & 0 \\ 0 & \{1\}_M & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \{1\}_M \end{bmatrix} \tag{5.7}$$

N is the number of color channels, C the number of classes and M the number of samples in each class. Tian et al. solve this problem in a straightforward way using a unified pseudoinverse algorithm [161].

The advantage with this approach is that we may add constraints to the optimization problem. In our case we will add constraints ensuring that we only have one element in each column of W, i.e. that a given sensor row is only used once. The optimization problem can thus be formulated as:

$$\min \quad J(W, S) = \|(WS) \cdot D - T\|^2 \tag{5.8}$$

The matrix $(WS) \in \Re^{C:N}$ is given by:

$$(WS) = \begin{bmatrix} w_{11}s_{11} & \ldots & w_{1j}s_{1j} & \ldots & w_{1N}s_{1N} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ w_{i1}s_{i1} & \ldots & w_{ij}s_{ij} & \ldots & w_{iN}s_{iN} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ w_{C1}s_{C1} & \ldots & w_{Cj}s_{Cj} & \ldots & w_{CN}s_{CN} \end{bmatrix} \tag{5.9}$$

where $s_{ij} \in \{0, 1\}$. The constraint that each sensor row should only be used once can now be formulated as:

$$\sum_{i=1}^{C} s_{ij} \leq 1 \qquad j = 1 \ldots N \tag{5.10}$$

A solution to the optimization problem using the gradient based CONMIN algorithm [123] in combination with a heuristic iterative procedure to handle the constraint is described by Österberg et al. [124]. This algorithm was used when the PGP-MAPP spectrograph was first presented (Åstrand et al. [18]). It was shown that the basic analogue weighted sum calculations worked properly but the system was only capable of classifying objects with very clear color differences.

The problem was found to be the limited dynamic range of the 8 bit A/D-converter. The lens aperture and/or the exposure time should be adjusted so that the largest sum (applied on the brightest object) gives values close to 255. However, since the positive and negative contributions are A/D-converted separately, the exposure must be adjusted so that any positive *or* negative contribution does not reach 255. This is equivalent to a digital summation where we only use 8 bits precision, which is clearly a limitation. The result when the positive and negative partial sums are squeezed into 8 bits is that the difference between them gives numerically very small values. In the initial experiments performed in [18], it was found that the numerical results from the discriminant functions were in the range 3-7 even for spectrally very different objects. More spectrally demanding problems, such as the tablet classification described in Section 5.4.11, gave virtually 0 for all discriminant functions.

An obvious solution to the problem is to use a sensor with better dynamic performance. It will probably take some years though before a 12-bit MAPP is available. However, it turns out that much can be done in the calculation of the weight coefficients. The objective function in equation 5.8 only tries to maximize the classification distances but does not in any way reward solutions which use the dynamic range in the MAPP environment in an efficient way.

Another problem with the optimization used in [18] was that the solution procedure was very time consuming. Neither did the iterative heuristics perform well when the number of classes increased and the color channel exclusion constraint became more severe. It was clear that a completely new approach to the weight coefficient calculation was required.

5.4.6 Handling the limited dynamic range of the A/D-conversion

The new optimization approach was developed in cooperation with Österberg et al. [124]. Ideally, we would like to model the MAPP implementation by simulating the 256-level discretization of the positive and negative contributions respectively. The discretization should then be normed by the maximum response of any of the contributions applied on the training data. This is equivalent to adjusting the aperture of the lens so that the maximum contribution makes full use of the dynamic range of the A/D-converter.

Thus, the optimization problem should be to maximize the classification distance in MAPP levels based on the simulation. Especially, we would like to maximize the classification distance for the most difficult combination of patterns. The critical classification distance can be defined by:

$$Q_{dist}(W) = \min_{c=1,\ldots,C} \left\{ \operatorname*{mean}_{\substack{p=1,\ldots,CM, \\ I(p)=c}} \left\{ \min_{\substack{t=1,\ldots,C, \\ t \neq c}} (O_c(p) - O_t(p)) \right\} \right\} \tag{5.11}$$

where

$$O_c(p) = \sum_{j=1}^{N} w_{cj} D_{jp} \tag{5.12}$$

is the output from the c^{th} linear discriminant function, defined by the coefficients w_{cj}, on the input pattern p. N is the number of color channels, C the number of classes and M the number of training patterns in each class. $I(p)$ is the class membership of pattern p. The inner minimization in equation 5.11 computes the smallest classification marginal for the p^{th} input pattern and the mean operation computes the mean of the distances over all input patterns belonging to the c^{th} class. The outer minimization computes the smallest mean distance over the C classes $c = 1, \ldots, C$.

Since we want to maximize the classification distance in relation to the maximum response of any positive or negative contribution, we may define the amplification factor:

$$Q_{amp}(W) = \max_{\substack{c=1,\ldots,C,\\ p=1,\ldots,CM}} \left\{ \sum_{\substack{j=1\\ w_{cj}\geq 0}}^{N} w_{cj} D_{jp}, \; -\sum_{\substack{j=1\\ w_{cj}<0}}^{N} w_{cj} D_{jp} \right\} \tag{5.13}$$

and the task to establish the weight coefficients can then be formulated as an unconstrained optimization problem:

$$\max_{W} \frac{Q_{dist}(W)}{Q_{amp}(W)} \tag{5.14}$$

Since we will use a gradient based optimization technique, the partial derivatives:

$$\frac{\partial}{\partial w_{ij}} \frac{Q_{dist}(W)}{Q_{amp}(W)} \tag{5.15}$$

have to be continuous and analytically computable, which is not the case for the *min* and *max* operations. Instead we use an approximation by noting that if $x_l \geq 0, l = 1, \ldots, L$, then we have:

$$\begin{aligned} max\{x_1, x_2, \ldots, x_L\} &= \lim_{k\to\infty} \left((x_1)^k + (x_2)^k + \ldots + (x_L)^k \right)^{\frac{1}{k}} \\ &\approx \left((x_1)^{\bar{k}} + (x_2)^{\bar{k}} + \ldots + (x_L)^{\bar{k}} \right)^{\frac{1}{\bar{k}}} \end{aligned} \tag{5.16}$$

where $\bar{k}$ is a large number such that overflow does not occur. To approximate the *min* operation we observe that if $x_l \leq 0, l = 1, \ldots, L$, then we have:

$$min\{x_1, x_2, \ldots, x_L\} = -max\{-x_1, -x_2, \ldots, -x_L\} \tag{5.17}$$

In the real implementation we have to force the *max* arguments to be positive and the *min* arguments to be negative by adding and subtracting a large number respectively. We also have to take care of the discontinuity of the derivative when $w_{ij} = 0$. For details refer to [124].

5.4.7 Handling the color channel exclusion constraint

The formulation above handles the problem with the limited dynamic range of the A/D-conversion but does not handle the constraint that a given color channel only may be used once. If we implement the classification using time multiplexing, i.e. by calculating the discriminant functions one by one, we may use the resulting coefficients W directly. Otherwise we have to add constraints ensuring that all but one coefficients are zero for a given color channel. A formulation and a solution procedure for the new constrained optimization problem was indicated by Rönnqvist [140] and further developed and implemented by Österberg et al. [124]. The new problem can be formulated as:

$$\max_{W} \frac{Q_{dist}(W)}{Q_{amp}(W)} \tag{5.18}$$

subject to

$$w_{ij} \leq K \cdot s_{ij} \qquad i = 1, \ldots, C \qquad j = 1, \ldots, N \tag{5.19}$$

$$-w_{ij} \leq K \cdot s_{ij} \qquad i = 1, \ldots, C \qquad j = 1, \ldots, N \tag{5.20}$$

$$\sum_{i=1}^{C} s_{ij} = 1 \qquad j = 1, \ldots, N \tag{5.21}$$

$$s_{ij} \in \{0, 1\} \qquad i = 1, \ldots, C \qquad j = 1, \ldots, N \tag{5.22}$$

where K is a large number. The constraints (5.19)-(5.22) ensure that only one weight w_{ij} is non-zero for a given j, i.e. that a given color channel is only used once. s_{ij} is a binary variable which is 1 if sensor row j is used in filter function i and 0 if it is not used.

The optimization problem can be solved using Lagrangian relaxation (Fisher [57,58]) which is a method to solve large structured integer problems. The basic idea is to move one or more of the constraints to the objective function giving a less constrained problem. In our case the so called Lagrangian subproblem can be formulated as:

$$\max \quad L(W, S, \lambda, \eta) \tag{5.23}$$

$$= \frac{Q_{dist}}{Q_{amp}} + \sum_{i=1}^{C}\sum_{j=1}^{N} \lambda_{ij}(K \cdot s_{ij} - w_{ij}) + \sum_{i=1}^{C}\sum_{j=1}^{N} \eta_{ij}(K \cdot s_{ij} + w_{ij})$$

where the first two constraint sets have been moved to the objective function with associated multiplier sets λ and η. A solution to this problem is in general not feasible but gives an upper bound of the solution to the original problem. By employing a heuristic procedure to generate feasible solutions from the solutions to the Lagrangian subproblems we will get lower bounds and the purpose of the Lagrange relaxation scheme is to iteratively generate better and better upper and lower bounds. This will not be discussed further here, for details refer to Österberg [124].

The procedure was found to generate good solutions after only a few iterations. An absolute upper bound is given by a solution to the unconstrained problem and the solution to the constrained problem rapidly approaches the unconstrained solution. This means that the method-imposed constraint that a given color channel only may be used once is not a serious limitation. This could be expected since the 256 color channels probably contain a lot of redun-

dant information. Recall that even a sharp spectral peak will be spread out on a number of sensor rows since the effective spectral resolution of the PGP is lower than the resolution of the MAPP sensor.

5.4.8 Maximizing the light usage

The procedure outlined in Section 5.4.6 and Section 5.4.7 does not automatically handle the light usage problem described in Section 5.4.3. The two solutions in Figure 5.9 both give the same levels in the A/D-conversion and there is nothing in the optimization that favors the solution in Figure 5.9b. This problem can be included in the optimization formulation by adding a light usage term to the objective function which will indicate when we have small differences in coefficients. An obvious such measure is the variance of the coefficients. However, as the objective is to use as much light as possible, it would probably be better to take the sum of the coefficients and relate to the biggest one, which is the one giving the cycle time:

$$U(W) = \frac{\sum_{i,j}^{C,N} abs(w_{ij})}{N \cdot max(abs(w_{ij}))} \tag{5.24}$$

This will give a direct measure of the fraction of the light used in the classification process and will give a severe penalty to the case we want to avoid, namely where we have one single large coefficient. The new objective function can thus be formulated as:

$$\max_{W} \frac{Q_{dist}(W)}{Q_{amp}(W)} + \gamma \cdot U(W) \tag{5.25}$$

where the constant γ must be established experimentally. This technique has not been implemented and tested so far. It will probably depend on the application whether it is required or not.

5.4.9 Performance estimates

In most practical situations, the factor limiting the speed of the system will be the illumination. This means that the MAPP processor array will be idling during certain exposure intervals. However, if the light is unlimited, what is the maximum speed of the system? Alternatively, if the classified pixel data are to be further processed within MAPP concurrently with the exposure of the next line, how much processing capacity is required for the readout and classification of each line? It turns out that the two cases give slightly different answers.

So far we have assumed that the readout and A/D-conversion are performed instantaneously which is not the case in reality. Figure 5.10 shows the sensor reset, readout and A/D-conversion timing assuming that we calculate the different sums one at a time as shown in Figure 5.8a. Let us first consider the second case above where the sensor reset cycle in Figure 5.10 is much longer than the readout and A/D-conversion cycles.

In the sensor reset cycle, the different sensor rows are reset in appropriate time slots illustrated by the short thick line segments. The reset operations require in total m+2 MAPP instruction cycles, where m is the number of color channels included in the sum. The remaining time during the sensor reset cycle can be used for other processing in MAPP. The reset instructions can be inserted in the main instruction stream without overhead since no MAPP registers are affected.

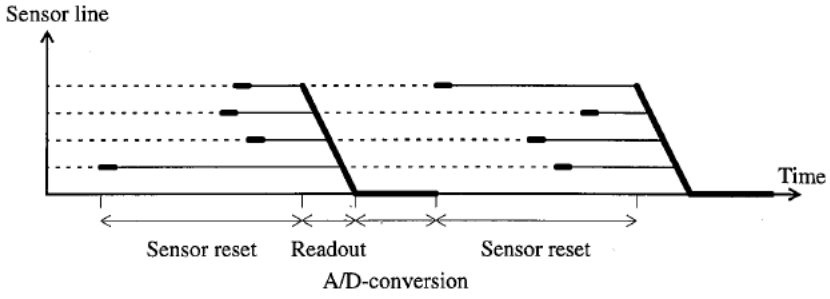


Figure 5.10 Timing sequence when the sums are calculated one at a time. Thick line segments denote time intervals when the readout logic is occupied.

The next cycle is the analogue summation readout, indicated by the thick slanted line. This can be performed in m+4 cycles including initialization of the A/D-conversion. The A/D-conversion itself, indicated by the horizontal thick line, runs asynchronously and concurrently to other MAPP processing and requires T_{ad} = 15 µs. The A/D-conversion can be interleaved with other processing, but not with operations using the readout logic. Thus, the sensor resets can not be performed while an A/D-conversion is in progress.

The classification also requires some digital processing. Assume that we have c classes which means that we have to calculate 2c weighted sums in each cycle (one positive and one negative for each class). After A/D-conversion we then need a bit serial subtraction of the positive and negative contributions which requires 3b+3 cycles where b is the number of bits. This is followed by an addition of a constant offset term (3b+1) and a comparison with discriminant function values from previous sums (2b+1). Finally we need a conditional storage of the maximum result and the class number requiring 3b and $3 \cdot {}^2\log c$ cycles respectively.

The total number of instruction cycles for the case where MAPP is also used for additional processing is summarized below. It is assumed that all A/D-conversion can be performed transparent to other processing.

Sensor reset:	$2c(m+2)$
Readout:	$2c(m+4)$
Subtraction of pos. and neg. contribution:	$c(3b+3)$
Addition of offset term:	$c(3b+1)$
Comparison of result:	$c(2b+1)$
Conditional storage of max value:	$c \cdot 3b$
Conditional storage of max class:	$c \cdot 3 \cdot {}^2\log c$

This gives the total processing time for the classification:

$$T = c\,(4m + 11b + 3 \cdot {}^2\log c + 17) \cdot T_M \qquad (5.26)$$

The specified maximum MAPP instruction frequency is 4 MHz giving an instruction cycle time T_M = 250 ns, although it is possible to run selected chips up to 8 MHz. Assuming 4 MHz, c = 4, m = 64 and b = 9 gives 378 µs for the total classification cycle.

This was the case when MAPP was occupied with other processing as well. If we only want to output classified pixels and the light is unlimited, what is the minimum cycle time? The digital computations will remain the same but since two A/D-conversions are required for each class, this will now be the limiting factor. The sensor reset and readout can not be interleaved with the A/D-conversion and the total time will thus in theory be:

$$T = c\,((4m+12)\cdot T_M + 2T_{ad}) \tag{5.27}$$

which with the figures above gives 388 µs as the total cycle time of the system corresponding to 2.6 kHz line frequency or a classified pixel rate of 0.7 MHz. This is, however, a theoretical figure. Consider the timing sequence in Figure 5.10. Running at full speed means that we only have m available time slots in the sensor reset section. Since only one sensor row can be reset in each cycle, this will severely limit the possible combinations of implemented coefficients. A small consolation is the fact that the readout of the coefficients can be performed in any order enabling many different coefficient combinations by reordering the color channels. The new situation could perhaps be formulated as additional constraints in the weight coefficient optimization but this has so far not been investigated.

The situation is somewhat different if we calculate the discriminant functions simultaneously as shown in Figure 5.11. The total number of instruction cycles will be the same but the time slots available for sensor resets will be spread out on the gaps between the readout and A/D-conversion cycles. This gives an even more complicated set of constraints on the weight coefficients. As in Figure 5.10, the thick lines should be interpreted as time intervals where the readout logic is occupied. Two such line segments may therefore never overlap.

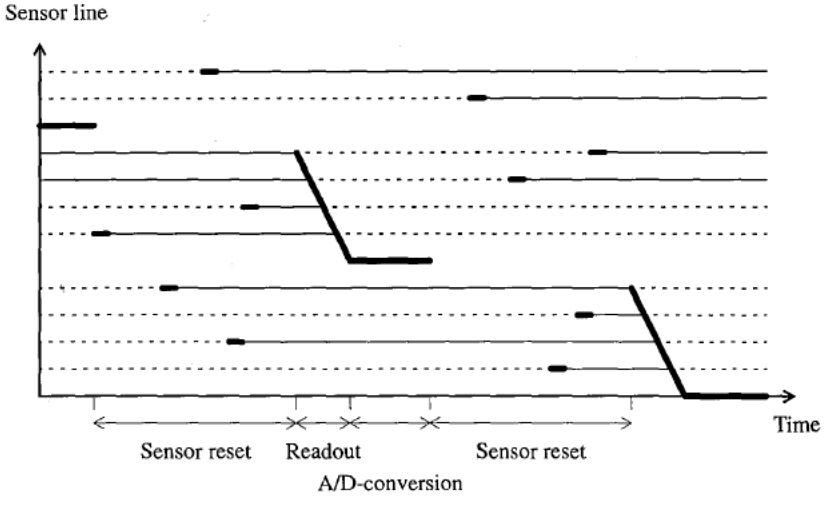


Figure 5.11 Timing sequence when the sums are calculated simultaneously.

The situation can be improved if we allow ourselves a small modification to the MAPP readout logic. By inserting a sample and hold stage between the readout amplifier and the comparator as indicated in Figure 5.12, the result is that sensor reset operations can be performed

concurrently with A/D-conversion and we will thus be able to position the resets more freely in time. As mentioned in Chapter 4, this modification would also improve the performance of other MAPP algorithms, such as the FBM sheet-of-light range imaging.

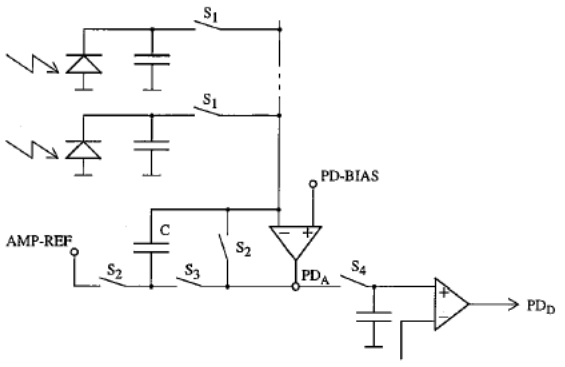


Figure 5.12 Adding a sample and hold stage controlled by the S_4 switch enables concurrent sensor reset and A/D-conversion.

5.4.10 Experiments

As pointed out in Section 5.2, pixelwise classification of wood features requires careful selection of classes and training data. The limited experiments to be presented here serve therefore only to illustrate the procedure, and can not be used to judge the spectral discriminating performance of the system. The experiments were performed on a number of spruce boards using four classes, namely sound knots, dry knots, blue stain and clear wood.

The illumination was supplied by two Fostec 8900 series Lightlines with cylindrical lens attachments as shown in Figure 5.13. The lightlines consist of a bundle of optical fibres arranged in a line with the other end connected to a Fostec 8360 series light source.

To reduce the amount of data and to increase the effective pixel size on the MAPP sensor, the sensor rows were grouped four by four and the intensity in each group of four lines was accumulated using the analogue summation feature before further processing. We then effectively had a 64 color channel system. A training data stack was acquired and 32 training pixels were selected from each class. The mean spectra for the four classes are shown in Figure 5.14. However, since the constant offset term was not used, the algorithm becomes insensitive to luminance variations and the normalized spectra in Figure 5.15 are therefore more interesting to study.

The spectra show very little similarities with the spectra in Figure 5.1 since they are severely affected by the spectral characteristics of the system, i.e. the lamp, the PGP and the MAPP sensor. Of special interest is the periodic variations that can be seen in all spectra. This is according to [80] caused by interference in the coating layer of the sensor chip and will occur more or less in all commercially available matrix sensors. However, from a classification point of view, the interference does not constitute a problem since the result is only a constant attenuation of certain wavelength bands and since the training data is acquired using the same optical system as in the classification, this is just another characteristics embedded in the training data.

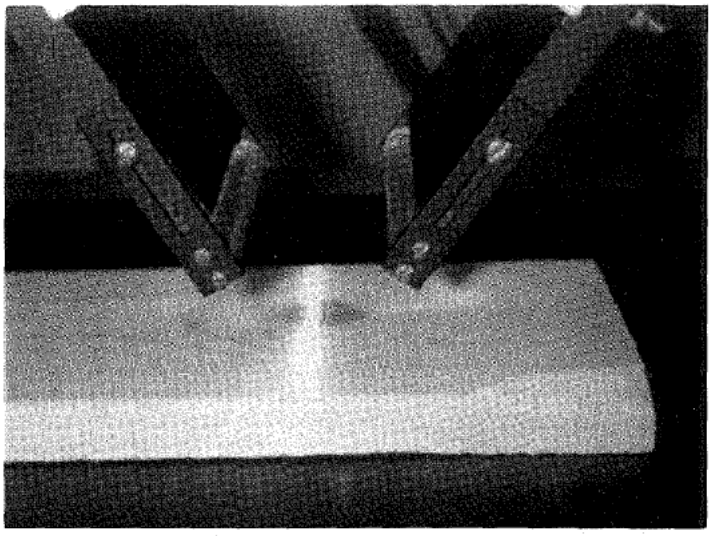


Figure 5.13 The Fostec fiber-optic Lightline Illumination used in the PGP-MAPP experiments.

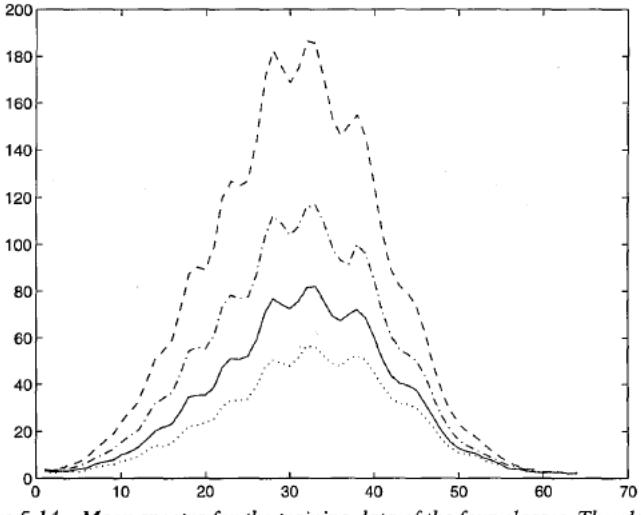


Figure 5.14 Mean spectra for the training data of the four classes. The classes are from top to bottom clear wood, sound knots, blue stain and dry knots.

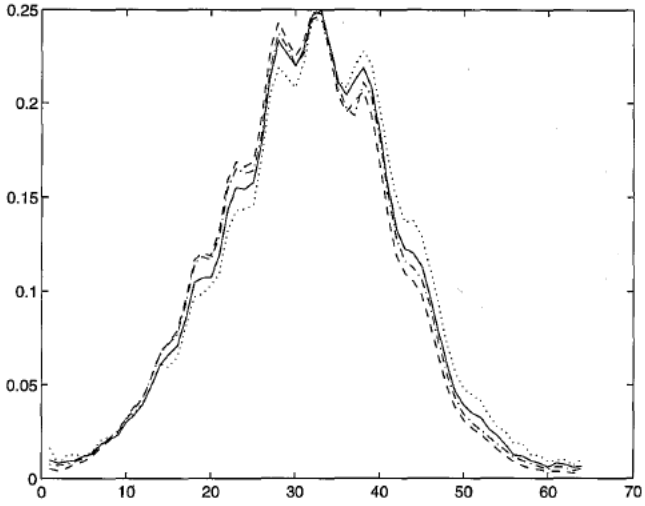


Figure 5.15 Normalized spectra for the classes in Figure 5.14.

A set of filter functions were calculated using the procedure described in Section 5.4.7. The coefficients for the different functions are shown in Figure 5.16. Note that any given color channel is only used in one sum. The coefficients are scaled in the range 0.0-1.0 meaning that 1.0 gives a full cycle exposure, 0.5 gives a reset readout after half the cycle etc.

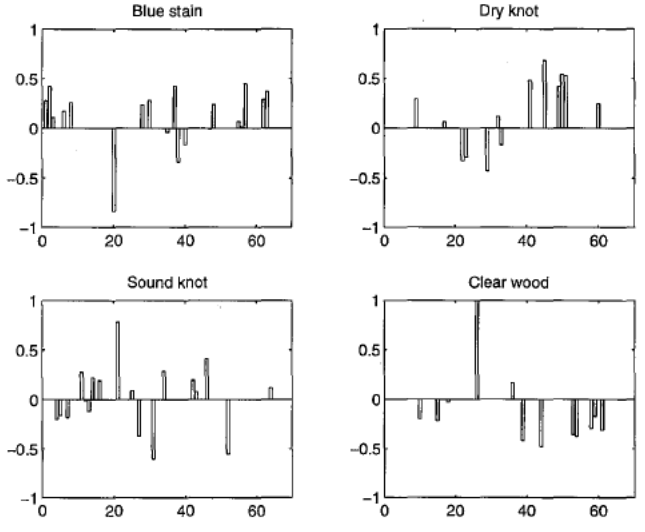


Figure 5.16 Filter functions for the four classes.

The filter functions were separated into their positive and negative parts and then converted to suitable timing schemes. The result was then tested on a number of boards in the real time PGP-MAPP environment. In these experiments the PC version of the MAPP camera was used which limited the instruction frequency to around 700 kHz. To run at full speed, i.e. at the maximum specified MAPP instruction frequency, would require a dedicated instruction sequencer. However, using the Fostec illumination the minimum cycle time in order to use the full dynamic range of the A/D-converters was found to be around 6 ms. This is much slower than the theoretical performance of the system, and the MAPP instruction frequency is therefore not a limiting factor. Clearly, the available amount of light will be the limiting factor in most practical cases.

The classification results are shown in Figure 5.17-5.21 (see also Appendix A). The left image in each figure is one of the components from the training stack and the right image the result of the classification. White represents clear wood, light gray sound knots, dark gray blue stain and black dry knots. The images do not match entirely in the vertical direction due to the different cycle time and conveyor belt speed used in the training data acquisition and the run time classification. The training data had to be acquired very slowly since all individual color channels must be A/D-converted and output from the sensor, and the low shift register bandwidth in the PC version of the MAPP camera will thereby become a limiting factor.

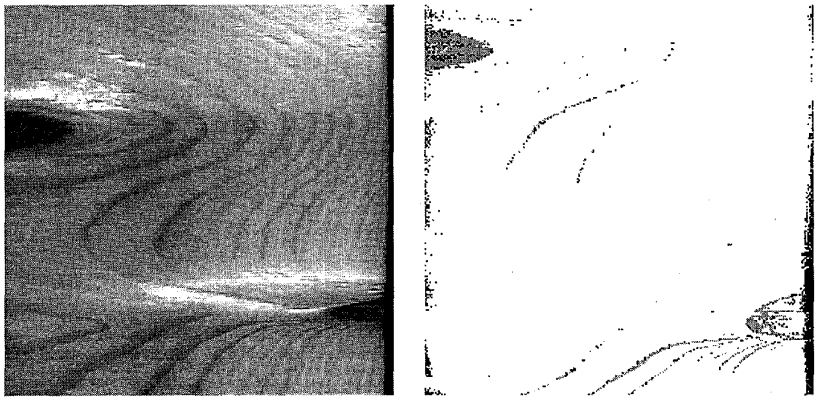


Figure 5.17 Classification of spruce boards using the real time PGP-MAPP spectrograph. White represents clear wood, light gray sound knots, dark gray blue stain and black dry knots. See also Appendix A.

The non-homogenous nature of the defects can be clearly observed in these images. It is especially difficult to distinguish dark clear wood areas, i.e. the grain, from sound knots. This can be seen in Figure 5.17 containing two sound knots where part of the grain is classified as sound knot and part of the knot as clear wood. Figure 5.18 contains a large sound knot and a blue stain area which are clearly discriminated, although the blue stain area contains a lot of more or less isolated dry knot pixels which must be filtered.

The problem with dark grain can also be seen in Figure 5.19 where one section of the grain is classified as blue stain. It seems that the clear wood should have been partitioned into at least two classes, one for the dark and one for the bright areas. Figure 5.20 contains a large blue stain area and two black-ringed knots, one with a very narrow ring and one with a wider ring.

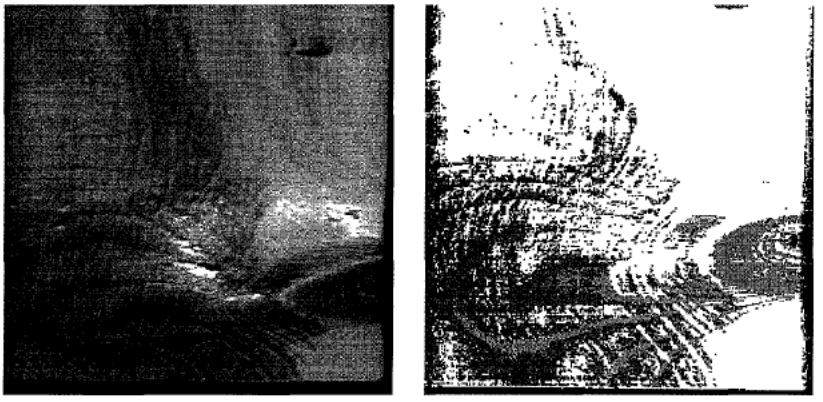


Figure 5.18 A board containing a blue stain area and a sound knot. Note that the blue stain contains a lot of pixels classified as dry knot.

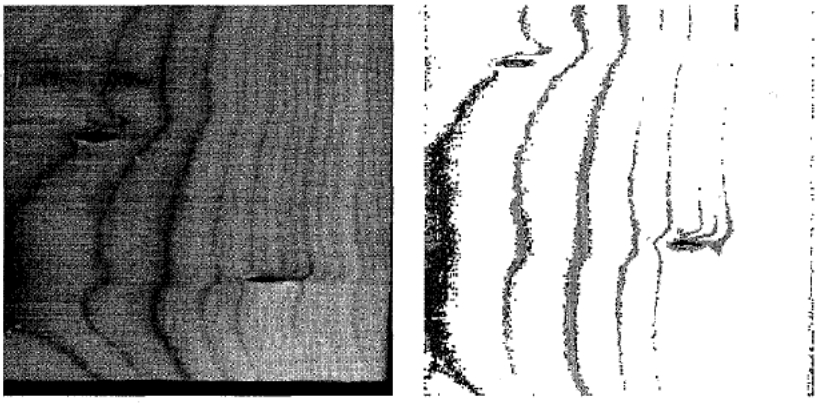


Figure 5.19 A board illustrating the non-homogenous nature of the clear wood. The bright areas are correctly classified whereas the dark grain is classified as either sound knot or blue stain.

The ring areas are classified as black knot while the rest of the knots are classified as sound knot, except for a section of the right knot which is missed. Finally, the dry horn knot in Figure 5.21 is clearly identified.

Once again, to really verify the potential of the pixelwise spectral approach to wood inspection, the physical properties of the different defects must be carefully examined in order to model the defects in terms of underlying spectral classes.

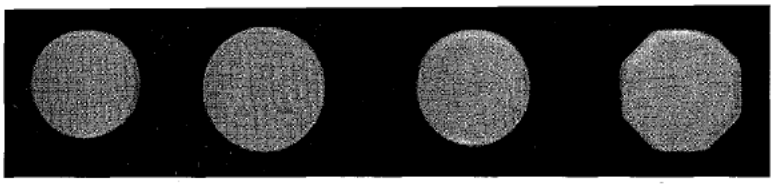


Figure 5.22 A set of four tablets to be sorted using the PGP-MAPP spectrograph. See also the color image in appendix A.

A training data stack was acquired and 16 pixels were selected from each tablet as training data. The sensor rows were not grouped in this case, i.e. a full 256 channel system was used. The resulting normalized mean spectra for the four tablets are shown in Figure 5.23. Figure 5.24 shows the total set of spectra for each one of the four tablets. Here we can clearly see that the in-class intensity variations are much larger than the differences between the classes. Most of the variations within the classes are due to the curvature of the surface of the tablets.

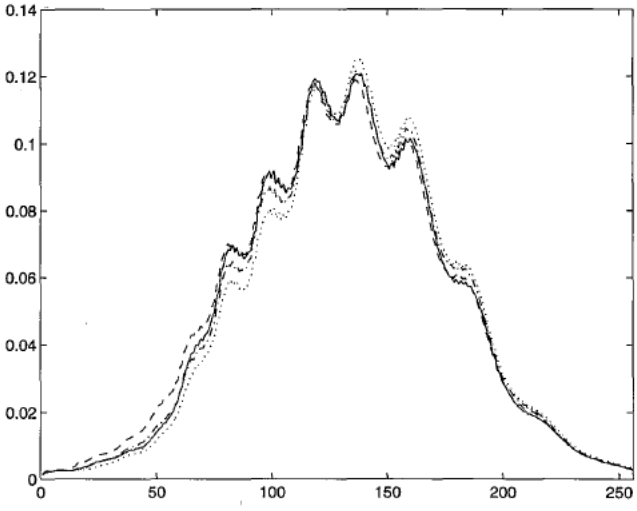


Figure 5.23 Normalized mean spectra for the four tablets.

The four different filter functions were established using the procedure in Section 5.4.7, and the resulting discriminant functions are shown in Figure 5.25. The filter functions were then converted to a timing scheme and implemented in the real time environment. The system was tested on the tablets and the result is shown in Figure 5.26.

To separate the tablets from the background, a threshold was applied to the maximum of any of the positive or negative contributions, i.e. the maximum result from any A/D-conversion. This prevents random classifications of the dark background pixels.

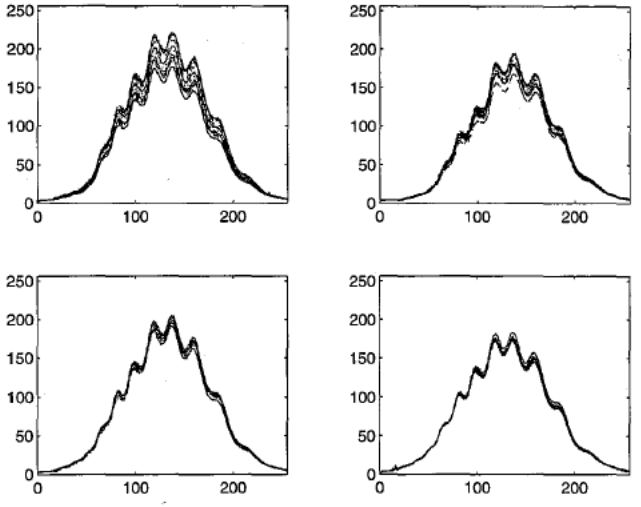


Figure 5.24 Spectra variations within each tablet.

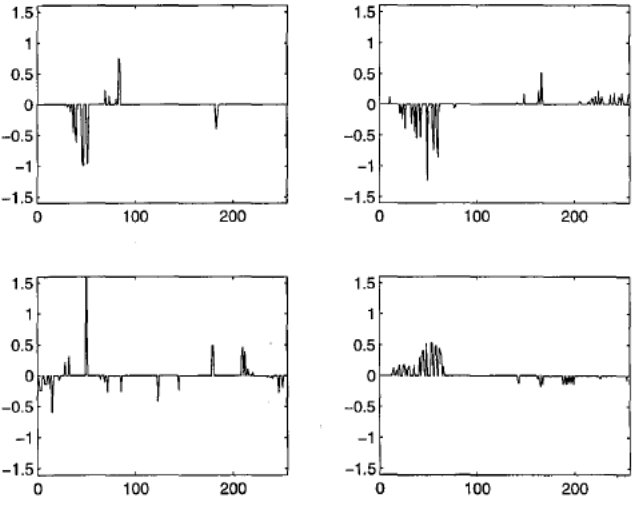


Figure 5.25 Discriminant functions for the tablet classification application.

The resulting classification is very good with only a few misclassifications, mainly along the edges. Unfortunately, only one tablet of each type was available, which means that the tablets classified in Figure 5.26 are the four used in the training process. On the other hand, only 16 training pixels were selected from each tablet and the run time classification is of course a completely different exposure of the tablets.

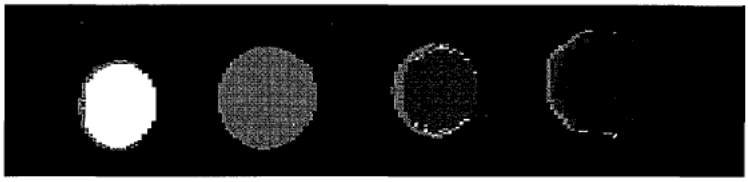


Figure 5.26 Result of classification of the tablets in the real time environment.

It could be interesting to study the actual values from the filter functions. A stationary profile of the tablets was acquired using the PGP-MAPP in the training data acquisition mode. The filter functions were applied off-line giving four response profiles. The 256-level quantification in the MAPP A/D-converter were then simulated, i.e. the positive and negative contributions were calculated separately and the maximum value from any of the four sums was given the value 255. This corresponds to adjusting the aperture of the lens (or the exposure cycle time) so that the maximum positive or negative contribution uses the full dynamic range of the A/D-converter. All other values were then scaled accordingly. Finally the subtraction was performed and the resulting profiles are shown in Figure 5.27. With the tablets still at the same positions on the conveyor belt, similar profiles were then acquired in the real time classification mode using the previously established timing scheme. The aperture was adjusted so that the full range of the A/D-converter was used. The resulting profiles are shown in Figure 5.28. Note that the maximum response from any A/D-conversion was thresholded also in this case to separate the tablets from the background. If the maximum response was below the threshold, the value zero was assigned to all filter functions.

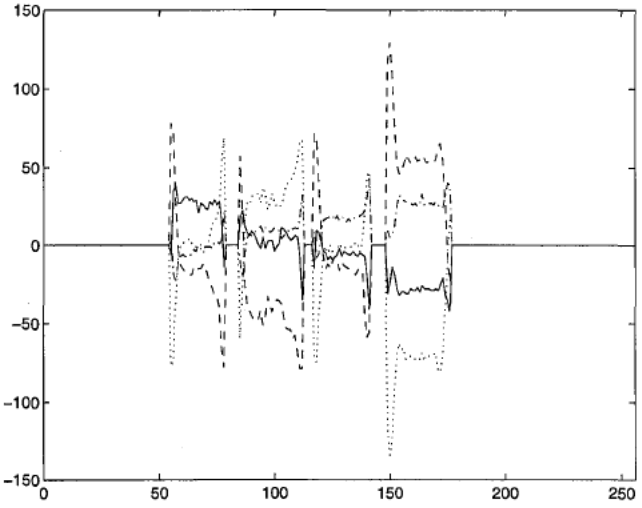


Figure 5.27 Simulated response profiles for the four filter functions applied on one line of the tablets.

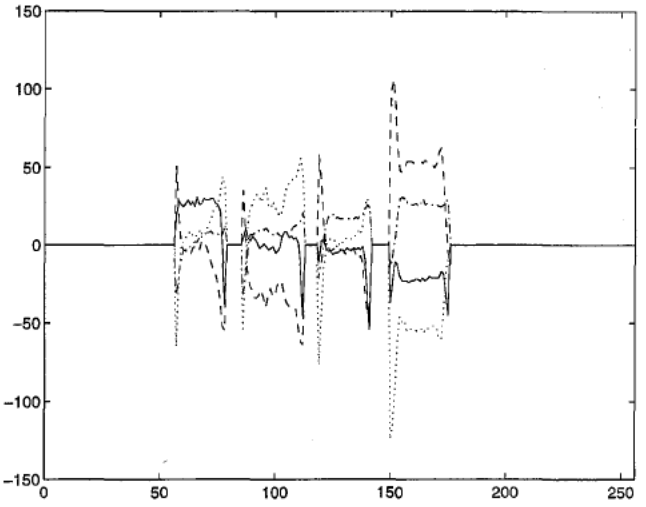


Figure 5.28 Actual response profiles for the four filter functions applied on one line of the tablets.

The profiles in Figure 5.27 and 5.28 clearly show the conformity between the analogue real time calculation of the discriminant functions and the digital simulation. In other words, the analogue implementation really performs the calculations it is supposed to do.

5.5 Discussion

We have presented a tool for high speed parallel pixelwise spectral classification. The line scanning nature of the system makes it well suited for classification of moving scenes, which is normally the case in wood inspection applications. We have also shown that the system has a high spectral discrimination capability, but the non-homogenous nature of many wood defects makes it difficult to apply pixelwise classification, at least at a defect level. The different defects must be carefully studied and modelled in terms of subclasses which represents different physical properties of the wood surface. This work is in progress by Hagman [77].

We have also shown that the PGP-MAPP spectrograph is a general tool for implementing different optical filter functions. As such it is a useful research tool. If we are able to isolate a number of spectral channels which are significant for the wood inspection problem, we may omit the PGP in a production system. Instead we add color filters directly on the MAPP sensor as indicated in Section 3.9. The analogue weighted sum calculations may still be used but here we do not have the constraint that a given color channel only may be used in one filter function since the color filter may cover several sensor rows. The spectral classification may thus be an integrated part of the total sensor system outlined in Chapter 3.

Another observation from this chapter is the significance of the procedure to establish the weight coefficients. When the system was originally presented using the procedure in Section 5.4.5, it was only capable of discriminating very clear colors. When the limited dynamic range of the MAPP sensor was considered in the formulation of the optimization problem, the spectral discrimination capability was drastically increased.

Chapter 6

Segmentation and defect classification

6.1 Introduction

One of the main contributions in this thesis is the sensor framework and improving the sensor data will always simplify the subsequent image analysis, i.e. simpler (and thereby faster) defect detection algorithms can be employed. Nevertheless, the sensor system itself does not solve any wood inspection problems. Segmentation of the input image followed by classification of the objects (defects) is unavoidable.

The MAPP sensor framework described in the previous chapters can be regarded in two ways. We can either see it as a pure sensor system delivering image data of different types, i.e. grayscale images, difference images, tracheid effect images, range images, spectrally pixel-wise classified images etc. The subsequent defect detection is then implemented in a separate processing unit and the full range of image processing operations and architectures can thereby be employed.

The other approach is to reduce as much of the information as possible directly on the sensor. The goal should be to reduce the data output from MAPP to the extent that the rest of the problem can be solved by an ordinary PC. Since the MAPP processor array is not in any way well suited for higher level image processing operations, we cannot expect to do without the host computer.

This almost stand-alone MAPP system approach is the one chosen in this chapter. The algorithms that can be employed are of course of limited complexity compared to a full fledged system. To a great extent, the defect detection is based on thresholding in the different images, since both thresholding and segmentation of binary images can be efficiently implemented in MAPP. The segmentation and classification scheme is partly inspired by the defect detection scheme described in Hall [78], but has been extended to include tracheid and difference data.

The system described in this chapter does not use the full set of the sensor data as described in Chapter 3. The grayscale and the tracheid images are used to detect normal defects. The difference image is used to detect wane edges but the surface roughness detection is not implemented. Neither are the range image and the corresponding intensity image directly used in the system. However, the acquisition of the images are included in the performance estimations and a limited scheme to extract features from the range data is outlined. The color analysis part is not included since a separate MAPP camera would be required for the PGP. The acquisition of color data, given that suitable filters are applied directly on the MAPP chip, is included in the performance calculations, but was not tested in reality since no such filters were available.

6.2 Segmentation of the wood sensor images

6.2.1 General observations

The purpose of the segmentation step is to locate areas of the board which are potential defects. We hereby assume that the defects are isolated islands on a clear wood background. Assume that we want to detect and distinguish the following defects: sound knots, dry knots, splits, pith stripes, pitch pockets and wane edges. By studying the images in Chapter 3 (as well as numerous other images) we can observe that both sound and dry knots turn dark in the tracheid effect

image. Both types of knots can thus easily be segmented by thresholding. We also note that a dry knot usually is much darker, or at least contain a very dark area, compared to the sound knot. The difference is more pronounced in the grayscale image indicating that a thresholding in this image could be used to classify the type of the knot. To distinguish the knots from longer defects, we may use some elongation measure.

We also observe that long defects, i.e. splits, pith stripes and pitch pockets, all are quite dark in the grayscale images whereas pitch pockets and splits virtually disappear in the tracheid images. Pith stripes are very dark in the tracheid image but here we can observe, for instance in Figure 3.13a, that the core section of the board is quite dark also outside the pith stripes themselves. We therefore conclude that the long defects should be segmented by thresholding in the grayscale image, but that we may use a threshold in the tracheid image to distinguish pitch pockets from pith stripes. Small splits are narrower than pitch pockets and pith stripes and can therefore be distinguished by measuring the width. Wider splits are on the other hand darker in the grayscale image than pith and pitch pockets and can therefore be classified using an additional threshold in the grayscale image.

Wane edges can be detected by thresholding in the difference image, provided that the wane edge reflects enough light. If it is dark, i.e. if it is covered by bark, it can instead be detected by thresholding in the grayscale image. Here we may take advantage of the fact that the wane is located along the edge of the board which never is the case for pith stripes and very seldom for pitch pockets.

The surface roughness detection procedure discussed in Section 3.5 can be implemented by first applying a [1 -1] filter in the crosswise direction and then thresholding the absolute value of the response. Since the result will be a number of small disconnected areas, we then apply a number of expansion and shrinking operations to create objects where we have a concentration of thresholding responses whereas isolated pixels will disappear.

All in all we need seven different thresholds. We need three in the grayscale image, one to segment the long defects (splits, pith, pitch pockets, dark wane edges) from the clear wood, one to distinguish sound and dry knots and one to distinguish wide splits from pith and pitch pockets. We need two thresholds in the tracheid image, one to segment knots from the clear wood and one to distinguish pith from pitch pockets. Finally we need two thresholds in the difference image, one to segment light wane edges and one to segment surface roughness defects.

6.2.2 Thresholding in MAPP

Thresholding is a very straightforward operation in MAPP. The readout logic was described in detail in Chapter 3. A sensor row is transferred to the readout logic using the READPD instruction and a global threshold level can be set by the SETV instruction. The thresholded binary data is then directly available in the PD register. However, the hardware thresholding scheme can not be used in this case since we use the two main sensor lines to calculate both the sum and the difference. Instead we must A/D-convert the two lines and threshold the sum and the difference digitally.

We also have a problem with shading in the laser illumination, at least if we use gaussian laser line projectors. The MAPP sensor does not contain any shading correction logic and to implement shading correction in the processor array would require a multiplication which is time consuming. Since the only thing we do with the data is thresholding, we may instead shading correct the thresholds in advance. This is not possible if we use the hardware thresholding feature, but can easily be implemented in the digital thresholding by precomputing and storing different thresholds for the different processing elements.

In any case, the result from each threshold is a binary bit vector containing a one in those processing elements where the image data is *below* the threshold for the grayscale and tracheid images and *above* the threshold for the difference image. We also need an additional threshold to detect the edges of the board. This threshold level will not be very critical if we have a dark background and we may thus use the hardware threshold. Preferably, we use the left illuminated line to detect the left board edge and the right illuminated line to detect the right edge. To detect the leftmost and rightmost edges in a bit vector is easily done using the MAPP GLU. By using the COUNT feature, which counts the number of ones in the accumulator, we may locate the positions of the edges and store for later processing, see the pseudo code below. Apart from that, in the segmentation step it will be convenient if we keep an edge mask as a bit vector together with the other thresholded data. This mask should be shrunk one or a few pixels to avoid false data along the edges.

In the lengthwise direction we also need to use the board edge threshold to trigger the scanning of a new board and to detect when the board end is reached. This is simply done by thresholding and using the COUNT feature in MAPP to count the number of pixels above the threshold.

All in all, the thresholding part can be described by the following MAPP pseudo code. For details on the MAPP instruction syntax, refer to [111]. A graphical illustration of the edge detection sequence is shown in Figure 6.1.

```
SETV <egde_threshold>          /* set edge threshold */
SETR <left_line>               /* read left line */
READPD
LD LMARK PD                    /* detect left edge */
ST A, R0
left_edge = COUNT
A/D-convert and store
SETR <right_line>              /* read right line */
READPD
LD RMARK PD                    /* detect right edge */
right_edge = 256 - COUNT
OR R0
LD (000) A                     /* invert and shrink */
LD (111) A                     /* shrink again */
ST A, R0                       /* store edge mask in R0 */
A/D-convert and store

Add left and right lines       /* apply gray thresholds */
Compare 1st gray threshold
ST result, R1
Compare 2nd gray threshold
ST result, R2
Compare 3rd gray threshold
ST result, R3

Subtract left and right lines  /* apply diff. threshold */
Take absolute value
Compare 1st diff. threshold
ST result, R4
Load subtracted line again
Apply [-1 1] filter
Take absolute value
```

```
Compare 2nd diff threshold
ST result, R5

SETR <first_trachline>          /* read tracheid image */
RESET
INTEG
SETR <2nd_trachline>
...
SETR <nth_trachline>
HOLD
A/D-convert
Compare 1st trach threshold     /* apply trach thresholds */
ST result, R6
Compare 2nd trach threshold
ST result, R7
```

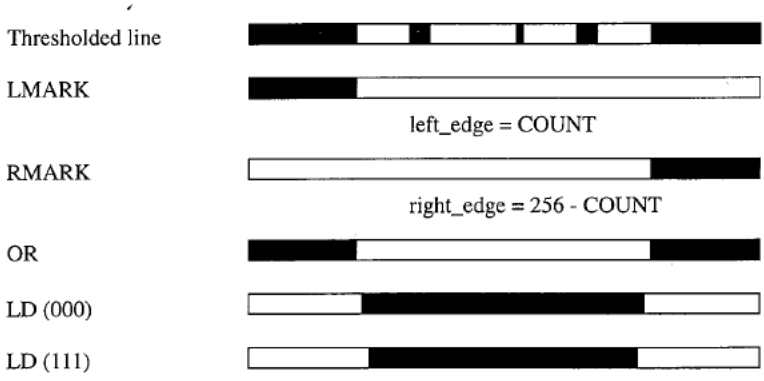


Figure 6.1 Detection of board edges and creation of an edge mask using the MAPP NLU and GLU.

After the sequence above, the results from the different thresholds together with the edge mask are contained in register R0-R7. We may now directly apply the segmentation based on this bit vector. Alternatively, we may use the shift register to store the thresholded line in an external memory for later processing. As will be evident from Section 6.2.4, the time needed for the segmentation is data dependent and if we apply the segmentation directly we must be able to handle the worst case. If we just shift out the bit vector we may scan the entire board first and segment afterwards. This will of course prevent us from scanning a new board directly after the first one since the segmentation takes some time. Alternatively, we may adopt in interrupt structure on the MAPP level where we use the external memory as a FIFO buffer and where the lines are segmented asynchronously with the scanning. This solution is quite complicated though. After all it is a trade-off of how much processing should be included at the MAPP level. Perhaps the thresholded bit vector is a suitable level of data reduction at the sensor and that the segmentation should be implemented somewhere else in the system.

6.2.3 Adaptive thresholding

Fixed threshold levels will always give problems. First, if the illumination is not stable, i.e. if the intensity from the lasers varies in time, we will threshold at incorrect levels. It also turns out that different pieces of wood, even from the same species, look very different and require different threshold levels to segment the features correctly.

There are basically two ways to make the system adaptive. We could either adapt the exposure time or the threshold levels. Varying exposure time could in turn be handled in two ways. Either we use a fixed exposure cycle and reset the sensor lines at a given time slot within the cycle or we can adapt the entire cycle time. The first solution is quite complicated to implement and we will also waste available light since the full exposure cycle is not used. The second solution is simpler but results in a varying board scale in the lengthwise direction. This does not need to be a problem since we may mark each line with its true position using a rotary encoder connected to the conveyor belt.

There is however a difficulty in the variable cycle time solution in that the grayscale image and the tracheid image must be adapted separately. Depending on the surface structure of the board, the grayscale and tracheid image may show very different absolute levels and we can of course not have different cycle times for the two images.

The other solution to adaptivity is to keep a constant exposure time and adapt the thresholds themselves instead. If we use the hardware thresholding feature it will be quite simple to change levels from cycle to cycle, but as was pointed out above, we can not use hardware thresholding because of the difference image and the shading problems. Instead we must adapt the individual digital thresholds. Since a multiplication is very time consuming in MAPP, the easiest solution is to pre-calculate a number of thresholds, based on the shading profile, and then to load the thresholds via the shift register when they are to be updated. Grayscale and tracheid thresholds can thus be adapted individually.

Regardless of the type of adaptation, we need a measure of the darkness of the board from line to line. A scheme to calculate the total average level of one entire line, for instance in the grayscale image, is given below. We will exclude the pixels which are located outside the board edges as well as pixels that are part of a potential defect, i.e. below the first grayscale threshold. Care must be taken in this case if the exposure is totally out of range. In the following we assume that the grayscale line is located in MAPP register R0-R7, with the least significant bit in R0. The procedure utilizes COUNT in each bit plane and multiplies with appropriate weights.

```
LD R(<edge_mask>)                    /* create mask */
ANDI R(<1st_gray_Thresh>)
ST A, R(<adapt_mask>)
n = COUNT                            /* n = number in mask */
sum = 0
for (i=0; i<8; i++) {
    LD R(i)
    AND R(<adapt_mask>)
    sum = sum + COUNT * 2^i
}
average_level = sum / n
```

This measure is calculated for each line. We will then apply suitable filtering in the adaptation process, but this will not be discussed further here. For a general discussion on adaptive exposure control in MAPP, see Åström [21].

6.2.4 Run-length encoding in MAPP

Binary images can be segmented in MAPP using a standard line by line segmentation scheme described by Åström [21]. In our case we have four different thresholds that may generate objects, the first grayscale threshold, the first tracheid threshold and the two difference thresholds. This corresponds to four bit positions in the 8-bit vector resulting from the thresholding stage. The different types of objects are segmented separately.

The MAPP segmentation scheme works on a line by line basis. The first step is to encode the objects along the line using a run-length code, i.e. to record the start position and length of each object along the line. The MAPP GLU is well suited for this and the procedure for one single threshold can be described as follows. The procedure is illustrated graphically in Figure 6.2

```
LD R(<line>)                    /* binary thresholded line */
ADD R(<edge_mask>)              /* no objects outside board */
ST A, R0
LEDGE A                         /* how many objects? */
n = COUNT
for (i=0; i<n; i++)
    LEDGE LFILL R0              /* left edge of left object */
    RFILL A                     /* get position */
    pos(i) = COUNT
    MARK R0                     /* get length */
    length(i) = COUNT
    XOR R0                      /* remove object from line */
    ST A, R0
}
```

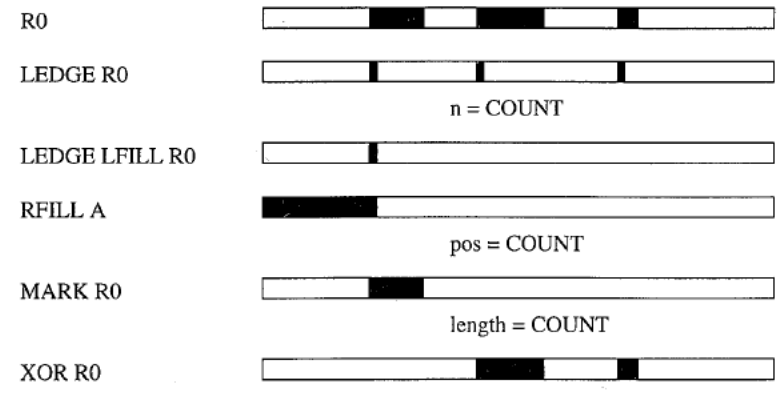


Figure 6.2 Run-Length encoded segmentation in MAPP.

The result from the first segmentation step is a list of objects for each line. The next step is to match and merge objects from line to line. It is possible to invent MAPP-based schemes for this but it turns out that this is no advantage compared to doing it in the host computer, in our case the PC. Since we no longer process full lines of data we can not take advantage of the SIMD structure in MAPP.

The PC processes one line at a time and each new object that appears is assigned a label. We then build up two-dimensional objects by letting the labels propagate to connected objects on the next line and so on. There are some special cases we must take care of, e.g. when two different objects with different labels are found to be connected and should be merged. The full procedure can be found in Åström [21] or in Hall [78].

6.3 Feature extraction

6.3.1 General

In a traditional image processing system, the segmentation step is followed by a feature extraction step where we calculate a number of feature values from the segmented regions. The features are then used to classify the regions in a subsequent classification step. In our case we only scan the images once and the feature extraction is therefore an integrated part of the segmentation. This means that we except from the position and width for each object along a given line also acquire additional features during the run length encoding. Typical low level features are number of pixels below additional thresholds etc. When all lines of an object have been processed, we then calculate a number of high level features which are used in the classification. The high level features include shape parameters, darkness parameters etc.

In the following sections, the features used in our experimental system will be described. No features for roughness detection, deviating grain direction, geometrical properties or color defects are included in the present feature extraction, although different alternatives are discussed and included in the total system performance estimation.

6.3.2 Low level object features

During the segmentation we will collect information about the objects in a table. Each object has an entry in the table and if several objects need to be merged, the corresponding table entries will be merged as well. The table contains the following low level features:

type Indicates from which threshold the object is generated. 0 is the first grayscale threshold, 1 the difference threshold and 2 the first tracheid threshold.

min_x Minimum x coordinate (in pixels from left image edge).

max_x Maximum x coordinate.

min_y First line of the object.

max_y Last line of the object. The min and max features give a bounding rectangle around the object.

area Number of pixels in the object, i.e. the number of pixels below the object generating threshold.

gr2 Number of pixels below the second grayscale threshold. This value is easily obtained for each line of an object by AND'ing the corresponding bit vector with the object and reading COUNT.

gr3 Number of pixels below the third grayscale threshold.

tr2 Number of pixels below the second tracheid threshold.

edgecnt Number of lines within the object which touch the edge. To test if a line touches the edge, expand the object once and AND with the inverse of the edge mask. Then read the result using GOR (Global Or).

6.3.3 High level object features

From the low level features we calculate a number of high level features which are used in the classification. This involves changing coordinates from pixel positions in the image to true positions along the board. To be able to do this we need to include additional information such as board edge positions and rotary encoder positions for each line.

The feature conversion also includes a filtering step where objects for one reason or another are removed from the list. A typical example is very small objects which only cover one or a few pixels. The high level features used in our system are:

y Lengthwise position for the defect mid point given in mm from the beginning of the board.

x Crosswise position for the mid point in mm from the board edge.

length The length of the object in mm calculated as the encoder position of the last line of the object minus the encoder position of the first line.

width The width of the object, or more correct of the bounding rectangle, given in mm.

type Same as the low level **type**.

elong Elongation, i.e. the relation between the length and the width of the object.

rel_gr2 The relation between the number of pixels below the second grayscale threshold and the object generating threshold, i.e. the fraction of the object area below the second threshold.

rel_gr3 Fraction of the object below grayscale threshold 3.

rel_tr2 Fraction of the object below tracheid threshold 2.

rel_edge The relation between the number of lines touching the edge and the total number of lines in the object.

6.3.4 Geometrical features

Apart from the thresholded object features, we may also calculate some geometrical features based on the edge positions and the range image information. These features can then be used to calculate geometrical parameters as those given in Figure 4.1 in Chapter 4. This has so far not been implemented in our system but a typical low level geometrical feature extraction from the range image could be to detect the corner and mid positions for each cross section of the board as shown in Figure 6.3. Here we only use the range information for global shape features and not to detect wane edges. If we have a wane edge we will have false results from the corner measurements but since wane edges are detected in the difference image we know where this occur and may disregard the data.

Assume that we want to get the average range values from a five pixel neighborhood in each one of the positions in Figure 6.3. This can be implemented in MAPP as follows:

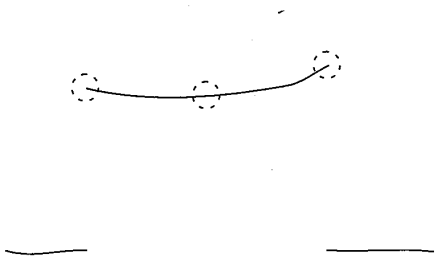


Figure 6.3 Extracting geometrical features from a cross section of the range image.

```
LFILL R(<thresh_line>)          /* get position of left edge */
lpos = 256 - COUNT
LEDGE
LD (1XX) A                      /* shift right two steps */
LD (1XX) A
LDI (00X) A                     /* expand right four steps */
LDI (00X) A
LDI (00X) A
LDI (00X) A
ST A, R(<mask>)
lval = 0
for (i=0; i<8; i++) {           /* read and sum bit planes */
    LD R(<range_bit_i>)
    AND R(<mask>)
    lval += COUNT * 2^i
}
lval = lval / 5
```

A similar procedure is used to detect and measure the right edge. To measure the mid position we must first create a mask. This can be done by pre-loading a ramp in a set of MAPP registers, i.e. each processing element is identified by an 8-bit number. By AND'ing and ANDI'ing the corresponding bit planes according to the position, the result will be a single one bit where the processor number match. The mask is created by expanding the bit two steps in each direction and the value is then read using COUNT in the bit planes.

We can also use the range data as well as the edge positions to detect dimension faults, i.e. where the width and/or height of the board does not meet stipulated limits. A convenient way to handle these dimension faults is to let them generate objects which are added to the list for subsequent classification. Here we can use the **type** field to indicate that this is a dimensional object.

6.3.5 Color features

If suitable color filters were available, i.e. filters mounted directly on the sensor, we could integrate the real time spectrograph methods, described in Chapter 5, into our sensor system. In this case it would be a limited system where color is only used to detect predefined color defects, such as blue stain. The result from the color measurement is classified pixels, and the classes would typically be one or a few color defects plus clear wood.

Since at least blue stain covers large areas without any distinct borders, we would probably need some kind of filtering where we expand and shrink to fill holes and to remove isolated pixels. MAPP is very well suited for this type of binary image processing. After filtering we can apply the segmentation scheme to build objects which are marked with yet another code in the **type** field.

6.4 Defect classification

6.4.1 General

The classification of the objects into defects is a traditional classification problem. Each object is characterized by a feature vector. In some cases the classification is trivial, e.g. a dimension fault or a blue stain area is given directly by the type code of the object. Other distinctions, for instance between a sound and a dry knot, can be more difficult to make.

A fact that complicates things is the lack of a correct answer. Different human graders can have very different opinions on for instance the categorization of a knot. On the other hand, if the graders are allowed to communicate with each other they could probably agree on the classification. This indicates that there are overlapping zones with classifications that can be considered correct and that the system is quite forgiving when it comes to the borderline cases. This must be considered when evaluating the results. Experiments have shown [108] that if a human grader first assigns classes to a number of defects and we then compare with the machine, the result will be much worse than if the grader just compares the output from the machine with the boards.

Another problem is that there is often not a one to one relationship between classified defects and "real" defects. A typical example is a long split which physically often is divided into a number of smaller splits. If the length of the split is a parameter in the grading of the board we must be able to merge the splits into one single defect. The opposite case arise when two knots are located close to each other and the system detects them as one single defect.

6.4.2 Secondary defects

Grading rules may state that a single defect of a certain type may be allowed but that a cluster of similar defects is not. We may also have a rule like a knot being admitted if it does not go through the board, i.e. if we can not find a corresponding knot on the other side. These situations can be handled in two ways. One way is to just classify the individual defects and leave it to the subsequent optimization to decide what to do. The other approach is to define the set of defects as a *secondary defect*, i.e. a defect based on other defects. The secondary defect is then handled as any other defect in the subsequent optimization. A cluster of worm holes or a knot through the board is probably better handled as a secondary defect while rules stating that a certain number of sound knots are allowed per meter of the board probably should be handled by the optimization. This will be further discussed in Chapter 7.

6.4.3 Classifier alternatives

A lot has been written about pattern classification. For a survey see Duda and Hart [52] or Schalkoff [144]. In the wood inspection field we can note from the review in Chapter 2 that the systems are approximately equally divided into those using some kind of learning classifier based on a number of training samples and those using a knowledge based approach.

A learning classifier, be it a statistical classifier or a neural net, is attractive in the sense that it is conceptually simple. Just show a number of training samples and then everything works. In reality it is not that simple. The performance of any classifier is heavily dependent on the

features selected. If the features have good discriminating capability, then any classifier will do but if the features do not contain enough information then the most sophisticated classifier will fail. We thus believe that the selection of features are more important than the selection of the classifier.

The selection of training samples is also very important. A problem here is that even in a quite large material, it could be difficult to find enough representatives for some less frequently occurring defects. We must also cope with long term variations in the material, i.e. it may be necessary to collect training samples during a longer period in time. Another problem is the inhomogenity of the classes themselves. A round sound knot can show very little similarity with a sound horn knot even though they belong to the same class. Just as in the case of the pixel classification in Chapter 5, it may be necessary to define subclasses of given defects.

The alternative to a learning classifier is a knowledge based network in which we apply decision rules on the features. Typically this can be implemented as a tree classifier where we at each node test on a given feature and then follow different branches. The major drawback with this approach is that we have a large number of parameters to set and experience has shown that the trimming of the decision parameters is an endless task where one never gets satisfied. This is most undesirable in a commercial system where the classifier must be robust without too much parameter adjustments in every individual installation. The system also easily becomes unstable with a lot of thresholding effects where one gray level of a single pixel may determine the classification of the defect and thereby indirectly the grading of an entire board.

On the other hand, the rule-based classifier gives us better possibilities to directly manipulate the classification where the problems occur and to incorporate new features to discriminate between new classes, e.g. if we need to refine the subdivision of different types of knots. If we use a learning classifier we need to re-train the entire system, and since the original training data may not be available with the new feature, we may need to start from the beginning again with the collection of training samples. We therefore believe that a rule-based approach is the method of choice, at least during the development of the system where different feature combinations are tested and where we continuously add new defects and new features. It is however possible that a learning classifier is preferable in a commercial system when robustness in installations is a key factor.

6.4.4 Error handling

An important aspect on the classification is the handling of erroneous classifications. The economical consequence of an error can vary a lot depending on the type of defect and the inspection situation. A false positive split may for example waste an entire board in a sorting situation while it in a cutting situation only causes a local degradation with all evidence of the error disappearing on the waste belt. A missed split on the other hand may be quite harmless in a green sorting application where the boards are to be sorted again after drying, but can be very expensive in a window production if an entire window must be discarded in the final inspection.

This dependency is most undesirable since we want the classification of the defects to be independent of the subsequent decision stage. Unfortunately we have to accept that incorrect classifications always will occur to a certain extent and we therefore need to incorporate a mechanism to handle the cost. In the rule-based classifier we may modify certain threshold levels to adjust the relation between false positive and false negative classifications. If we use a statistical classifier we need to include the cost or loss function for given defect combinations into the classifier itself.

6.4.5 Contextual dependencies

Most of the systems reviewed in Chapter 2 classify the defects on an individual basis. However, by exploiting knowledge about how different defects appear on a board we may improve the classification. If we for instance have detected and classified an obvious split on a board, then there is an increased probability that the long split-like object next to it also is a split. Another example is that sound and dry knots are unlikely to be randomly mixed on a single board. It is also very unlikely that a lengthwise cut horn knot is located just next to a perfectly round knot growing in a perpendicular direction.

This type of contextual information is not very easy to include in a classifier. Probably we need to classify the defects in two steps where we first classify the obvious cases and then in a second pass use the result to classify the remaining objects.

A very interesting hierarchical classification procedure where contextual information can be included has been proposed by Pölzleitner [130]. The decision rules are formulated using a special grammar and the classification network is then compiled into efficient code. Context information is recorded in a global working memory and can be accessed by the decision rules. The procedure also has a number of other interesting features. First, each decision rule returns a *certainty factor* indicating the confidence level of the decision. The certainty factors are defined as fuzzy sets, i.e. they have values in the interval [0,1] where 1.0 represents a perfect decision and 0.5 total uncertainty whether the decision is correct. Second, each decision rule may employ a number of different procedures to perform the decision. The paper includes three procedures: interval tests, Fisher discriminant analysis and *km*-nearest neighbor classification. The procedures all return decisions combined with certainty factors. We thus have a hybrid classification network which both employs statistical classifiers as well as a priori knowledge and context information.

The paper does not give any decision rules for the particular wood inspection problem. Instead it should be regarded as a tool for experimentation. We believe that this approach looks very promising providing a framework into which decision rules, a priori knowledge and contextual information can be included. Much work remains however to really find the appropriate decision structure to implement in the network.

6.4.6 A rule-based tree classifier

In our system we have implemented a relatively simple rule-based tree classifier which in each node performs a test on a given feature value and which finally reaches a decision on the class. Alternatively, the network may find that the object in question does not correspond to a defect, in which case it is disregarded.

The classifier is not claimed to be the final solution to the wood defect classification problem. Contextual information has for instance not been included. Neither have color defects, surface roughness and geometrical properties other than wane edges been included. The main purpose is to show that the sensor data provides so much significant information compared to a traditional system, that even a relatively simple segmentation and classification scheme can produce relatively good results. It can thus serve as a starting point for further research and improvements. It is not unlikely that we will end up with a classifier having a structure similar to Pölzleitner's [130].

The present structure of the classifier is given below. If a branch in the tree is omitted, this means that the corresponding object is left without further action. Many of the threshold values in the different nodes can of course also be subject to adjustments in different situations. Note also that a given defect may appear in several sections of the tree. A wane edge can for instance be detected both in the grayscale and the difference image.

```
If type = 0     /* grayscale */
  If elong > 3.0
    If elong > 4.0 And rel_gr2 > 0.2 And
       width < 4.0 And rel_edge < 0.05
      SPLIT
    Else
      If rel_edge < 0.05 And length > 12.0
        If rel_tr2 < 0.2
          PITCH_POCKET
        Else
          PITH
      Else
        If length > 10.0 And rel_edge > 0.05
          WANE_EDGE

If type = 1     /* difference */
  If rel_edge > 0.05 And length > 10.0
    WANE_EDGE

If type = 2     /* tracheid */
  If width > 4.0 And length > 4.0
    If elong < 3.0 Or (rel_edge > 0.05 And elong < 6.0)
      If rel_gr3 < 0.1
        SOUND_KNOT
      else
        DRY_KNOT
```

The classification tree basically implements the classification according to the discussion in Section 6.2.1. An elongation limit is used to distinguish between short defects, i.e. knots, and long defects such as wane edges and pitch pockets. Splits are basically distinguished from other long defects using the second grayscale threshold in combination with a width limit. Splits are not allowed to touch the edge since we often, especially when scanning rough boards, have splinters along the edges which incorrectly can be classified as splits.

Wane edges on the other hand are required to touch the edge whereas pith and pitch pockets should not. Pith and pitch pockets themselves are distinguished by the second tracheid threshold.

Knots are allowed to be more oblong if they touch the edge since the knot may be cut in the middle. Small knots, in this case less than 4 mm, are disregarded but this limit can of course be changed. Finally, the distinction between sound and dry knots is based on the third grayscale threshold.

6.5 Experiments

The complete system, from sensor level to defect classification, was tested on a number of spruce boards of the dimension 130×20 mm. The length of the boards were approximately one meter. Ten boards containing representative defects were first selected to set the parameters in the classification tree as well as other parameters such as threshold levels, adaptation filter parameters etc. The ten boards were repeatedly scanned followed by iterative adjustments of the parameters.

Then another 20 boards were selected for the final test. The spruce boards were planed on one side and rough on the other and both sides were tested without any change in parameters in between. 40 board faces were thus scanned all together. The result of the defect classification is shown in Table 6.1.

System / Real	Sound knot	Dry knot	Split	Pith	Pitch pocket	Wane	Missed	Total
Sound knot	122	3	0	2	1	1	7	136
Dry knot	3	39	0	0	0	0	1	43
Split	0	0	2	0	4	0	5	11
Pith	0	0	0	7	0	0	0	7
Pitch pocket	0	0	0	0	17	2	0	19
Wane	1	0	0	0	1	27	0	29
False	11	1	0	0	5	4	-	21
Total	137	43	2	9	28	34	13	266

Table 6.1 *Result from the classification test on planed and unplaned spruce boards. "False" means that a piece of dirt or something else was incorrectly detected as a defect while "missed" means that a real defect was missed by the system.*

The judgement of the system was quite forgiving, e.g. if only 75% of a split was detected it was considered correct. If a wane edge was detected as five wane edges with small gaps in between this was also OK. Knots were accepted if the type were correct, even though the size in some cases were up to 30% wrong. The table will thus not give the full picture but some conclusions can still be drawn from it. The main conclusion is that the detection of knots were quite reliable, even very light sound knots on unplaned surfaces. This is significant since no other methods to detect light knots on rough surfaces have been reported.

The detection of splits is on the other hand very poor which can be explained by the low crosswise resolution of the system, 0.75 mm in our experimental setup. Many of the splits were quite narrow resulting either in a complete miss or in an object where the pixels were only partly covered by the split. This means that the object appears lighter and is therefore sometimes mistakenly classified as a pitch pocket.

The detection of pith and pitch pockets are reasonably good. Most of the false detections actually occurred in the beginning of the boards since it took some time for the adaptive thresholds to settle. Recall that the boards were planed on one side and unplaned on the other and since the two sides of a board were scanned after each other, the adaptation had to adjust the thresholds all the time.

Most of the wane edges were detected even though it must be admitted that they in many cases where subdivided into several sections. This could be handled in a post processing stage where the small sections are merged.

6.6 Speed performance

Of key interest is the speed performance of the complete system. In the experiments, the PC version of the MAPP2200 camera was used, which limits the instruction frequency to 700 kHz and which does not permit simultaneous access to the shift register. Neither can processing in MAPP be performed simultaneously with other processing in the PC, e.g. the defect classification, since the PC must send the MAPP instructions one by one. The run time experiments can therefore not be used to measure the performance of the system.

We could, however, count the number of instruction cycles required at the MAPP level, and thereby estimate the performance of a full-scale system. If a dedicated hardware controller is available, we can assume that the MAPP instructions can be fed to the chip at full speed and without overhead. It is also reasonable to assume that the PC processing for a given board, i.e. from the high level segmentation to the defect classification, can be performed within the time limit of the low level MAPP processing. We will therefore concentrate on the MAPP cycle count and assume that the system can be pipelined so that the PC operates on one board while the MAPP system processes the next. It is also assumed that the processing in the MAPP processor array is the limiting factor, i.e. A/D-conversions and shift register operations on the chip are considered to be transparent to other processing. However, move operations to and from the shift and A/D registers are included.

The cycle counts for the different sensor principles were discussed in Chapter 3, 4 and 5 respectively. In Table 6.2, we summarize the cycle count for the complete system. Low level acquisition of range and color data, as well as the roughness detection, are included although the experimental system did not make use of it. The range data for instance is only thresholded with features extracted according to the scheme in Section 6.3.4, even though it is quite likely that more advanced operations actually should be performed. The lengthwise grayscale measurement is not included in the table since the acquisition of the raw data is an integrated part of the range measurement and the length grayscale data itself has so far not been used.

The range image algorithm used is TIM. This is because this algorithm gives range data directly on the MAPP sensor. The faster algorithms FBM and USM require additional hardware support to handle table lookup operations.

The total cycle time of the system is data dependent since the number of objects on a line varies. If the thresholds are properly adapted we will have very few objects, in most lines none at all. However, if a threshold happen to be approximately at the average board level we could end up with a lot of false objects, in the worst case 128 object in each of the three lines. Since this is very unlikely to occur we could probably set a limit on the number of objects we can handle on each line. If more objects occur we just disregard them. We thereby run the risk of missing real objects and depending on the application this could be more or less serious. Since we know when this occurs, we always have the option to generate an artificial "object overrun" defect which can be forbidden in all quality classes. It would then just be a matter of statistics how much we loose due to the object number limitation.

Another approach instead of limiting the number of objects is to use some kind of buffering mechanism where we shift out the thresholded bit vector in each exposure cycle and then reload and process the lines asynchronously with the exposure. This is quite complicated since we will have several MAPP processes running simultaneously. It is, however, possible with a suitable instruction sequencer. Interrupts in MAPP are not so difficult to handle since only the accumulator and the carry registers need no be saved. A little bit more difficult is the synchronization of the shift register which will be a narrow resource. This will not be discussed further here.

Measurement	Operation	Cycles	Total b=8
Grayscale	Acquisition	5	
	Storing	2b	
	Summation	3b	
	Thresholding	2b+1	62
Difference	Subtraction	3b+2	
	Absolute value	4b-2	
	Thresholding	2b+1	
	[1 -1] filter	3b+2	
	Absolute value	4b-2	
	Thresholding	2b+1	
	Exp. and shrink	20	165
Tracheid	Acquisition	14	
(10 lines)	Thresholding	2b+1	31
Range	Acquisition	645	
(TIM, 160 lines)	Gray conversion	2b	
	Thresholding	2b+1	
	Edge features	10b+23	781
Color	Acquisition	7	
(3 filters, 3 classes)	Subtraction	3b+2	
	Comp. and store	4b+18	83
Adaptation	Average value calc.	3b+4	
(gray and trach.)	Threshold reload	5b+5	73
Segmentation	R-L encoding	28n+20	
(n objects per line)	Feature extraction	36n	64n+20
Total			64n+1215

Table 6.2 *Cycle count estimations in the integrated MAPP wood inspection system. n is the number of objects on the line and b the number of bits used in the calculations.*

Let us return to Table 6.2 and assume that we set a limit on the number of objects to 10 in each line of the grayscale, difference and tracheid images. The total cycle count will thus be approximately 2000 instructions for one scanning cycle. Assuming 4 MHz MAPP instruction frequency gives 500 µs cycle time or 2000 lines per second. We can thus run the boards at a speed of 2 m/s and still use a 1 mm resolution in the lengthwise direction.

It should however be noted that the range image acquisition is responsible for 40% of the cycle time. If we instead use FBM, the range imaging speed can be drastically increased but since FBM requires a 16 to 8 bit table lookup, we then also need additional hardware support outside MAPP. If this is arranged, we could probably reach approximately 350 µs, or nearly 3000 lines per second.

6.7 Conclusions

We have shown that a compact wood inspection system can be built around only a MAPP camera and a PC. The speed of the system meets the industrial demands and the defect detection capability is sufficient except for splits where the low resolution of the MAPP sensor sets a limit. Apart from the poor split detection, we believe that both the speed and the defect detec-

tion capability of the system is in range with present commercial systems, whereas the total cost of the system is much lower. When a higher resolution sensor is available, the system will be fully competitive.

The system is however not in any way fully developed. Future work will concentrate on the full integration of the remaining sensor principles, i.e. dimension data, deviating grain and color. A major concern is to find the optimum balance of processing within and outside the MAPP chip. Much also remains to be done in the classification stage.

Chapter 7

Cross cut optimization

7.1 Introduction

7.1.1 Overview

In Chapter 1 we used the term optimization to denote the decision stage where we make a decision about the use of the board based on the positions and types of different defects. The type of decision varies from application to application. In a sorting application the decision is just a single grade assigned to the entire board while in a cutting application the result is a complete list of cutting positions and grades of the different pieces. In this chapter we will take a closer look at the cross cutting application.

The cross cutting application in general was described in Section 1.3.3 and a real world example from the cutting of window casement parts was given in Section 1.3.4. The manual alternatives using either a completely manual saw or crayon marking were discussed. It was found that an automatic environment, where we have full knowledge of every individual defect, which is not the case when using crayon marking, gave rise to new optimization possibilities having no counterpart in the manual environment. In this chapter we will present a dynamic programming optimization algorithm which can take full advantage of the new possibilities with different quality requirements on the different sides as well as within different zones in both the cross- and lengthwise direction of the pieces. The algorithm will also be extended to handle a full mixture of fixed and variable length products.

In this chapter, the term *product* will be used to denote the wood pieces resulting from the cutting operation while *the board* refers to the incoming material which is subject to the optimization. The term *quality rules* is defined as the set of rules describing the products, i.e. what is allowed and what is not allowed in terms of defects such as knots and splits.

7.1.2 Previous work

The problem of stock cutting in one or more dimensions has been the subject of several books, conference papers and articles since the classic work by Gilmore and Gomoroy in 1965 [65]. A review of packing problems in general is given by Dowsland and Dowsland [50] and a review of industrial packing applications is given by Whelan and Batchelor [166]. Cutting wood in one or more dimensions has been addressed by several authors [29, 34, 61, 95, 169] but these references deal only with the optimal use of clear wood areas from a material containing inadmissable defects. Defects are handled in a more general way by Sarker [143], who discusses value as a function of defect contents and Sweeney and Haessler [157], who introduce multiple quality grades. Although none of the two latter references address cutting of wood directly, the problem discussed by Sweeney and Haessler is actually identical to the old crayon marking optimization. None of the algorithms reported are capable of handling more complicated quality rules, such as those described in Section 1.3.4.

7.1.3 Comparison with the crayon marking optimization

In a crayon marking system, the operator partitions the board into zones in the lengthwise direction using fluorescent crayon. Alternatively, the operator may mark the zones by pointing with a laser line. Each zone can be given a quality label, say A-E, even though many crayon marking installations only use two, namely usable wood and waste. In the general case, each product is assumed to correspond to one single given quality grade. To be able to cut that particular product from the board, the quality label of the board must be the same or higher along the entire length of the product as shown in Figure 7.1.

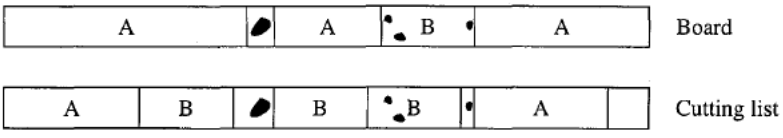


Figure 7.1 *Crayon marking optimization. The board is partitioned into quality zones. To cut a product with a given quality label, all matching zones of the board must be of equal or higher quality.*

The crayon marking system can not handle different quality requirements in different zones of a single product. In our system, we allow an arbitrary number of quality zones within each product, both in the cross- and lengthwise direction. Each zone is then defined in terms of admissible defects. Thus, the optimization problem will not only include the placement of different products, but must also take the possible rotations of all products into account.

It should be noted that the hierarchical structure of the quality labels used in a crayon marking system does not exist in our system. A given cutting is either admissible or not at a given position along the board, depending on the location of the defects, but this does not imply that a given product is of higher quality than another, they are just different. The only way in which such an hierarchy is reflected is in the values of the different products in the cutting bill. The quality zone labels A-E in Table 1.3 on page 12, should therefore only be regarded as *different* quality zones without any implicit relation.

7.1.4 A simple example

To illustrate the optimization procedure, a rather simple example will be used in this chapter. The products are defined according to the following. See also Figure 7.2.

- Boards with a square cross section, say 10×10 cm.
- Three different products, 40, 60 and 80 cm respectively.
- Knots larger than 30 mm are not allowed at all in the 40 and 60 cm products.
- Knots smaller than 30 mm are allowed, but only on one side on the 40 and 60 cm products.
- The first and last 10 cm of the 40 and 60 cm products must be free of knots on all sides.
- The 80 cm product may contain any knots.
- The values of the products are 10, 20 and 5 respectively.

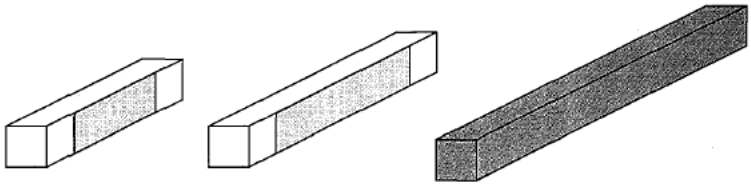


Figure 7.2 *A simple example of quality rules to illustrate the optimization procedure. White = no knots allowed, light gray = small knots allowed and dark gray = all knots allowed. Values are 10, 20 and 5 respectively.*

7.2 Generating an allocation list

7.2.1 Quality description of the board

The generation of the cutting list from defect data is divided into two steps. The first step takes the defect list as input and generates an *allocation list* showing where and how the different products may be positioned and oriented along the board without violating the quality rules. The second step uses this list and generates an optimal cutting list based on the values of the different products.

The allocation list consists of a sequence of bit fields covering different zones along the board. A single bit set in a certain zone indicates that the *starting point* of a given product with a given orientation may be allocated in that zone. Before scanning the board, we assume that it is completely defect free. This means that all allocation bits will be set to one along the entire board as long as the products fit within the total length. In our example from Section 7.1.4, we had three different lengths, one totally symmetric and two which can be turned in four different ways. Note that the number of turnings would have been eight if the lengths were asymmetric in the lengthwise direction. In this example we need a total of nine allocation bits, one for the 80 cm piece and four for the 60 and 40 cm respectively. An initial allocation list for a "clean" board is shown in Figure 7.3.

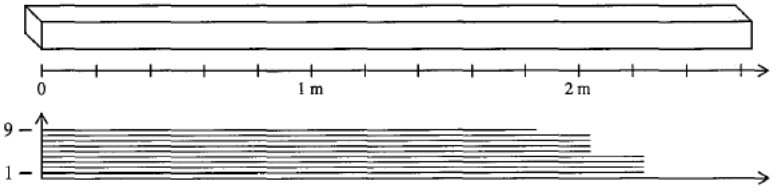


Figure 7.3 *Initial allocation bits.*

The scanning of the board results in a list of defects, all with a given type, size and position. Each such defect disqualifies one or more allocation bits in one or more zones of the board. An example showing the result of a big knot is given in Figure 7.4. Note that the knot is allowed in the 80 cm length. When we have gone through the entire defect list, the result is the final allocation list which is passed to the optimization.

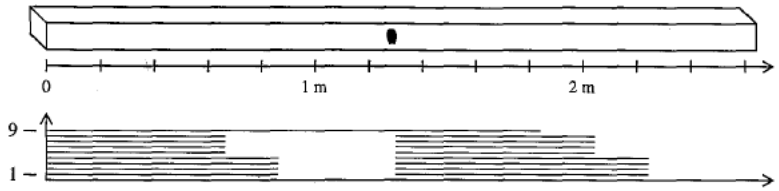


Figure 7.4 Allocation bits disqualified by a big knot.

7.2.2 Disqualifying bit lists

In order to efficiently apply the sometimes rather complicated quality rules to the allocation bit list of the board, some kind of preprocessing of the quality rules is necessary. The objective is to reduce the amount of computations needed during the actual production phase. This can be done by transforming the quality rules into *disqualifying bit lists*, which can be merged into the allocation bit list of the board using simple logical AND operations. An example of a disqualifying list is shown in Figure 7.5. The example corresponds to a small knot on one certain side in the example from Section 7.1.4.

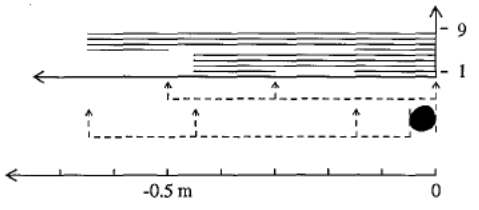


Figure 7.5 Disqualifying bit list for a defect found on a given side of the board.

A solid line in Figure 7.5 indicates that the corresponding product may not be allocated at that position, relative to the defect, without violating the quality rules. Recall that in our example we could turn the 40 cm product in four different ways, allowing the knots on one of the four sides. Bits 2 to 4 correspond to the three turnings allowing knots on other sides that the one where this particular knot was detected. Therefore, these cuttings may not be allocated at all within a zone starting at a position 40 cm in front of the knot and ending just behind the knot. Bit 1, however, corresponds to the turning that allows knots on this side as long as they do not affect the ends. The result is a gap in the disqualifying list where the product may be allocated, resulting in a product where the knot does not touch any of the ends. Bits 5 to 8 have a similar appearance, but correspond to the four different turnings of the 60-cm product. Finally, bit 9 represents the 80-cm lower grade product. The absence of a solid line in Figure 7.5 for bit 9 indicates that this particular knot does not affect the 80-cm product at all.

The disqualifying bit lists are generated from the quality rules and the cutting bill as a preprocessing before production starts. They cannot, however, be fully established in advance as they also depend on the size of the defect. As shown in Figure 7.5, all positions in the list are relative to either the beginning or the end position of the defect, indicated by the dashed lines. During production, these values are taken from the input list of defects. The final disqualifying

list can then be established, applied to the correct position along the board and merged into the allocation list using logical AND operations. Note that if the defect covers a longer zone, positions in the list may change places as shown in Figure 7.6. Note also the difference in scale in the figures and that the solid lines in Figure 7.5 represents the zones that will be cleared in Figure 7.3.

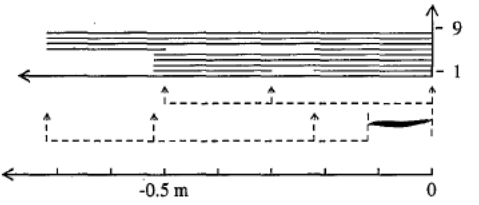


Figure 7.6 Bit list zones overlapping each other when the defect is long.

Two other parameters must also be considered when generating the allocation list. First, due to the environmental and mechanical exposure of the board ends, a trim cut, e.g. 5 mm, must normally be done at each end. This can be included when generating the initial allocation list. Second, the width of the kerf, e.g. 4 mm, must be taken into account. The easiest way to handle this is to make all products four mm longer and add a four mm zone at the end when generating the disqualifying bit lists. By allowing any defects within this small zone, no usable wood is wasted in the kerf when a product is directly followed by a defect.

7.2.3 Decision tree

Different disqualifying lists are required for different defects. Depending on the quality rules, different lists may also be needed depending on the size of the defect, on which side the defect was detected and sometimes also on the position of the defect in the crosswise direction on a certain side. One example of the latter case is the rules shown in Figure 1.13 on page 12, where on several sides there are two different zones with different quality requirements.

During the preprocessing of the quality rules, a decision tree is formed for each defect type. The tree includes tests on the different defect properties and the leaf nodes contain different disqualifying lists into which we only have to add two offset values before they are merged into the board allocation list. The appearance of the tree corresponding to our example from Section 7.1.4 is shown in Figure 7.7. In this particular case, no crosswise position dependence exists.

Tests on the position in the cross direction could be handled in two different ways as shown in Figure 7.8. If a defect affects two zones, we could either have exclusive tests in the tree leading to a single list, or we could use several paths in the tree simultaneously. The resulting lists would in that case be merged together before they are merged into the board allocation list.

When all detected defects have been processed, the result is the final allocation list for the board. A sample board, which for simplicity only contains knots on the two visible sides, is illustrated in Figure 7.9. The resulting allocation list for this board is also shown.

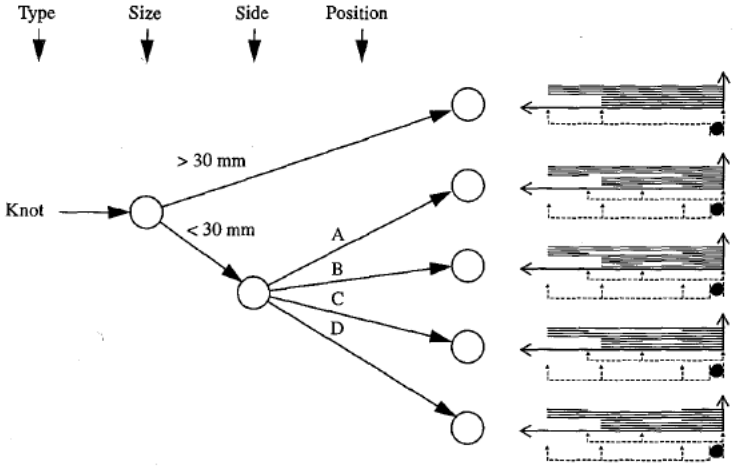


Figure 7.7 Decision tree to find the disqualifying lists for the simple example.

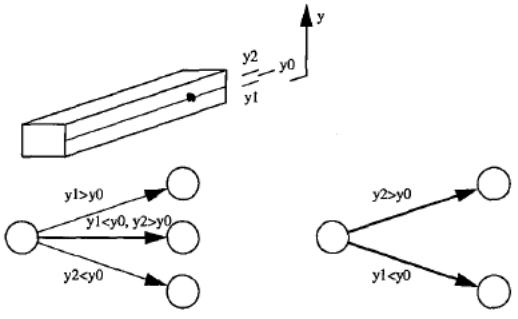


Figure 7.8 Two ways to handle position tests.

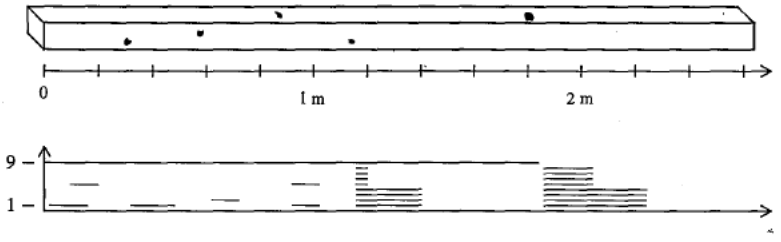


Figure 7.9 Final allocation list.

7.3 Optimization

7.3.1 Mathematical model

The optimization problem can be formulated in a number of different mathematical models, which are more or less suitable for a given solution method. Our formulation is based on the allocation bit model described in Section 7.2. A discrete model is chosen where the board is divided into n discrete points. The size of each step could vary but is typically 1 mm as this is reasonable resolution for both the scanning system as well as the saw. Moreover, the m products in the cutting bill are also normally given in mm.

Introduce the following notations:

$$x_{ij} = \begin{cases} 1 \text{ if product } i \text{ is allocated at pos. } j & i = 1,...,m \quad j = 1,...,n \\ 0 \text{ otherwise} \end{cases} \tag{7.1}$$

$$a_{ijp} = \begin{cases} 1 \text{ if product } i \text{ allocated at pos. } j \text{ covers pos. } p & p = 1,...,n \\ 0 \text{ otherwise} & i = 1,...,m \quad j = 1,...,n \end{cases} \tag{7.2}$$

$$c_i = \text{value of product } i \quad i = 1,...,m \tag{7.3}$$

The problem can now be formulated as:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} c_i x_{ij} \tag{7.4}$$

$$\text{s.t.} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ijp} x_{ij} \leq 1 \quad p = 1,...,n \tag{7.5}$$

$$x_{ij} \in \{0, 1\} \quad i = 1,...,m \quad j = 1,...,n \tag{7.6}$$

This problem is a so called set packing problem, see e.g. Nemhauser and Wolsey [120]. The constraints ensure that any position is covered by at most one allocation. Note that no constraints are included to explicitly ensure that an allocation does not violate the quality requirements. This is because the feasibility of each x_{ij} variable can be determined a priori from the board's allocation bit list. Those x_{ij} corresponding to inadmissable allocations are simply not regarded as part of the problem and will never be included in the solution procedure.

In our application, a_{ijp} has a certain structure which can be exploited in the solution procedure:

$$a_{ijp} = \begin{cases} 1 \text{ if } j \leq p \leq j + l_i & i = 1,...,m \quad j = 1,...,n \quad p = 1,...,n \\ 0 \text{ otherwise} \end{cases} \tag{7.7}$$

where

$$l_i = \text{length of product } i \tag{7.8}$$

7.3.2 Solution procedure

A solution procedure for this problem based on Lagrangian Relaxation was proposed by Rönnqvist [139]. Here we will solve the problem using classical Dynamic Programming which essentially decomposes a large problem into a series of more tractable smaller problems. General references on DP are e.g. Dreyfus and Law [51], Dernando [49], and Martello and Toth [116]. In DP nomenclature the problem has the following characteristics: The problem is divided into stages, represented by the discretization elements at which a decision is required. At every stage, the model can be in a number of states. States provide information needed to make an optimal decision at every stage. The decision made at every stage describes how the state at the current stage is transformed into the state of the next stage. Most importantly, the decision made at the current state is independent of previously reached states or previously chosen decisions. This is called the *Principle of Optimality.* In this application there is a one-to-one correspondence between stages and states. Finally, there will be a recursion which relates the reward gained from the current stage to that of previous stages. At each stage the decisions made to reach each state is recorded. A links or arc from one stage to another correspond to a product, and the value assigned to that link correspond to the product value.

In our formulation we have a discretization of n elements, and each such element will give a step in the solution procedure. Introduce the following notation:

$$S(k) = \text{the value of the maximum allocations in the interval } [1,k-1] \tag{7.9}$$

$$L(k) = \text{number of the allocation ending in } k-1 \text{ giving } S(k) \tag{7.10}$$

$$R(k) = \text{start point of the allocation ending in } k-1 \text{ giving } S(k) \tag{7.11}$$

$$S_{max} = \text{total maximum value} \tag{7.12}$$

$$R_{max} = \text{position giving max value} \tag{7.13}$$

The algorithm can be described as:

Initialize $S(k) = 0$, $R(k) = 0$, $L(k) = 0 \quad \forall k$, $S_{max} = 0$, $R_{max} = 0$
for $k = 1$ **to** $n+1$
  if $S(k) > S_{max}$
    $S_{max} = S(k)$
    $R_{max} = k$
  else
    $S(k) = S_{max}$
    $R(k) = R_{max}$
    $L(k) = 0$
  endif
  for each *admissible* allocation i at position k
    if $S_{max} + c_i > S(k + l_i)$
      $S(k + l_i) = S_{max} + c_i$
      $R(k + l_i) = k$
      $L(k + l_i) = i$
    endif
  endfor
endfor

When the algorithm terminates, S_{max} gives the optimal value, and R_{max} is the position of the last cut. The complete cutting list, i.e. the final x_{ij} values, is determined by following $R(k)$ backwards. The algorithm can be interpreted as walking along the board from left to right. At each

position, all admissible alternatives are tested and new values are updated at their end points respectively. The new value for each alternative is the sum of the maximum value reached so far and the value of the product. If when we arrive at a new position, the recorded value is lower than the maximum value achieved so far, the recorded allocation is replaced by an artificial zero value waste length from the position of the maximum value to the current position. This is done in the **else** branch in the first **if**.

The number of operations in the procedure is proportional to the length of the board (number of discrete points) and the number of products. The worst case is if the board is completely free from defects, which means that all products are allowed at all positions. Using real world numbers however, the complexity is not higher than what could be handled. The key is the structure of the input data which directly gives the feasibility of the allocations, otherwise the testing whether an allocation is possible would have taken too much time.

The number of required operations could however be drastically reduced by a simple observation. If we at any position of the board allocate a product following a waste section, then there can never be any point in not allocating it as early as possible, i.e. making the waste section as small as possible. This rather self-evident observation means that the only positions we have to investigate are positions where new possible allocations start and all the subsequent positions representing all length combinations of admissible products. In terms of the procedure above, only the first position, a position which previously has received a value and a position where a new allocation possibility starts need to be investigated. Furthermore, if a position has a previous value but this value is lower than the maximum value achieved so far, then this value can be disregarded. Note also that if a position is investigated only because of new possibilities, only the new ones need to be tested. Introducing the additional notation:

$$T = \text{set of allocations to test} \tag{7.14}$$

the modified algorithm can be described as:

Initialize $S(k) = 0$, $R(k) = 0$, $L(k) = 0 \quad \forall k$
$k = 1$, $S_{max} = 0$, $R_{max} = 0$
repeat
  if $S(k) > S_{max}$
    $S_{max} = S(k)$
    $R_{max} = k$
    $T =$ all admissible allocations at position k
  else
    $S(k) = S_{max}$
    $R(k) = R_{max}$
    $L(k) = 0$
    $T =$ all new allocations at position k (i.e. not admissible at $k-1$)
  endif
  for all allocations i in T
    if $S_{max} + c_i > S(k + l_i)$
      $S(k + l_i) = S_{max} + c_i$
      $R(k + l_i) = k$
      $L(k + l_i) = i$
    endif
  endfor
  $k =$ next position with a previously assigned value or new allocations
until $k > n + 1$

Like in the previous procedure, S_{max} gives the final value and the cutting list can be unwound via R_{max} and $R(k)$. $L(k)$ indicates backwards which product is allocated at each position. The value 0 denotes a waste section.

The procedure can be improved even further by observing that if a number of allocations with equal lengths and equal values are admissible at a given position, only one of them has to be tested. This situation will frequently occur as the allocations often are different orientations of each other. Equivalently, an allocation does not need to be tested if another allocation with a shorter (or equal) length and a higher (or equal) value is admissible at the same position.

This can be handled as a pre-processing of the allocation list where we remove all redundant allocation possibilities. Allocation bits corresponding to different orientations of the same product can from the optimization point of view simply be OR'ed together, but we may need to keep the orientation information for each zone, for instance if the products are to be automatically turned in a subsequent profiling operation.

After this step, we may remove allocations of lower value, i.e. if in a given zone a certain allocation is possible, then all other allocations with longer or equal length and lower or equal value can be removed. However, this does not necessarily lead to an improvement. Consider the situation in Figure 7.10a with one high and one low value length. If we exclude the low value length as shown in Figure 7.10b, then we introduce a new position where a new allocation possibility starts and this may lead to new subsequent unnecessary combinations being evaluated. Clearly, positions where new allocation possibilities start should be interpreted as positions where new and *better* allocations start.

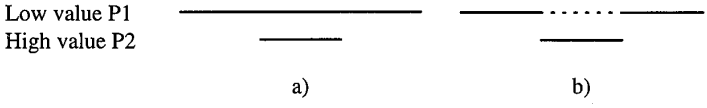


Figure 7.10 Excluding a low-value allocation P1 where a high-value product P2 is admissible may introduce a position where new allocations starts, and thereby unnecessary allocation attempts.

Finally it should be noted that one must be careful if the values of the products are updated dynamically during the production, e.g. if the value of a certain product depends on the number produced so far. Since the length/value relation between different products may change, the allocation exclusion pre-processing may need to be different from board to board.

7.3.3 An example

Applying the procedure on our example from Section 7.1.4 gives the final value 70 after investigation of 13 positions and a total of 16 allocations. Figure 7.11 shows the complete sequence together with the final cutting list. Note that the products are free of knots at the ends. Note also that the number of positions and allocation attempts are independent of the discretization of the board.

7.3.4 Experimental results

The implementation of the algorithm is straightforward. In our tests, the main part of the optimization was implemented using only some 40 lines of C-code. The procedure was evaluated using real defect data, quality rules and cutting bills from the Swedish window manufacturer

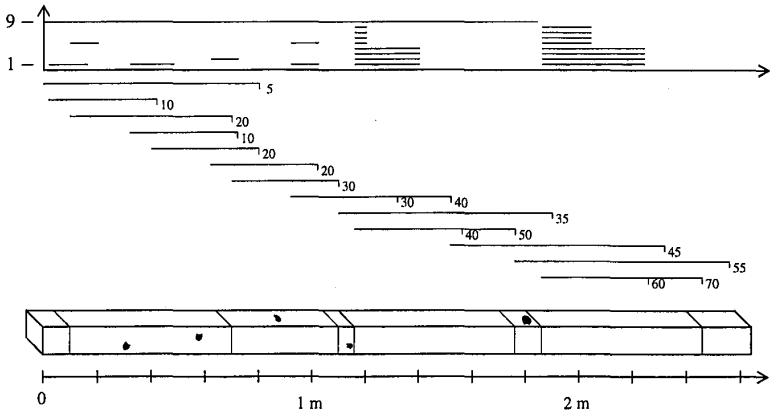


Figure 7.11 Illustration of the optimization sequence.

mentioned in Section 1.3.4. This particular manufacturer has installed a Woodeye system [142] connected to a commercial optimizing cross cut saw. The Woodeye system did not give defect information directly. Instead, an intermediate format based on *usage bit lists* (see Åstrand [7]) was used. This format was converted directly to allocation bit lists and the disqualifying list part of the algorithm was therefore not applicable.

20 boards with varying defect contents were run through their existing system and the total value for each board were recorded for comparison. The cutting bill used included five different lengths of casement parts and two different low quality frame parts. Both inner and outer casement parts were included, which means that there were eight different ways to cut each casement length. All together, the number of possible product alternatives was 42. The value and number of combinations for each product is given in Table 7.1.

Length	Value	Comb.
500	53	8
530	57	8
730	80	8
930	104	8
1330	155	8
680	1	1
1180	23	1

Table 7.1 Length, value and number of combinations for different products.

The output from the Woodeye were converted to allocation bit lists using the same cutting bill and the same parameter values for end trim cut (5 mm) and kerf width (4 mm). These lists were then run through the new optimization procedure using the values from the original cut-

ting bill, and the resulting lists were manually compared to the input data to verify that the products were in accordance with the quality rules. The test was run on a 33 MHz 486 PC and for each board, the total value, the number of investigated positions, the number of operations (allocation attempts) and execution time were recorded. The results are shown in Table 7.2. The execution time, given in ms, is measured for the main part of the procedure including initialization but excluding the output of the cutting list. The resolution was 1 mm, i.e. the number of discrete points equals the length of each board. The old value is the value given by the existing cross cut saw.

Board no.	Length (mm)	Value	Old value	Positions	Operat-ions	Execution time (ms)
1	4496	412	402	37	91	16
2	4796	427	396	45	106	17
3	5094	579	569	99	323	23
4	4195	420	420	55	199	17
5	5387	602	569	84	233	20
6	4799	455	419	39	98	15
7	4495	463	421	42	88	15
8	4492	453	341	64	150	19
9	5086	288	266	19	36	14
10	4193	449	447	68	200	19
11	4189	424	364	53	115	17
12	5090	500	478	65	179	19
13	4493	399	397	80	189	21
14	3899	373	367	44	81	15
15	5091	430	334	39	85	16
16	5386	185	184	19	37	14
17	4495	473	471	55	133	17
18	4793	522	522	55	240	17
19	5385	472	419	54	148	18
20	4794	105	105	11	20	13
Average	4732	422	395	51	138	17

Table 7.2 Results from tests with real production data.

7.3.5 Performance discussion

Table 7.2 shows that our proposed algorithm finds better solutions than the existing system in 17 out of 20 boards. Only in three cases were the optimal solution found by both systems. In this particular installation, or at least this particular cutting situation, the average profit could be increased by nearly 7%. This sounds a little bit too good to be true. A more careful inspection of the produced cutting lists shows that in those cases where the difference in value is

large, the existing system produced a number of long low quality lengths, even though it was obvious that shorter high value pieces could be produced instead. The result was a higher utilization, i.e. less waste, but a lower total value. The existing system was indeed run in the maximum value mode (minimum waste was also available), but the question is of course which optimization criterion was actually used. On the other hand, in our test 10 of the 20 boards had both a higher value and a higher utilization. In seven cases, our algorithm gave more waste but higher value and for the remaining three boards, the solutions were identical.

Table 7.2 also shows that the procedure is very fast. The key to the efficiency is the structure of the input data, combined with the small number of allocations that really need to be investigated. One could expect that the number of points to investigate, or in DP terminology number of stages to update, would increase very rapidly as we move from left to right along the board, especially if the products are many and short. It is true that a lot of positions will receive values in the right end, but many of the values will be too small for further investigation when we arrive at that position. Besides, a major defect in the middle of the board will effectively stop the combinatorial "explosion" and make everything start from the beginning again.

One of the most difficult cases is actually when the board is completely free from defects. In order to "provoke" the algorithm, five additional tests were performed. In all cases, a synthetic 6000 mm defect free board was used. Table 7.3 shows the results using the following cutting bills:

1. Same cutting bill as in the previous test.
2. Same cutting bill as above but with two short products added, 157 and 271 mm. The values were 14 and 27 respectively, i.e. relatively lower values (per length unit) than the other products.
3. Same cutting bill as above but using the values 22 and 35 for the new products, i.e. relatively higher values than the other products.
4. Same cutting bill as above but using the values 220 and 350 for the new products, i.e. much higher values than the other products.
5. A list of 15 prime number products, 1-43 mm. The values were identical to the length of each product.

Board no.	Length (mm)	Value	Old value	Positions	Operations	Execution time (ms)
1	6000	679	-	88	412	20
2	6000	679	-	132	840	37
3	6000	836	-	108	688	42
4	6000	8360	-	108	213	22
5	6000	6000	-	6000	89733	248

Table 7.3 Additional tests with a synthetic defect free board.

As expected, a long board free of defects takes in general longer time than a short with defects, although some defect combinations may give a slightly more complicated situation than a clean board. Adding two short low value products does not affect the final solution but

twice as many operations are needed. When the values of the short products are increased until the final list contains only those products, the number of operations is reduced. The reason for this is that when we move along the board, many previously recorded values will be lower than the accumulated value and no allocation attempts will take place. When the values are increased above the values of the longer products, the pre-processing exclusion of allocation will prevent the longer products from being evaluated at all and the number of operations will decrease even more.

The last case approaches the theoretical worst case when the number of operations equals the number of positions times the number of products. Still, only 248 ms were needed in this extreme case.

7.4 Secondary defects

The disqualifying list procedure described in Section 7.2.2 was based on the assumption that each individual defect has its own influence on the allocation list. There are however cases where a single defect does not affect the quality but clusters of similar defects do. In Section 6.4.2, the term *secondary defect* was introduced, i.e. a defect consisting of other defects. A common example is when a maximum number, say three, of a certain defect is allowed within one meter of the board. A zone less than a meter containing four such defects could therefore be considered a defect in itself. Such a defect is handled in the optimization stage just as any other defect and it will of course have its own disqualifying list.

There are, however, situations when this approach leads to undesirable results. Consider the situation shown in Figure 7.12. Suppose that clusters of at least three knots are inadmissible. A cut between the knots may however make both pieces OK. This means that if the secondary defect spans a significant length of the board, we need a better mechanism.

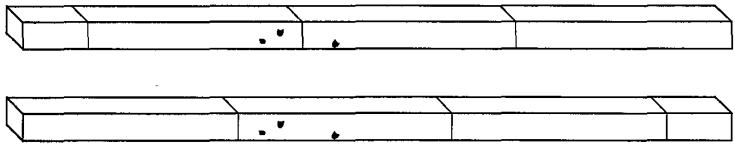


Figure 7.12 Three knots giving a secondary defect.

What we need is a feature to disqualify only if a given number of a certain defect appear within one single product. This can be implemented by a modification to the allocation bit data structure. Originally, we had a single bit indicating whether an allocation of a product with a given orientation was allowed or not. If we instead assign a number of bits for each allocation, we can treat them as a *counter field*. When applying the disqualifying bit list, we decrement the counter within the relevant zone instead of just AND'ing. We will thus count the number of times a given allocation was excluded within a certain zone. Initially, the counter is set to the maximum allowed number of the defect and if it reaches zero, the allocation is not allowed at that position. The actual allocation bit used in the subsequent optimization is then a logical OR of the counter bits. An example of the counter allocation list for the board from Figure 7.12 is shown in Figure 7.13. In the final allocation list, an allocation is only excluded if all three knots end up within one single product. Note that scheme requires one counter field for *each*

defect that has limitation in number. This means that if all different defects were limited by numbers, the complexity would increase rapidly. Fortunately, this is not the case in most practical situations.

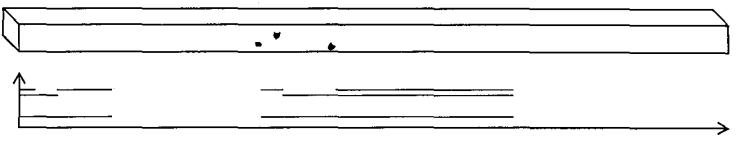


Figure 7.13 Implementing knots per product. The top two bits represent a counter giving the number of times a given allocation is excluded. The third bit is a logical OR which is used in the optimization stage.

This scheme handles number of defects per product. However many quality rules state number of defects per meter, and this can be interpreted in different ways. In sawmill grading one usually means number of defects per worst meter but in the cutting of window parts, where the products are shorter, it is not very clear what a maximum of 3 knots per meter means on a 60 cm product. Common sense says that we should allow maximum 2 knots on this product. The number per meter should thus be transformed to number per product based on the length of the product. On the other hand, if the rules state a maximum of three knots per meter, would then six knots in the end of a two-meter product be allowed? In a practical implementation, one should probably have the possibility to use both secondary defects as well as limitations in number and which one to use will depend on the situation.

7.5 A limited two-dimensional case

The optimization presented in this chapter is primarily one-dimensional. However, the partition of a board often involves a combination of cross and rip cutting. To find the best combination of products is therefore a two-dimensional optimization problem. It has some restrictions though since either the cross or rip cuts must go through the entire width or length of the board, so called guillotine cutting. Most of the work reported in the field of wood cutting [29, 34, 61, 95, 169] cover this problem where we using a sequence of cross and rip cuts try to maximize the yield of defect free pieces from boards containing inadmissable defects. This mostly applies to large dimension hardwood applications. In the softwood dominated Scandinavia, the dimensions are smaller which means that more than one cross and/or rip cut are seldom used.

The proposed one-dimensional algorithm can be extended to the limited case, where we first perform a cross cut and then a rip cut using a fixed rip pattern. An example of this is manufacturing of glued joint boards as shown in Figure 7.14.

The boards are produced in different dimensions with varying width, length and thickness. This means that when the raw material is cut, different length correspond to different fixed rip patterns. When a number of each length has been produced, all pieces of a given length are rip cut using the rip pattern corresponding to that particular length. Then the saw blades of the rip saw are adjusted to another pattern and then all length of that type are cut, and so on.

The objective is to find the combination of lengths in the cross cutting giving the highest yield after rip cutting. A single defect may disqualify one or more of the pieces depending on its position. If the pieces are to be glued as shown in Figure 7.14, a defect may even be OK if it appears completely within a piece, as it will be hidden, but may disqualify two pieces if it

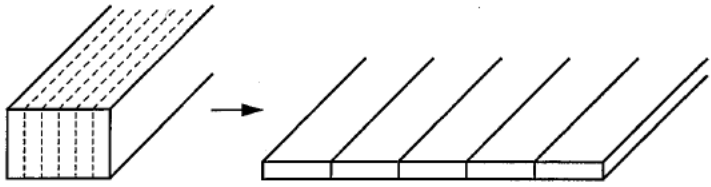


Figure 7.14 Glued joint boards.

appears in a cut. This is illustrated in Figure 7.15. The exact position of a defect may therefore have a great impact on the total value. This is a situation which is impossible to handle in a manual saw.

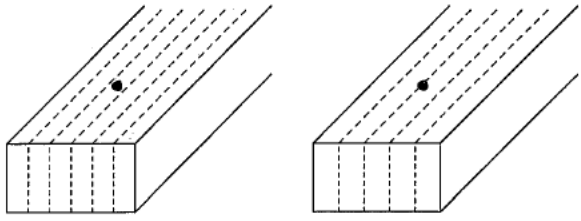


Figure 7.15 Selection of rip patterns.

The described optimization procedure can be extended to this limited two-dimensional case with fixed rip patterns by the following modifications:

- In the first phase when the allocation bit list is established, every individual piece within each possible rip pattern is represented by its own allocation bit. Appropriate quality requirements will be applied on the crosswise zone corresponding to the individual piece, whereas the rest of the width may contain any defects. The allocation list generation procedure is thus identical, but the number of products will be larger.
- Before optimization, a pre-processing of the allocation list is performed where the different groups of allocation bits corresponding to certain rip patterns are logically OR'ed into single allocation bits. We also count the number of bits set within each such zone and store as a *zone factor*.
- The optimization is then performed as before. The only difference is that when an allocation is evaluated, its value is no longer fixed. Instead it is a product of the zone factor and the value of a single piece of the particular type.

7.6 Variable length products

7.6.1 A simple solution

Another extension is the possibility to define variable length products, e.g. finger joint pieces, see Figure 1.12 on page 11. Finger joint lengths should normally be made defect free and as large as possible between defects that must be removed. We also normally have a minimum meaningful length, below which the entire section goes to waste. In a real situation where only finger joint lengths are produced, the quality requirements are normally not very complicated. A defect will either be allowed or not, so this is no longer an optimization problem.

It is not uncommon though that a combination of fixed and finger joint lengths are produced simultaneously. In most cases where this is done, the finger joint lengths are considered to be low value by-products and only a way to make use of material which cannot be used in any fixed length products, e.g. defect free zones that are too short. The optimization can in this case be performed as before, but with a finger joint post-processing stage where we inspect the waste sections to see if it is possible to use them as finger joint material. This will of course only work if the value per length unit of the finger joint pieces is lower than the values of the fixed length products.

If the finger joint pieces have significant values, this can be handled in a sub-optimal way by assuming that all finger joint lengths are a multiple of the minimum length. This length is then treated as a fixed length product in the optimization but we then add a post-processing stage where we merge adjacent finger joint pieces and also try to expand them into waste sections where the allocation list would have allowed allocation. This is illustrated in Figure 7.16. The result is a system where we can make full use of the raw material, but where the values of the finger joint products from the optimization point of view always are a multiple of the value of the minimum length. The solutions are thereby not guaranteed to be optimal but will probably be good enough for all practical purposes.

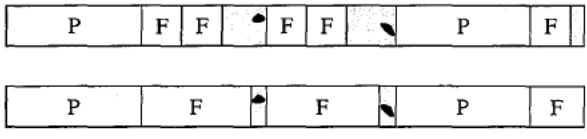


Figure 7.16 Finger joint lengths (F) can be handled as a short length in the optimization and then merged and expanded in a post-processing stage.

7.6.2 A general solution

The scheme described above can not handle the case where the variable length products have different quality requirements in the lengthwise direction. It is quite common though that finger joint pieces allow small defects in general but must be completely defect free in the ends to enable routing of the fingers, see Figure 7.17. This can be regarded as if each piece contains two fixed length zones and one variable length zone in between.

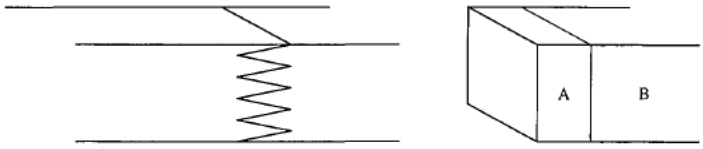


Figure 7.17 Finger jointing. The A section may have higher quality requirements than the rest of the piece.

What we really want is to extend the optimization to handle fixed as well as variable length products in a general way so that any product can be defined as a combination of any number of fixed and variable zones, or perhaps with a restriction of one varying zone. The products will thus have a minimum and a maximum length and a fixed length product is just a special case where the minimum and maximum lengths are the same.

The dynamic programming approach used in the original algorithm seems inappropriate in this general case since we now at each position must evaluate a huge number of length alternatives, which of course also means that the number of positions subject to evaluation will increase rapidly. However, by choosing a clever representation of the data, the algorithm can actually still be very efficiently implemented. The key is the assumption that the value of a variable length product is proportional to its length, i.e. it can be expressed as $k \cdot l + c$, where l is the length of the product and k and c constants. The constant c will typically be negative since we normally want one long product to have a higher value than two short products of the same type. If the value can not be expressed as a straight line, e.g. if the value should be progressive with respect to the length, we may approximate by using piecewise linear values and divide into several different products covering different length intervals.

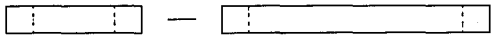
Even though the optimization algorithm is basically the same, the implementation, which is the key to the efficiency, is rather complex. The algorithm will be intuitively described here using figures and examples. A more formal description will be available [141].

7.6.3 The allocation list

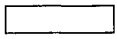
The allocation list in the original algorithm was a binary one-dimensional structure for each product. An allocation was either allowed or not at a given position. In the variable length version, all allocations are given in terms of the minimum and maximum lengths. Consider the simple example shown in Figure 7.18.

When allocating a variable length product, the allocation can either be limited by the minimum and maximum lengths of the product or it can be limited by defects in any direction. The following different cases of allocations between position Pos1 and Pos2 in Figure 7.19 exist.

- No defects affect the allocation. Minimum and maximum lengths are the same within the interval (Figure 7.19a).
- A defect limits the allocation in the forward direction. The maximum length gradually decrease within the interval (Figure 7.19b).
- A defect limits the allocation in the backward direction. This is possible if a defect is allowed in the middle but not in the end. The minimum length gradually decrease within the interval (Figure 7.19c).
- The allocation is limited by defects both in the forward and the backward direction (Figure 7.19d).



Product P1: Variable length 5-10, value 4-9, small knots allowed in the mid section.



Product P2: Fixed length 4, value 6, no knots allowed.

Figure 7.18 Example of quality rules to illustrate the variable length optimization procedure.

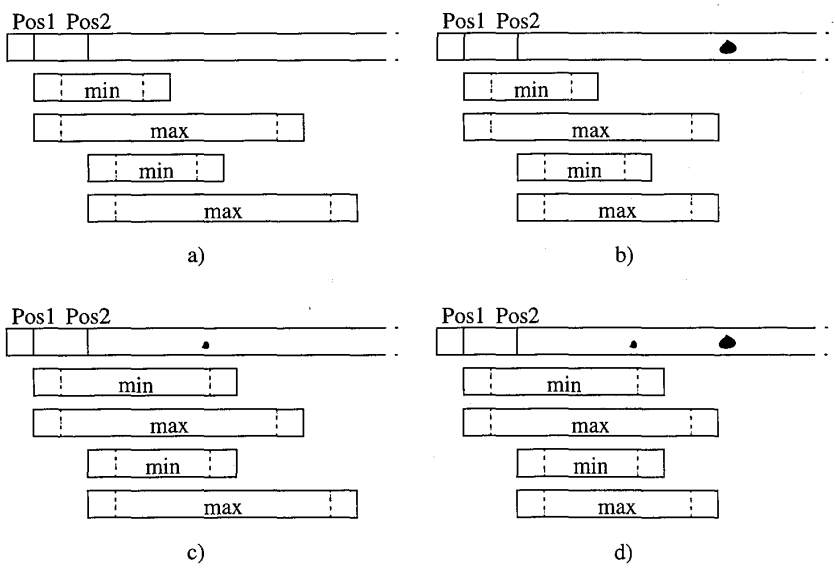


Figure 7.19 Four allocation alternatives from the interval Pos1-Pos2. a) limited by product. b) forward limited by defect. c) backward limited by defect. d) forward and backward limited by defects.

The allocation list in the variable length case is thus a two-dimensional structure which can be graphically illustrated as shown in Figure 7.20. The allocation list should be interpreted in the following way: The horizontal axis is the position along the board and the vertical axis the length of the allocation from a given position. The shaded areas represent possible allocations of the variable length product P1, e.g. from the position one unit from the left, we may allocate a P1 of length 5-7 or a P1 of length 9-10. The straight lines at length 4 in the diagram represent allocations of the fixed length product P2.

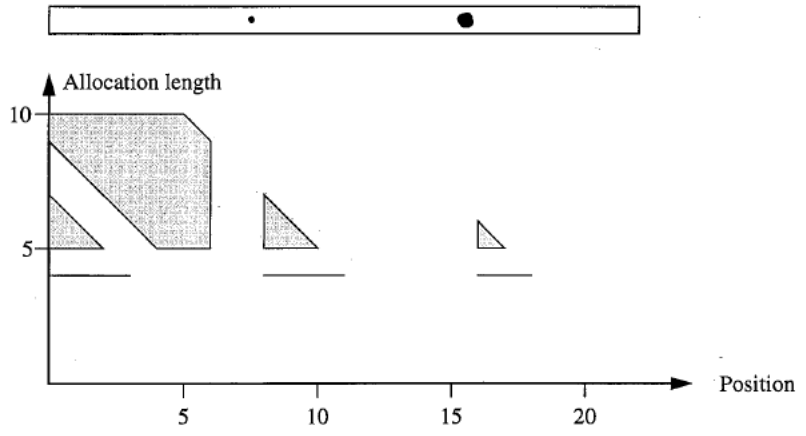


Figure 7.20 A sample board and the corresponding allocation list. The shaded areas indicate the minimum and maximum lengths of different products from different positions.

The disqualifying lists are also transformed into two-dimensional structures as shown in Figure 7.21. The example shows the disqualifying list for a small knot applied on product P1. The process of establishing the allocation list is thus a matter of finding appropriate two-dimensional disqualifying lists from the decision tree, just as in the original algorithm, and applying them in sequence to the allocation list. Note that shaded areas in Figure 7.21 correspond to areas that should be excluded in Figure 7.20.

7.6.4 The optimization procedure

In the original algorithm we evaluated *positions*. Each such position was defined by a current value, a back pointer which indicated from where which product was allocated to give this value and a set of allocation bits to indicate where to go from here. In the new implementation, we instead evaluate *intervals*. The interval is the basic data structure in the variable length optimization and is defined by the following data, which also are illustrated in Figure 7.22.

- First and last position of the interval along the board.
- Current value for the first position and the last position. The current value for a position in between is computed as a linear combination of the end values.
- Back pointers from the first and the last position giving the current value. Back pointers in between are also linear combinations.
- An allocation list for the zone according to Section 7.6.3.

The allocation list is thus an integrated part of the board optimization list. Each interval defines a continuous zone along the board where everything is linear, i.e. current value, back pointers and allocation.

The actual optimization is basically performed in the same way as in the original algorithm, except that we evaluate entire intervals in single operations. When we evaluate an interval we go through the list of possible allocations. In the original algorithm, an allocation from

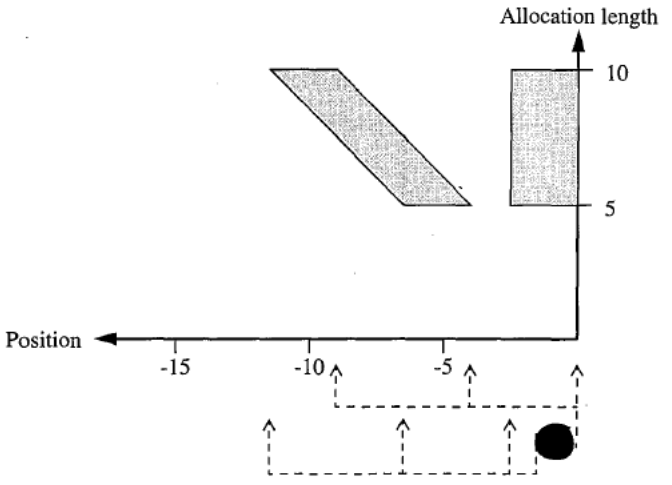


Figure 7.21 Disqualifying bit list in the variable length algorithm.

a given position resulted in a new value at another position along the board. We then compared the value with the previously stored value at that position and replaced the value and the back pointer if the new value was larger. In the new implementation, the allocation from a *board value interval* results in a new *allocation value interval*, with values and back pointers, which should be compared to the list of previously stored values between the two positions given by the start position of the originating interval plus the minimum length of the allocation and the end position of the originating interval plus the maximum length of the allocation. The procedure is illustrated in Figure 7.23.

The section of the board list subject to comparison can of course consist of several intervals or, as in Figure 7.23, be a subsection of an interval. We must thus first split all intervals in matching sections. We then compare the lines of the sub-intervals one by one. Three cases are then possible:

- The allocation interval has a lower value along the entire length (case c in Figure 7.23). In this case we just disregard the new interval.
- The new interval has a higher value along the entire length (case a), and here we replace that section of the list with the new interval. The back pointers from the beginning and the end of the interval are updated accordingly.
- The value lines of the board and the allocation cross each other in any direction (case b). Here we split the existing interval and replace the section where the new value is larger.

The attentive reader has noted that the allocation attempt in Figure 7.23 corresponds to a fixed length. Allocation of a variable length normally gives a two-section allocation value interval as shown in Figure 7.24. If the originating board interval has a lower value derivative than the product being tested (Figure 7.24a), then we should start the allocation as early as pos-

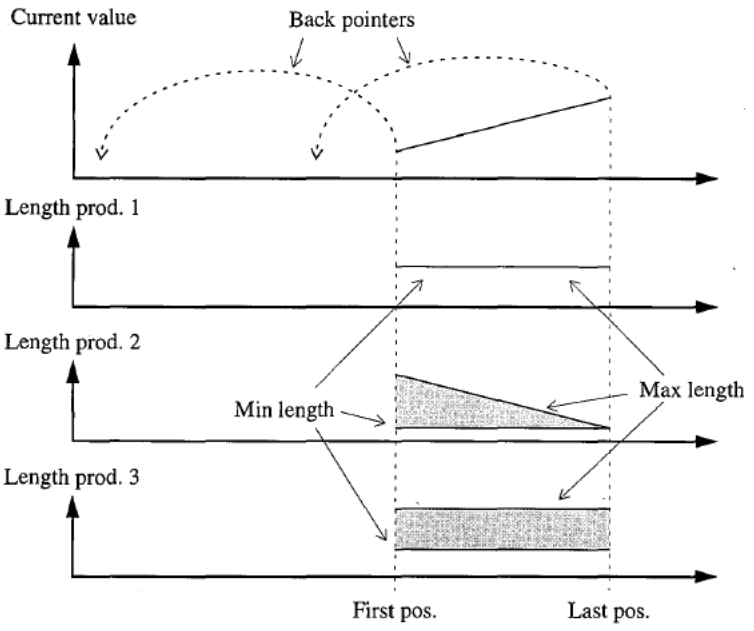


Figure 7.22 An interval in the variable length optimization. Current value, back pointers and allocations vary linearly within the interval. The shaded areas indicate possible allocations of different products.

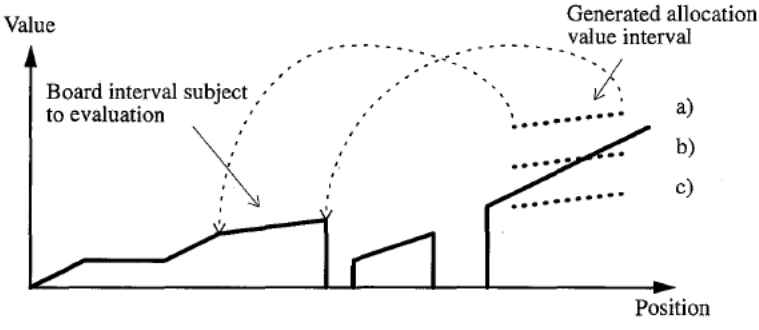


Figure 7.23 Evaluating an allocation. a) replace with new value. b) split the interval and replace the first section. c) disregard the allocation.

sible. On the other hand, if the derivative of the originating interval is higher, then the allocation should be made as late as possible (Figure 7.24b). There could never be any reason for doing something in between.

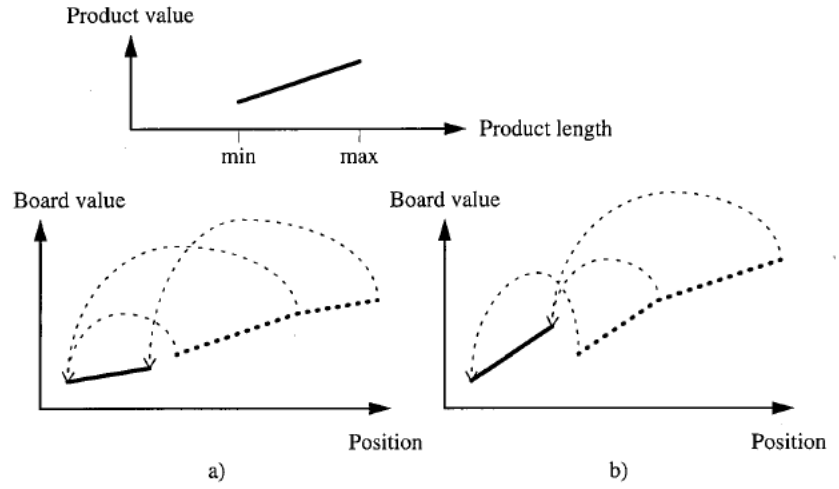


Figure 7.24 Allocation of a variable length normally gives a two-section value interval. The back pointer from the mid point depends on the derivative of the board value and the allocation value.

The potential value from a general allocation actually constitute a rhomb as illustrated in Figure 7.25a and b. The point A is the first position in the interval plus the minimum allocation length, B the first position plus the maximum length, C the last position plus the minimum length and D the last position plus the maximum length. The maximum value allocation will always follow either the path A-B-D or the path A-C-D depending on the derivatives of the originating value and the product value.

A number of degenerated cases from this general case exist. If it is a fixed length allocation, then minimum and maximum lengths will be equal and the value of the allocation will look like Figure 7.25c. Variable length allocations which are limited by a defect in either the forward or backward direction give a rhomb which is cut off as shown in Figure 7.25d, e and f. This results in two of the points ending up at the same position along the board.

When all allocations of an interval have been evaluated, we walk along the board to the next interval. If the value of this interval (or a subsection thereof) is lower than the maximum current value achieved so far, we replace the values and the back pointers of this interval (possibly after splitting) to allocate a waste section instead, just as in the original algorithm. We then proceed with evaluation of the allocations in this new interval. Note that each evaluation of an interval may generate a number of new intervals along the board, and can also result in a splitting of the interval itself being evaluated. Note also that the discussion on allocation exclusion in Section 7.3.2 is valid also in the variable length case.

The entire algorithm can thus be viewed as putting a number of straight line segments on top of each other while always keeping the highest. When we have gone through the entire list of intervals, the optimum cutting list can be unwound by following the back pointers. We may then need to interpolate back pointer values from positions within intervals.

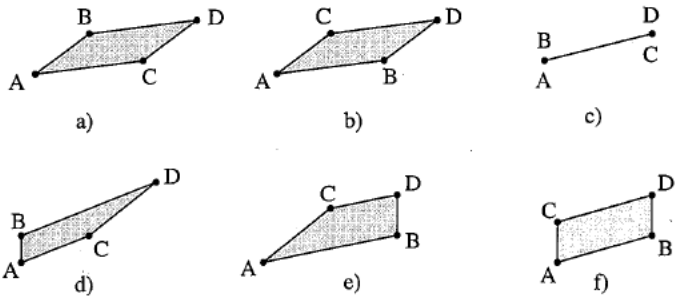


Figure 7.25 Possible allocation patterns. a) and b) general cases. c) fixed length. d) backward limited by a defect. e) forward limited by a defect. f) forward and backward limited. A is the first position + min length, B first position + max length, C last position + min length and D last position + max length.

7.6.5 A variable length example

The procedure will now be illustrated by the example from Section 7.6.3. When starting the optimization, we will first evaluate the initial three intervals as illustrated in Figure 7.26. An interval is defined as a zone in which everything is linear and no events occur, such as the beginning of new allocations or previously stored values. The evaluation of such an interval generates new value intervals (thick lines) and corresponding back pointers (thin arrows). Here we can see that it is not meaningful to allocate a short P1 from the beginning of the board since a P2 allocation has a higher value for the entire interval. A long P1 can still be meaningful though. For simplicity, in the following steps only the highest value line in each interval will be shown in the diagrams.

In Figure 7.27 we evaluate the next two intervals giving a new solid value line which is merged into the list. Note that the main section of the line is sloping since it is a variable length allocation while the last section is horizontal since we have reached the maximum length. Note also the back pointers which point to an identical position for the entire variable section while a back pointer within the constant section is given as a linear combination.

In Figure 7.28 we evaluate three more intervals. Note that the second interval does not correspond to an event in the allocation list but is the result of the beginning of a new current value interval due to a previous allocation.

In Figure 7.29 we evaluate the final two intervals. Here we have the situation that the previously stored value (zero in this case) is lower than the maximum value achieved so far. The new allocation evaluations will thus start from this maximum value and the back pointers inserted will denote a waste section.

Figure 7.29 also shows the resulting cutting list which is established by starting at the highest value and following the back pointers. In one step we have to start from the middle of an interval but the calculation of the new back pointer is trivial since the back pointers in both the beginning and the end point at the same position.

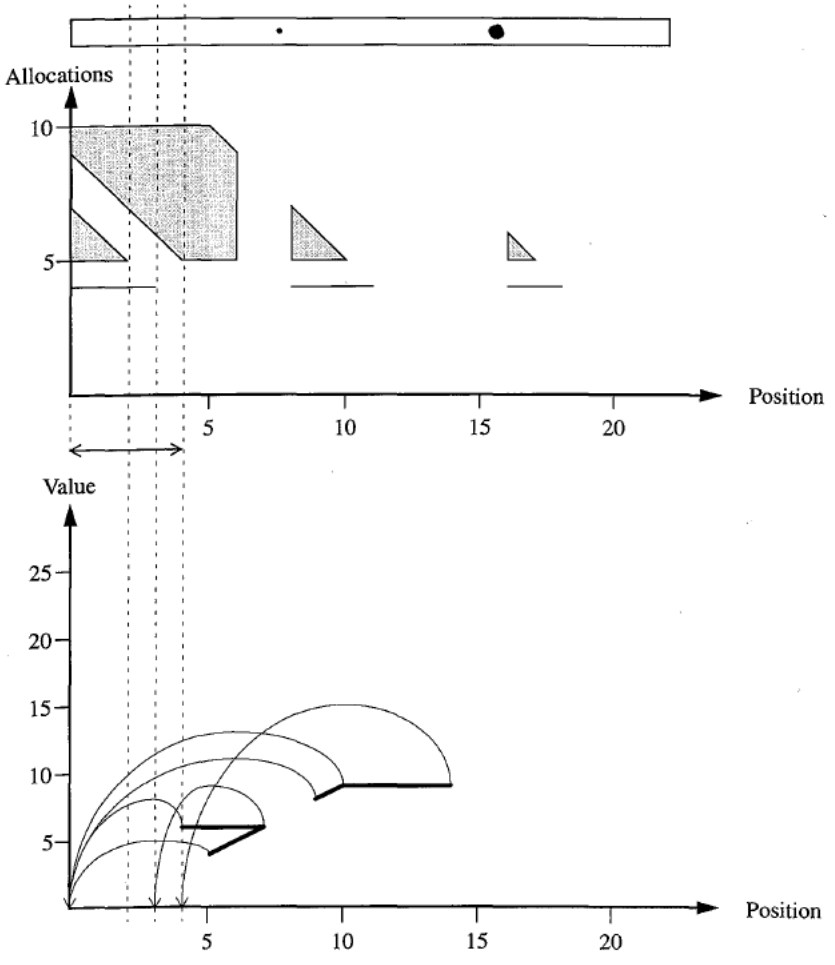


Figure 7.26 *Evaluating the first three intervals of the sample board. Thick lines represent value intervals and thin arrows back pointers.*

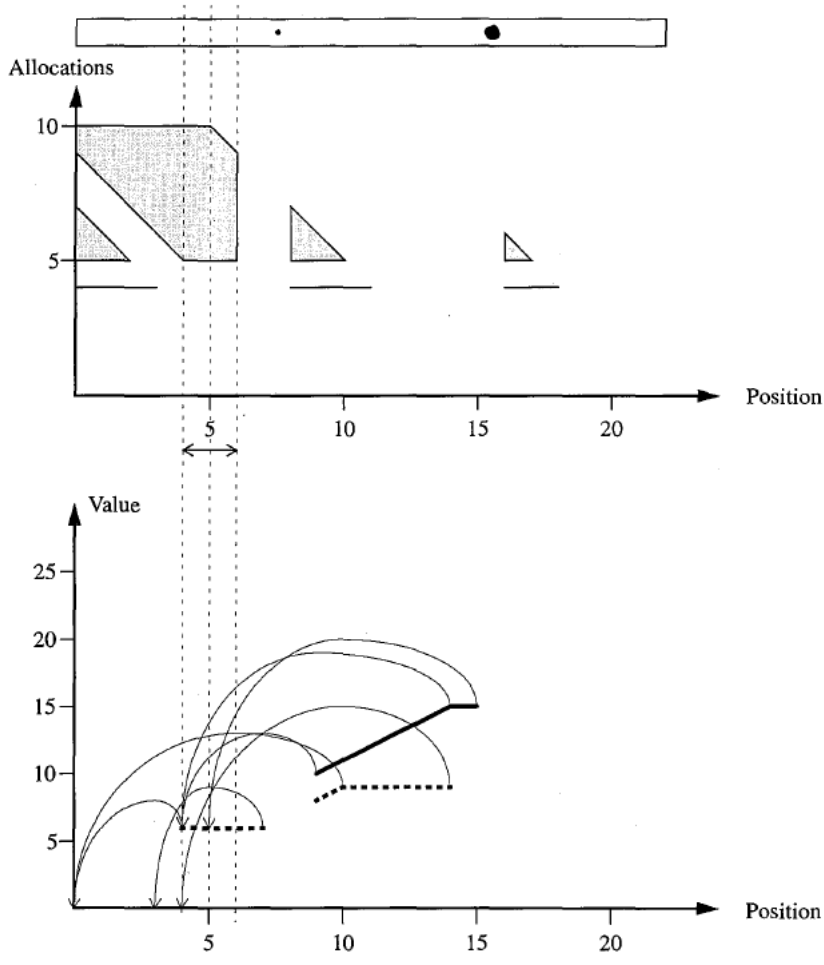


Figure 7.27 Evaluating the next two intervals giving a new value line which is merged into the list.

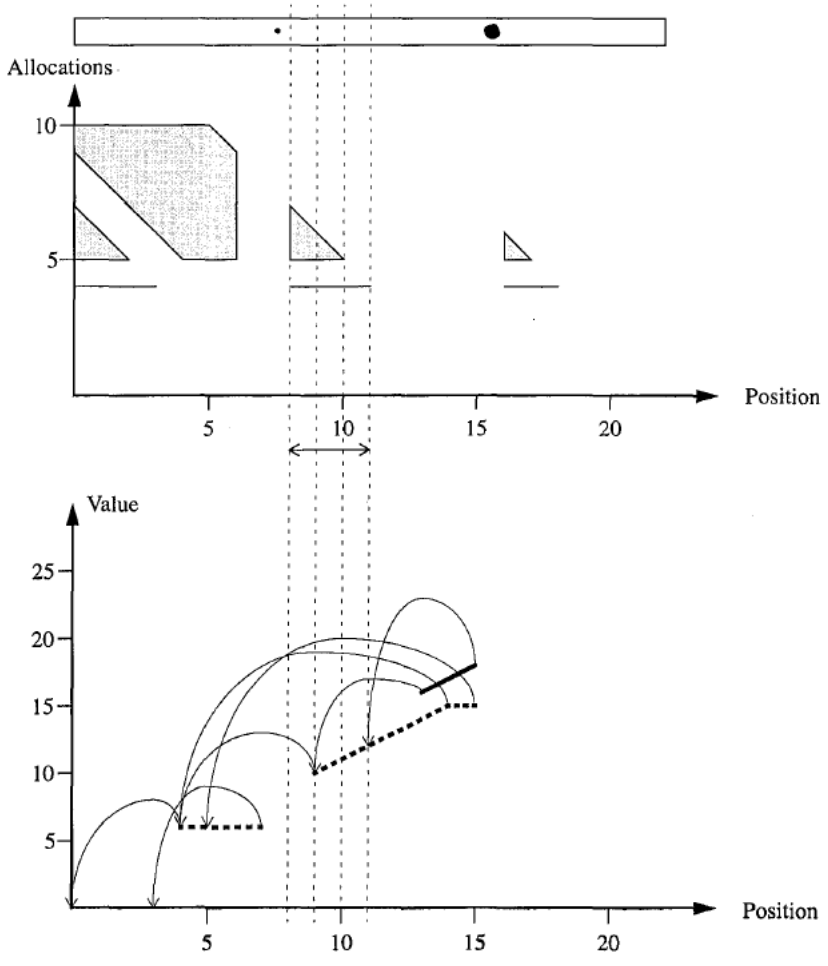


Figure 7.28 Evaluating three additional intervals. Note that the second interval is not due to an event in the allocation list but the result of a split due to a previous allocation.

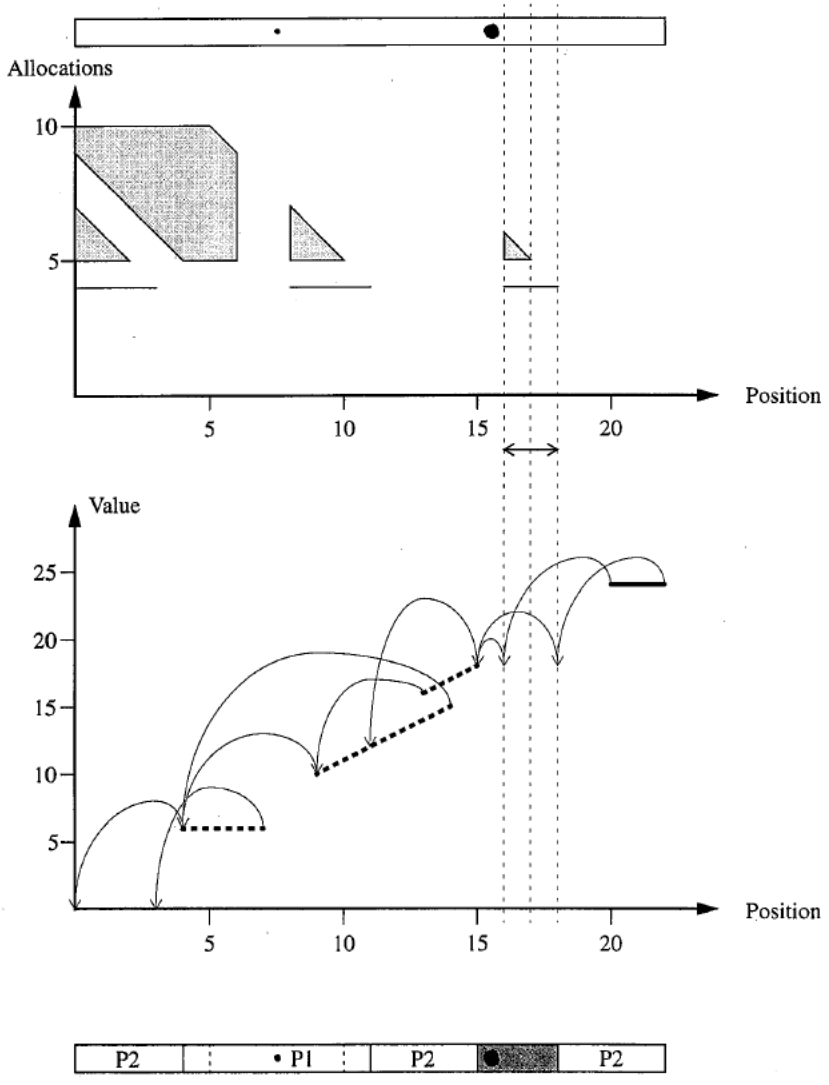


Figure 7.29 *Evaluation of the final intervals and the resulting cutting list established from following the back pointers. If we end up in the middle of an interval we use a linear combination of the back pointers.*

7.6.6 Implementation aspects

As pointed out before, the key to the variable length optimization is the implementation and the data representation. The optimization itself is still based on ordinary dynamic programming but the representation permits us to handle entire intervals in single operations. The original algorithm was implemented using arrays where we directly could index different positions. The board optimization list in the variable length version is preferably implemented as a dynamic double linked list as shown in Figure 7.30, where each zone along the board is represented by a record containing position, values and back pointers. Note that the back pointers are implemented as position numbers, *not* pointers. This implementation makes it easy to walk from zone to zone and to split and merge different zones.

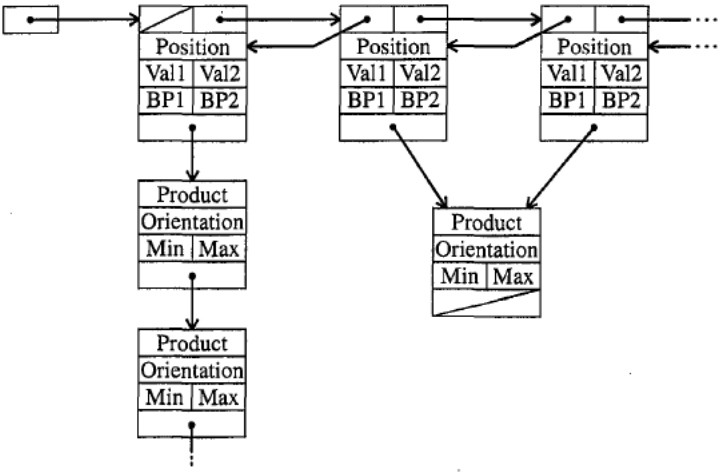


Figure 7.30 The optimization is implemented using a double linked list where zones of different length easily can be split and merged.

The allocations with their minimum and maximum lengths can then be implemented as linked lists originating from each zone. For each allocation, the product, the orientation and the minimum and maximum length is stored. The orientation is a bit field of eight bits indicating the possible orientations of the product. Hence, only one allocation record is required for all orientations of each product.

As shown in Figure 7.30, two zones may share a common allocation list. This occurs when a zone is split due to the assignment of a value from a previous allocation. The zones may be merged again if a later allocation is found to cover both of them with a single higher value line. The list will thus be a very dynamic structure during the optimization with zones continuously being split and merged.

There are also some algorithmic aspects that should be considered. A critical operation is for instance to find the interval which contains a given position without extensive linear search. A simple solution is to keep a pointer to the zone last subject to an allocation attempt and to search linearly from there, assuming that the next allocation will be in the neighborhood. Another possibility is to use an index table pointing to all zones crossing even position bound-

aries, say each 100 mm as illustrated in Figure 7.31. If a split or merge operation involves a zone crossing one or more boundaries, then the index table is updated accordingly. This works good in practice even though the list tend to be rather unbalanced. An alternative could therefore be to use a B-tree [167] as indexing structure. However, experiments have shown that if all searchs are performed linearly from the beginning of the list, around 40% of the execution time will be spent on searching for positions. If an index table is employed, the figure is reduced to around 5-10% so the gain in using more advanced structures is marginal.

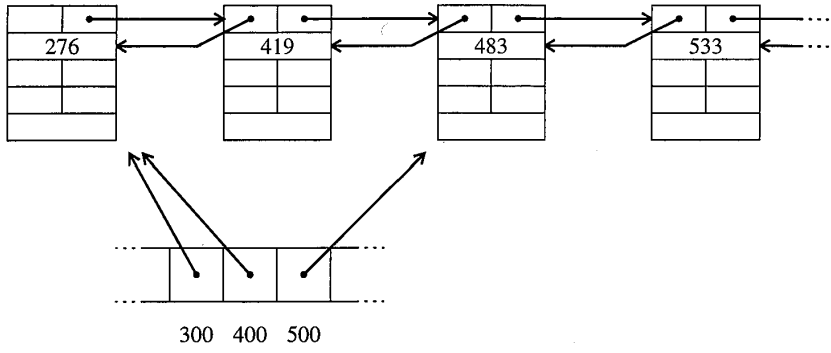


Figure 7.31 Index table to speed up search for a given position. If a zone containing an even boundary is modified, the table is updated accordingly.

7.6.7 Numerical results

An implementation of the variable length optimization procedure is actually used in the commercial WoodEye system from Innovativ Vision AB, Sweden, and is currently running in full production in several installations around the world. As in the fixed length case, the input to the procedure is usage bit lists (Åstrand [7]) and not defects. The disqualifying bit list structure is therefore not applicable here. The usage bit lists are instead converted directly to allocation lists.

To compare the performance of the new algorithm to the original one, exactly the same boards as in Section 7.3.4 were run through the system using identical quality rules and cutting bill, except for the end trim cuts. Any variable length products were not included in this first test since we want to compare using identical input data.

The results from the optimization of the 20 boards is shown in Table 7.4. The number of evaluated intervals, which corresponds to the number of positions in Table 7.2, is shown together with the number of allocation attempts, i.e. the number of generations of "value intervals" that possibly should be merged into the list.

The tests were run on the same 33 MHz 486 PC as the tests in Section 7.3.4. However, in the previous tests, a precision timer was used to measure the execution speed. In the new tests, the internal timer of the PC was used, where each timer tick corresponds to 50 or 60 ms. The measurement is therefore not exact and can even be zero if the entire board is optimized during one single timer tick. Still, the figures give a rough feeling for the decrease in performance in the new implementation.

Board no.	Length (mm)	Value	Intervals	Operat-ions	Execution time (ms)
1	4496	412	125	232	60
2	4796	427	145	670	110
3	5094	579	533	4233	600
4	4195	424	207	1708	220
5	5387	602	237	957	170
6	4799	455	117	379	50
7	4495	471	177	616	110
8	4492	455	288	1419	270
9	5086	288	88	210	50
10	4193	453	296	2161	330
11	4189	424	478	890	220
12	5090	500	243	1074	220
13	4493	399	559	3445	550
14	3899	373	170	549	110
15	5091	430	143	349	110
16	5386	185	91	210	0
17	4495	473	233	1146	220
18	4793	522	89	597	50
19	5385	472	157	554	110
20	4794	105	37	69	0
Average	4732	422	221	1073	178

Table 7.4 *Results from tests of the general variable length implementation with real production data and a fixed length cutting bill.*

It can be noted that some boards have slightly higher values in this test due to the absence of the trim cuts, i.e. the boards were 10 mm longer. The actual average values were 421.55 and 422.45 respectively, giving a 0.2% value increase. Since 10 mm is 0.2% of the average board length 4732, this seems reasonable.

If we compare the figures in Table 7.2 and Table 7.4 we can see that the total execution time increases roughly a factor 10 for the identical fixed length problem. To evaluate the performance of the system on variable lengths, let us instead use the cutting bill from Table 7.1 and define variable lengths between the points given by the fixed lengths, i.e. we define four different variable length products as shown in Figure 7.32 and as defined in Table 7.1. The two low value products were also included but were kept as fixed lengths.

The optimization was once again applied to the 20 boards using this new cutting bill. A fixed 50 mm defect free zone was defined in each end of the product, just as in the fixed length case. The result is shown in Table 7.4.

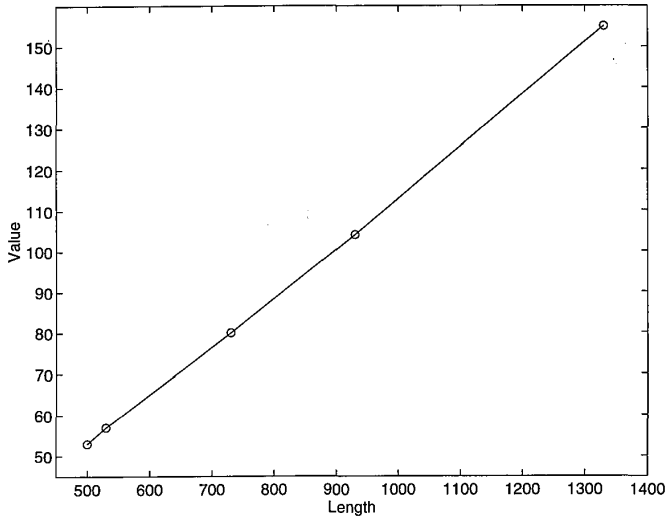


Figure 7.32 Four variable lengths products derived from the five fixed length product in Table 7.1.

Min length	Max length	k	c	Comb.
500	530	0.1333	-13.667	8
530	730	0.115	-3.95	8
730	930	0.12	-7.6	8
930	1330	0.1275	-14.575	8
680	680	0	1	1
1180	1180	0	23	1

Table 7.5 Cutting bill for the products in Figure 7.32. The value is given as k·l+c, where l is the length. The two low quality products are of fixed length.

As could be expected, the average value is slightly higher since we now can make full use of the material. The average execution time is only slightly higher, indicating that variable lengths are not significantly more complex to handle than fixed lengths. In fact, both the number of intervals and the number of operations are lower than in the previous test, but it seems that each operation takes longer time. This is reasonable since an evaluation of a variable length in general gives a two-section value interval whereas a fixed length evaluation only gives one.

Board no.	Length (mm)	Value	Intervals	Operat-ions	Execution time (ms)
1	4496	451	107	628	160
2	4796	459	84	534	110
3	5094	586	291	2850	770
4	4195	428	132	1386	280
5	5387	610	178	1353	280
6	4799	487	83	577	110
7	4495	494	96	564	160
8	4492	475	159	1275	330
9	5086	308	52	166	0
10	4193	470	170	1472	330
11	4189	436	140	1248	270
12	5090	515	159	1319	330
13	4493	411	239	2408	600
14	3899	387	101	551	170
15	5091	444	108	547	110
16	5386	203	48	150	0
17	4495	486	118	812	220
18	4793	534	54	474	60
19	5385	526	83	451	50
20	4794	125	25	53	0
Average	4732	442	121	941	217

Table 7.6 Results from tests of the general variable length implementation with real board data and a variable length cutting bill.

Even though an evaluation of a variable length is not significantly more complex than the evaluation of a fixed length from the optimization point of view, the allocation list itself may become more complex in the variable case. If we have a lot of small knots which are allowed in the mid section of a product but not in the end sections, then there will be a lot of small intervals in the allocation list, but there will also be a lot of different allocation possibilities for the same product from each interval. Adding yet another small knot to the board will add another zone but will also add another allocation to all other zones located in front of the new knot within the range of the maximum length of the product. This "quadratic" complexity will not apply to fixed length products.

7.6.8 Some remarks

In Dynamic Programming terminology, each interval evaluation comprises a stage, even though the total set of stages in the problem is dynamically updated as the procedure progresses along the board. Since each stage is an interval, the problem does not have the same

discrete structure as the fixed length version where the board was partitioned into fixed discrete elements. Even if a fixed discretization is used in the variable length case, it turns out that the number of operations for a given board is not in general dependent on the discretization. This was actually true also in the final version of the fixed length case even though it was not explicitly pointed out. The simple example in Section 7.3.3 would for instance still give 13 positions and 16 allocations even if the discretization was finer.

It therefore seems that floating point numbers could be used to define the positions of the intervals. However, one should be careful about this. One of the key features of dynamic programming is that there can be several ways to reach a given stage, but only the optimal one is stored. This is the main reason why the combinatorial complexity can be kept low. In terms of our cross cutting problem, there are two situations in which we have multiple paths to the same stage. The first case is when a given combination of a number of products are allocated in different orders. Then it is obvious that product A allocated after product B should give the same position as B after A, regardless of the discretization. However, if floating point arithmetic is used, there is actually no guarantee that A+B = B+A. One should in general be careful about testing for equality of floating point numbers.

The other case is when different combinations "by accident" end up in the same position, e.g. if one product allocated directly after a defect ends up in exactly the same position as another allocation which was not affected by the defect at all. It can be assumed that if we use a reasonably fine resolution, the latter case will seldom occur, at least not in the fixed length version. In the variable length version, this case will often occur implicitly, but note that the combinatorial complexity in this case is given only by the explicit combinations of the fixed length and the minimum and maximum lengths of the variable length products. The conclusion is that we should use a discrete structure but we may use as fine discretization as needed to match the precision of other machinery, e.g. 1 mm.

However, if we regard the theoretical worst case performance, i.e. where we have to test all products at all positions, then it is obvious that a finer resolution gives a huge complexity. On the other hand, the worst case will become more unlikely to occur. Still, we may need a strategy to handle these cases where the complexity grows too rapidly. Depending on the buffering structure of the system, i.e. if we are able to halt the infeed of the boards, quite long computation times may be acceptable if this does not happen too often. A more drastic solution is to set a time-out limit on the optimization and then just use the maximum value achieved so far. The remaining section of the board we just throw on the waste belt. It would then only be a matter of statistics how frequently this will happen. It is admitted though that such a solution may be psychologically less attractive to the customer, even if the average loss is marginal.

Another aspect that deserves a remark is the assignment of values to the products. The dynamic programming algorithms are only guaranteed to give optimal cutting lists with respect to the assigned values. However, we will probably very seldom be able to put price tags directly on the products. Instead it is often a matter of producing given numbers of different pieces to be combined in a final product, e.g. the different parts of a window.

This is an aspect which is often overlooked in existing commercial systems. For instance, the documentation of the system from which the test data was collected provides no guidelines whatsoever about the value assignment. If a fixed set of products are produced for a longer period of time, the values can probably be manually adjusted so that the relation between the different products reaches a steady state. Alternatively, if different parts are kept in stock we may manually adjust the cutting bill according to the present stock situation. However, if we cut according to customer specific orders, we probably need a dynamic update of the values

depending on the numbers produced so far. Dynamic values are possible in our algorithms as long as the values remain the same within one single board. This is no problem as long as the size of the order is large compared to the number of products obtained from each board.

How the values should relate to the produced numbers is not obvious. Probably one should assign initial values based on an estimated difficulty to cut a given product, e.g. directly related to the length. The values could then be updated as the piles of produced products grows so that the product in a larger pile receives a relatively lower value. When the quota of a given product is full, the value should be set to zero or the product simply removed from the cutting bill. The commercial system referred to above uses fixed values which abruptly are set to zero when the quota is full. This unfortunately gives a very low yield in the end of the order unless new subsequent orders are initiated. On the other hand, how should the values in the new order relate to the previous one? The new order may use completely different quality rules. Furthermore, if we only use produced number relative to the number in the order to determine the value, then this may result in the last pieces of the previous order not being produced at all. If we have only a few remaining pieces from an order, then maybe we should assign very high values on these pieces in order to close the order? This value assignment problem will be subject to future research where different real world cutting situations must be investigated.

Another problem is the fact that pieces may be rejected due to missed defects, handling damage etc. If we produce an exact number of pieces according to orders we need a practical mechanism to produce additional pieces afterwards. There is no point in chasing the last percentage of yield in the optimization if we by routine have to add a few spare pieces on a hundred piece order.

7.7 Conclusions

We have shown that the fully automatic environment where we have full information on all defects gives new possibilities to optimize the cutting of boards. Even if quite complex quality rules, such as the window casement parts described in Section 1.3.4, existed also in the previous manual environment, it was not possible to take full advantage of the possibility to "hide" different defects in non-visible zones. Using our optimization procedure where any combination of zones can be defined, it is even possible to produce parts as shown in Figure 7.33, where a big defect is allowed only within a very small zone. This situation can not be handled in a manual saw or a crayon marking system.

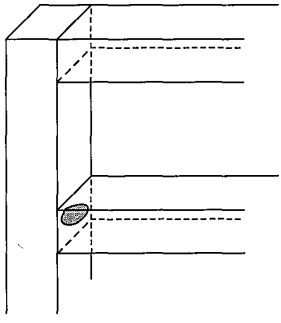


Figure 7.33 A (somewhat theoretical) case where the new optimization algorithm enables the hiding of a defect by placing it within a short interval.

In the automatic environment we may thus use raw material of lower quality (i.e. with more defects) to produce the same amount of parts. If the material is almost defect free, then the advantage of the new system compared to a crayon mark system would be lower.

The proposed dynamic programming algorithm produces optimum cutting lists well within the time limit given by the industrial requirements. We have also shown that a number of extensions are possible, such as secondary defects, limited two-dimensional cases and full integration of fixed and variable lengths. We have thereby shown that the general variable length optimization, which at a first glance seems unsuitable for a dynamic programming solution, can be very efficiently implemented using the appropriate data structures.

Chapter 8

Discussion and conclusions

8.1 The smart image sensor concept

A substantial part of this thesis has been devoted to processing image data as close to the sensor as possible, sometimes even before A/D-conversion. Hereby, an early data reduction takes place and the system is relieved from much overhead and I/O problems. This means that even if the processing capacity of the smart sensor is limited, it can significantly contribute to the overall performance of the system by reducing both bandwidth requirements as well as the computational load on subsequent processing stages.

However, this does not automatically imply that the low level algorithms employed on the sensor level are different in scope from those used in a traditional image processing system, but we have shown that the algorithms have to be adapted with ingenuity when they are to be put to work close to the sensor. Algorithms where the sensor itself is an integrated part of the process and where the exposure timing plays a key role we call *near-sensor* algorithms, indicating that they are related to the Near Sensor Image Processing paradigm as outlined by Åström and Forchheimer [21, 22, 60].

The concept of *sensor fusion* takes a somewhat new interpretation in this thesis. In the integrated sensor system described in Chapter 3, we have transformed the essentially two-dimensional matrix sensor to a multi-principle line scanning sensor system. Hence, in this case sensor fusion means the fusion of several sensor principles, or imaging modalities, onto a *single* two-dimensional sensor chip.

8.2 MAPP2200

An alternative title to this thesis could be "101 ways to use the analogue summation feature in MAPP2200". The tracheid measurement, the different sheet-of-light range imaging algorithms as well as the near-sensor spectrograph classification, all make use of the analogue summation in one way or another. Another key feature is the random access to the sensor rows, which indirectly gives the possibility to different exposure times for the different sensor rows. However, none of these features are particularly emphasized in the MAPP2200 documentation [111]. The analogue summation is mentioned in one sentence as a side-effect of the readout logic, indicating that its potential was not fully realized when the chip was designed.

Even if the existing MAPP2200 chip is very powerful, we have concluded that the resolution is still too low for the wood inspection application. There are also some problems with the MAPP2200 *camera*, i.e. the camera electronics surrounding the MAPP chip. First, the shift register output is not connected at all which means that data has to be output from the camera via the instruction bus. Therefore, shifting can not take place simultaneously with other processing. Furthermore, the A/D-conversion does not give 8 bits precision due to high noise levels, especially when A/D-conversion is performed concurrently to other processing. This is not so much of a problem when the sensor is used to acquire traditional images, but both the spectrograph and the FBM sheet-of-light algorithm are very dependent on the A/D-converter accuracy, and the noise levels will thereby directly affect the performance of the algorithms.

Another problem for our applications is the instability of the PD-bias voltage. For some reason, the noise level seems to be proportional to the number of rows used for analogue summation. The situation is acceptable if the rows are A/D-converted one by one, but when a large number of rows are accumulated, which is the case in both the spectrograph and several of the sheet-of-light algorithms, further stabilization of PD-bias is required.

As a result of the algorithms and experiments presented in this thesis, we conclude that to make full use of the more advanced possibilities of the MAPP sensor, the camera need to be redesigned with careful attention to noise levels and stabilization of reference voltages.

8.3 Other sensors

The algorithms and methods presented in this thesis require a sensor with random row addressing, flexible integration control and analogue summation. Are there other cameras or sensors on the market with these properties? The digital processor array in MAPP, or at least the GLU, is of less importance in our particular application since it is only the segmentation and feature extraction stage that makes use of it. However, in Chapter 6 we noted that the complexity of the segmentation stage is dependent on the number of objects on the line. Consequently, if we use a fixed cycle time we must be able to handle a worst case situation and as a result, the processors will be idling most of the time. It was therefore concluded in Chapter 6 that in order to achieve full performance of the system, we need a buffering stage where the bit vectors resulting from the thresholding stage are shifted out from the camera and then shifted back again asynchronously. It must be admitted though that it seems odd to shift data back again into the camera. A better solution might be to use a separate MAPP-like processor without the sensor and preferably also including a bit-serial multiplier which would significantly increase the ability to perform other image processing operations, such as convolutions and other filtering. Such a processor would also simplify the often inevitable shading correction. This processor actually exists, namely the VIP processor (Åström [20], Johannesson [89] and Hall [78]) shown in Figure 8.1. The WVIP version also contains a MAPP-like GLU.

A third alternative would be to use traditional digital signal processors which continuously are becoming more and more powerful. In either case, the conclusion is that the sensor does not really need the full digital processing capability of MAPP, but the bandwidth between the sensor and the next processing stage will be critical.

So far we have not been able to find any other sensors on the market which directly can be utilized to implement the sensor principles described in this thesis, even if there are some sensor designs which show a similar structure of the readout logic and where the columnwise analogue summation is possible, at least in principle. One such sensor is the VVL sensor described by Denyer et al. [48] and shown in Figure 8.2. The read-out logic is very similar to MAPP but the different rows are addressed using a shift register which selects the rows one at a time in sequence. If this sensor was modified to include a more flexible addressing structure it could very well be used to implement the near-sensor algorithms described in this thesis. Note also that the read-out logic of this chip includes the sample and hold stage discussed in Section 5.4.9.

Another sensor based on a similar read-out logic is the CIVIS sensor described by Ricquier et al. [137] and shown in Figure 8.3. This sensor is especially designed to perform analogue summation of pixel data in both direction. The main purpose is to obtain a programmable resolution imager so that regions of for instance 2×2, 4×4 or 8×8 pixels can be averaged directly on the sensor. However, since both the rows and columns are fully addressable, nothing prevents the accumulation of charges from entirely different regions of the sensor even if this was not

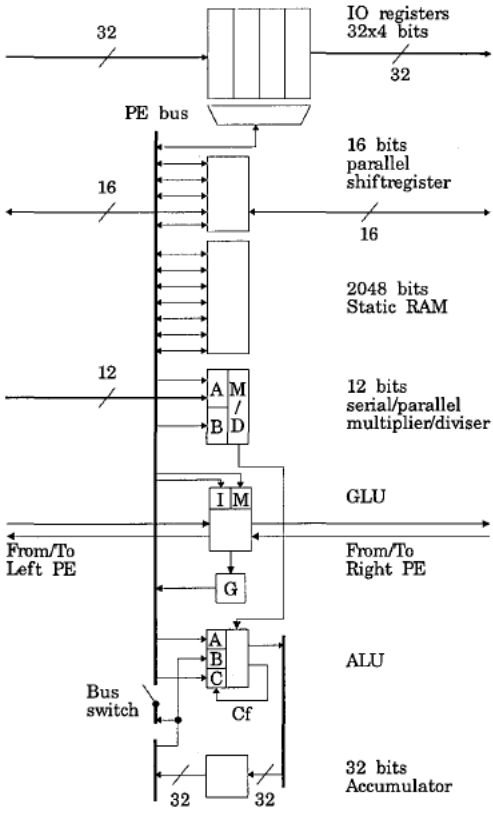


Figure 8.1 *The bit-serial VIP processor which compared to MAPP includes more memory, a multiplier and considerably higher I/O-bandwidth. From Hall [78].*

the intention when the sensor was designed. It is also fully possible to apply the reset pulses at any time intervals and variable integration times can thereby be implemented. Experiments with the CIVIS sensor and the spectrograph will be performed in the near future.

8.4 The wood inspection application

Judging from the literature study in Chapter 2, we claim that the current state-of-the-art in wood inspection is rather immature. Many of the suggested solutions are very ad hoc. We see two major reasons for this. First, many of the solutions emanates from research groups rooted in wood technology having little experience or training in signal and image processing. The result is a clear understanding of the problem but in some cases a less systematic approach to the solutions. The second reason is the very time critical nature of the problem which may prohibit the use of many traditional and more systematic image processing methods.

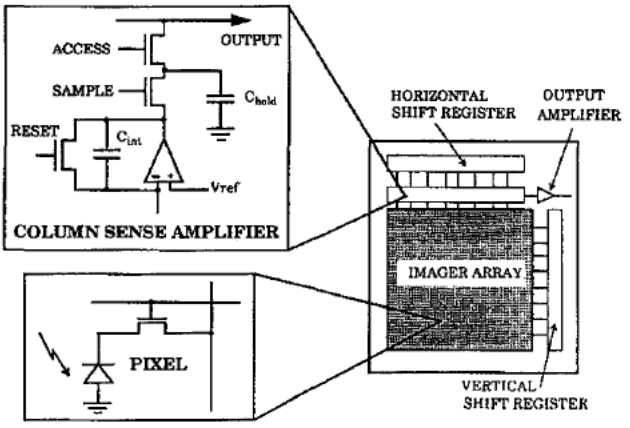


Figure 8.2 *The VVL sensor which has a read-out logic similar to MAPP but which lacks the flexible row addressing capabilities. From Denyer et al. [48].*

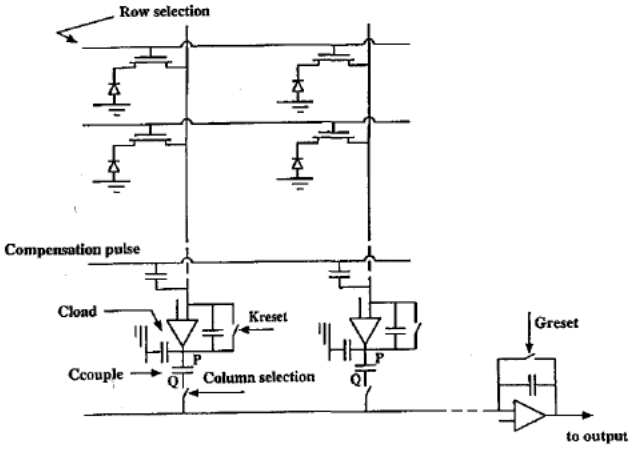


Figure 8.3 *The CIVIS sensor which was specifically designed to perform analogue summation of pixel data. From Ricquier et. al [137].*

This thesis does contribute not so much to high level defect detection but rather to techniques for executing low-level image processing at high speed and low cost. It is shown though that the new types of sensor data simplify the subsequent defect detection and that a system with reasonable performance can be designed using very simple threshold-based algorithms. Nevertheless, future research will be directed towards more advanced processing in the segmentation and defect detection stage.

The sensor level contributions can be divided into three fields, the integrated wood sensor system with the tracheid measurement as the main feature, the 3D-profiling measurement algorithms and the PGP-MAPP spectrograph.

The *tracheid measurement* has proven to be very good for detecting sound knots, especially on rough surfaces. Detection of light sound knots has long been known to be a problem with the systems on the market today and therefore this is one of the most significant results in this thesis. The tracheid effect in itself was discovered some 20 years ago (Matthews and Beech [117]), but since then no industrially feasible way to utilize the effect has been reported.

Other defects can be detected by the methods described in Chapter 6 at roughly the same confidence level as other systems, except for the detection of splits where the low resolution of the MAPP2200 sensor sets a limit. The conclusion is that a combination of the integrated sensor system and a traditional high resolution line scan camera will probably be needed in a full-scale system. A complete stand-alone system based on our technique would require MAPP-like sensors of higher resolution.

The *sheet-of-light range imaging* is an integrated part of the total sensor system shown in Figure 3.4, but can also be used stand-alone. Compared to the solutions based on standard cameras, the algorithms presented in this thesis give much higher line frequencies and thereby much higher resolution in the lengthwise direction when scanning moving objects. Note also that some of the proposed algorithms are a magnitude faster than the sheet-of-light range imaging implementations used in presently available commercial products based on MAPP2200 [1, 134].

The *spectrograph* system is a somewhat different approach to wood inspection than the integrated sensor system. It is not clear whether the wood surface actually contains enough spectral information to really discriminate the different defects. This is the subject of ongoing as well as future research. If the discrimination effect is there, then the PGP-MAPP imaging spectrograph is indeed a powerful tool to implement the spectral classification in real time. However, the spectrograph can also be regarded as a research tool to find significant spectral regions to detect some specific color defects. If such wavelength bands or combinations of bands can be identified, then appropriate optical filters can be designed and applied directly on the sensor chip. The analogue calculation of discriminant functions as described in Chapter 5 will then still be applicable even if there are fewer color channels.

The *cross cut optimization* chapter is technically different from the other material in the thesis. There we have shown that a fully integrated defect detection and optimization gives completely new possibilities to optimize the use of the raw material. The new optimization problem is fundamentally different from the traditional cross cutting optimization. We have shown that optimality can be obtained using dynamic programming and that the cutting lists are produced well within the time limits given by the industrial requirements. It is also quite pleasing to know that both the fixed length and the variable length versions are presently used in full production in a number of installations around the world.

Scanning for cross cutting in the secondary wood industry is the prime target application for the results in this thesis. Nevertheless, the different sensor principles are perfectly applicable in other wood inspection applications, at least as long as we are scanning using length transport. The specific sensor setup described in Chapter 3 can not be directly used in a cross transport situation. Especially the tracheid measurement is difficult to apply since it is based on the longitudinal transmittance of light along the tracheid cells and will not give the same effect if the laser line is oriented along the board. This means that the sensor system is less suitable for sawmill applications using cross transport. On the other hand, given that the mechanical problems can be handled, the speed performance of the system is such that one could allow

length transport and still be able to handle the sawmill volumes. The defects themselves on which the grading is based are the same in the sawmills as in most types of secondary processing.

The spectrograph could also be employed in applications where overall color of the wood piece is more of an issue than regular defect detection. A typical example is sorting of parquet floor tiles, but there are probably other applications where aesthetic appearance is important.

8.5 Other applications

The integrated sensor system is by nature focused on the wood inspection application, even though we have shown that the "tracheid effect" can be utilized to perform measurements on other semi-transparent materials. The idea of using a randomly addressable matrix sensor as a multi-principle line-scanning sensor could also be applicable in other applications even if the measurements actually employed could be very different.

Measurements of 3D-profile are relevant in many other applications than wood. For a survey of techniques, see Johannesson [89]. The different algorithms presented in this thesis have different merits and drawbacks concerning precision, speed, grayscale dependency and acquisition, error detection etc. and which algorithm to choose will depend on the application.

Classification according to color is also applicable to other fields than wood. The proposed system fills a gap between standard RGB-cameras where the spectral information is limited and traditional spectrophotometers which are precision instruments concerning the spectral information but which do not give the spatial dimension. The applications where the PGP-MAPP spectrograph is applicable are those where color differences are small and the speed constraints highly emphasized. The spatial dimension should also be a part of the problem, e.g. by requiring measurements not only of color but also of color homogeneity. Such applications could be found for instance in paper production or in the graphical industry, e.g. to monitor a printing process. However, the present PGP-MAPP spectrograph is a classification tool and not a calibrated precision instrument. For that, a better sensor with much lower noise levels and better dynamic range would be required.

8.6 Final conclusion

To summarize the thesis, the claimed contributions to the field are the following:

- The power of the smart image sensor concept is identified. It is shown that new algorithm possibilities appear when we use a flexible and fully addressable image sensor and when we can use the exposure time as a part of the algorithm.
- An integrated sensor system is presented where a number of different properties of the wood surface are measured simultaneously on a single sensor chip. The main feature of the sensor system is an industrially feasible method to utilize the so called tracheid effect, which is based on transmittance of light within the wood surface. Using this method, light sound knots can be reliably detected on rough surfaces, which is not the case in other reported systems.
- A number of sheet-of-light range imaging algorithms for dimension measurements are proposed giving range pixel speeds a magnitude higher than systems reported elsewhere. The high speed range imaging algorithms are applicable to many other fields than wood inspection.
- A real-time imaging spectrograph is presented. The system provides a tool for high-speed pixelwise classification based on an extended number of color channels. The sys-

tem has better color discrimination capability than solutions based on standard RGB cameras, while retaining the spatial dimension and a high speed compared to traditional spectrophotometers. The system is applicable in any situation where a fast line-based classification according to color is desired.

- It is shown that a fully automatic inspection system gives new possibilities to optimize the utilization of the raw material with respect to the final products. A dynamic programming optimization algorithm is presented enabling complex definitions of the products in terms of admissible defect contents in different zones in fixed as well as variable length products. It is shown that the algorithm produces optimal cutting lists well within the time limits set by the industrial requirements.

To conclude, we do not claim that we have presented the ultimate solution for wood surface inspection. However, we do believe that we have pushed the state of technology in the field a bit further in that direction. The process of automating the inspection tasks in the wood industry has only begun and the thesis should therefore be regarded as a starting point for further research and development, both academically and commercially.

References

[1] *A new view of surface and edge inspection for flooring panels*, SVEP. Svenska Elektronikprodukter AB.

[2] Adel M, Wolf D, Vogrig R, Husson R, *Evaluation of Colour Spaces in Computer Vision Application of Wood Defects Detection*, IEEE Int. Conf. on Systems, Man and Cybernetics, Le Touquet, France, Oct. 17-20, 1993.

[3] Alapuranen P, Westman T, *Automatic Visual Inspection of Wood Surfaces*, Proc. 11th Int. Conference on Pattern Recognition, The Hauge, The Netherlands, Aug 30-Sep 3, 1992.

[4] Almqvist B, *Sagan om VASA*, Bonniers förlag, Stockholm, 1965. (in Swedish)

[5] Andersson H, *An Investigation of radiation hardness and optical performance for smart optical sensors*, Master Thesis, LiTH-IFM-Ex-584, Linköping University, 1993.

[6] Araki K, et. al. *High speed and continuous 3-d measurement system*, 11th Int. Conference on Pattern Recognition, Vol. 4, Hague, 1992.

[7] Åstrand E, *Detection and Classification of Surface Defects in Wood - A Preliminary Study*, Report LiTH-ISY-I-1437, Linköping University, 1992.

[8] Åstrand E, Rönnqvist M, *Crosscut optimization of wooden boards based on automated inspection*, proc. SPIE, vol. 2063, Vision, Sensors, and Control for Automated Manufacturing Systems, Boston Sep. 7-10, 1993.

[9] Åstrand E, Rönnqvist M, *Crosscut optimization of boards given complete defect information*, Forest Products Journal, vol. 44, no. 2, p15-24, Feb. 1994.

[10] Åstrand E, Åström A, *An Intelligent Single Chip Multisensor Approach for Wood Defect Detection*, proc. SSAB Symposium on Image Analysis, Halmstad 1994.

[11] Åstrand E, Åström A, *An Intelligent Single Chip Multisensor Approach for Wood Defect Detection*, proc. M^2VIP, Mechatronics and Machine Vision in Practice, Toowoomba, Australia, Sep. 13-15, 1994.

[12] Åstrand E, Åström A, *A Single Chip Multi-Function Sensor System for Wood Inspection*, proc. 12th ICPR, International Conference on Pattern Recognition, Jerusalem, Oct. 9-13, 1994.

[13] Åstrand E, Johannesson M, Åström A, *Five Contributions to the Art of Sheet-of-light Range Imaging on MAPP2200*, report LiTH-ISY-R-1611, Linköping University, 1994.

[14] Åstrand E, Johannesson M, Åström A, *Two novel methods for range imaging using the MAPP2200 smart sensor*, proc. SPIE, vol. 2423, Machine Vision Applications in Industrial Inspection III, San Jose, Feb. 8-9, 1995.

[15] Åstrand E, Johannesson M, Åström A, *An ultra-fast algorithm for range imaging using the MAPP2200 smart sensor*, proc. SSAB Symposium on Image Analysis, Linköping 1995.

[16] Åstrand E, Åström A, *Metod och apparatur för detektering av defekter i trä*, Swedish Patent no. 501650, 1995.

[17] Åstrand E, Åström A, *Metod för spektralbaserad representation av bildelementen på en linje*, Swedish Patent no. 502197, 1995.

[18] Åstrand E, Österberg M, Hagman O, Åström A, *Near-Sensor Real Time Spectral Classification for Industrial Applications*, Proc. 9th Scandinavian Conference on Image Analysis, vol. 1, pp 33-40, Uppsala, Sweden, June 6-9, 1995.

[19] Åstrand E, Österberg M, Åström A, *Near-sensor real-time imaging spectroscopy for industrial applications*, proc. SPIE, vol. 2661, Real-Time Imaging, San Jose, Jan. 27 - Feb. 2, 1996.

[20] Åström A, Danielsson P-E, Chen K, Ingelhag P, Svensson C, *Videorate signal processing with PASIC and VIP*, proc. Barnaimage 91, Barcelona, Spain, 1991.

[21] Åström A, *Smart Image Sensors*, Ph.D. Thesis, Linköping University, 1993.

[22] Åström A, Forchheimer R, Eklund J-E, *Global Feature Extraction Operations for Near-Sensor Image Processing*, IEEE Transactions on Image Processing, vol. 5, no. 1, 1996.

[23] Aune J, *An X-ray Log-Scanner for Sawmills*, Proc. 2nd Int. Seminar/Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug. 14-16, 1995

[24] *Autopos - en smidig kantautomat*, Product sheet, Söderhamns Verkstäder, Sweden. (in Swedish)

[25] Bates T, *Scanner Installation Complexities from a User's Point of View*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[26] Blais F, Rioux M, *Real-time numerical peak detector*, Signal Processing 11, 1986.

[27] Blais F, Lecavalier M, Domey J, Boulanger P, Courteau J, *Application of the BIRIS Range Sensor for Wood Volume Measurement*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[28] Brazakovic D, Beck H, Sufi N, *An Approach to Defect Detection in Materials Characterized by Complex Textures*, Pattern Recognition, vol. 23, no. 1/2, pp 99-107, 1990.

[29] Brunner C C, White M S, Lamb F M, Schroeder J G, *CORY: a computer program for determining dimension stock yields*, Forest products journal, vol. 39, no. 2, pp 23-24, 1989.

[30] Brunner C C, Shaw G B, Butler D A, Funck J W, *Using Color in Machine Vision Systems for Wood Processing*, Wood and Fiber Science, vol. 22, no. 4, pp 413-428, Oct 1990.

[31] Brunner C C, Maristany A G, Butler D A, VanLeeuwen D, Funck J W, *An evaluation of color spaces for detecting defects in Douglas-fir veneer*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[32] Brunner C C, Maristany A G, Butler D A, Funck J W, *Enhancing color-image data for wood-surface feature identification*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[33] Butler D A, Brunner C C, Funck J W, *A dual-threshold image sweep-and-mark algorithm for defect detection in veneer*, Forest Products Journal, vol. 39, no. 5, pp 25-28, May 1989.

[34] Carnieri C, Mendoza G A, Luppold W G, *Optimal cutting of dimension parts from lumber with a defect: a heuristic solution procedure*, Forest Products Journal, vol. 43, no. 9, pp 66-72, 1993.

[35] Chang S J, Guddanti S, *Application of High Speed Image Processing in Hardwood Sawing Research*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[36] Cho T-H, Conners R W, Araman P A, *A Computer Vision System for Analyzing Images of Rough Hardwood Lumber*, Proc. 10th Int. Conference on Pattern Recognition, Atlantic City, 1990.

[37] Cho T-H, Conners R W, Araman P A, *A Computer Vision System for Automatic Grading of Rough Hardwood Lumber Using a Knowledge-based Approach*, Proc. IEEE Int. Conference on System, Man and Cybernetics, Los Angeles, 4-7 Nov 1990, pp 345-50.

[38] Choffel D, Goy B, Martin P, Gapp D, *Interaction Between Wood and Microwaves - Automatic Grading Application*, Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[39] Choffel D, Martin P, *Automatic Grading of Structural Wood*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[40] Conners R W, McMillin C W, Lin K, Vasquez-Espinoza R E, *Identifying and Locating Surface Defects in Wood: Part of an Automated Lumber Processing System*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, no. 6, pp 573-583, Nov 1983.

[41] Conners R W, McMillin C W, Vasquez-Espinoza R, *A Prototype Software System for Locating and Identifying Surface Defects in Wood*, Proc. 7th Int. Conference on Pattern Recognition, Montreal 1984, vol. 1, pp 416-29.

[42] Conners R W, Cho T-H, Araman P A, *Automatic Grading of Rough Hardwood Lumber*, Proc. 3rd Int. Conference on Scanning Technology in Sawmilling, Oct 5-6, Burlingame CA, 1989.

[43] Conners R W, Ng C T, Cho T-H, McMillin C W, *Computer Vision System for Locating and Identifying Defects in Hardwood Lumber*, SPIE vol 1095, Applications of Artificial Intelligence VII, pp 48-63, 1989.

[44] Conners R, Cho T-H, Ng C T, Drayer T H, Araman P A, Brisbin R L, *A machine vision system for automatically grading hardwood lumber*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[45] Dashner B, *3D Log Scanning - The Next Generation*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[46] Davis J, Wells P, *Computed tomography measurements on wood*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[47] Davis J, *The GLASS LOG Project*, Project description, Monash University, Australia, 1995.

[48] Denyer P B, Renshaw D, Gouyu W, Ying L M, Anderson S, *On-chip CMOS sensors for VLSI imaging systems*, Proc. VLSI-91, 1991.

[49] Dernando E, *Dynamic Programming: Theory and Applications*, Prentice Hall, Englewood Cliffs, 1982.

[50] Dowsland K A, Dowsland W B, *Packing problems*, European Journal of Operational Research, 56:2-14, 1992.

[51] Dreyfus S E, Law A M, *The Art and Theory of Dynamic Programming*, Academic Press, New York, 1977.

[52] Duda R O, Hart P E, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[53] *EDGAR - en avancerad kantautomat*, Product sheet, Söderhamns Verkstäder, Sweden. (in Swedish)

[54] Ersbøll B K, Conradsen K, *Automated Grading of Wood-slabs Using Visual Inspection*, Proc. 2nd workshop on Industrial Machine Vision, Kuusamo Finland, March 29-31, 1992.

[55] Ersbøll B K, Conradsen K, *Automated grading of wood slabs: The development of a prototype system*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[56] Ersbøll B K, Conradsen K, *A Strategy for Grading Natural Materials using a Two Step Classification Procedure*, SPIE vol. 1821, Industrial Applications of Optical Inspection, Metrology, and Sensing, pp 318-329, 1992.

[57] Fisher M L, *The langrangian relaxation method for solving integer programming problems*, Management Science, vol. 27, no. 1, pp 1-18, 1981.

[58] Fisher M L, *An application oriented guide to langrangian relaxation*, Interfaces, vol. 15, no. 2, pp 10-21, 1985.

[59] Forchheimer R, Ingelhag P, Jansson C, *MAPP2200, a second generation smart optical sensor*, SPIE vol. 1659, 1992.

[60] Forchheimer R, Åström A, *Near-Sensor Image Processing: A New Paradigm*, IEEE Transactions on Image Processing, vol. 3, no. 6, 1994.

[61] Foronda S U, Carino H F, *A heuristic approach to the lumber allocation problem in hardwood dimension and furniture manufacturing*, European Journal of Operational Research, 54:151-162, 1991.

[62] Forrer J B, Butler D A, Funck J W, Brunner C C, *Image sweep-and-mark algorithms. Part 1. Basic algorithms*, Forest Products Journal, vol. 38, no. 11/12, pp 75-79, Nov/Dec 1988.

[63] Forrer J B, Butler D A, Brunner C C, Funck J W, *Image sweep-and-mark algorithms. Part 2. Performance evaluations*, Forest Products Journal, vol. 39, no. 1, pp 39-42, Jan 1989.

[64] Garcia D, et. al., *Flatness defect measurement system for steel industry based on a real-time linear-image processor*, Proc of IEEE/SMC'93, Le Touquet 1993.

[65] Gilmore P C, Gomoroy R E, *Multistage cutting stock problem of two and more dimensions*, Operations Research, 13:94-120, 1965.

[66] Granström P, Hilbers H, *Undersökning av metod för utvärdering av automatsorteringsutrustningar av trä*, Projektrapport, Högskolan i Luleå, Inst. för Träteknik, Skellefteå, 1992. (in Swedish)

[67] Green J C, *3-D Log Scanning and Optimization*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[68] *Gröna boken - Sortering av sågat virke av furu och gran*, Föreningen Svenska Sågverksmän, 1976. (in Swedish)

[69] Grönlund U, *Quality Variations in the Chain Log-Board-Blank Caused by Biological Properties and Subjective Grading Procedures*, Licentiate Thesis, Luleå University of Technology, 1992.

[70] Grönlund U, *Quality Improvements in Forest Products Industry*, Ph.D. thesis, Luleå University of Technology, 1995.

[71] Grundberg S, Grönlund A, *Log Scanning - Extraction of Knot Geometry*, Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[72] Grundberg S, Grönlund A, *The development of a Log-Scanner for Scots Pine*, Proc. 2nd Int. Seminar/Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug. 14-16, 1995

[73] Gruss A, Carley L R, Kanade T, *Integrated Sensor and Rangefinding Analog Signal Processor*, IEEE Journal of Solid State Circuits, Vol 26, pp 184-191, 1991.

[74] Hagman P O G, *Multivariate methods to classify wooden surfaces*, Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[75] Hagman P O G, *Automatic quality grading of logs with TINA, a Gamma Ray Log Scanner*, Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[76] Hagman P O G, *Prediction of Wood Quality Features by Multivariate Nodels based on Scanning Techniques*, Licentiate thesis, Luleå University of Technology, 1993.

[77] Hagman P O G, *Multivariate Prediction of Wood Surface Features using an Imaging Spectrograph*, Proc. 2nd Int. Seminar/Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug. 14-16, 1995.

[78] Hall M, *A Study of Parallel Image Processing on the VIP Architecture*, Licentiate Thesis LIU-TEK-LIC-1995:41, Linköping University, 1995.

[79] Herrala E, Okkonen J, Hyvärinen T, Aikio M, Lammasniemi J, *Imaging spectrometer for process industry applications*, SPIE vol. 2248, Optical Measurements and Sensors for the Process Industries, 1994.

[80] Herrala E, Oral communication, 1995.

[81] Huber H A, Ruddell S, Mukherjee K, McMillin C, *Economics of cutting hardwood dimension parts with an automated system*, Forest Products Journal, vol. 39, no. 5, pp 46-50, May 1989.

[82] *Imaging PGP-Spectrograph*, Product data sheet, Specim Ltd., Finland, 1995.

[83] Jansson C, Ingelhag P, Svensson C, Forchheimer R, *An Addressable 256x256 Photodiode Image Sensor Array with an 8-bit Digital Output*, ESSIRC '92, Copenhagen 1992.

[84] Jansson C, Oral communication, 1994.

[85] Johannesson M, *Sheet-of-light range imaging experiments with MAPP2200*, Report no. LiTH-ISY-I-1401, Linköping University, Sep 1992.

[86] Johannesson M, Åström A, *Sheet-of-light range imaging with MAPP2200,* Proc of SCIA -93, Tromsø 1993.

[87] Johannesson M, *Sheet-of-Light Range Imaging*, Licentiate Thesis LIU-TEK-LIC-1993:46, Linköping University, 1993.

[88] Johannesson M, Åstrand E, Åström A, *A Fast and Programmable Sheet-of-light Range Imager using MAPP2200*, Proc. SSAB Symposium on Image Analysis, Halmstad 1994.

[89] Johannesson M, *SIMD Architectures for Range and Radar Imaging*, Ph.D. thesis, Linköping University, 1995

[90] Jonsson L, *Recording Annual Ring Development Using Image Analysis,* Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[91] Kauppinen H, Silvén O, *Development of color vision based solutions for lumber grading*, SPIE vol. 2423, Machine Vision Applications in Industrial Inspection III, 1995.

[92] Kauppinen H, Silvén O, *A Color Vision Approach for Grading Lumber*, Theory & Application of Image Processing II - Selected papers from the 9th Scandinavian Conference on Image Analysis, World Scientific, 1995.

[93] Kim C W, Koivo A J, *Hierarchical Classification of Surface Defects on Dusty Wood Boards*, Proc. 10th Int. Conference on Pattern Recognition, Atlantic City, 1990.

[94] Kim C W, Koivo A J, *Hierarchical classification of surface defects on dusty wood boards*, Pattern Recognition Letters, vol. 15, No. 7, pp 713-721, 1994.

[95] Klinkhachorn P, Franklin J P, McMillin C W, Huber H A, *ALPS: yield optimization cutting program*, Forest Products Journal, vol. 39, no. 3, pp 53-56, 1989.

[96] Klinkhachorn P, Kothari R, Huber H A, *A Modified Vision System for the Automated Lumber Processing System (ALPS)*, Proc. 4th Int. Conference on Scanning Technology in the Wood Industry, San Francisco, Oct 28-29, 1991.

[97] Koivo A J, Kim C W, *Classification of Surface Defects on Wood Boards*, IEEE Int. Conference on System, Man and Cybernetics, Oct 1986, pp 1431-36.

[98] Koivo A J, Kim C W, *Parameter Estimation of CAR Models for Classifying Wood Boards*, Proc. 1988 IEEE Int. Conference on System, Man and Cybernetics, Beijing and Schenyang, China, 8-12 Aug 1988, pp 1376-79, vol. 2.

[99] Koivo A J, Kim C W, *Robust Image Modelling for Classification of Surface Defects on Wood Boards*, IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 6, pp 1659-66, 1989.

[100] Koivo A J, Kim C W, *Automatic classification of surface defects on red oak boards*, Forest Products Journal, vol. 39, no. 9, pp 22-30, Sep 1989.

[101] Labeda A, *The Implementation of Neural Networks in Rough Lumber Grading*, Report, Trätek, Swedish institute for wood technology research, 1991.

[102] Labeda A, *Practical Implementations of Cognitive Processing Methods in Wood Scanning*, Proc. 5th Int. Conf. on Scanning Technology & Process Control For the Wood Products Industry, Atlanta, Oct. 25-27, 1993.

[103] Lampinen J, Smolander S, Silvén O, Kauppinen H, *Wood Defect Recognition: A Comparative Study*, Proc. Workshop on Machine Vision for advanced production, Oulu, Finland, June 2-3, 1994.

[104] Lampinen J, Smolander S, *Wood defect recognition with self-organizing feature selection*, SPIE vol. 2353, Intelligent Robots and Computer Vision XIII: Algorithms and Computer Vision, 1994.

[105] Lee S C, Qian G S, Chen J-B, Hay D A, *Scanning Logs for Knots*, Proc. 7th Scandinavian Conference on Image Analysis, Aalborg, Denmark, Aug 13-16, 1991.

[106] Lindgren O, *Medical CT-Scanners for Non-Destructive Wood Density and Moisture Content Measurements*, Ph.D. thesis, Luleå University of Technology, 1992.

[107] Lycken A, Uusijärvi R, *Automatisk sortering och optimering av sågat virke*, Kontenta, Trätek, Institutet för träteknisk forskning, Stockholm, Sweden 1992. (in Swedish)

[108] Lycken A, Uusijärvi R, Hagman O, Åstrand E, *Automatoptimering av sågat virke - sammanfattning av AO-projektet 920101-931031*, L-rapport, Trätek Stockholm, 1993. (in Swedish)

[109] Manetsch T J, Huber H A, *An economic and system analysis of the automated hardwood lumber processing system (ALPS)*, Forest Products Journal, vol. 42, no. 3, pp 51-56, March 1992.

[110] Månson M, *Technical Achievements Offering New Grading Principles for Softwood Lumber*, Proc. 4th Int. Conference on Scanning Technology in the Wood Industry, San Francisco, Oct 28-29, 1991.

[111] *MAPP2200 Technical Description*, Integrated Vision Products AB, 1992.

[112] *MAPPC Mapp Compiler*, Soliton Elektronik AB, 1995.

[113] Maristany A G, Butler D A, Brunner C C, Funck J W, *Exploiting Local Color Information for Defect Detection on Douglas-Fir Veneer*, Proc. 4th Int. Conference on Scanning Technology in the Wood Industry, San Francisco, Oct 28-29, 1991.

[114] Maristany A G, Lebow P K, Brunner C C, Butler D A, Funck J W, *Classifying wood-surface features using dichromatic reflection*, SPIE vol. 1836, Optics in Agriculture and Forestry, 1992.

[115] Marszalec E, Pietikainen M, *Colour Analysis of Defects for Automated Visual Inspection of Pine Wood*, SPIE vol. 1907, Machine Vision Applications in Industrial Inspection, pp 80-89, 1993.

[116] Martello S, Toth P, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley, 1990.

[117] Matthews P C, Beech B H, *Method and apparatus for detecting timber defects*, U.S. Patent no. 3976384, 1976.

[118] *Measuring & benefits - the LIGNA magazine from selcom*, Selcom AB, 1995.

[119] Milota M R, *Methods for the measurement of bow and crook*, Forest Products Journal, vol. 41, no. 9, pp 65-67, Sep 1991.

[120] Nemhauser G, Wolsey N, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.

[121] *Nordiskt Trä - Sorteringsregler för sågat virke av furu och gran*, Föreningen Svenska Sågverksmän, 1994. (in Swedish)

[122] Ojala P, Pietikäinen M, Silvén O, *Edge-Based Texture Measures for Surface Inspection*, Proc. 11th Int. Conf. on Pattern Recognition, The Hague, The Netherlands, Aug. 30-Sep. 3, 1992.

[123] Österberg M, *Quality Functions for Parallel Selective Principal Component Analysis*, Ph.D. thesis, Linköping University, 1994.

[124] Österberg M, Åstrand E, Rönnqvist M, Åström A, *Computation of Linear Discriminant Functions in the MAPP Spectral Classification System*, Report to be published, Linköping University, 1996.

[125] Paul D, Hättisch W, Nill W, Tatari S, Winkler G, *VISTA: Visual Interpretation System for Technical Applications - Architecture and Use*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, pp 399-407, 1988.

[126] Penman D, Olsson O, Bowman C, *Automatic inspection of reconstituted wood panels for surface defects*, SPIE vol. 1823, Machine Vision Applications, Architectures, and Systems Integration, 1992.

[127] Pölzleitner W, *A Hough Transform Method to Segment Images of Wooden Boards*, Proc. 8th Int. conference on Pattern Recognition, Paris, Oct 1986.

[128] Pölzleitner W, Schwingshakl G, *Real-time classification of wooden boards*, SPIE vol. 1384, High Speed Inspection Architectures, Barcoding and Character Recognition, pp 38-49, 1990.

[129] Pölzleitner W, *Real-time Syntactic Segmentation with Applications to Images of Wooden Boards*, Proc. Workshop on Digital Image Processing and Computer Graphics, Vienna, Academy of science, Oct 1991.

[130] Pölzleitner W, *Knowledge Representation for 2-D Real-Time Object Recognition*, Proc IEEE Int. Conf. on Systems, Man and Cybernetics, Charlotteville, VA, 1991.

[131] Pölzleitner W, Schwingshakl G, *Real-time surface grading of profiled wooden boards*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[132] Pölzleitner W, *Convex shape refinement using dynamic programming for surface defect classification on wooden materials*, SPIE vol. 2055, Intelligent Robots and Computer Vision XII, 1993.

[133] Portala J-F, Ciccotelli J, *Nondestructive testing techniques applied to wood scanning*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[134] *RANGER RS2200 Technical Description*, Integrated Vision Products AB, 1994.

[135] Regalado C, Kline D E, Araman P A, *Value of defect information in automated hardwood edger and trimmer systems*, Forest Products Journal, vol. 42, No. 3, pp 29-34, 1992.

[136] Rice R W, Steele P H, Lumar L, *Detecting knots and voids in lumber with dielectric sensors*, Industrial Metrology, vol. 2, no. 3-4, pp 195-218, 1992.

[137] Ricquier N, Debusschere I, Dierickx B, Alaerts A, Vlummens J, Claeys C, *The CIVIS sensor: a flexible smart imager with programmable resolution*, SPIE vol. 2172, Charge Coupled Devices and Solid State Optical Sensors IV, 1994.

[138] Riihinen J, *Automatic grading and optimizing for green trimming and green sorting - practical applications in production lines*, Proc. Workshop on Machine Vision for advanced production, Oulu, Finland, June 2-3, 1994.

[139] Rönnqvist M, *A method for the cutting stock problem with different qualities*, European Journal of Operational Research, 83, 57-68.

[140] Rönnqvist M, Åstrand E, Åström A, Österberg M, *A solution approach for the spectral classification problem*, Internal report, Dept. of Engineering Science, University of Auckland, New Zealand, 1996.

[141] Rönnqvist M, Åstrand E, *Integrated defect detection and optimization for cross cutting of wooden boards*, Submitted to European Journal of Operational Research, 1996.

[142] Rowa P, *Automatic Visual Inspection of Lumber*, Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[143] Sarker B R, *An optimum solution for one-dimensional slitting problems: a dynamic programming approach*, Journal of Operational Research Society, 39(8):749-755, 1988.

[144] Schalkoff R, *Pattern Recognition - Statistical, Structural and Neural Approaches*, John Wiley & Sons, 1992

[145] Schmoldt D L, Li P, Abbott A L, *Log Defect Recognition Using CT Images and Neural Net Classifiers*, Proc. 2nd Int. Seminar/Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug. 14-16, 1995

[146] Seltman J, *Indication of Slope-of-grain and Biodegradation in Wood with Electromagnetic Waves*, Proc. Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug 31-Sep 1, 1992.

[147] Serra J, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1980.

[148] Silvén O, Kauppinen H, *Color Vision Based Methodology for Grading Lumber*, Proc. 12th Int. Conf. on Pattern Recognition, Jerusalem, Israel, Oct. 9-13, 1994.

[149] *Skogsstatistisk årsbok*, Skogsstyrelsen, 1995. (in Swedish)

[150] Sobey P J, Semple E C, *Detection and sizing visual features in wood using tonal measures and a classification algorithm*, Pattern Recognition, vol. 22, no. 4, pp 367-380, 1989.

[151] Sobey P J, *The Automated Visual Inspection and Grading of Timber*, Ph.D. thesis, University of Adelaide, March 1989.

[152] Soest J, *Geometry Scanner for Secondary Breakdown Applications*, Proc. 4th Int. Conference on Scanning Technology in the Wood Industry, San Francisco, Oct 28-29, 1991.

[153] Soest J, Matthews P, Wilson B, *A Simple Optical Scanner for Grain Defects*, 5th int. conf. on Scanning Technology & Process Control for the Wood Products Industry, Atlanta, Oct 25-27, 1993.

[154] Steele P H, Neal S C, McDonald K A, Cramer S M, *The slope-of-grain indicator for defect detection in unplaned hardwood lumber*, Forest Products Journal, vol. 41, no. 1, pp 15-20, Jan 1991.

[155] Svalbe I, Som S, Davis J, Grant J, Tsui K, Wells P, *Internal scanning of logs for grade evaluation and defect location*, Proc. 2nd Int. Seminar/Workshop on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug. 14-16, 1995.

[156] *Svensk Limfogsguide*, Träindustrins allmänna Grupp, Stockholm 1985. (in Swedish)

[157] Sweeney P E, Haessler R W, *One-dimensional cutting stock decisions for rolls with multiple quality grades*, European Journal of Operational Research, 44:224-231, 1990.

[158] Tatari S, Hättisch W, *Automatic Recognition of Defects in Wood*, Proc. SPIE vol. 804, Advances in Image Processing, pp 229-36, 1987.

[159] Tatari S, *Selection and Parametrization of Texture Analysis Methods for the Automation of Industrial Visual Inspection Tasks*, SPIE vol 829, Applied Digital Image Processing X, pp 86-94, 1987.

[160] Tatari S, Paul D, *Verfahren und Geräte zur schnellen automatischen Prüfung texturierter Oberflächen*, Technisches Messen tm, 55. Jahrgang, Heft 12/1988. (in German)

[161] Tian Q, Fainman Y, Gu Z H, Lee S H, *Comparison of statistical pattern-recognition algorithms for hybrid processing. 1. Linear-mapping algorithms*, Journal of Optical Society of America, Vol. 5, No. 10, pp 1655-69, 1988.

[162] *Typical Defects of Wood*, http://ee.oulu.fi/~olli/Projects/Wood.Defects.html.

[163] Vaarala T, Moring I, Herrala E, *Imaging Spectroscopy for On-line Wood Quality Inspection: An Experimental Study*, 2nd Int. Seminar on Scanning Technology and Image Processing on Wood, Skellefteå, Sweden, Aug. 14-16, 1995.

[164] Verbeek P.W, et al., *Video-speed Triangulation Range Imaging,* NATO ASI Series, Vol F 63, pp 181-186, Springer-Verlag Berlin, Heidelberg 1990.

[165] Westman T, Alapuranen P, *Experimental System for Automatic Plywood Inspection,* Proc. 2nd Workshop on Industrial Machine Vision, Kuusamo Finland, Mar. 29-31, 1992.

[166] Whelan P F, Batchelor B G, *Automated packing systems: review of industrial implementations,* SPIE vol. 2064, Machine Vision Applications, Architectures and Systems Integration II, 1993.

[167] Wirth N, *Algorithms + Data Structures = Programs,* Prentice-Hall, Englewood Cliffs, 1976.

[168] Yläkoski I, Visa A, *A Two-Stage Classifier for Wooden Boards,* Proc. 8th Scandinavian Conference on Image Analysis, Tromsø, Norway, May 25-28, 1993.

[169] Zscheile M, *Einsatz eines mathematischen Modells der zweidimensionalen Zuschnittoptimierung zur rechnergesteuerten Verarbeitung von Vollholz,* Ph.D. thesis, Technische Universität Dresden, Germany, 1987. (in German)

Appendix A

Color images

Figure 1.29 Blue stain.

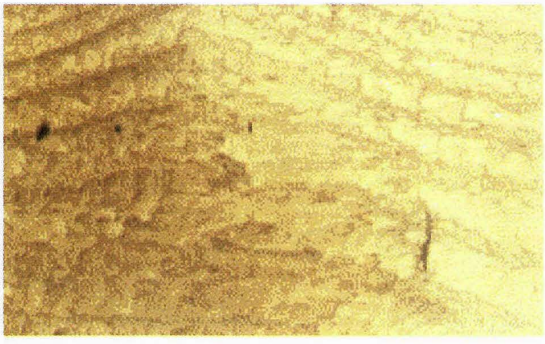


Figure 1.30 Brown stain.

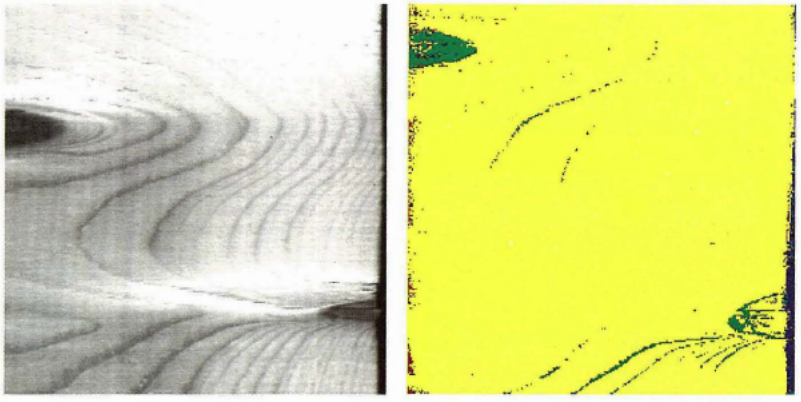


Figure 5.17 *Classification of spruce boards using the real time PGP-MAPP spectrograph. Yellow represents clear wood, green sound knots, red blue stain and blue dry knots.*

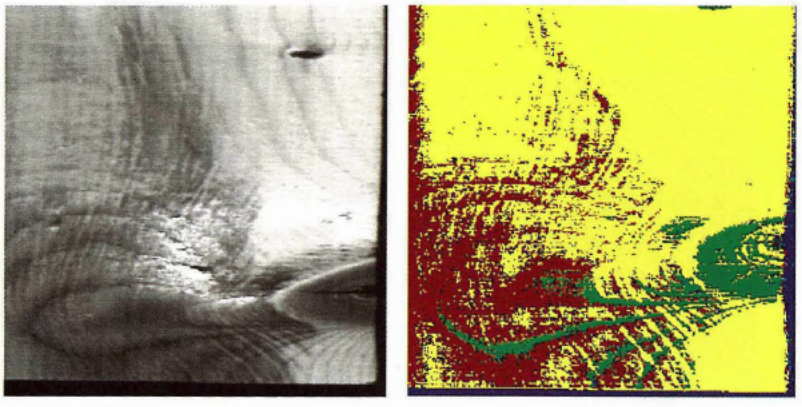


Figure 5.18 *A board containing a blue stain area and a sound knot. Note that the blue stain contains a lot of pixels classified as dry knot.*

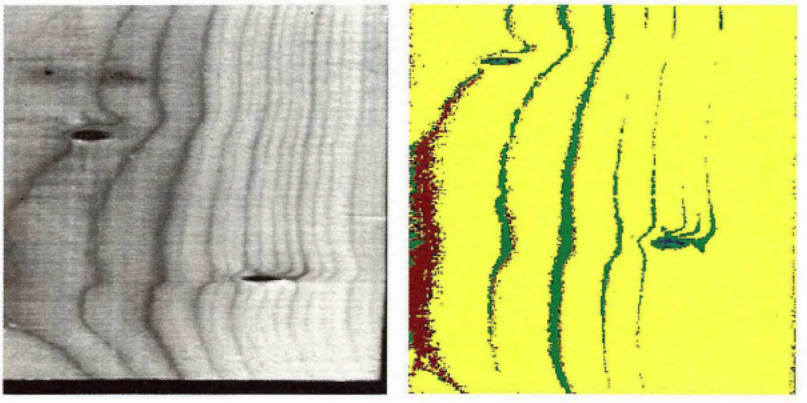


Figure 5.19 *A board illustrating the non-homogenous nature of the clear wood. The bright areas are correctly classified whereas the dark grain is classified as either sound knot or blue stain.*

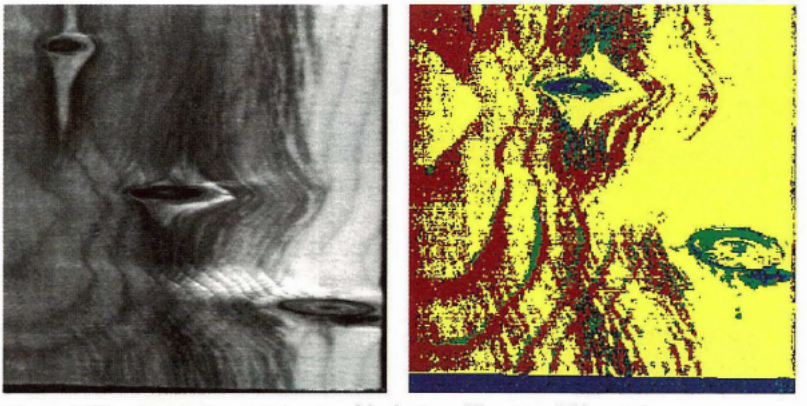


Figure 5.20 *A board containing two black-ringed knots and blue stain.*

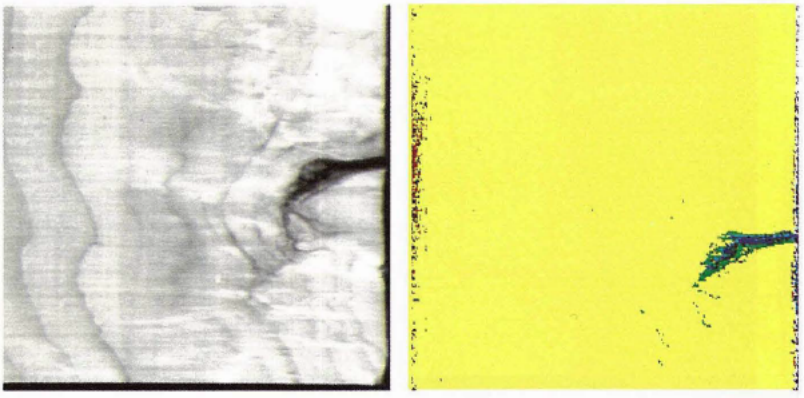


Figure 5.21 A dry horn knot which is clearly identified.

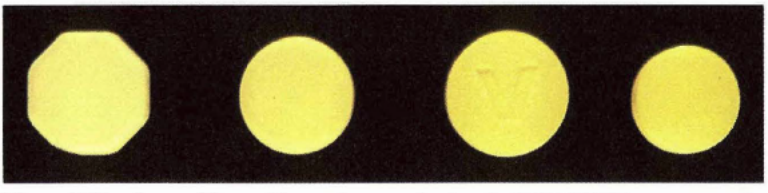


Figure 5.22 A set of four tablets to be sorted using the PGP-MAPP spectrograph.

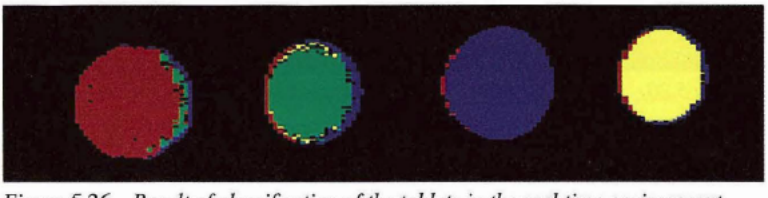


Figure 5.26 Result of classification of the tablets in the real time environment.

Ph.D Dissertations

Image Processing Laboratory, Department of Electrical Engineering
Linköping University, Sweden

Björn Kruse, *Design and implementation of a picture processor.* Linköping Studies in Science and Technology. Dissertations No. 13, 1977.

Kameswara Rao, *Pattern recognition techniques applied to fingerprints.* Linköping Studies in Science and Technology. Dissertations No. 19, 1977.

Björn Gudmundsson, *Contributions to the design of a picture processing system.* Linköping Studies in Science and Technology. Dissertations No. 42, 1979.

Roger Cederberg, *On coding, processing and display of binary images.* Linköping Studies in Science and Technology. Dissertations No. 57, 1981.

Olov Fahlander, *Development towards animation in computer graphics.* Linköping Studies in Science and Technology. Dissertations No. 78, 1982.

Reiner Lenz, *Reconstruction, processing and display of 3D-images.* Linköping Studies in Science and Technology. Dissertations No. 151, 1986.

Björn Lindskog, *PICAP3. An SIMD architecture for multi-dimensional signal processing.* Linköping Studies in Science and Technology. Dissertations No. 176, 1988.

Qin-Zhong Ye, *Contributions to the development of machine vision algorithms.* Linköping Studies in Science and Technology. Dissertations No. 201, 1989.

Olle Seger, *Model building and restoration with applications in confocal microscopy.* Linköping Studies in Science and Technology. Dissertations No. 301, 1993.

Ingemar Ragnemalm, *The Euclidean distance transform.* Linköping Studies in Science and Technology. Dissertations No. 304, 1993.

Anders Åström, *Smart Image Sensors.* Linköping Studies in Science and Technology. Dissertations No. 319, 1993.

Maria Magnusson (Seger), *Linogram and other direct Fourier methods for tomographic reconstruction.* Linköping Studies in Science and Technology. Dissertations No. 320, 1993.

Mats Österberg, *Quality functions for parallel selective principal component analysis.* Linköping Studies in Science and Technology. Dissertations No. 345, 1994.

Mats Gökstorp, *Depth Computation in Robot Vision,* Linköping Studies in Science and Technology. Dissertations No. 378, 1995.

Mattias Johannesson, *SIMD Architectures for Range and Radar Imaging,* Linköping Studies in Science and Technology. Dissertations No. 399, 1995.

ISBN 91-7871-693-4 ISSN 0345-7524