Sheet-of-light Range Imaging

Mattias Johannesson

Reprint of Part I of

Linköping Studies in Science and Technology. Dissertations No. 399

SIMD Architectures for Range and Radar Imaging

Linköping 1995

Reprint of Part I of
"Linköping Studies in Science and Technology. Dissertations No. 399
SIMD Architectures for Range and Radar Imaging
by Mattias Johannesson
ISBN 91-7871-609-8, ISSN 0345-7524"
for IVP, Integrated Vision Products AB, Linköping, Sweden.
Printed in Sweden by L.J. Foto & Montage, Linköping/VTT-Grafiska, Vimmerby, 1995

Abstract

This thesis describes sheet-of-light range imaging and Doppler radar signal processing. The focus is on implementing algorithms on bit-serial SIMD processor arrays.

The reduction of 2D sensor data to 1D position data in sheet-of-light range imaging is a task well suited for smart optical sensors combining image acquisition and image processing on a single chip. Several smart sensor approaches aimed at sheet-of-light range imaging are presented. Among these is a novel sheet-of-light range imaging architecture based on the Near-Sensor Image Processing concept. We predict that this architecture is capable of range imaging with range pixel (rangel) frequencies up to 25 MHz while keeping a relative accuracy in the range of 1/500 and delivering intensity data concurrently with range. We also present several sheet-of-light range imaging algorithm implementations using the commercial smart image sensor MAPP2200. With this system we can reach up to 5 MHz rangel frequency with a relative accuracy of 1/500. General sheet-of-light system setup and calibration issues are also covered.

We also present how three different Doppler radar algorithms can be implemented on different versions of the VIP, Video Image Processor, architecture. VIP is a family of bit-serial SIMD processor arrays aimed at real-time signal processing. We present the design of three VIP architectures, RVIP, FVIP, and MVIP. The Radar VIP, RVIP, is aimed at ground-based surveillance radar applications. The Flight VIP, FVIP, is aimed at airborne search radars. The Matrix VIP, MVIP, finally, is aimed at airborne space-time adaptive processing in phased array antenna applications. Common for all these radars is that the real-time processing demands are very high, ranging from 2 to 40 Giga MAC operations per second, and that the IO data rates are high, up to 1.5 Gbit per second. FFTs are an important part of Doppler radar signal processing and FFT implementations on a VIP array are therefore presented in detail.

The RVIP architecture has been manufactured by Ericsson Microwave systems for use in their future radar systems.

Acknowledgments

First I wish to thank my supervisor Professor Per-Erik Danielsson and co-supervisor Doctor Anders Åström for their help, suggestions and valuable discussions.

A very special hats-off thanks to Doctor Mats Gökstorp for all discussions, pep-talk and the offer to proof read large parts of this thesis (of course, I read parts of his so what would you expect...). Special thanks also to Michael Hall for innumerous VIP discussions, especially during the summer of '95 when we both were writing VIP theses.

Thanks to Per Ingelhag, Christer Jansson, Anders Edman and Jan-Erik Eklund at the department of physics, IFM, for help with the VLSI architecture issues, both within the sheet-of-light and the VIP projects.

Thanks to Erik Åstrand who devised several of the algorithms presented in chapter 7, and has also given invaluable input to many other smart MAPP algorithms.

Without the initiative and help of Ragnar Arvidsson, Tor Ehlersson, Bo Lyckegård, Ulf Näsström, and several others at EMW there would never have been any radar VIP architectures, and no second part of this thesis, thank you very much.

I must also thank everybody at IVP for giving me the opportunity to take my research into the "real world" when implementing the RANGER system.

Thanks to everybody who has contributed to the Elauto atmosphere through the years, especially to the "Acats" Mats G., Magnus H., and Mikael W., and to the system administrators and secretaries.

The financial support of CENIIT's (Center for industrial information technique) VLSI for vision project and NUTEK's (Swedish National Board for Technical Development) grant No. 9303459 are gratefully acknowledged.

Last but not least, thanks to Myself, I couldn't have done this without Me.

Contents

Preface to this edition

This book is a reprint of the first half of my Ph.D. dissertation. This half concerns sheet-of-light range imaging. The other, not included, half, concerns the development of the VIP architecture. VIP is a bit-serial SIMD architecture for real-time image and signal processing. For an orientation of the contents of the complete dissertation, we refer to the section "About this thesis" which gives an overview of the dissertation.

About this thesis

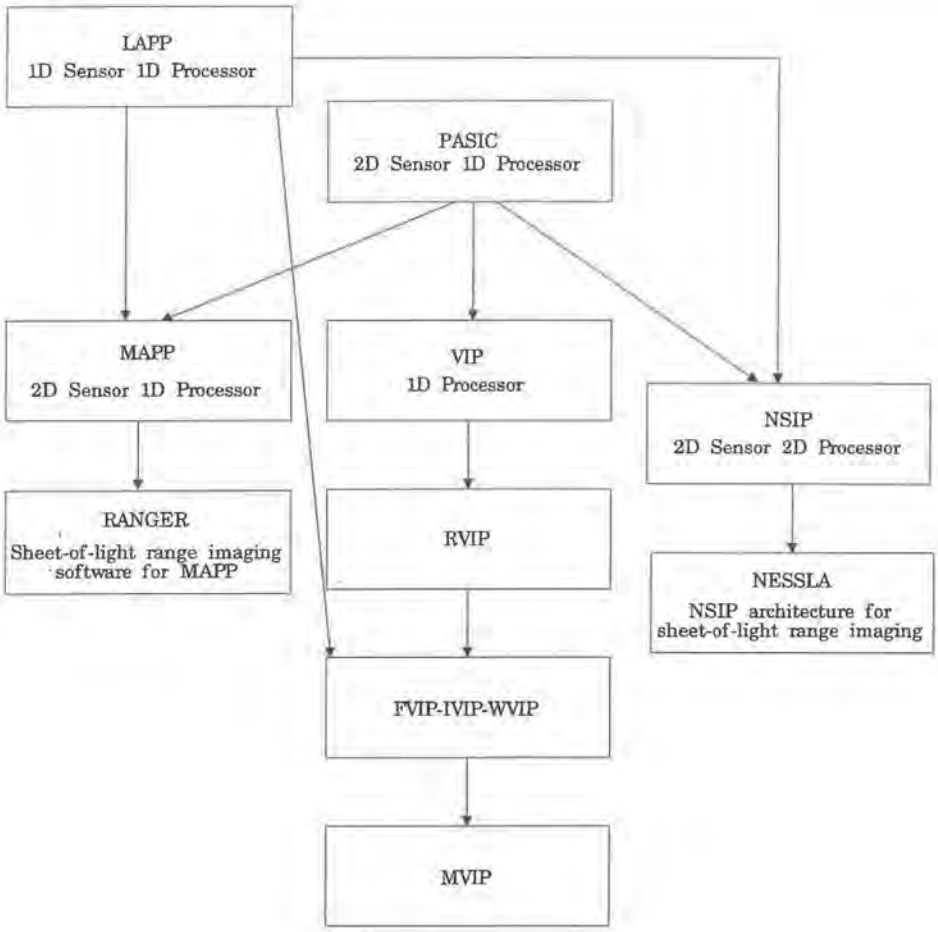
Background

This thesis describes work in two seemingly very disparate areas, triangulation-based range imaging using sheet-of-light illumination, and bit-serial Single Instruction stream Multiple Data stream, SIMD, architectures aimed at Doppler radar signal processing. However, both these projects share the same origins in the SIMD-architecture/Smart-Sensor research at Linköping University [5]. All architectures described in this thesis evolved at least partly from the PASIC project which started in the late eighties at Linköping University [2], [4], [29], see Figure 1. PASIC, which stands for Processor, A/D-converter and Sensor Integrated Circuit, combined a fully addressable 128x128 photo diode sensor with a linear array of A/D converters and bit-serial Processing Elements, PEs, on a single chip. PASIC and LAPP1100, Linear Array Picture Processor, [12] together evolved into the MAPP2200, Matrix Array Picture Processor, [9], [32], a commercial smart sensor from Integrated Vision Products, IVP, in Linköping. MAPP2200 contains a 256x256 photo diode array and a linear array of 256 A/D converters and PEs on a single chip.

A novel imaging and signal processing idea called Near-Sensor Image Processing, originally implemented in the LAPP architecture, then led to the evolution of the 2D NSIP architecture [10], [11], [30], [33]. In the NSIP architecture the sensor and processing element is combined into a compact Sensor-Processor Element, SPE, and the SPEs are arranged in a 2D architecture. The NSIP architecture is capable of general low-level image processing but a simplified version of the NSIP architecture called NESSLA [16], [23] was developed for sheet-of-light range imaging applications. This architecture, together with several sheet-of-light range imaging algorithm implementations on the MAPP2200 sensor [16], [17], [18], [22], [27], [28] are presented in **Part I** of this thesis. We also cover optical design, calibration and general signal processing aspects of sheet-of-light range imaging as well as comparisons with other smart sensor architectures aimed at this particular application. This is an reworked edition of my licentiate thesis [16] together with work on novel MAPP2200 algorithms [18], [27], [28] and calibration [19]. Three of the MAPP2200 algorithms and the calibration routines have been implemented in a sheet-of-light range imaging software package called RANGER commercially available from the MAPP2200 manufacturer IVP since mid 1994 [26]. The NSIP work in [16] have been updated to include experimental results from two general NSIP chips that have been designed and tested to verify the concept of NSIP as a good sheet-of-light range imaging architecture [7], [8].

During the work with PASIC, its processor part was discovered to be very powerful on its own. This resulted in a processor design called VIP, Video Image Processor [3], [6], [31]. VIP is a 1-Dimensional bit-serial SIMD architecture. The combination of many small but powerful Processing Elements, PEs, on a single chip gives a very effective row-parallel image processing computer for demanding low-level real-time applications.

FIGURE 1.



The PASIC-VIP family tree.

The VIP design also proved to be effective for Doppler radar signal processing [15], and **Part II** of this thesis describes the Radar-VIP, RVIP, architecture aimed at ground-based surveillance radar signal processing [1], [25] and its successor for airborne applications, the Flight-VIP, FVIP, [24], [35]. We also present ideas for the Matrix-VIP, MVIP, architecture aimed at future airborne space-time adaptive phased-array processing.

The RVIP chip has been implemented by Ericsson Microwave systems, EMW, (formerly Ericsson Radar Electronics) in Mölndal for use in ground based surveillance radar applications. Two other VIP architectures, the Wood-VIP (WVIP) [13] and the Infra-red-VIP (IVIP) [34], [21] have also been developed but are not presented in depth here. For more information on the IVIP and WVIP architectures we refer to [14]. The FVIP, WVIP and IVIP designs have been developed concurrently and are very similar. We also present implementations of Fast Fourier Transforms, FFTs [20], on VIP arrays since this is an important part of the Doppler radar signal processing.

Thesis overview

Part I describes sheet-of-light range imaging, and the goal is fast imaging. Using a standard video sensor gives us very slow imaging speed, and the work focuses on effective peak detection algorithms and their smart sensor implementations. After the introduction about different active range imaging techniques in *chapter 1,* I present a thorough investigation of the optical design of a sheet-of-light system in *chapter 2.* I derive equations mapping from sensor to world coordinates and describe how to apply the Scheimpflug condition so that the whole light-sheet can be viewed in perfect focus.

In *chapter 3* I present a novel sheet-of-light camera calibration scheme based on approximations of the triangulation equations derived in chapter 2.

Different suitable peak detection algorithms are studied in *chapter 4*, and I present simulations of the accuracy of the algorithms with and without signal degradation by noise.

An overview of different sensor designs aimed at sheet-of-light range imaging are given in *chapter 5.*

In *chapter 6* I present the novel NESSLA sensor architecture. NESSLA is a special purpose NSIP-type architecture with a 2D array of Sensor-Processor Elements and it is capable of very fast sheet-of-light range imaging. Experimental results obtained using a general 16x16 SPE NSIP chip are given to verify the NESSLA concept.

Chapter 7 presents implementations of different peak detection algorithms on the MAPP2200 smart sensor, and I describe how to achieve fast and accurate imaging. The algorithms in sections 7.10-7.13 were mostly devised by E. Åstrand.

I have implemented the presented calibration routine and the algorithms in sections 7.3-7.6 in the RANGER sheet-of-light range imaging software package available from the MAPP2200 manufacturer IVP.

Part I ends with *chapter 8* which is a short discussion of the presented smart sensor architectures.

Part II of this thesis describes the evolution of different VIP architectures for Doppler radar signal processing applications. Much of this work was performed in a large group of people from Ericsson Microwave systems, the VLSI group at Linköping University, and from the Image Processing Laboratory at Linköping University, where I work.

Chapter 1 gives an overview of the history of the VIP architecture and describes two similar linear array SIMD architectures.

Chapter 2 describes a few of the principles and the signal processing algorithms of pulsed Doppler radars.

Chapter 3 describes an LPD algorithm and how this algorithm led to the evolution of the RVIP architecture. The RVIP PE redesign mainly affected two parts of the PE, the IO-register and the Multiplier-Accumulator. I was involved to a high degree in the evolution of the Multiplier-Accumulator design, where I studied several different designs to find the best trade-off between high performance and small area. The combination of the parallel coefficient load port and the dual-ported accumulator increased the PE performance for multiply-accumulate operations three to four times compared with the original VIP PE design. We also present an overview of the system design with several chips and a local micro controller packaged together in a 512 PE system. For this part I worked on the micro controller design.

Chapter 4 then presents an MPD algorithm and how this led to the evolution of the FVIP architecture. There was very little architecture development in FVIP and my work was concentrated on the problem of mapping the 2D input image on a linear array architecture. A complication was that the image size varied continuous between three different aspect ratios, and that the architecture and algorithm mapping had to be able to handle all three formats.

I also studied how 1D FFTs, which are a major part of the MPD algorithm, could be implemented efficiently on FVIP, and this work is presented in *chapter 5*. One of the main problems with the FFT implementation was that the organization of the samples within the array made it impossible to neglect the bit-reverse sorting and butterfly-scheme communication problems.

In *chapter 6* I present ongoing work on how to map a space-time adaptive processing (STAP) algorithm on a new VIP architecture called MVIP. Here I have devised a possible mapping of the 3D data matrix and algorithm on an MVIP array. To solve the difficult communication problems the linear array has been augmented with limited 2D communication capacity. A new floating-point version of the MVIP PE is also presented.

Part II ends with *chapter 7* which compares the VIP performance with two other architectures, and describes possible future work within the VIP project. There is also a short list of abbreviations used in Part II.

Publications

The work presented in this thesis has previously appeared, in different form, in these publications

[i] Johannesson M., Åström A., Danielsson P.E., *An Image Sensor for Sheet-of-light Range Imaging*, Proc. of MVA '92, Tokyo, 1992.

[ii] Johannesson M., Åström A., *Sheet-of-light range imaging with MAPP2200*, Proc. of SCIA '93, Tromsö, 1993.

[iii] Johannesson M., Åström A., Ingelhag P., *The RVIP Image Processing Array*, Proc. of CAMP '93, New Orleans, 1993.

[iv] Johannesson M., *Fast, programmable, sheet-of-light range finder using MAPP2200*, Proc. of San Diego SPIE '94, San Diego, 1994.

[v] Åstrand E., Johannesson M., Åström A., *Two novel methods for range imaging using the MAPP2200 smart sensor*, Proc. of Machine Vision Applications in Industrial Inspection, SPIE '95, San José, 1995.

[vi] Åström A., Johannesson M., Edman A., Ehlersson T., Nässtrom U., Lyckegård B., *An Implementation Study of Airborne Medium PRF Doppler Radar Signal Processing on a Massively Parallel SIMD processor architecture*, Proc. of RADAR '95, Washington 1995.

[vii] Johannesson M., *Calibration of a MAPP2200 sheet-of-light range camera*, Proc of SCIA '95, Uppsala, 1995.

[viii] Johannesson M., *FFTs on a linear SIMD array*, Proc. of EURO-PAR'95, Lecture Notes on Computer Science 966, Springer Verlag, 1995.

Related, but not direct included, publications are

[ix] Johannesson M., *Can Sorting Using Sheet-of-Light Range Imaging and MAPP2200*, Proc. of IEEE/SMC '93, Le Touquet, 1993.

[x] Johannesson M., Gökstorp M., *Video-rate pyramid optical flow computations on the linear SIMD array IVIP*, Proc. of CAMP '95, Como, 1995.

References

[1] Arvidsson R., *Massively Parallel SIMD Processor for Search Radar Signal Processing*, Proc. of RADAR '94, Paris, 1994.

[2] Chen K., Danielsson P.E., Åström A., *PASIC a Sensor/Processor Array for Computer Vision*, Proc. of Application Specific Array Processors, Princeton, 1990.

[3] Chen K., Svensson C., *A 512-processor array chip for video/image processing*, Proc. of "From Pixels to Features II", Bonas, 1990.

[4] Chen K., Åström A., Danielsson P.E., *PASIC a Smart Sensor for Computer Vision*, Proc. of Pattern Recognition, Atlantic City, 1990.

[5] Danielsson P.E., *Smart Sensors for Computer Vision*, Report No. LiTH-ISY-I-1096, Linköping, 1990.

[6] Danielsson P.E., Emanuelsson P., Chen K., Ingelhag P., Svensson C., *Single-Chip High-Speed Computation of Optical Flow*, Proc. of MVA '90, Tokyo, 1990.

[7] Eklund J-E., Svensson C., Åström A., *Near-Sensor Image Processing, a VLSI Realization*, Proc of Transducers95, Eurosensors IX, Stockholm, 1995.

[8] Eklund J-E., Svensson C., Åström A., *Near-Sensor Image Processing, a VLSI Realization*, Proc of 8th Annual IEEE Int. ASIC Conference., Austin, 1995.

[9] Forchheimer R., Ingelhag P., Jansson C., *MAPP2200, a second generation smart optical sensor*, SPIE, vol 1659, 1992.

[10] Forchheimer R., Åström A., *Near-Sensor Image Processing. A new Approach to low level image processing*, Proc. of ICIP, Singapore, 1992.

[11] Forchheimer R., Åström A., *Near-Sensor Image Processing. A New Paradigm*, In IEEE trans. on Image Processing, november 1994.

[12] Forchheimer R., Ödmark A., *Single chip linear array processor*, in Applications of digital image processing, SPIE, vol 397, 1983.

[13] Hall M., Åström A., *High Speed wood inspection using a parallel VLSI architecture*, Proc. of 3rd Int. Workshop on Algorithms and Parallel VLSI Architectures, Leuven, 1994.

[14] Hall M., *A study of Parallel Image Processing on the VIP Architecture*, Lic Thesis, LiTH-ISY-LIC-1995:41, Linköping, september 1995.

[15] Ingelhag P., Åström A., Chen K., Svensson C., Ehlersson T., *RADAR Signal Processing Using a 512-Processor Array Chip*, Proc. of ICDP91, Florence, 1991.

[16] Johannesson M., *Sheet-of-light Range Imaging*, Lic thesis No 404, Linköping, 1993.

[17] Johannesson M., *Can Sorting Using Sheet-of-Light Range Imaging and MAPP2200*, Proc. of IEEE/SMC '93, Le Touquet, 1993.

[18] Johannesson M., *Fast, programmable, sheet-of-light range finder using MAPP2200*, Proc. of San Diego SPIE '94, San Diego, 1994.

[19] Johannesson M., *Calibration of a MAPP2200 sheet-of-light range camera*, Proc of SCIA '95, Uppsala, 1995.

[20] Johannesson M., *FFTs on a linear SIMD array*, Proc of EURO-PAR'95, LNCS 966, Springer Verlag, 1995.

[21] Johannesson M., Gökstorp M., *Video-rate pyramid optical flow computations on the linear SIMD*

array IVIP, To appear in Proc of CAMP '95, Como, 1995.

[22] Johannesson M., Åström A., *Sheet-of-light range imaging with MAPP2200*, Proc. of SCIA '93, Tromsö, 1993.

[23] Johannesson M., Åström A., Danielsson P.E., *An Image Sensor for Sheet-of-light Range Imaging*, Proc. of MVA '92, Tokyo, 1992.

[24] Johannesson M., Åström A., Edman A., *FVIP, An implementation study of an MPD radar algorithm*, Report LiTH-ISY-R-1698, Linköping, 1994.

[25] Johannesson M., Åström A., Ingelhag P., *The RVIP Image Processing Array*, Proc. of CAMP '93, New Orleans, 1993.

[26] *RANGER RS2200 Product Information*, IVP, Linköping, 1994.

[27] Åstrand E., Johannesson M., Åström A., *Five contributions to the art of sheet-of-light range imaging*, Report No. LiTH-R-ISY-1611, Linköping, 1994.

[28] Åstrand E., Johannesson M., Åström A., *Two novel methods for range imaging using the MAPP2200 smart sensor*, Proc. of Machine Vision Applications in Industrial Inspection, SPIE '95, San José, 1995.

[29] Åström A., *A Smart Image Sensor. Description and Evaluation of PASIC.*, Lic. Thesis. LIU-TEK-LIC-1990:57, Linköping 1990.

[30] Åström A, *Smart Image Sensors*, Ph.D. thesis No. 319, Linköping, 1993.

[31] Åström A., Danielsson P.E., Chen K., Ingelhag P., Svensson C., *Videorate signal processing with PASIC and VIP*, Proc. of 2nd Specialist Sem. on design and Appl. of Parallel Digital Processors, Lissabon, 1991.

[32] Åström A., Forchheimer R., *MAPP2200 Smart Vision Sensor: Programmability and Adaptivity*, Proc. of MVA '92, Tokyo, 1992.

[33] Åström A., Forchheimer R., Ingelhag P., *An Integrated Sensor/Processor Architecture Based on Near-Sensor Image Processing*, Proc. of CAMP '93, New Orleans, 1993.

[34] Åström A., Isaksson F., *Real-time Geometric Distortion Correction and Image Processing on the 1D SIMD architecture IVIP*, Proc of MVA '94, Kawasaki 1994.

[35] Åström A., Johannesson M., Edman A., Ehlersson T., Näsström U., Lyckegård B., *An Implementation Study of Airborne Medium PRF Doppler Radar Signal Processing on a Massively Parallel SIMD processor architecture*, Proc of RADAR '95, Washington 1995.

Part I

Sheet-of-light Range Imaging

Chapter 1

Introduction

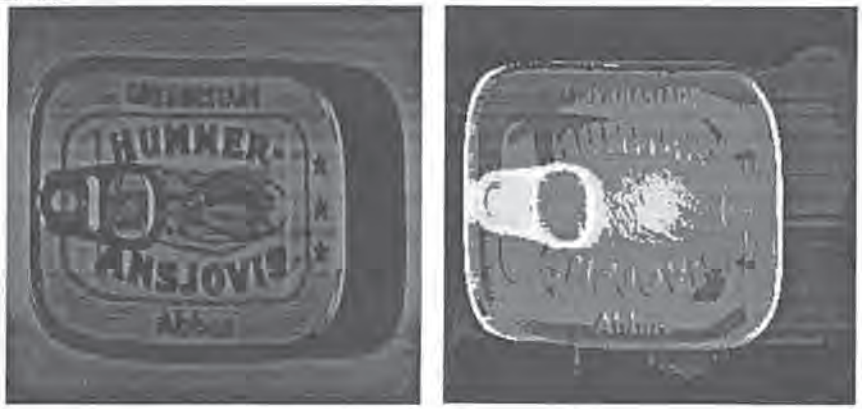
1.1 Range Imaging

Normally, the image from a camera depicts the intensity distribution of the viewed scene as in the left image in Figure 1.1. A range camera on the other hand, depicts the distance from the camera to the objects of the scene as in the right image in Figure 1.1. In this range image the printed text is no longer visible, while the can opener and edges show up because of their difference in height. Interestingly, range and intensity images may also be combined as in Figure 1.2.

Range images are used in many applications, e.g. road surface surveying [9], industrial inspection [4], [50], [45], [65], and industrial robots [63]. In road surface surveys the goal is to measure cracks and ruts in the road surface. In industrial inspection the goals are often to determine if fabricated parts are within the tolerated sizes and/or orientations. In industrial robot applications we need to find locations and sizes of objects for navigation and/or object manipulation purposes.

Range images are obtained in many different ways. Traditionally these are separated into two categories, active and passive range imaging respectively. In active range imaging a dedicated and well defined light source is used in cooperation with the sensor. In passive range imaging no special light (except ambient) is required. The most common passive method is stereo vision where at least two images from different positions are used in a triangulation scheme. More detailed descriptions of different passive range imaging methods can be found in for instance [24], [1] and [22].

FIGURE 1.1.



An intensity and a range image of a fish can. Dark areas are far away from the camera, and light areas close.

FIGURE 1.2.

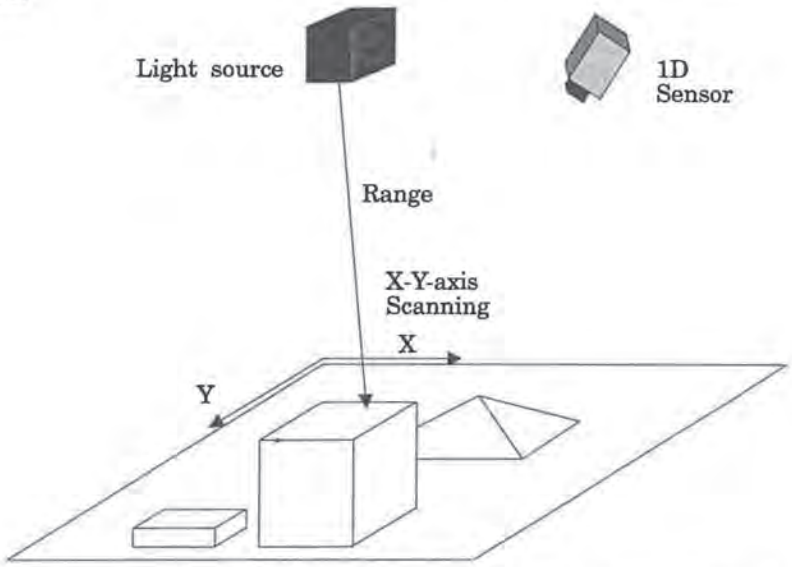
A combined range and intensity image.

1.2 Active range imaging

Many active range imaging techniques use a *triangulation* scheme where the scene is illuminated from one direction and viewed from another. The illumination angle, the viewing angle, and the baseline between the illuminator and the viewer (sensor) are the triangulation parameters. The most common active triangulation methods include illumination with a *single spot, a sheet-of-light* and *gray coded light.* A thorough discussion on the different methodologies can be found in for instance [64], [6] and [47]. The most common non-triangulation method is time-of-flight (radar), where the time for the emitted light pulse to return from the scene is measured. The high speed of electromagnetic waves makes time-of-flight methods difficult to use for high accuracy range imaging since small differences in range have to be resolved by extremely fine discrimination in time.

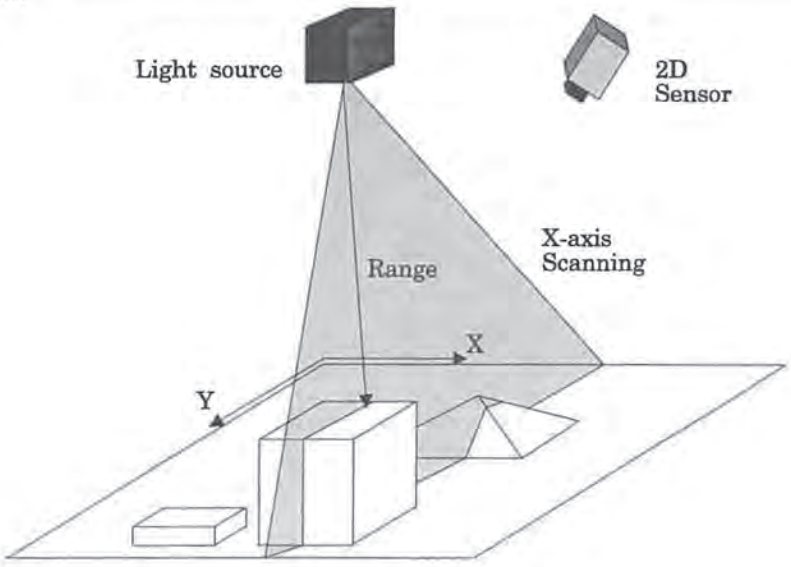
In single spot range imaging a single light ray is scanned over the scene and one range datum (rangel) is acquired for each sensor integration and position of the light, see Figure 1.3. Thus, in order to obtain an *MxN* image *MxN* measurements and sensor integrations have to be made. In sheet-of-light range imaging a sheet (or strip) of light is scanned over the scene and one row with *M* rangels is acquired at each light position and sensor integration, see Figure 1.4. In this case only *N* measurements and integrations are needed for an *MxN* image. Finally in a gray-coded range camera the scene is illuminated with *logN* gray coded patterns and therefore only *logN* integrations are needed to form one *MxN* image, see Figure 1.5.

FIGURE 1.3.



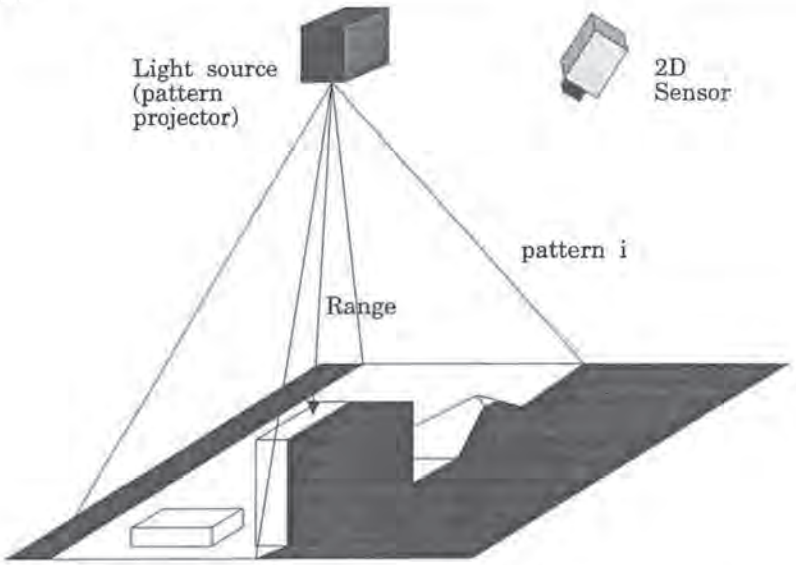
Single spot range imaging. Note that the range can also be defined as the distance between the sensor and the illuminated object instead of the distance between the light source and the object.

FIGURE 1.4.



Sheet-of-light range imaging.

FIGURE 1.5.



Range imaging with gray coded light.

It would seem that the gray coded light method has speed advantages over the other active triangulation methods since fewer sensor integrations are needed. However, one problem seems to be how to make a high intensity projector that can switch between patterns as fast as the sensor can integrate images. Hence, the trouble spot is the design of the illuminator rather than the sensor. Furthermore, any object movements between the integrations can give errors in the range image since the triangulation is performed *after* the *logN* integrations under the assumption that the illuminated scene was static. Admittingly, the errors are minimized and not catastrophic due to the properties of the Gray code. In the other methods, however, each single sensor integration gives one set of range values. Therefore any object movement between integrations only gives a time dependent range image, where each range datum was correct when it was acquired.

In gray-coded light systems ordinary tungsten or halogen lamps are predominant instead of the laser light predominant in the other methods. We note that tungsten and halogen lamps are less harmful for the eyes than laser light and therefore better suited for measurements of human body parts.

The single spot technique requires advanced mechanics to allow the spot to reach the whole scene. At least two synchronous scanning mirrors are required. The advantage is that a relatively simple linear sensor, e.g. of the Position Sensitive Device type, can be used. But, the use of only one single sensing element often reduces the acquisition speed; the system presented by Rioux [49] reaches a rangel (range pixel) speed of 10 KHz. However, a recent system by the same research group actually outputs range data at RS-170 compatible video rate [5].

In the case of sheet-of-light systems, the projection of the light can be made with one single scanning mirror which is considerably simpler than the projector design for gray coded light, or the two mirror arrangement for single spot. Actually, in the sheet-of-light systems presented

in this thesis the sheet-of-light is not swept at all. Instead the apparatus itself or the scene is moving. Thus, the camera system itself has no moving parts. In principle, however, the sensor designs to be presented are just as applicable to sweeping sheet-of-light.

Finally, it should be noted that a moving scene or moving apparatus rule out the gray-coded methods since they require a static scene during the *logN* integrations.

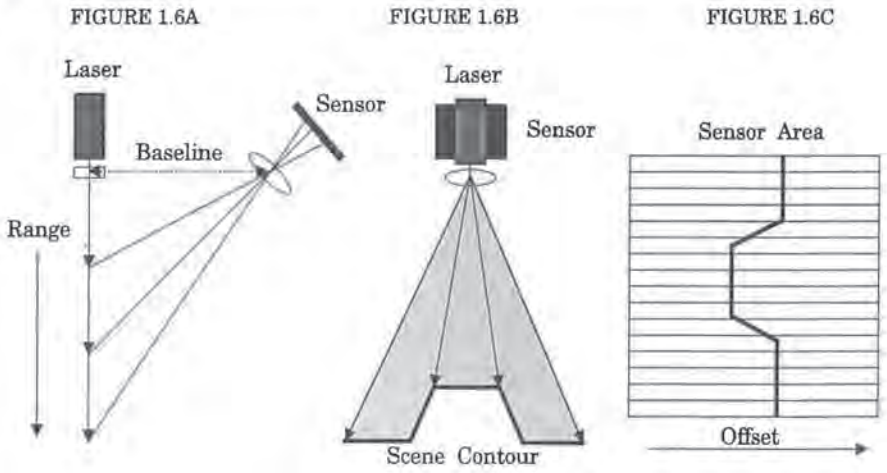
1.3 Sheet-of-light range imaging

In a sheet-of-light system range data is acquired with triangulation. The offset position of the reflected light on the sensor plane depends on the distance (range) from the light source to the object, see Figure 1.6. Using trigonometry we can solve the equations for the range if we know the distance between the laser and the optical center of the sensor (the *baseline*) and the direction of the transmitted ray. The third triangulation parameter, the direction of the incoming light ray, is given by the sensor offset position.

For each position of the sheet-of-light the depth variation of the scene creates a contour which is projected onto the sensor plane, see Figure 1.6B-C. If we extract the position of the incoming light for each sensor row we obtain an offset data vector that serves as input for the triangulations.

To make a sheet-of-light the pencil sharp laser spot-light passes through a cylindrical lens, see Figure 1.6A-B. The cylindrical lens spreads the light into a sheet in one dimension while it is unaffected in the other. A sheet-of-light can also be made using a fast scanning mirror mechanism [4], an array of LED's [45], or with a slit projector [45].

FIGURE 1.6.



A range camera using sheet-of-light. Figure 1.6A shows the illumination with the sheet-of-light perpendicular, Figure 1.6B parallel to the plane of the paper. Figure 1.6C is a snapshot of the sensor area.

As mentioned, two-dimensional range images can be obtained in at least three ways: by moving the apparatus over a static scene as in a road surface survey vehicle, by moving the scene as a conveyor belt, or by sweeping the sheet-of-light over a static 3D-scene using a mirror arrangement. In the two first cases the distance can be computed from the offset position in

each row using a simple range equation or a precomputed lookup table. The third case is more awkward since the direction of the emitted sheet-of-light changes so that a second triangulation parameter becomes involved for each light sheet position.

In any case, for each illumination position and sensor integration, in the first processing step, the output from the 2D image sensor should be reduced to a 1D array of offset values.

Many commercial sheet-of-light range imaging systems are based on video-standard sensors. Such a sensor has a preset frame-rate, which in the PAL standard is specified as 50 Hz. This limits the range profile frequency to 50 Hz assuming that the processing can be made in real-time. A 256x256 range image is then obtained in 5 seconds, and the range pixel frequency is 12.8 KHz. Even if the frame rate could be increased the sensor output is still serial which would require very high output and processing clock frequencies and possibly increase the noise level.

In [6] Besl reports on a few commercial sheet-of-light range imaging systems with CCD video-type sensors, but none of these have higher rangel frequencies than 5 KHz. As will be shown below, combining sensing and processing on the same chip implies that shorter integration times and considerably faster range imaging can be obtained.

1.4 Smart sensors

Part I of this thesis deals mainly with smart optical sensors and some initial comments on the concept seem appropriate. A smart sensor is a design in which data reduction and/or processing is performed on the sensor chip. Sheet-of-light range imaging is well suited for smart sensor architectures since each row of sensor data shall be reduced to one offset position value for the incoming light sheet.

In Linköping the first smart sensor was LAPP1100 [19], conceived in the early eighties. LAPP1100 combines a linear photo diode array with a linear array of bit-serial processing elements. It was followed by the PASIC design [68] in the late eighties. PASIC was a general programmable smart sensor, consisting of a 2D sensor array combined with a 1D array of A/D converters and processing elements. No PASIC chip was built, but the ideas from this design, combined with the LAPP1100 architecture became the MAPP2200 sensor [40], [17] now being manufactured by IVP, Integrated Vision Products, in Linköping.

Chapter 2

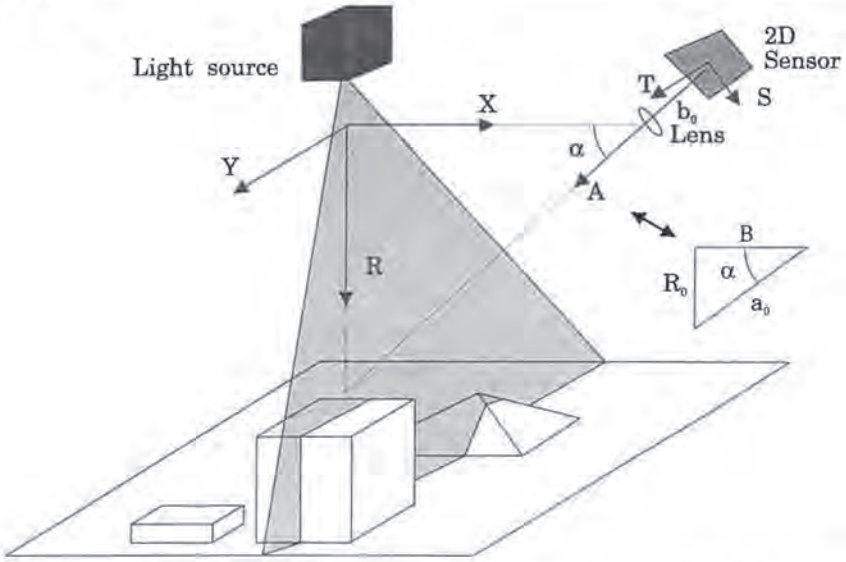
Optical design for sheet-of-light

2.1 Geometry definitions

In Figure 2.1 we show the coordinate system to be used in the sequel. The world coordinate system is (X,Y,R) and the sensor coordinate system is (S,T,A). The transformation between the coordinate systems can be described by two translations and one rotation (Translation B along the X-axis, rotation 90-α around the Y-axis and translation -b_0 along the A-axis). As will be shown later, there might also be a skew of the (S,T,A) system so that A is non-orthogonal to the (S,T) plane.

Since the Y and T axes are parallel, range measurements along the R-axis are independent of the sensor t-coordinate. This reduces all geometric range discussions to 2D, and we can view the system as in Figure 2.2 (for geometry definitions see Table 2.1). Here we only consider a non-scanning light-sheet and therefore the illumination angle γ is set to 90° unless otherwise is stated. This simplifies the range expressions somewhat. Also, the laser axis is considered vertical unless stated otherwise. A discussion on different scanning techniques can be found in [31].

FIGURE 2.1.



A Sheet-of-light system with world and sensor coordinate system definitions.

TABLE 2.1.

Word	Short	Definition
Optical center	OC	Position of the center of the sensor system lens.
Optical axis	OA	The *A*-axis, passing through the optical center.
Laser axis	LA	Optical axis of the laser system (the *R*-axis).
Laser origin	LO	Position along the laser axis closest to the optical center. Origin of world coordinate system.
Baseline	B	Distance between the optical center and the laser origin.
View angle	α	Angle between the optical axis and the baseline. Also the angle between the laser sheet and the normal of the optical axis if $\gamma = 90°$.
Illumination angle	γ	Angle between the baseline and the laser sheet. 90°unless deflected by a mirror.
Sensor angle	β	Angle between the normal of the optical axis and the sensor *s* axis.
Sensor position	s,t	Sensor coordinates along rows and columns. Origin at optical axis
Focal length	f	Focal length of the lens.
Lens aperture	d	Effective diameter of the lens.
Sensor distance	b_0	Distance along the optical axis between the optical center and the sensor.
Focus distance	a_0	Distance along the optical axis between the optical center and the sheet-of-light.
Range	r(s)	Range (distance), parallel with the range axis, from LO to a point in the laser sheet from where a ray emanates and hits the detector at *s*. $r(0) = R_0$ is the range of ray going in the optical axis.
Sensor height	h_0	Height between the baseline-axis and the sensor center column position (s = 0)
Sensor baseline	b_{10}	The base in the triangle made by the optical axis, the sensor height and the baseline-axis.
Pixel pitch	Δx	Inter-pixel distance on the sensor.
Sensor row size	N	The number of pixels in a sensor row.
Light angle	ρ	Angle between a light ray and the optical axis
Max angle	ρ_m	Maximum angular deviation from the optical axis.
Range interval	R_T	Total range covered by light rays offset with $\|\rho\| < \rho_m$ from the optical axis.

FIGURE 2.2.

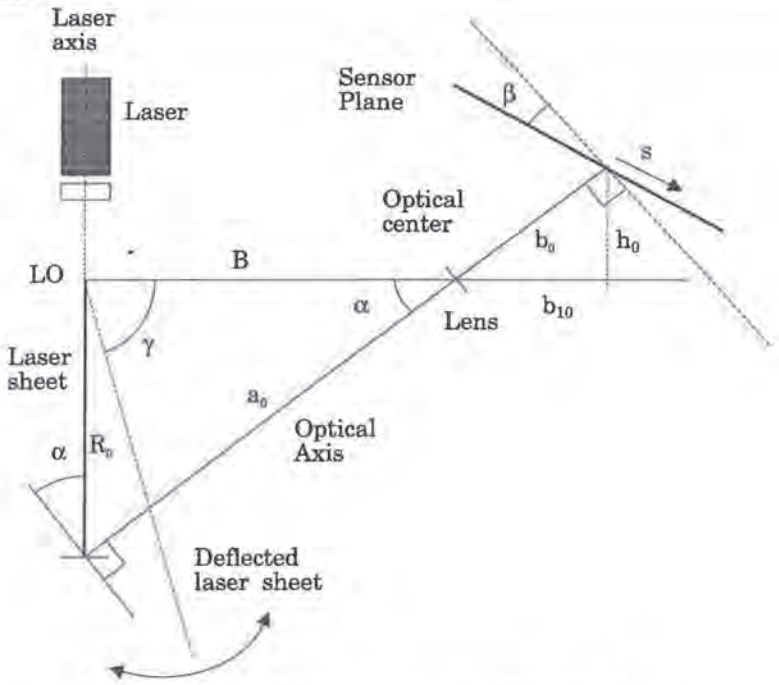


Illustration to the definitions made in Table 2.1.

Numerical examples are used in the sequel to give a better intuitive understanding of the problems and the derived equations. We consider two systems with equal view angles and approximately equal range interval but with different optics. One system utilize a "normal" lens with $f = 18$ mm, the other one a "zoom" lens with $f = 75$ mm. We also consider two apertures for each lens. The light sensitivity of a lens is expressed by the f-stop (relative aperture) f/d. The used f-stops are 5.4 and 1.8 which give the lens apertures 3.3 and 10 mm for the 18 mm lens and 14 and 42 mm for the 75 mm lens. Typical parameters are

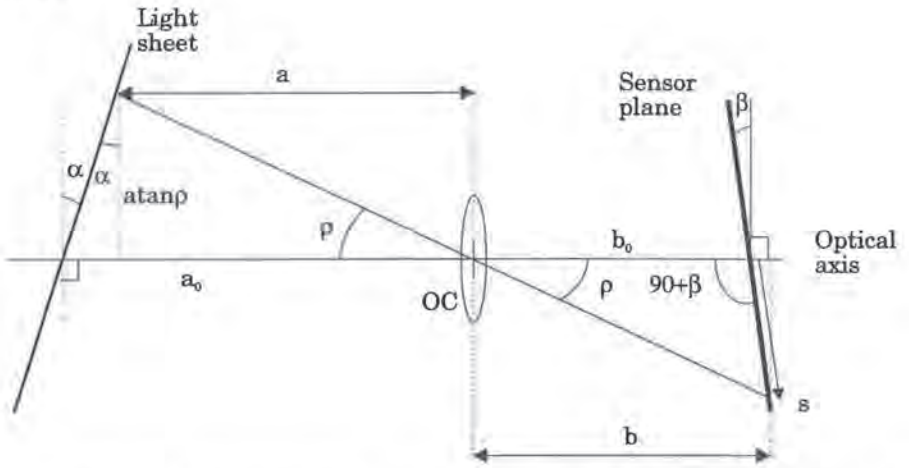
View angle	α	45°
Range interval	R_T	500 mm
Sensor row size	N	256
Pixel pitch	Δx	32 μm
Focal length	f_1	18 mm
Lens aperture	d_A	3.33 mm
Lens aperture	d_B	10.0 mm
Base line	B_1	520 mm
Focal length	f_2	75 mm
Lens aperture	d_A	13.9 mm
Lens aperture	d_B	41.67 mm
Base line	B_2	2282 mm

2.2 The Scheimpflug condition

In Figure 2.3 we see the sheet-of-light geometry for arbitrary laser and sensor angles, and for an arbitrary incoming light ray. The relation between the focal length and the sensor and focus distances is dictated by the lens law

$$\frac{1}{f} = \frac{1}{a_0} + \frac{1}{b_0} \tag{2.1}$$

FIGURE 2.3.



A setup with arbitrary sensor and laser angles and an incoming light ray at an angle ρ.

For an arbitrary light ray making an angle ρ with the optical axis and hitting the sensor at s, we can find the corresponding orthogonal distances a and b from the optical center to the laser sheet and the sensor. The equations are

$$a = \frac{a_0}{(1 + \tan\rho\tan\alpha)} \tag{2.2}$$

$$b = \frac{b_0}{(1 - \tan\rho\tan\beta)} \tag{2.3}$$

If we want the ray going to s to be focused on the sensor plane, a and b must satisfy the lens law (2.1). From (2.2) and (2.3) we get

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{a_0}(1 + \tan\rho\tan\alpha) + \frac{1}{b_0}(1 - \tan\rho\tan\beta) \tag{2.4}$$

$$= \frac{1}{f} + \tan\rho\left(\frac{\tan\alpha}{a_0} - \frac{\tan\beta}{b_0}\right) \tag{2.5}$$

We see that the lens law is satisfied for arbitrary angles ρ if

$$\frac{\tan\alpha}{a_0} = \frac{\tan\beta}{b_0} \tag{2.6}$$

This is satisfied in the special case

$$\alpha = \beta = 0 \tag{2.7}$$

and in general for

$$\tan\beta = \frac{b_0 \tan\alpha}{a_0} \tag{2.8}$$

The condition in (2.7) implies that the sensor is orthogonal to the optical axis and that the optical axis is orthogonal to the laser axis. The condition in (2.8) is known as the *Scheimpflug condition* [6], and defines the tilt of the sensor plane to achieve focus over the whole sensor plane on the sheet-of-light when the optical axis is not orthogonal to either plane. In normal cameras the sensor angle is fixed and orthogonal to the optical axis, which means that only the special condition in (2.7) can be satisfied.

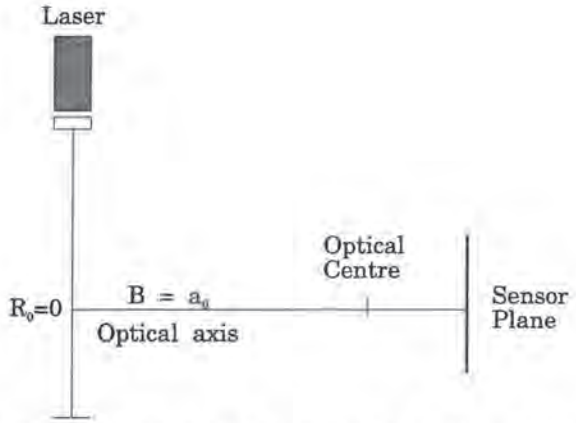
If we have a setup which satisfies the Scheimpflug condition in (2.8), how large is the angle β? Numerical examples are given in Table 2.2. As expected we find that β is very small for view angles below say 70°. Notice also that the angle is almost independent of the lens parameters for a given range interval and view angle.

TABLE 2.2.

f	b_0	a	b
18 mm	18.4 mm	45°	1.43°
75 mm	76.8 mm	45°	1.36°
18 mm	18.4 mm	63°	2.81°
75 mm	76.8 mm	63°	2.67°
18 mm	18.4 mm	85°	15.98°
75 mm	76.8 mm	85°	15.21°

As said before, in normal cameras the sensor plane is orthogonal to the optical axis and if we focus the camera on the light plane as in Figure 2.4 the focus condition in (2.7) is satisfied. Unfortunately this arrangement is not very attractive. If we assume that the objects in the scene are planar and parallel with the optical axis, then when an object is higher than the range R_0 the laser light will by all likelihood be occluded, and no light will reach the sensor.

FIGURE 2.4.

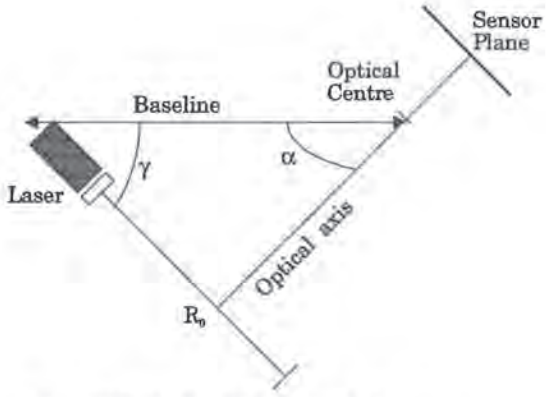


A system with the sensor aligned with the optical axis orthogonal to the light plane.

However, there might be applications where the surface orientations of the objects have little variation. The system in Figure 2.4 should then be arranged so that the object surface normal approximately bisects the 90° angle between the laser sheet and the optical axis. For a case where the 3D scene moves horizontally the setup should be tilted 45° as shown in Figure 2.5.

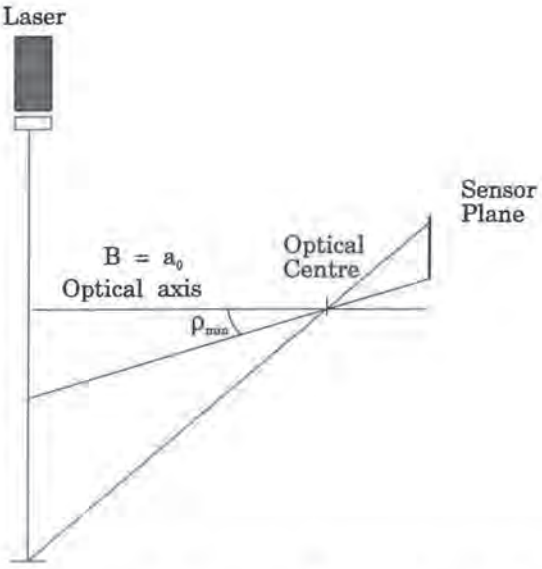
If we use the setup in Figure 2.4 but move the sensor so that the whole sensor is above the optical axis we still satisfy (2.7). We then get the setup in Figure 2.6. However, normal lenses attenuate the intensity across the image plane with the factor $\cos^4\rho$ [24]. Therefore, in Figure 2.6 where we utilize a shifted interval of the sensor plane off the optical axis, there is a price to pay in the form of uneven sensitivity along the sensor plane. Also, lenses are often less than ideal for light rays far away from the optical axis, which results in distorted images.

FIGURE 2.5.



The system in Figure 2.4 tilted 45°.

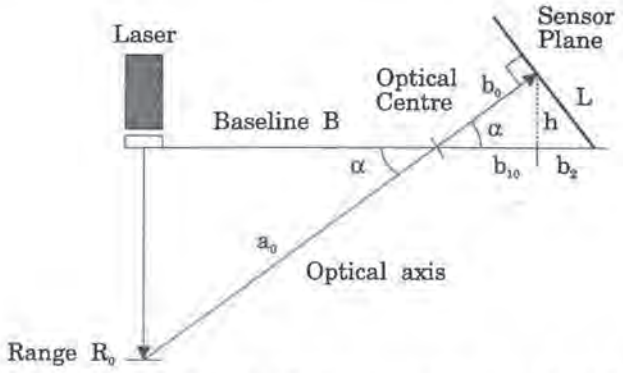
FIGURE 2.6.



A system with the sensor offset from the optical axis.

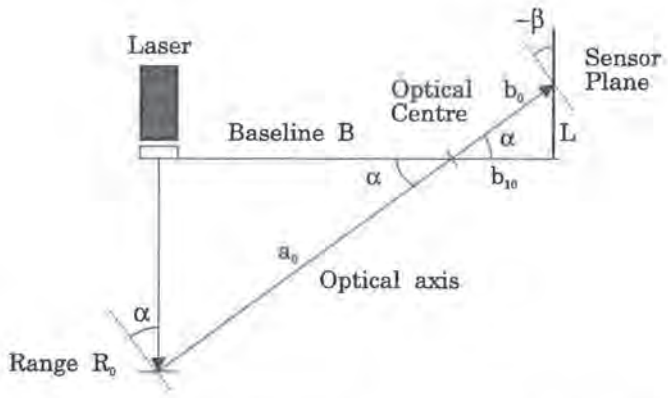
The most common arrangement for sheet-of-light systems is to tilt the camera and its optical axis so that the view angle α is substantially larger than zero, typically in the range of 30-60°, see Figure 2.7. This reduces the problem of occlusion, but the conditions in (2.7) or (2.8) are not satisfied. Furthermore, the linearity found between range r and sensor position s in the previous cases is lost since linearity is only found when the sensor and laser planes are parallel. To regain linearity we can tilt the sensor as in Figure 2.8. Here, the sensor and view angles α and β are equal but with opposite signs. Unfortunately this setup dissatisfies the conditions in (2.7) and (2.8) even more, and therefore gives more unfocused (unsharp) images. If we want a focused design we should use a setup as in Figure 2.7 but instead tilt the sensor according to the Scheimpflug condition (2.8).

FIGURE 2.7.



A system with the sensor aligned with the view-angle approximately 40°.

FIGURE 2.8.



A setup for linearity between the sensor position and the range data.

2.3 Equations for range

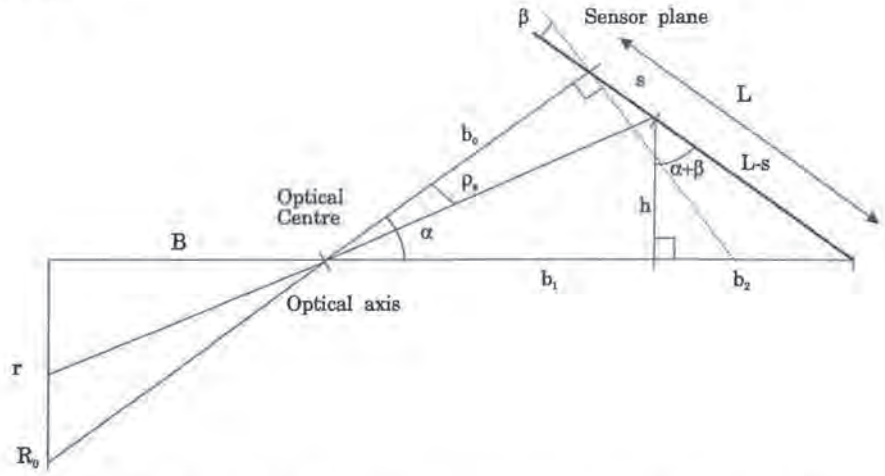
From Figure 2.2 we have

$$R_0 = \frac{Bh_0}{b_{10}} \tag{2.9}$$

For an arbitrary light ray, offset from the optical axis with an angle ρ_s and hitting the sensor at s, we get the geometry in Figure 2.9 where

$$r = \frac{Bh}{b_1} \tag{2.10}$$

FIGURE 2.9.



Geometry for an arbitrary sensor alignment, and an arbitrary input ray.

The geometry for the triangles give

$$b_1 + b_2 = \frac{b_0 \sin(90+\beta)}{\sin(90-(\alpha+\beta))} = \frac{b_0 \cos\beta}{\cos(\alpha+\beta)} \tag{2.11}$$

$$L = \frac{b_0 \sin\alpha}{\sin(90-(\alpha+\beta))} = \frac{b_0 \sin\alpha}{\cos(\alpha+\beta)} \tag{2.12}$$

$$h = (L-s)\cos(\alpha+\beta) \tag{2.13}$$

$$b_2 = (L-s)\sin(\alpha+\beta) \tag{2.14}$$

Using (2.11) - (2.14) in (2.10) yields

$$r = \frac{B\left(\frac{b_0 \sin\alpha}{\cos(\alpha+\beta)} - s\right)\cos(\alpha+\beta)}{\frac{b_0 \cos\beta}{\cos(\alpha+\beta)} - \left(\frac{b_0 \sin\alpha}{\cos(\alpha+\beta)} - s\right)\sin(\alpha+\beta)} \tag{2.15}$$

This rather cumbersome equation holds for all sensor and laser alignments including the Scheimpflug geometry. For the special geometry in Figure 2.4 we have $\alpha = \beta = 0$, which simplifies (2.15) to

$$r = \frac{-Bs}{b_0} \tag{2.16}$$

For the geometry in Figure 2.7 we have $\beta = 0$, which reduces (2.15) to

$$r = \frac{B(b_0 \tan\alpha - s)\cos\alpha}{\frac{b_0}{\cos\alpha} - (b_0 \tan\alpha - s)\sin\alpha} = B\frac{b_0 \tan\alpha - s}{b_0 + s\tan\alpha} \tag{2.17}$$

For the geometry in Figure 2.8 we have $\alpha = -\beta$ which gives the range equation

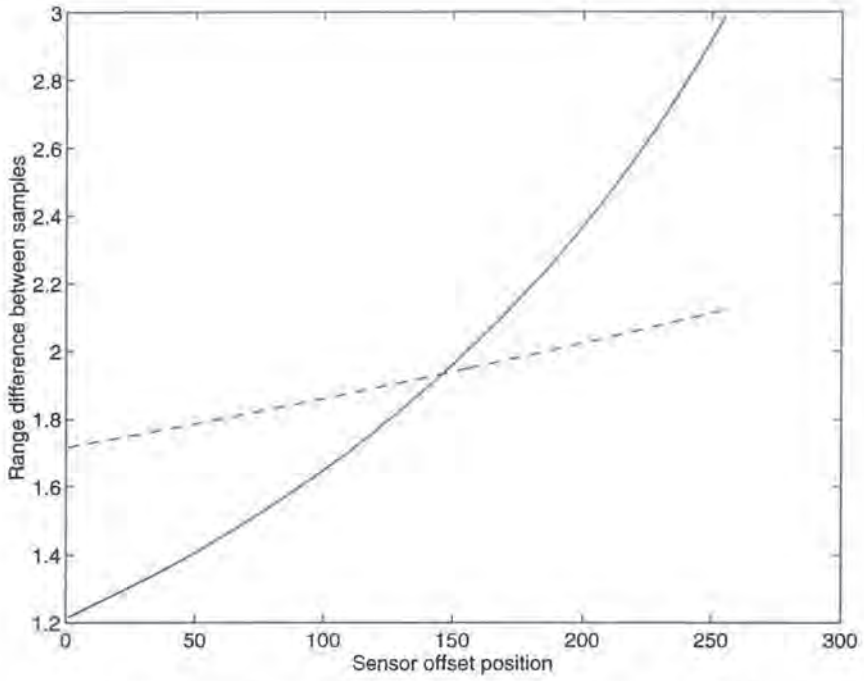
$$r = \frac{B(b_0 \sin\alpha - s)}{b_0 \cos\alpha} \tag{2.18}$$

As mentioned above the most common setup is the one in Figure 2.7, with the non-linear range equation (2.17). The non-linearity means that a constant precision Δs in the sensor position s results in a variable precision Δr

$$\Delta r = \Delta s \frac{-Bb_0}{(b_0 \cos\alpha + s\sin\alpha)^2} \tag{2.19}$$

The system should be designed with this in mind. The function $|\Delta r|$ is plotted in Figure 2.10 for the two cases $f = 75$ mm and $f = 18$ mm and the range interval ~350 mm. Here we see that the difference in Δr is much larger for the smaller f value. This is an indication that a larger f (a zoom lens) gives better linearity.

FIGURE 2.10.



Plot of range increments $|\Delta r|$ as a function of the sensor offset using two different lenses f_1 =18 (solid) and f_2 = 75 (dotted). s = 0 is sensor offset position 128.

To find the range interval R_T for a given setup we insert the extreme values of s (+/- $N\Delta x/2$) into equation (2.17) and after simplification we obtain

$$R_T = \frac{4Bb_0N\Delta x\,(1 + (\tan\alpha)^2)}{4b^2{}_0 - (N\Delta x\tan\alpha)} \approx \frac{4BfN\Delta x\,(1 + (\tan\alpha)^2)}{4f^2 - (N\Delta x\tan\alpha)} \tag{2.20}$$

This equation can be useful if we want to determine the range interval of a given system, or for instance if we want to redesign a system keeping the same range interval but using a different focal length or view angle.

2.4 Equations for width

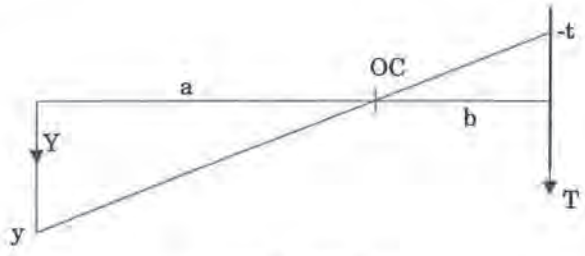
In Figure 2.11 we see the correspondence between sensor coordinate t and world coordinate y which can be expressed as

$$y = \frac{a \cdot t}{b} \tag{2.21}$$

where a and b as functions of ρ are defined in (2.2) and (2.3) and as shown in Figure 2.12 tanρ is found using

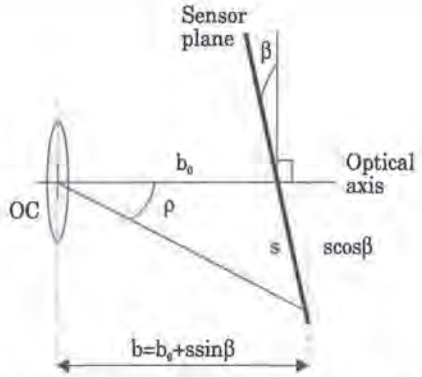
$$\tan\rho = \frac{s\cos\beta}{b_0 + s\sin\beta} \tag{2.22}$$

FIGURE 2.11.



The width geometry.

FIGURE 2.12.



The focus distance as a function of the incoming light angle ρ

If we combine these equations we obtain

$$y = \frac{-t \cdot B}{\cos\alpha\,(b_0 + s\sin\beta)\left(1 + \dfrac{s\cos\beta\tan\alpha}{b_0 + s\sin\beta}\right)} \tag{2.23}$$

If β=0 (2.23) can be simplified to

$$y = \frac{-t \cdot B}{b_0\cos\alpha\left(1 + \dfrac{s\tan\alpha}{b_0}\right)} = \frac{-t \cdot B}{b_0\cos\alpha + s\sin\alpha} \tag{2.24}$$

From (2.23) and (2.24) it is evident that y decreases for increasing s and constant t.

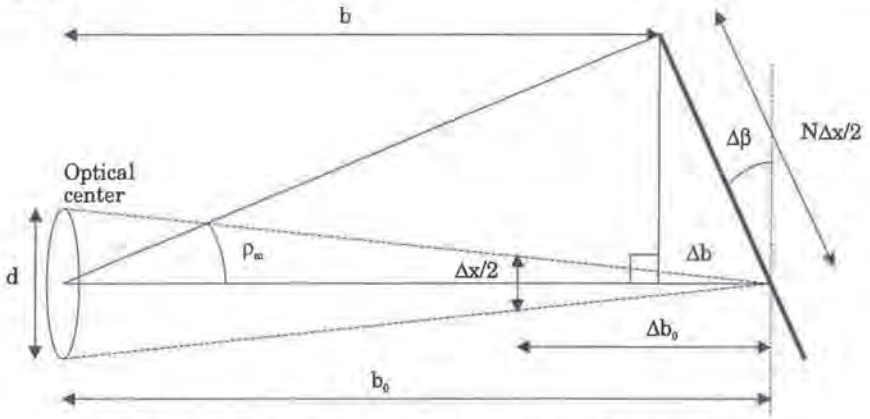
As we have shown the range r and width y can be determined from the sensor offset position (s,t) if we know the system parameters. However, in practice these range equations are not often used to obtain the range in a sheet-of-light range imaging system. Instead the range values (r,y) and their corresponding offset values (s,t) are registered in a calibration procedure, and stored in a look-up table which later can be addressed with the sensor offset positions. The advantage of the calibration solution is that it may automatically compensate for lens and sensor aberrations [55] which are not accounted for in the formulas above. Sheet-of-light range camera calibration is discussed in chapter 3. Nevertheless, the formulas (2.15) - (2.24) should give a solid qualitative understanding of the design problems for sheet-of-light based ranging systems.

2.5 Depth of focus

The lens parameters affect the depth of focus for the system. To correctly design a system we must know the tolerances for the sensor and sheet-of-light alignment. Specifically, we want to know how much the sensor can deviate from the Scheimpflug condition (2.8) without loosing much focus. To this aim we define the sensor to be in focus when the focusing ray-cone has a cross section width which is less than 1/2 pixel pitch, see Figure 2.13.

To be in focus, the sensor must be aligned so that the maximum deviation Δb from the correct distance b_0 is smaller than the depth of focus distance Δb_0. For simplicity, in these computations we only consider the case β=0 but the results should be valid for most β within the narrow range given in the first four rows of Table 2.2 above.

FIGURE 2.13.



Depth of focus geometry. The broad line depicts the sensor plane.

The depth of focus distance Δb_0,where the ray-cone has a cross section diameter which is half the pixel pitch Δx, is

$$\Delta b_0 = \frac{b_0 \Delta x}{2d} \tag{2.25}$$

The length Δb as a function of the sensor mis-alignment $\Delta\beta$ is

$$\Delta b = \frac{N\Delta x \sin\Delta\beta}{2} \tag{2.26}$$

and it denotes how much the distance b deviates from b_0 at the edge of the sensor. If this change in b is smaller than Δb_0 according to (2.25) the sensor is in focus, which gives a condition for the maximum deviation angle $\Delta\beta_m$

$$\sin\Delta\beta_m = \frac{b_0}{Nd} \approx \frac{f/d}{N} \tag{2.27}$$

The only parameters that determines $\Delta\beta_m$ is the sensor row size N, the lens diameter d and the focus distance b_0 (the sensor row size N and the f-stop f/d). Numerical examples are given in Table 2.3. We notice that the sensor angular alignment is rather critical, especially for small f-stops f/d, but that for f/d = 5.4 $\Delta\beta_m$ is almost as large as β.We can conclude the depth of focus discussion by noting the following. If we have high laser intensity (good signal to noise ratio) we can use a large f-stop, and then we can possibly achieve sharp (focused) images without

using Scheimpflug sensor geometry. If, on the other hand, we are forced to use a small f-stop or a large view angle we should definitely align the sensor according to the Scheimpflug condition to achieve sharp range images.

TABLE 2.3.

f/d	$\Delta\beta_m$	Δb_0
1.8	0.41	0.03
5.4	1.24	0.09

2.6 Accuracy limitations

The limited resolution of the sensor limits the possible ranging accuracy. The range uncertainty Δr as a function of the sensor uncertainty Δx (= Δs if no sub-pixel accuracy is used) and the sensor position s is obtained as follows, see also Figure 2.14.

$$s = b_0 \tan\rho \tag{2.28}$$

$$\Delta\rho_s = \frac{\Delta x (\cos\rho)^2}{b_0} = \frac{\Delta x b_0}{b_0^2 + s^2} \tag{2.29}$$

$$r = B\tan(\alpha + \rho) \tag{2.30}$$

$$\Delta r = \frac{\Delta(\alpha+\rho)B}{(\cos(\alpha+\rho))^2} = \Delta\rho_s \frac{B^2 + r^2}{B} = \frac{\Delta x b_0}{b_0^2 + s^2}\frac{B^2 + r^2}{B} \tag{2.31}$$

Let us increase the angle α and assume $r >> B$, $b_0 >> s$ and $b_0 \approx f$. Then (2.31) reduces to

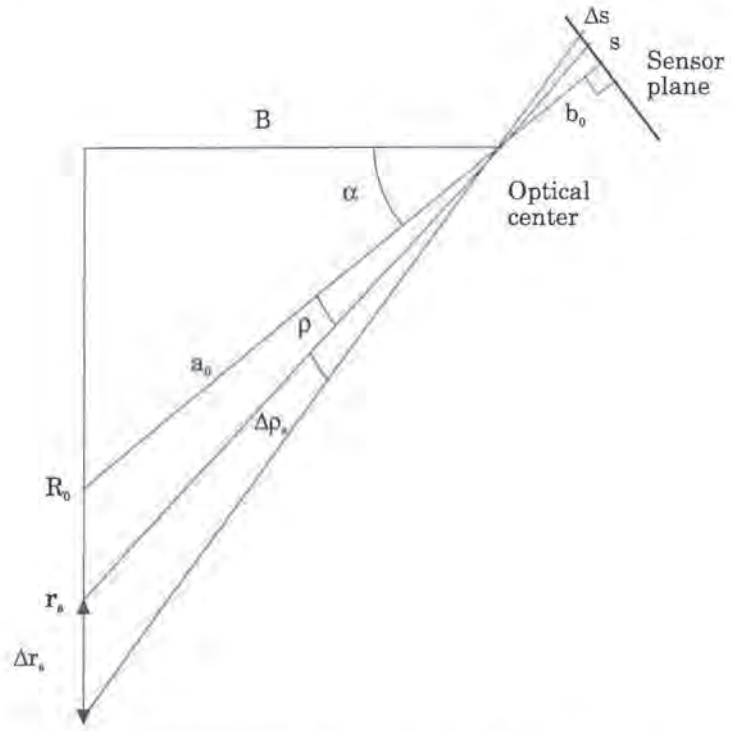
$$\Delta r = \Delta x \frac{r^2}{fB} \tag{2.32}$$

This equation shows us that the range uncertainty increases proportionally to the square of the range r. This makes it clear that range imaging based on triangulation only is attractive for short range distances where the baseline distance is at least in the same order of magnitude as the range distances. Further discussion on accuracy limitations due to sensor resolution can be found in [28].

Another limiting factor is the thickness of the laser sheet, see Figure 2.15. To give an unambiguous range value the reflected laser light should only illuminate *one* pixel in each sensor row. Each pixel interval Δx corresponds to a range increment Δr according to (2.32). For a given maximum view angle $\alpha+\rho_m$ and resolution Δr we get a maximum laser sheet thickness m as

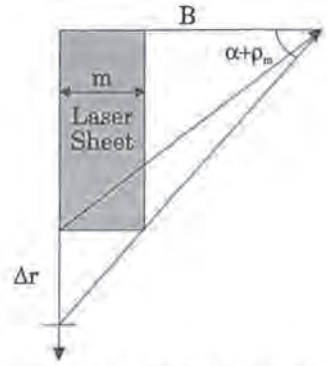
$$m \le \Delta r \tan(90 - \alpha - \rho_m) \tag{2.33}$$

FIGURE 2.14.



The accuracy limitations due to the sensor resolution.

FIGURE 2.15.



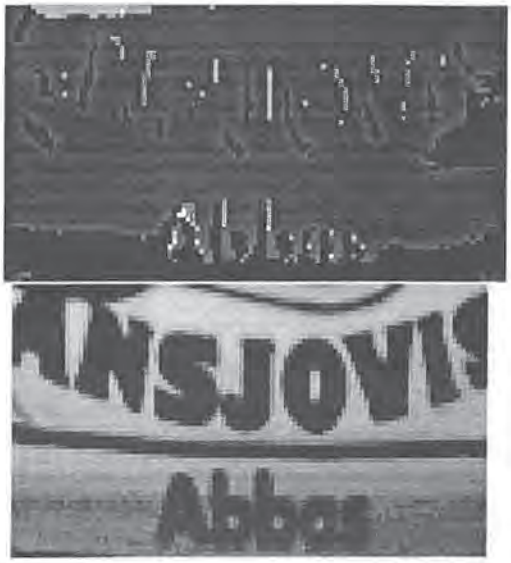
The resolution limit due to the thickness of the laser sheet.

In [44] a theoretical value for the maximum range error due to laser sheet width is given, but it is only valid for center-of-gravity sensor offset position computations. However, we can observe that the larger baseline we have, the smaller view angle α we get, and therefore better accuracy for a given laser sheet thickness. Figure 2.16 gives an example of what happens with a laser sheet which is thicker than the limit in (2.33). We see that the range values are incorrect where the maximum intensity change to/from very low values. When one part of the laser sheet illuminates an area of low reflectance and the other part illuminates an area of high reflectivity the maximum intensity along the sensor row gets shifted left or right away from the

center-plane of the sheet as shown in Figure 2.17. Off course, if the object reflectivity is constant a wider laser sheet can be used and therefore the offset position can be obtained with very fine sub-pixel accuracy, see section 4.1.

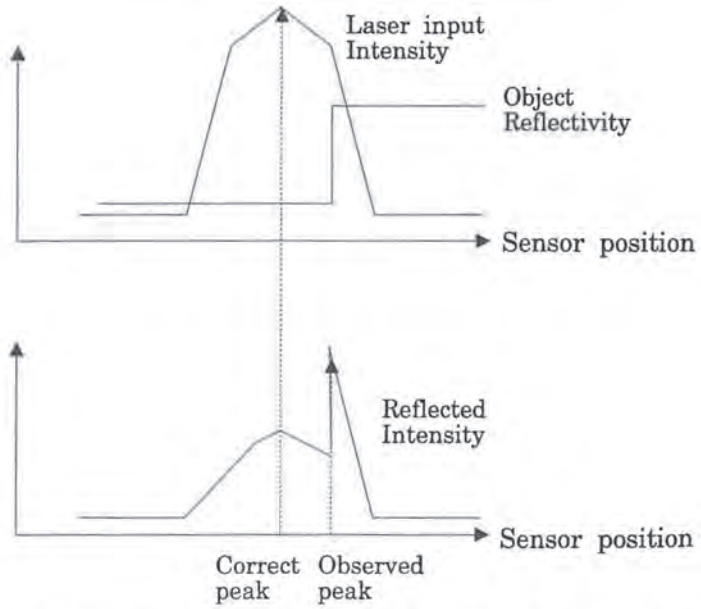
Another problem with a thick laser sheet is the risk for multiple reflections when the laser illuminates two different ranges.

FIGURE 2.16.



A range image where the laser sheet width is larger than the limit in (2.33). The range image depicts a flat surface. The top image is the range and the bottom image the intensity.

FIGURE 2.17.



Response from a thick laser sheet illuminating an object with strong reflectivity contrast.

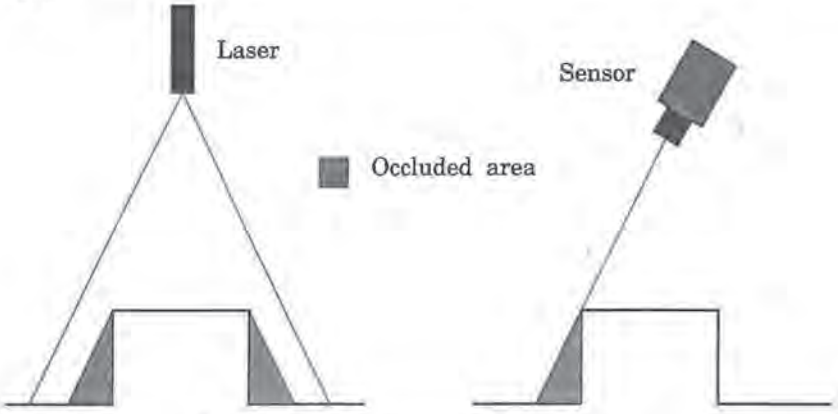
In theory the laser can be focused to a very thin sheet, limited only by the wavelength of the light. Unfortunately, some materials such as plastics spread the light so that a thick line is reflected nevertheless. If the 3D-object is made of a material which does not spread the light the potential range increment Δr may well be in the 20 μm range [45].

Since the laser sheet thickness changes over the range it is not possible to focus the laser precisely over a very large distance. In [44] dynamic laser focusing is suggested, but this requires that two range images are acquired. The first one is to achieve approximate range values so that the laser sheet focus distance can be adjusted and the second one to obtain the exact range.

2.7 Occlusion

A major problem in all triangulation methods is occlusion. In our case the problem manifests itself as one of the two following cases, see Figure 2.18. Either the laser light does not reach the area seen by the camera (laser occlusion) or the camera does not see the area reached by the laser (camera occlusion). In both cases the maximum reflection peak found in the sensor row data is the result of noise and/or ambient light. Any range value computed from such a maxima is totally insignificant as seen in Figure 2.19. We see large spurious range values computed for a large occluded area to the right of the can and smaller occluded areas around the other edges.

FIGURE 2.18.



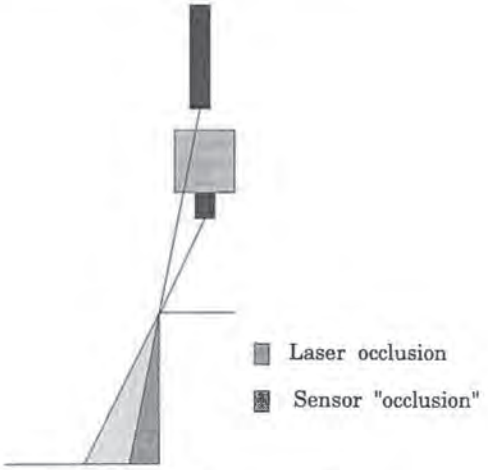
Laser (left) and camera (right) occlusion.

Both laser and camera occlusion can be minimized by careful placement of the laser and sensor. Laser occlusion is avoided by assuring that the laser reaches all areas seen by the sensor. Ideally, the baseline should be small so that the sensor and the laser can be considered as being in the same plane. Laser occlusion is then avoided by ensuring that the optical centre of the laser lens is further away from the scene than the optical centre of the sensor system, see Figure 2.20. Here the divergent sheet-of-light reaches all areas seen by the sensor since for any occluding object edge the area not illuminated by the laser is smaller than the area not seen by the sensor. Laser occlusion can also be avoided with multiple laser sources each illuminating the scene a little from the side, see Figure 2.21.

FIGURE 2.19.

A3D plot of an unfiltered range image of a fish conserve can illustrating the effects of occlusion.

FIGURE 2.20.

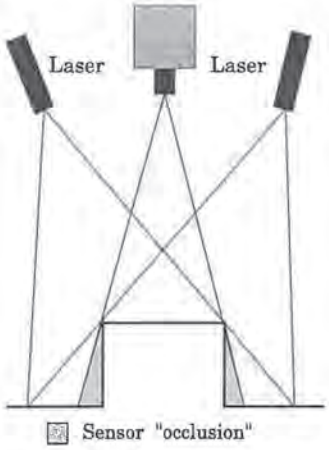


Laser occlusion avoidance.

Sensor occlusion occurs when the baseline increases from the ideal zero value. This can only be avoided using multiple sensors as seen in Figure 2.22. If two sensors are used, each viewing with a baseline B, but from separate sides of the laser most of the sensor occlusion is avoided. For more on the problems and benefits of a two-sensor system see for instance [51]. The only remaining sensor occlusion comes from deep holes where both sensors are occluded [57]. The system can be designed with only one sensor and mirrors reflecting the light from two virtual baselines as in Figure 2.23 [43]. The sensor must alternate between imaging as the

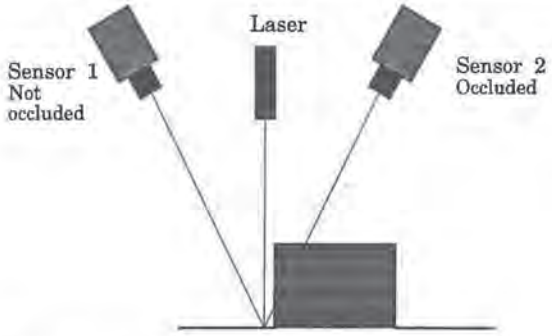
two virtual sensors since the sensor offset position will be different for the different virtual sensors. This technique also reduces the physical width of the system since the utilized baseline is wider than the system.

FIGURE 2.21.



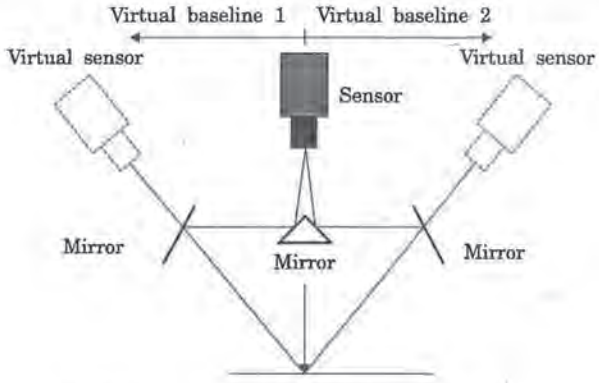
Laser occlusion avoidance using two lasers.

FIGURE 2.22.



Sensor occlusion avoidance using two sensors.

FIGURE 2.23.



Using one sensor with two virtual baselines.

Chapter 3

Sheet-of-light range camera calibration

3.1 Calibration Procedures

As was shown in chapter 2 the mapping from sensor to world coordinates can be found using trigonometric and geometric calculations. Often the geometry parameters are hard to measure, especially the interior camera parameters, which are the focal distance and sensor center offset position relative to the optical center. An alternative to setup measurements is to calibrate the system to find all unknown parameters in the equations. No direct (non-iterative) solution exists to find all the parameters, only iterative solutions [23]. However, a direct (non-iterative) solution exists to find 4x4 homogenous transformation matrix describing the sensor-to-world transformation [51], [55]. Alternatively, if the interior camera parameters are known all transformation parameters in the equations can be determined [41].

Another approach is to find the precise matching for a number of points in the scene and make an interpolation for all points in-between [57]. This method takes all lens and setup distortions into account, but it is often cumbersome, both in data acquisition and in the mapping from sensor to world coordinates. If this mapping is made as a direct lookup, a table with MxN entries is needed. A smaller table can be used, but then a 2D interpolation step is needed for each 3D value that is computed. For example, if $N = 256$ and $M = 512$ a complete table with 32-bit data precision would require 1Mb.

A third approach, which is used here, is to calibrate the system and find a polynomial approximation of the desired functions for range and width derived in chapter 2. This approach results in two small 1D lookup tables (one with N and one with M values) and requires only two table lookups and one multiplication to find the desired world coordinates for range and width. Another reason to choose this approach is that the resulting tables can be used with the existing *RANGER* range camera software [48]. However, this calibration scheme does not account for lens and setup distortions.

3.2 Function approximations

In our calibration we find the best polynomial approximations of the range and width equations (2.17) and (2.24) using least-square techniques. We wish to use a low order polynomial since it allows us to use fewer calibration points, and it minimizes the risk that the approximation fits closely to calibration data corrupted by noise. This is because a high-order polynomial tends to follow the noise. To find the suitable polynomial orders we have evaluated the maximum deviations between the correct equations and the least-square fitted polynomial functions. We used a setup with f=23 mm and B=300, and using the view angles α=30°,45° and 60°. In Table 3.1 we give the maximum deviations between the least-square fitted polynomial and the correct range equation (2.17). In this example each pixel corresponds to a range of roughly 0.5-3 mm and we aim at an accuracy of the resulting system which is within 1 pixel (the range camera algorithms used has a resolution of 1/2 pixel). Thus we propose to use at least second, third and fourth order approximations for the three cases α=30°,45° and 60° as this can give an approximation accuracy within +/-0.25 mm.

TABLE 3.1.

	Maximum approximation error (mm)				
View-angle	App deg 1	App deg 2	App deg 3	App deg 4	App deg 5
30°	4.3	0.24	0.01	0.0001	< 0.00001
45°	12.3	1.2	0.12	0.01	0.001
60°	50.9	9.0	1.5	0.25	0.04

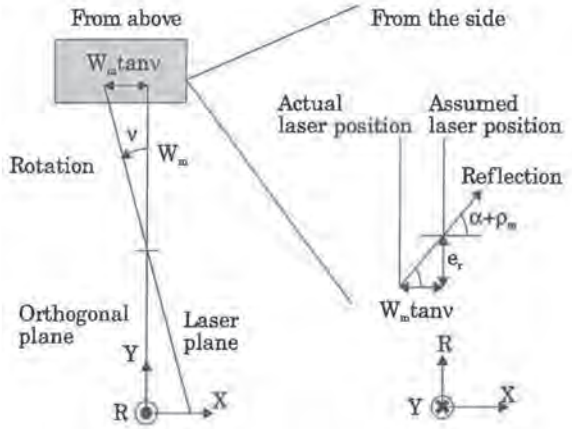
If we compare with Table 3.1 the maximum width error is approximately 10% of the maximum range error. Therefore we chose to use a first order approximation. However, for angles above ~45° it might be worthwhile to use a second order approximation.

3.3 Error sensitivity

In our calibration scheme we assume that the laser plane is parallel to one axis of the sensor. How sensitive is the system to violations of this assumption? If the laser plane is rotated an angle ν around the range axis (laser optical axis) this gives a translation $W_m \tan\nu$ of the laser plane reflected at the edge of the sensor, see Figure 3.1. This translation gives a projection range error which is

$$e_r = W_m \tan\upsilon \tan(\alpha + \rho_m) \tag{3.1}$$

FIGURE 3.1.



The projected range error when the laser plane is rotated.

If we input the system parameters, using the approximation that b_0 is approximately equal to f into equation (3.1) we obtain

$$e_r = \frac{BN\Delta x \tan\upsilon}{2f\cos\alpha_p^{\,2}} \tag{3.2}$$

where

$$\alpha_p = \alpha + \rho_m = \alpha + \text{atan}\left(\frac{N\Delta x}{2f}\right) \tag{3.3}$$

If we use the system considered before, with B= 300 and f=23, we find that for a 1° degree rotation the error is 1.6, 2.8 and 8 mm for the cases α = 30°, 45° and 60° degrees respectively. Thus, we see that the laser plane must be aligned very carefully, which has proved to be very hard.

3.4 Calibration procedure

To calibrate the system we need to know the corresponding world and sensor coordinates for at least one more point than the highest order of the approximations. We use a calibration object with bright patches at well defined 3D positions placed in the laser plane. In order to achieve an accurate calibration the calibration object should cover as large part of the field-of-view as possible.

A segmentation program implemented on the MAPP2200 smart sensor is then used to accurately find the center-of-gravity of these patches [69]. Least-square adaptions are then made to find the best approximations of the transfer functions. Since both functions are approximations around the origin and the exact location of the optical center (camera coordinate system origin) is unknown, we iterate the least-square adaption over several possible sensor offset positions and use the offset which gives the best least-square fit to the model. The whole procedure takes less than 1 minute using a standard PC. The resulting polynomials are then used to create lookup-tables compatible with the *RANGER* software.

3.5 Experiments

We give results for two setups, one with a baseline B = 290 mm, view angle α = 45° and one with a baseline B = 170 mm and view-angle α = 30°, both with an f = 23 mm lens. The first system has a total range resolution of approximately 200 mm and each pixel corresponds to approximately 0.7-1.1 mm. The second system has a total range resolution of 80 mm where each pixel corresponds to approximately 0.3-0.7 mm. The fit between the calibration data set and the second and third order approximations are found in Table 3.2. We see that both the maximum error and the standard deviation is more than halved using the third order approximation compared to the second order ditto.

In Table 3.3 we have collected the corresponding results when the calibration lookup tables were applied to actual range data. We can see that the resulting accuracy of the calibrated system is close to the theoretical (measured) system in the first case (in the second case no accurate setup measurements were made). Thus we believe that the observed errors are errors in the range data acquisition rather than in the calibration. In the second experiment the range acquisition was more accurate, i.e. the thickness of the sheet-of-light was wider, giving a more stable range algorithm output, and we see that the second order approximation actually gives better results than the third order ditto. This is probably due to the low number of points (5) used in the calibration. Here the calibrated range accuracy is within 1 pixel on the sensor.

TABLE 3.2.

Model	Max Error (mm)	Std dev (mm)	# points
2-deg app 1	1.18	0.56	12
3-deg app 1	0.46	0.24	12
2-deg app 2	0.28	0.17	5
3-deg app 2	0.08	0.05	5

TABLE 3.3.

	System 1		System 2	
Calibration	Max error (mm)	Std dev (mm)	Max error (mm)	Std dev (mm)
Equation	1.6	0.7	-	-
2-deg app	1.8	0.7	0.25	0.15
3-deg app	1.2	0.6	0.48	0.2

We feel that the resulting accuracy, around 0.5% of the measurement range is adequate and comparable to those reported in [57], [55] and [51]. However, we have noticed that our calibration scheme is very sensitive to sensor and calibration (laser) plane rotations. That is, if the Y and T axes are non-parallel no correct calibration can be made since the range values then depend both on the s and t axis position of the sensor data. We have not presented any results on the y-axis calibration accuracy but from our experiments we believe that it works as good as the range calibration.

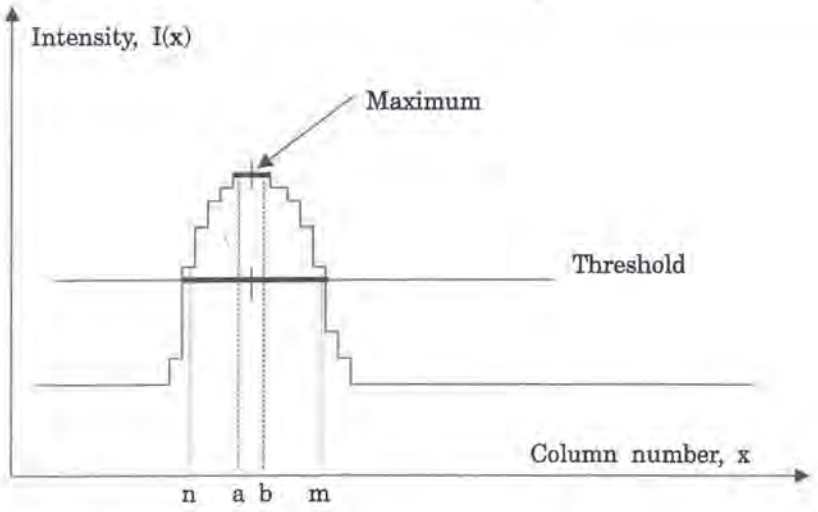
Chapter 4

Signal processing for sheet-of-light

4.1 Localizing the impact position on the sensor

Assume that the scene contour is reflected vertically along the image array as in Figure 1.6C. Along a row in the sensor the sheet-of-light reflected from the 3D scene may produce a signal as in Figure 4.1. Various rules and algorithms may be formulated to derive a unique impact position (*pos*) from this 1D signal. The most natural may be to find the position of the maximum intensity. Alternatively, we can threshold the signal and find the mid position of the pixels above the threshold. Analog devices such as position sensitive devices (see section 5.1) measure the center-of-gravity of the incoming light and this can also be computed with ordinary pixel-based sensors.

FIGURE 4.1.



An illustration of two different ways to find the laser reflection.

As indicated in Figure 4.1, even if the signal is well-behaved, the localization of the maximum of a discrete and quantized signal might be ambiguous. Assuming that the signal maximum is flat in the interval $a \leq x \leq b$ the position (*pos*) would naturally be chosen as

$$pos = \frac{a+b}{2} \tag{4.1}$$

In some smart sensors [11], [38] the position is actually given as either a or b whichever is easier to compute. Obviously, this simplification is allowed only if it is ensured that the interval $b-a$ is small.

If we use thresholding the peak mid position is given by

$$pos = \frac{n+m}{2} \tag{4.2}$$

where n and m are the first and last positions with a pixel value above the threshold respectively, see Figure 4.1. For signal peaks with a width of an even number of pixels equations (4.1) and (4.2) result in a position halfway between two pixel centers. This tells us that these formulas give answers with half-pixel resolution (sub-pixel resolution).

The center-of-gravity position using the total signal is given by

$$pos = \frac{\sum_x xI(x)}{\sum_x I(x)} \tag{4.3}$$

where x is the pixel column position and $I(x)$ the pixel intensity.

In [4] an accuracy of 1/20000 is claimed for a sensor with 388 pixels/row (which is 1/50 pixel sub-pixel resolution) using a White Scanner [54] with center-of-gravity computations. We find these claims somewhat doubtful and certainly impossible to hold up in the presence of noise and changing object reflectiveness. In [44] 1/16th of a pixel resolution is reported, but it is also stated that 1/2 pixel is a realistic value when viewing metallic surfaces with specular reflections. This resolution is acquired with a twin laser-sheet illumination scheme, and approximating (fitting) the signal bolus (c.f. the signal in Figure 4.1) with a Gaussian before the center-of-gravity computation. The double-sheet illumination design utilizes two lasers of slightly different wavelengths displaced slightly from each other and this scheme is used to avoid effects of abruptly changing object reflectiveness (c.f. Figure 2.17).

Both the maximum and center-of-gravity techniques can be combined with thresholding, so that only values above a threshold are considered in the computations. This makes the computations less sensitive to noise and background illumination. A compromise between center-of-gravity and thresholding, where multiple thresholds are used and the *pos* value is chosen as the mean center-of-gravity of the binary peaks obtained in the thresholding, is suggested in [20]. Maximum finding can also be combined with multiple thresholds to increase the position resolution.

A well known technique for finding the exact location of a peak is to find the position of the zero crossing of the first derivative. Using this for laser triangulation was suggested in [7]. Here several different derivative operators were evaluated using linear interpolation to determine the exact zero crossing location. The filter that gave the best performance was an eight-order FIR filter, with the convolution kernel [1 1 1 1 0 -1 -1 -1 -1]. Using this a resolution of 1/5 of a pixel was achieved in their experiments. If we assume a certain smoothness of the imaged scene it might be worthwhile to use filter kernels that cover several rows, for instance 3x3 or 5x5 Sobel filters, but this has not been evaluated.

Another elaborate scheme to find the range is to threshold the data at half of the maximum intensity and then use a linear interpolation as in [10]

$$pos = \frac{1}{2}\left[n - 1 + m + \frac{T - I_{n-1}}{I_n - I_{n-1}} + \frac{I_m - T}{I_m - I_{m+1}}\right] \tag{4.4}$$

The portion above the threshold is n to m, see Figure 4.1. According to the authors this gives very good resolution, but they also claim that their thresholding and center-of-gravity methods produced equally correct values. Furthermore, maximum finding was found to be more noise sensitive than thresholding.

In the MAPP2200 chapter we will discuss the implementation of an ad-hoc peak-detection algorithm called the odd-even method, OEM [66]. In this method the odd and even rows of the sensor are processed separately. Starting with the odd rows the charge from all pixels is added until the sum is above a certain threshold. The same technique is applied for the even rows, but starting on the other edge of the sensor. The idea is that each measurement gives the start position of the peak from one direction of the sensor, and that the correct peak position is the mean of these values. The algorithm is sensitive to noise since a small noise contribution from each pixel can give a large contribution to the sum which is compared with a threshold. The algorithm and its implementation is discussed in more detail in section 7.13.

To evaluate these different algorithms we have performed two sets of experiments. First we simulate the algorithm peak detection accuracy with and without the addition of white noise. Then, we test the algorithms on real data, where the noise cannot be said to be white. In the simulation results we can give very accurate error estimates, but the evaluation of the real data experiments is only qualitative.

Simulations

The simulations were performed using Matlab. The Signal-to-Noise Ratio, SNR, of an image row is measured as

$$SNR = 10\log\frac{\sigma_s^2}{\sigma_n^2} \tag{4.5}$$

where σ_s is the standard deviation of the signal and σ_n is the standard deviation of the Gaussian noise. The signal has 8-bit dynamic range. Since the algorithm performances might be dependent on the peak width we evaluated a narrow and a wide peak. The narrow peak covered approximately 2 pixels at half the maximum intensity and the broad peak 5. Results for the sub-pixel accuracy of two different peak-widths are found in Table 4.1 (narrow) and Table 4.2 (wide). In Table 4.1 the columns indicate maximum absolute error and mean error in pixels for zero noise, 43 and 10 dB SNR, left to right, and in Table 4.2 they indicate zero noise, 66dB SNR and 23 dB SNR. The added noise component is the same in the two simulations, but the peak width is changed, thus changing the SNR.

The simulated algorithms are thresholding, maximum finding, maximum finding with four extra thresholds, center-of-gravity with and without thresholding, the odd-even method, the interpolation method and the derivative method. The derivative method has been simulated using four different filter kernels, three derivative of Gaussian filters and the filter suggested in [7]. The derivative of Gaussian filters have the size three, five and seven pixels, and they are [1 0 -1], [1 2 0 -2 -1] and [1 3 3 0 -3 -3 -1] respectively.

TABLE 4.1.

	Algorithm accuracy measured in pixels					
SNR	∞ dB		43 dB		10dB	
Algorithm	E-max	E-mean	E-max	E-mean	E-max	E-mean
Thresholding	0.49	0.05	0.47	0	94.3	0.7
Max-position	0.49	0.04	0.76	0.03	1.01	0.03
Max-multi thr	0.18	0	0.3	0	0.49	0.01
CoG	0.01	0	5.15	0.66	6.6	0.70
CoG w thr	0.20	0.02	0.34	0	13.96	0.1
Odd/even	0.73	0.05	7.67	0.6	10.97	0.64
Interpolation	0.02	0	0.23	0	114.4	1.4
Der 3	0.03	0	0.18	0.01	0.41	0
Der 5	0.02	0	0.17	0.01	0.36	0
Der 7	0.02	0	0.19	0	0.35	0
Der 8	0.01	0	0.22	0	0.44	0

TABLE 4.2.

	Algorithm accuracy measured in pixels					
SNR	∞ dB		66 dB		23 dB	
Algorithm	E-max	E-mean	E-max	E-mean	E-max	E-mean
Thresholding	0.48	0.05	0.66	0.01	1.68	0.01
Max-position	0.8	0.05	1.36	0.05	1.89	0
Max-multi thr	0.19	0	0.44	0	0.79	0.01
CoG	0.01	0	4.2	0.73	6.37	0.74
CoG w thr	0.20	0.02	0.50	0.01	4.38	0.04
Odd/even	0.75	0.05	6.89	0.88	14.6	0.29
Interpolation	0.01	0	0.38	0	3.89	0.03
Der 3	0.05	0	0.65	0.01	1.20	0.02
Der 5	0.03	0	0.36	0	0.74	0.02
Der 7	0.02	0	0.28	0	0.61	0.01
Der 8	0.01	0	0.24	0.01	0.48	0.02

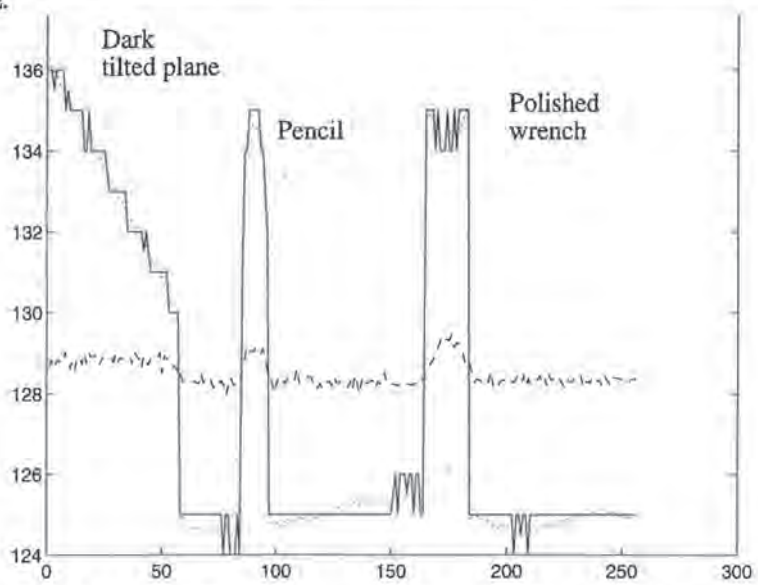
In the leftmost columns in these tables we see how the algorithms perform under ideal conditions. As can be expected the more elaborate algorithms give high accuracy, and the simple maximum and thresholding around 1/2 pixel accuracy. If we study the middle rows of the tables we see how the algorithms perform under fairly reasonable SNR values. We see that the center-of-gravity, CoG, and odd/even methods give large errors, and that maximum finding on a broad peak also gives quite large deviations. We also see that the claim from [7] of 1/5 of a

pixel should be possible with the derivative algorithm and an appropriate derivative kernel. If we study the rightmost columns we see how the algorithms perform under low SNR. We can see that maximum finding combined with multiple thresholds gives fairly good accuracy and that the derivative methods perform very well, but for the narrow peak the Gaussian derivative filters give slightly better accuracy than the filter suggested in [7]. We also note that the interpolation method and center-of-gravity with thresholding, which works well under moderate noise conditions deteriorates under heavy noise. As a conclusion we can state that maximum finding combined with multiple thresholds or a derivative method seems to give the best peak detection accuracy under very noisy conditions.

Experiments

We have tested the algorithms above on real images acquired using the MAPP2200 sensor. The images illustrate several of the error sources in sheet-of-light range imaging i.e. occlusion, uneven object reflectiveness, secondary reflections and illumination from ambient light. An example of range profiles acquired using maxima, center-of-gravity and derivative-based maxima detection can be found in Figure 4.2. As was expected the center-of-gravity is very inaccurate but the other two methods give similar, more correct, profiles. We also tried the algorithms subtracting two images, one obtained with the laser on and one with the laser off. This gives almost noise free images and all algorithms give performance in accordance to the table above. Another effective way to reduce the problem with ambient illumination is to use $\lambda/4$ optical interference wavelength filters which only allows light with a wavelength close to the laser wavelength. More experiments using real data are reported in [31].

FIGURE 4.2.



An example of range profiles. The scene contains three objects, a dark tilted plane, a wooden pencil and a polished wrench on a flat surface. The profiles were obtained using maximum (solid) derivative (dotted) and center-of-gravity (dashed) methods.

4.2 Multiple maxima considerations.

With all the above algorithms we get problems if two or more maxima appear in one sensor row. The most natural scheme would then be to simply remove or disregard the range results, or mark them as uncertain. However, this is not possible in all sensors. Some sensors/algorithms output the position of one of the maxima [11], or the mid-position between the two [2], [60].

In some applications it is desirable not to discard the values but to read out the range values from several peaks on one row, and leave the decision on which datum is correct to the next computational level. The reason that several peaks occur might not always be background illumination noise but secondary reflections from the laser reflecting on specular or transparent surfaces. In those cases more global information is needed to distinguish between the correct and the false reflections [46].

4.3 Post processing

The main purpose of the post processing is to detect, remedy, or alleviate the effects of occlusion and multiple reflections. Of course the true offset position for an occluded position can never be obtained, but once detected it might be approximated and/or interpolated, for instance using so called normalized convolution [39]. Alternatively the range data can be labeled as occluded (uncertain). We have used two different cues to detect erroneous (occluded) peak positions. *Error filtering* uses the error value, which we define as the number of detected maxima (peaks) in the sensor row. If occlusion occurred, the background illumination should be almost homogeneous and thus the probability that several peaks are found is large. This filtering is also very effective for detecting multiple reflections. Thus, when the number of maxima are more than one we know that the range datum is unreliable. Another way to detect and alleviate effects of occlusion is to use only those sensor rows where some pixels has an intensity above a certain maximum value. We define this as *intensity filtering*. If the laser is occluded very little light reaches the sensor row and the maximum intensity is low. Actually, if we use a thresholding algorithm intensity filtering is an integral part of the algorithm. An alternative to intensity filtering is to use the intensity as a confidence measurement in normalized convolution. When discussing different smart sensors and smart sensor implementations we will also discuss their ability for intensity and/or error filtering. Further discussion on different post processing techniques can be found in [31].

Chapter 5

Sensor designs for sheet-of-light

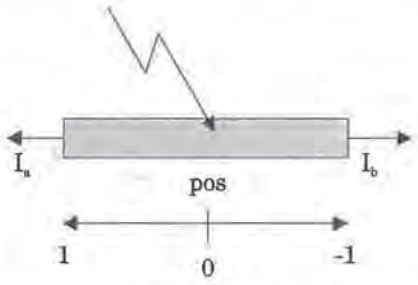
5.1 Position Sensitive Devices

A laterally extended photo diode is shown in Figure 5.1. This device is known as a Position Sensitive Device, PSD. The incoming light produces a photo-electric current in the p-n junction of the diode, and by measuring the currents flowing out from the edges of the diode the centroid position of the illuminated area can be obtained with the formula

$$pos = \frac{I_a - I_b}{I_a + I_b} \tag{5.1}$$

The sum I_a+I_b is proportional to the total incoming light intensity.

FIGURE 5.1.



A 1D PSD photo diode.

Sheet-of-light range finders based on use of multiple 1D PSDs has been presented by Verbeek et. al. [53], [60], [61] (1985) and Araki et.al. [2], [3] (1990), see Figure 5.2.

From Verbeek et. al. a 96-strip PSD array chip has been fabricated for which is reported 2 Mrangels/second with an accuracy of 1%. However, as of present, due to a bottle-neck in the analog processing only 8 of the 96 strips are utilized. The sensor chip from Araki et. al. has an array of 128 PSDs which is said to produce range images at 32 Hz with a range accuracy within 0.3%. If 128^2 images are obtained the range pixel frequency is 500 KHz. However, it is also stated that the PSD response time is 1 µs. Assuming that the processing is no bottleneck the theoretical maximum rangel frequency is in the 100 MHz range. But given the processing complexity a more realistic maximum rangel frequency ought to be more in the order of 1-10 MHz.

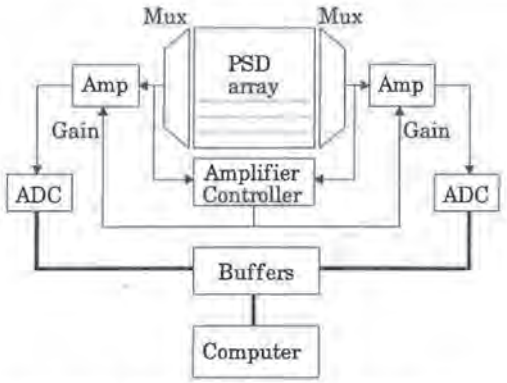
We notice that the primary sensor output from a PSD is the two analog values I_a and I_b which have to be transferred (in Figure 5.2 also multiplexed) and A/D converted. Analog data transfers are more sensitive to noise than digital data transfers which is a potential problem for PSD range finders.

Another problem is the following. If the background light intensity is high in parts of the scene, the centroid position is no longer the same as the peak position value as pointed out in section 4.1. To overcome this problem a PSD-system needs very high signal to noise ratios, and this can only be acquired by either increasing the sheet-of-light intensity and/or quelling

the ambient illumination. This is reflected in the use of a very high powered laser, 200 mW, in the case of [2], [3], and very short imaging distance, 50 mm, in the case of [53], [60], [61]. An image subtraction technique similar to the one discussed in section 4.1 can also be used with PSDs [59]. First the outputs of the array are recorded without any laser illumination. Then, when the sheet-of-light sweeps over the scene these values are subtracted from the sensor outputs prior to the pos calculations.

Since the intensity values might be read out they can be used as a confidence measurement. It is not possible however, to detect multiple maxima.

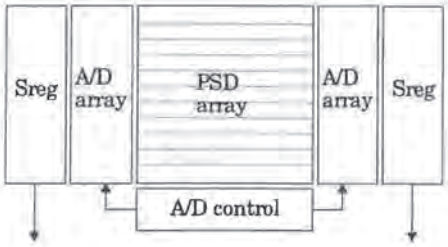
FIGURE 5.2.



A range finder architecture from Verbeek et. al. [60]. The amplifier control is used to continuously adapt the analog values to the working range of the eight-bit A/D converters.

To overcome the drawback of analog data transfers and the A/D conversion bottle neck in the proposed PSD array architectures we can combine a PSD array sensor with on-chip A/D conversion using row-parallel A/D converters, see Figure 5.3. The A/D converter and shiftregister layouts are very similar to the MAPP2200 design [17], [25], [27]. In [27] C. Jansson describes a 16-bit A/D converter design suitable for array implementations, and this precision would certainly be enough for high accuracy sheet-of-light range imaging. The conversion speed of the described circuit is 78 µs, giving an effective sample frequency of 3.3 MHz in a 256 cell array, but it is also stated that higher speeds are possible. Thus, this architecture should easily be able to output digital samples corresponding to 50 Hz 256^2 range image frame rate. Higher speeds are probably possible, especially if lower A/D conversion precision is used.

FIGURE 5.3.



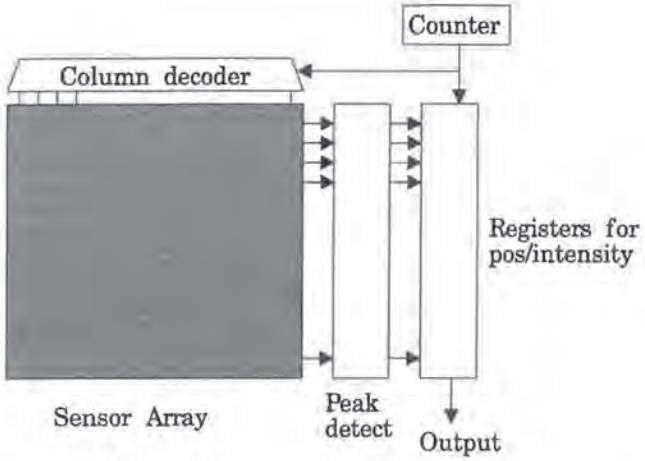
A novel sheet-of-light sensor combining a PSD array and on-chip A/D conversion.

5.2 Smart sensor architectures

We will present three devices below all of which attempts to compute position (offset) from a 2D discrete sheet-of-light sensor using on-chip circuitry. To be able to compare these non-implemented architectures we have assumed that we need 10 μs integration time and that the on-chip processing and analog data transfer frequency is 10 MHz.

The *column readout peak detection sensor* was originally described briefly in [11], and it has not been implemented. The chip combines an ordinary photo-diode or CCD sensor array with a column of peak detection circuits, see Figure 5.4. After integration of one sheet-of-light position in the sensor array the analog pixel values are read out column-wise, and the maximum positions are obtained during the column readout by the analog peak detection circuits. Thus, one column of data from all sensor rows are processed in parallel, not in series as is the case with a standard video sensor.

FIGURE 5.4.

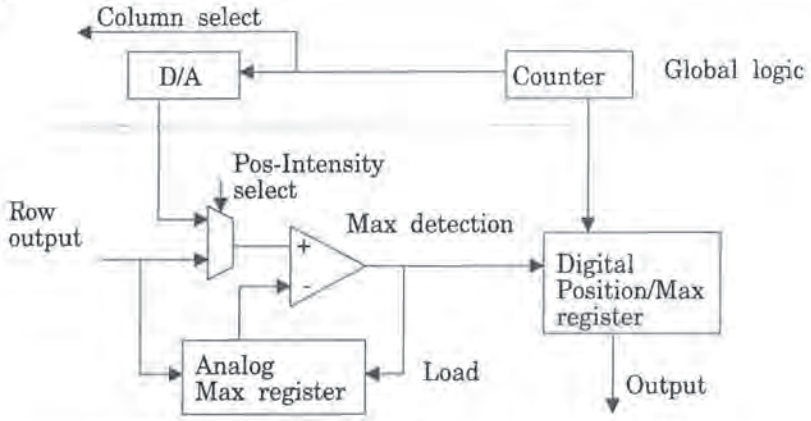


Column by column readout with peak detection.

The peak detection circuits is suggested to consist of a column of comparators which compare each row output with the previously recorded maximum intensity for that row, see Figure 5.5. If a new maximum is found, the analog value is saved in an analog register and the digital column position is saved in a position/max register.

After the position has been obtained the analog maximum registers contain the intensity values for the range data. To obtain the digital maximum intensity values a D/A converter instead of the row output is connected to the comparator so that the maximum value can be compared with a global ramp (staircase) signal from the D/A converter. When the ramp signal reaches the value of the analog max register the counter value is loaded to the corresponding register as seen in Figure 5.5. Thus, one column of range values for one sheet-of-light position is captured in a time frame which consists of integration plus sensor readout and peak detection plus position result output plus (optional) detection and readout of intensities. By doubling the digital registers sensor readout and peak detection can be made in parallel with the position result output. By again doubling the digital registers and also doubling the max-detection circuits and analog registers the intensity detection can be made in parallel with the next sensor read out and peak detection.

FIGURE 5.5.



One peak detection circuit with maximum value output.

If we have an *NxN* sensor one *NxN* range and intensity image can then be obtained in a time defined as

$$T_{im} = N\left(T_{Int} + \frac{N}{f}\right) \tag{5.2}$$

where f is the processing frequency. If we assume that the integration time is 10 μs, the processing frequency is 10 MHz and $N = 256$, a complete 256^2 range image can be obtained in approximately 9 ms. This is a range and intensity pixel frequency of 7.2 MHz.

If we redesign the sensor so that we can integrate and process in parallel we can achieve a range and intensity image frequency of approximately 150 Hz, corresponding to a rangel/intensity frequency of 10 MHz. This method might introduce errors since the maximum finding is done on columns integrated at slightly different time intervals, but we believe that this is no great problem. It is also possible to include peak width and multiple peak detection in the read-out circuits to decrease the noise sensitivity. If we preload the analog max registers with a threshold value before the sensor readout we obtain an automatic intensity filtering.

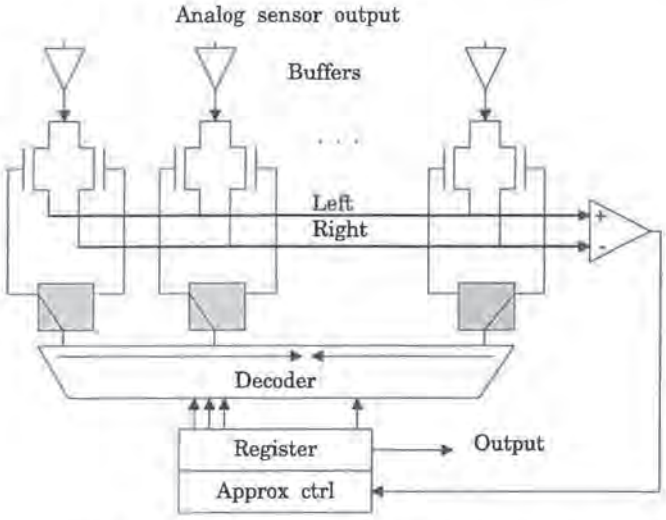
The *row readout successive approximation architecture* was also described briefly in [11], and it has not been implemented. It utilizes row-by-row readings of analog sensor data from an ordinary photo diode or CCD sensor array. For each row of analog sensor output we find the *mid-point position of the incoming light profile*. The analog values from the sensor outputs are multiplexed and summed to one of the two inputs of a comparator, see Figure 5.6. The output of the comparator controls a sequential network of type successive approximation which moves the approximative mid-point position left or right. First the array middle is tested as midpoint. If the left sum is larger than the right sum the assumed mid-point is moved so that the leftmost 25 percent are summed and compared to the rightmost 75 percent e.t.c. To get the output for an N bit row the network has to do $logN$ approximations. After the approximations the register indicates the pixel position where half of the total intensity is to the left and half to the right, that is, the mid-point. The mid-point is the integer approximation of the center-of-gravity position.

Sensor integration and read-out can take place in parallel, but the readout speed is not so high since *logN* cycles are needed for each rangel output. Another disadvantage is that no intensity information can be obtained. If we have an *NxN* sensor an *NxN* range image can be obtained in a time defined by

$$T_{im} = N\left(max\{T_{Int}, \frac{N\log N}{f}\} \right) \tag{5.3}$$

If we assume a 256x256 sensor using an integration time of 10 μs, having a clock frequency of 10 MHz, the maximum range pixel frequency is limited by the processing and is 1.25 MHz. This is a range image frequency of 20 Hz. Obviously the design is sensitive to background illumination since all pixels in a row always contribute to the result.

FIGURE 5.6.



Row-by-row read-out with analog/digital processing to find the mid-point of a sensor row.

An *analog processing sensor cell* design differs from "standard' sheet-of-light range imaging techniques which integrate and read out for each sheet-of-light position. With analog processing cells a whole 2D range, or at least point in time, image is captured before the 2D sensor array is read out. (The sensor cells measure the instant light intensity at any given time.) The light-sheet sweeps over the whole scene and each sensor cell records at which sweep position the maximum light intensity occurred. That is, each sensor cell records events in the scene along a ray from the sensor cell through the optical center (OC), the sight line. Using this and the recorded impact time translated to illumination angle a triangulation can be made to obtain the range value, see Figure 5.7. Thus, each full sweep single "integration" gives a complete 2D range image of the scanned scene.

A design for this purpose was proposed by Kanade et. al. [21], [38], see Figure 5.8. Peak detection and registration of the time when the peak occurred is done within each sensor cell. Actually, Figure 5.8 represents the simplified circuit implemented on a 28x32 sensor chip [21] and not the more complex circuit implemented with discrete components in [38]. Among other things the difference is that the latter had true peak detector and also the ability to register the

maximum intensity in each sensor cell. During the "integration" a well-defined analog ramp signal is distributed to the sensor cells. This value defines the time, and the ramp value is stored when a maximum occurs.

As pointed out true maximum detection is not performed. Rather a fixed threshold is used and the ramp (time) value is saved when the sensor output crosses the threshold the first time. This makes the circuitry sensitive to background illumination and secondary reflections since any peak can be chosen as the maximum. It also gives inaccurate results for a thick light sheet reflection since any part of the sheet can give a positive result.

In [21] it is argued that the time domain sampling, restricted by the time resolution in sampling of the ramp signal, in this circuit gives higher range accuracy than the traditional space domain sampling, restricted by the inter-pixel pitch of the sensor. However, this reasoning is only valid if the peak is found accurately, and this is not ensured since no real peak detection is made in the circuit. Furthermore, in space domain sampling it is often possible to determine the width of the peak to correct for laser sheet width and uneven reflectiveness as discussed in section 4.1.

FIGURE 5.7.

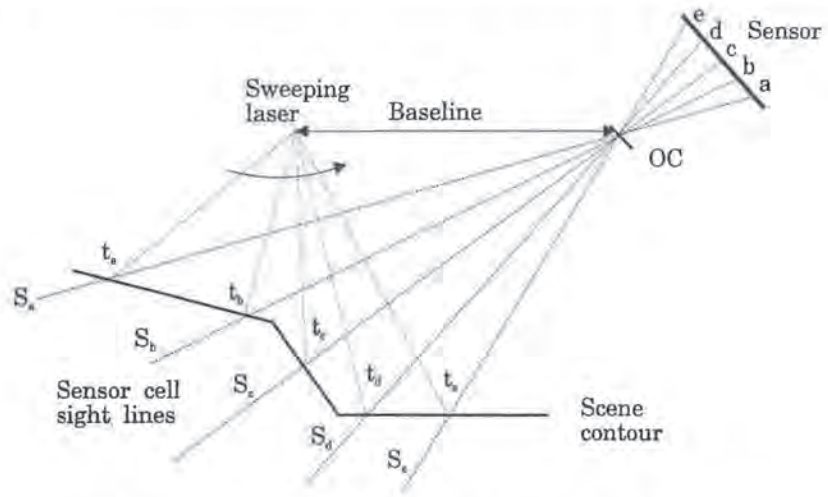
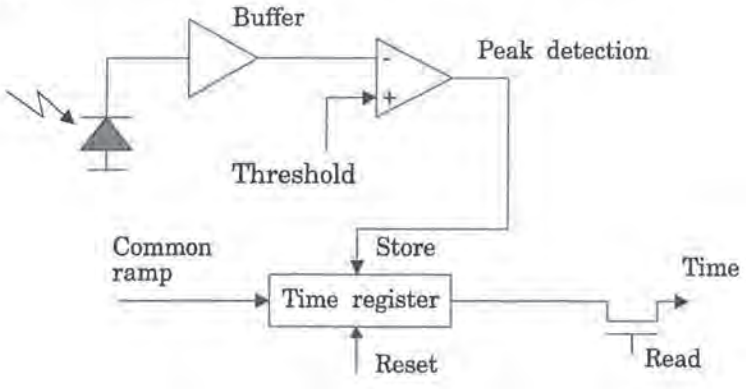


Image acquisition while the light sheet sweeps over the scene.

FIGURE 5.8.



Sensor cell due to Kanade et. al.

As we mentioned each sensor cycle gives data for one complete range image. Hence, every pixel in the 2D sensor must be read out in-between cycles which takes time and decreases the duty cycle for the system. Furthermore, analog processing is needed outside the chip to convert the analog time values to digital range ditto. Different triangulation parameters are needed for every sensor cell since each sensor cell represents one unique direction of the incoming light.

Due to the extensive hardware in each sensor cell this design is very hard to implement in standard VLSI. A recent report mentions a fabricated 28x32 array [21], where each sensor cell is 0.25 x 0.25 mm. Nevertheless, Kosuke Sato at the University of Osaka, who previously worked with Kanade, states that a similar design with 128^2 sensor cells is currently being manufactured by their laboratory [52].

In [21] it is stated that a range image speed of up to 1000 images per second can be achieved. 1000 images per second gives a range pixel frequency of approximately 900 KHz with the 28x32 sensor. If the same image speed is obtained in an 128^2 sensor the range pixel frequency is 16 MHz, and the requested readout frequency is 32 MHz if equal time is used for sweep and readout. Accurate readout of analog values at such high clock frequencies might be a problem. Even so, we admit willingly that this architecture has a potential for very high range image speeds. It is claimed [21] that the range accuracy is 1/256, based on eight bits of resolution in A/D conversion of the registered impact times.

Chapter 6

The NESSLA architecture

6.1 The NSIP paradigm

Near-Sensor Image Processing (NSIP) [18], [69], [70] is a new image processing concept where the individual sensor operation is an integrated part of the signal processing algorithm. In conventional smart sensors and image processing systems, after a fixed integration time, the sensor data is A/D converted and then processed digitally. In NSIP however the *binary* (thresholded) sensor cell outputs are repeatedly interrogated and processed *during* the integration and the results are obtained directly when the integration is completed. The use of binary signals seem like a disadvantage, but by processing the binary signals as they evolve over time we can nevertheless emulate grey-scale image processing. Actually, if we want a grey-scale output, by repeated interrogation of the binary outputs we can measure the time from the start of the integration until each binary sensor output switches from "0" to "1". The time until the sensor output switches is inversely proportional to the accumulated light intensity.

A general 2D NSIP architecture [69], [71] is capable of many low level image processing algorithms, but the architecture we present here is specialized to implement a row wise maximum finding algorithm. Thus the *Sensor-Processing-Element* (SPE) is much simpler and easier to implement than the general case. The architecture to be presented below can be implemented in full-scale with current CMOS technology. We call the architecture *NEar-Sensor Sheet-of-Light Architecture*, NESSLA [36], [37].

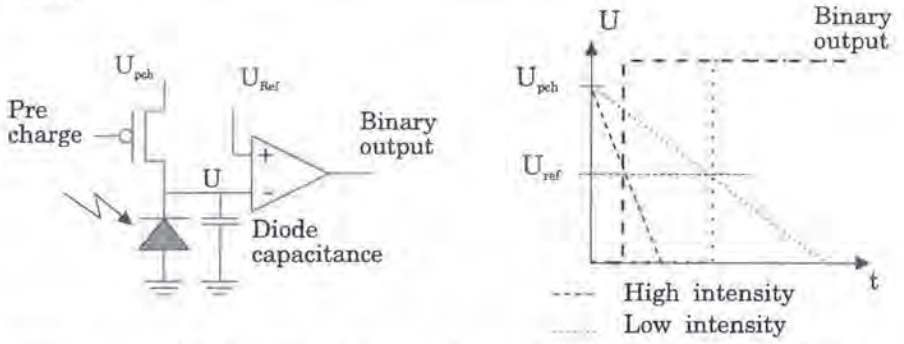
6.2 The NSIP sensor element

The basic NSIP sensor element is shown in Figure 6.1. It consists of a pre-charged photodiode connected to a high impedance voltage comparator with a binary output. If the diode voltage is below the reference voltage the comparator output is high (*one*), otherwise low (*zero*). After the diode precharge the incoming light will induce a current in the diode, proportional to the intensity. This current will discharge the capacitor of the diode and decrease the voltage. The time from the precharge of the diode to the time the diode voltage crosses the reference voltage of the comparator is inversely proportional to the light intensity, see Figure 6.1. Further discussions on NSIP sensors can be found in [69].

6.3 The NESSLA Sensor-Processing-Element

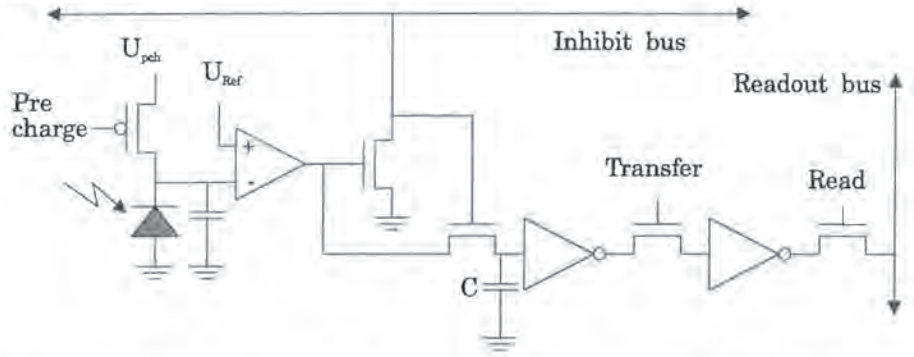
The design of a complete individual NESSLA Sensor-Processing-Element (SPE) is shown in Figure 6.2. The sensor area consists of a two-dimensional array of SPE's interconnected by inhibit and readout buses. An inhibit bus connects SPE's in a row for the purpose of maximum finding and a readout bus connects SPE's in a column for data readout. We shall see later that in an alternative design the readout bus is made horizontal, but this does not affect the design of the SPE itself.

FIGURE 6.1.



The NSIP sensor cell design. Circuit diagram for the diode and comparator (left). Voltages as a function of time and intensity (right).

FIGURE 6.2.



A complete single NESSLA SPE. The inhibit and readout buses are precharged high.

The integration starts when the photo diode capacitors in the SPE's have been precharged, after which they discharge proportionally to the incoming light intensities. When a photo diode voltage reaches the reference voltage its comparator output switches from low to high, and the inhibit bus goes low from its precharged high. Thus, at this moment all comparator outputs in the row are forced to freeze their potentials at the capacitor C. Interpreted as binary signals these values are *zero* except for the cell with the highest intensity and which took command over the inhibit bus. This cell has attained the value *one*.

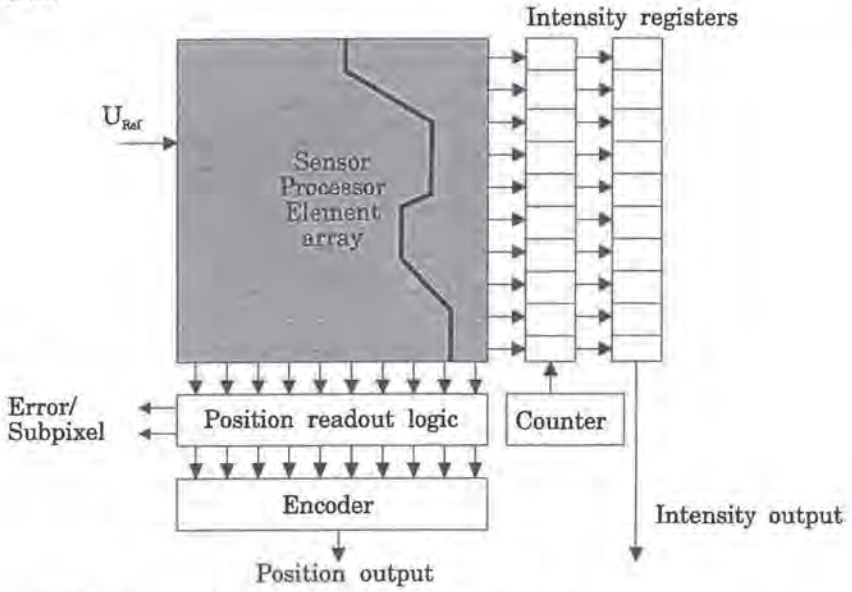
When the transfer transistors are enabled the binary data are moved from the first inverter to the second, and a new integration can begin in the photo diodes. The readout is then made concurrently with the next integration cycle in the sensor array.

We notice that the NESSLA SPE of Figure 6.2 so far is only capable of producing a maximum intensity *position* indication and we will see shortly how this is taken care of to obtain range data. However, already now we may also notice that if we would be able to record the point in time *when* the inhibit bus switches to low in a certain row, this information should be possible to translate to intensity. Hence, the NESSLA cell harbors the same possibilities for intensity *and* range data fusion as those presented in previous chapters.

6.4 Global architecture

The global architecture presentation is split in two parts; the position readout and the intensity readout respectively. An overview of the chip layout is shown in Figure 6.3. From the 2D-array of SPE's binary position data is read one row at a time, and the binary code of the position of the single *one* is obtained with an encoder. The encoder is preceded by the *position readout logic*, which inhibits all but a single *one* in a way defined below. Intensity values are read out to the right and stored in the intensity registers during integration. They are then read from the intensity registers concurrently with the position readout. As already indicated above, what is called intensity here is really time intervals inversely proportional to intensity.

FIGURE 6.3.



The NESSLA chip layout. Max position data are read out downwards from the sensor and intensity data are read out to the right.

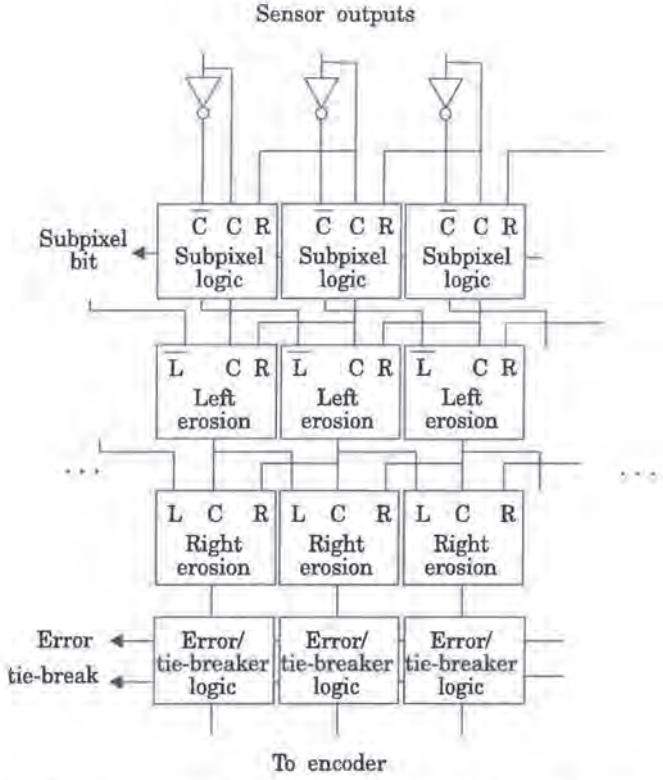
The SPE row is designed so that in most cases the resulting binary position vector contains only a single *one* which corresponds to the position of the maximum intensity. However, if several SPE's in a row are subjected to the same or nearly the same intensity (possibly because of a thickened, unfocused sheet-of-light or because of multiple reflections) more than one SPE could obtain a binary *one* output. Hence, some binary post-processing is necessary to obtain a clear-cut position of the maximum intensity.

Another case which must be accounted for is the case where no maximum intensity exists in the row, or rather when all intensities are so low that none of them are able to bring the photo-diode voltage down to U_{ref} in the allotted integration time, see Figure 6.1.

If the output from a SPE row contains one consecutive run of ones the correct output ought to be the mid-point value of this run. Ideally, the vector is expected to have one or maybe two ones if the light is focused and the object is a good reflector. We have chosen to accept up to three ones as an acceptable run while more than three consecutive ones will be indicated as an error. This rule comes from empirical experiments using MAPP2200 [29], [35].

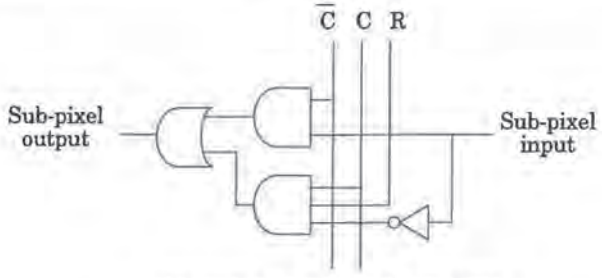
The position readout logic is pipelined into four stages, see Figure 6.4. The first operation on the output vector is a subpixel ripple chain. This is used to produce a sub-pixel position bit which extends the precision of the final position vector with an extra least significant bit, see Figure 6.5. The sub-pixel output is set to *one* if the sensor output vector has two consecutive ones. In such a case the maximum position is more likely to be the mid-point between these two pixels than the position of the leftmost or the rightmost of the two. This is exactly what the extra bit signifies.

FIGURE 6.4.



The position read-out logic pipeline. C stands for center pixel, L for left neighbor and R for right neighbor.

FIGURE 6.5.

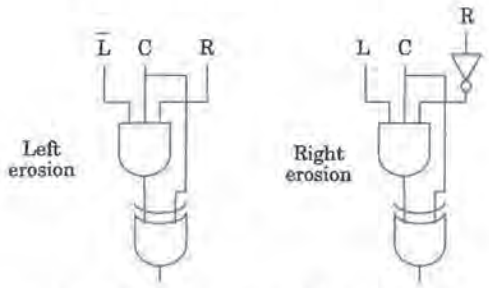


The sub-pixel ripple chain logic.

The second and third operations are one-dimensional erosion from the left and the right respectively, implemented with the logic seen in Figure 6.6. These stages erode two and three consecutive *ones* to a single *one*.

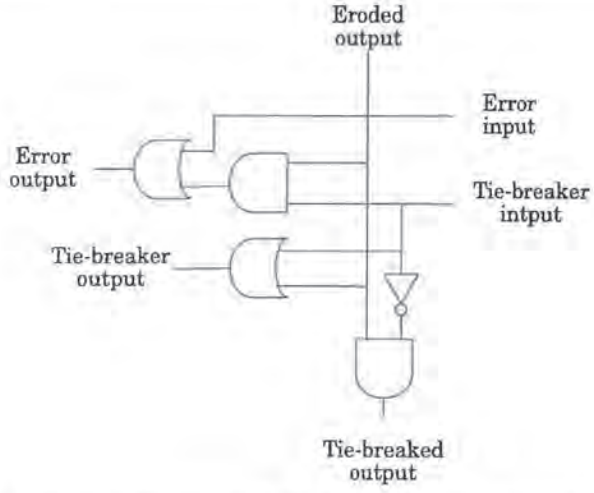
The last pipeline stage is the error indicator and tie-breaker logic, seen in Figure 6.7. The error indicator checks if there is more than one bit set to *one* after the erosion stage. Thus, the error bit will be set if there are more than three consecutive *ones* or more than one run of *ones* in the row read-out from the sensor. The tie-breaker ripple chain blocks all eroded outputs except the rightmost *one* to ensure proper function of the subsequent encoder in Figure 6.3. Hence, if there are more than a single *one* in the input vector to the tie-breaking logic it nevertheless gives the encoder a chance to deliver a properly coded position response. At the same time, potential multiples of *ones* in the input are indicated by the error output and the programmer can therefore decide wether the possibly flawed position value should be used or not. If no column position data is set to *one* the tie-breaker chain will deliver a *one* in the last position as a default. Hence, the encoder will always have a well defined output.

FIGURE 6.6.



One-dimensional erosion from the left and from the right. Erosion from the right is only performed if more than two consecutive ones are found.

FIGURE 6.7.

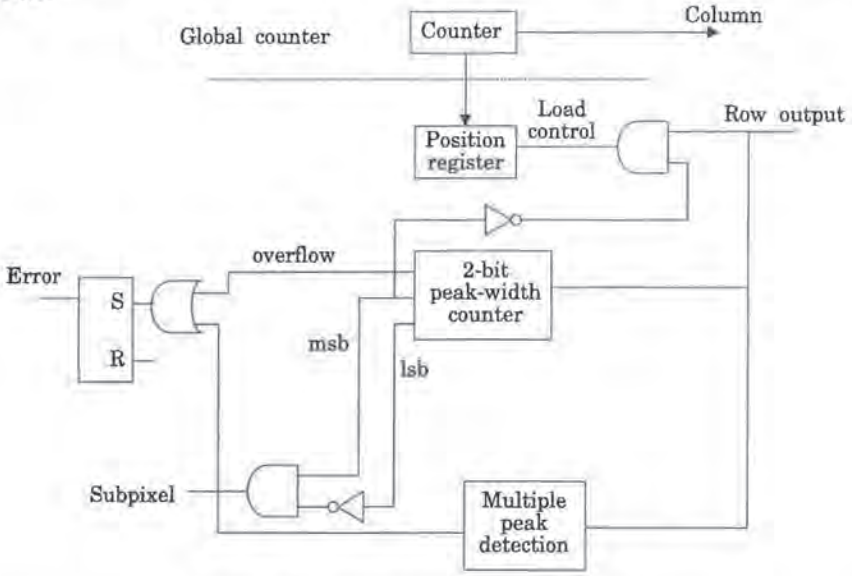


Error indicator and tie-breaker ripple chain logic. The error output is one if there is more than one bit set to one after the erosion stage.

Since the position readout logic has a gate-depth which is at least as large as the width of the sensor it is hard to use high clock frequencies. We must set aside time for the signals to ripple through the subpixel, error and tie-breaker ripple chains. One way to allow higher clock frequencies is to read out the position values column-wise and replace the position encoder by a serial peak detection in a manner that is vaguely similar to the column readout sensor in section 5.2. In fact, in the present case the peak detection is digital (binary) and therefore much easier to implement. The error and subpixel logic are already given as iterative networks and therefore ready to be implemented in a serial fashion.

One such column-wise readout circuit can be seen in Figure 6.8. To ensure that the position register receives a valid information, loading is only allowed twice. Thus a three pixel wide peak will get the correct position value. Error is indicated if more than three *ones* are found, that is when the two-bit width counter overflows, or when multiple peaks are detected.

FIGURE 6.8.



Serial column-wise position readout logic.

As pointed out before the time from the start of the integration to the activation of the inhibit bus is inversely proportional to the light intensity in the maximum intensity position. This intensity depends on the reflectance (brightness) of the surface illuminated by the sheet-of-light. Thus, a full reflectance map, an intensity image, can be obtained by saving these points of time for each row. To this effect the intensity registers in Figure 6.3 are utilized as follows.

A counter is started when the sensor diodes are precharged, and when the inhibit bus of a certain row is triggered the corresponding register is loaded with the counter value. These registers are then addressed simultaneously with the row output so that both the position and its intensity value are output concurrently. To be able to use this design together with the integrations, the registers have to be doubled so that one set of registers can be loaded from the inhibit bus while the other set is used for output.

6.5 NESSLA implementation

The implementation discussion is based very much on the results of two general NSIP chip prototypes presented in [13] and [14]. The two prototypes have 16x16 and 32x32 SPE cells respectively.

To achieve high sensor aperture the sensor should cover a large part of the area. However, to be able to implement the SPE logic effectively the area of the photo diode must be reduced. This is possible while actually increasing the aperture if we apply focusing micro lenses on top of the chip [15], [42], [58]. Such a lens is able to increase the effective light sensing area from around 1% to almost 100%. Other possible solutions are 3D VLSI processes [56] or thin-film amorphous silicon sensors applied on top of the chip [16]. The use of micro lenses is currently being tested on a prototype chip.

The comparator must be compact, use low power and be compensated against fabrication variations. The low power requirements makes it impossible to use standard current mirror comparators [26] and the proposed solution is to use an inverter with some sort of offset compensation [31], [13], [14]. The fabrication variations gives different threshold voltages for all transistors, and thus different thresholds for different inverters. The compensation can, for instance, be implemented by short-circuiting each inverter before the pre-charge, driving each inverter to its own individual threshold voltage.

Using the above comparator solutions in a 0.8 µm process a *general* 32x32 NSIP chip has been designed with 118x118 µm SPE size. The expected clock frequency is up to 100 MHz. For a NESSLA design the SPEs can probably be implemented on a 60x60 µm area, making it possible to implement a 256x256 SPE chip on a 15x15 mm chip [12]. The 16x16 prototype NSIP chip has been programmed for sheet-of-light range imaging, and an example of the results can be seen in Figure 6.9. For a 128x128 chip the predicted performance is 500000 processed frames per second, giving 64 MHz rangel frequency [14], [13]. However, if we consider the integration time the *NxN* range image speed for an *NxN* NESSLA sensor is given by the equation

$$T_{im} = N \cdot Max\{T_t, \frac{N}{f}\} \tag{6.1}$$

Hence a 256x256 SPE sensor array, 10 µs integration time and 100 MHz processing speed this gives a range image speed of 390 Hz, and a 25 MHz pixel frequency for both range and intensity values.

FIGURE 6.9.

An example of a sheet-of-light range profile acquired with a 16x16 general NSIP prototype. Left shows maximum finding results, mid after erosion and right after tie-breaking. Here the inhibit bus, left-right erosion and tie-breaker is implemented on the SPE array using general NLU and GLU operations. The grey levels of the pixels indicate the reflected brightness.

Chapter 7

MAPP2200 as range image sensor

7.1 MAPP2200 architecture

The smart sensor MAPP2200 [40], [17], is a 256x256 photo diode sensor array combined with a linear array of 256 A/D converters and 256 bit-serial processing elements (PE's) on one chip, see Figure 7.1. Each PE has an eight-bit A/D converter output register, an eight-bit parallel shiftregister, 96 bits of memory and an ALU, all connected via a one-bit bus. The A/D converters have direct outputs to the A/D registers so that pixel-row A/D conversions can be made independent of, and parallel with, the PE processing. The shiftregister can also run autonomously so that row transfers to/from MAPP2200 are independent of the processing. Each ALU has a Point Logic Unit (PLU) for 1-3 input logic operations, a Neighborhood Logic Unit (NLU) for left-right neighbor dependent binary operations and a Global Logic Unit (GLU) which sees MAPP2200 as one 256 bit wide processor. The sum of the accumulator contents of the ALUs (the number of *ones*) is called COUNT and can be read via MAPP2200s status register. The Global Or (GOR) output is *one* if the contents of the accumulator register is non-zero.

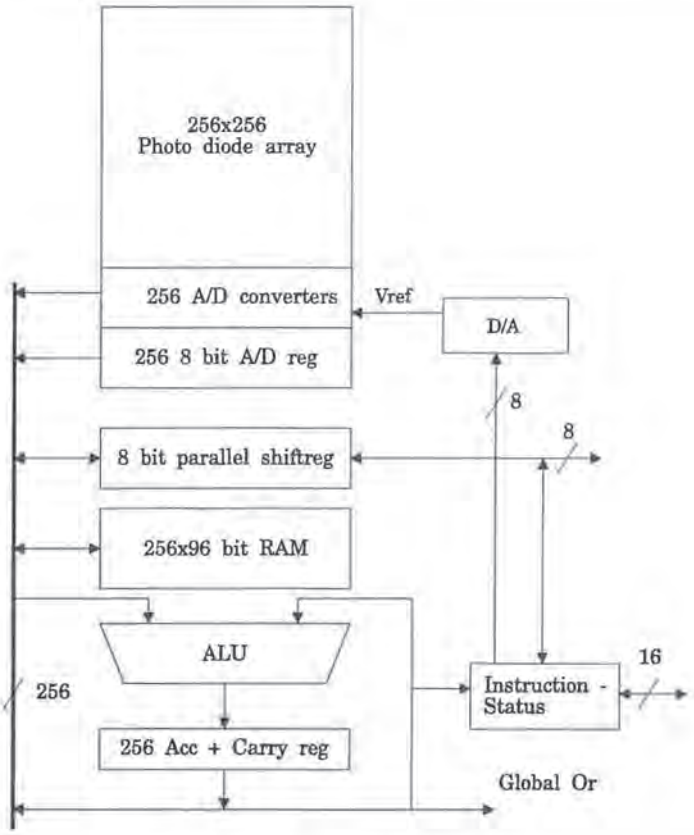
The row parallel Single Instruction stream Multiple Data stream, SIMD, PE array in MAPP2200 can be used to process the sensor data with a variety of low level image processing algorithms. For algorithms such as gradient (Sobel), high-pass (Laplace), low-pass and median filters video rate processing with eight-bit data precision is possible with 3x3 kernels. In [69] MAPP2200s performance for many low level algorithms are listed. Bit-serial addition of b *by* b bit words requires $3b$ operations and an increment of a b bit word requires $2b$ operations. However, if we use a Gray coded counter we can implement an increment operation with only an exor operation in 2 clock cycles. A conversion from Gray code to ordinary binary representation requires $2b$ operations.

During normal sensor row readout the read transistors are only activated a short time to allow the photo diode voltage to be polled, see Figure 7.2. Then as long as the integrate transistors are activated the pixel charge is stored in the capacitance C. V_{ref} changes linearly from 0-5 V (0-255) via a counter and an internal D/A converter. The D/A counter value is stored when the PD-output switches from zero to one. The A/D converter resolution is programmable, so that any A/D conversion precision b between 1 and 8 bits can be used. A b-bit A/D conversion requires 2^b clock cycles.

A special mode allow MAPP2200 to be used for Near-Sensor Image Processing. The processor capacity is hardly sufficient to give NSIP support to the full 2D-array of sensor elements. However, it is possible to repeatedly interrogate the binary comparator outputs while integration is going on in one row of pixels. The readout stage of one MAPP2200 pixel column can be seen in Figure 7.2. One row of sensor elements are activated by one row of read transistors. By keeping both the read and integrate transistors *on* and V_{ref} constant the comparator output (PD-out) is the NSIP cell output for the activated sensor row. Thus, we can emulate NSIP architectures on a row by row basis. One thing to note about the sensor layout is that the sensor rows are grouped two and two, i.e. the distance between consecutive rows alternate between

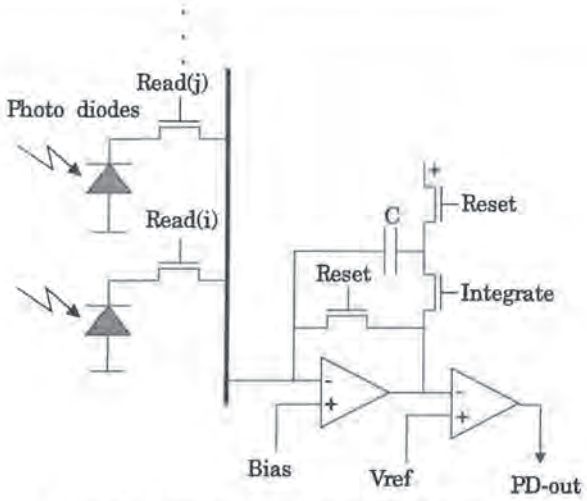
two different values. This will affect the performance of some algorithms. Also, the pixel aperture, i.e. effective sensor area, is only around 50%.

FIGURE 7.1.



Block diagram of MAPP2200 chip.

FIGURE 7.2.



Sensor column readout design on MAPP2200.

The NLU/GLU operations in MAPP2200 to be used in the sheet-of-light implementations are demonstrated in Figure 7.3. The NLU operation LEDGE is used to find the number of left edges, that is, the number of separate runs of ones, in a binary vector. We see that after the GLU operation LFILL the number of ones in the accumulator is the same as the position of the leftmost one in the original binary row. This number is the COUNT value. The MARK operation operates on two vectors, *image* and *mask*, and objects in image vertically connected to any set bit in mask are kept. The MARK operation can for instance be used to extract single objects from a binary vector (if the *mask* register is the result of an LFILL and a LEDGE operation the leftmost object will be extracted). The NLU operations LSHIFT and RSHIFT shift the binary vector one PE left or right respectively. This can be used in convolutions since it can be used to input data from the left/right neighbor to a PE. Thus it is easy to implement 3x1 convolution kernels using the NLU.

FIGURE 7.3.

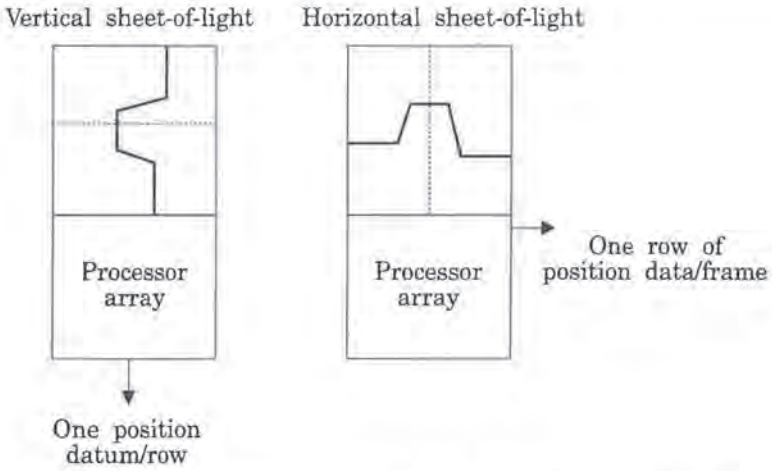
The NLU-GLU functions used in this chapter.

MAPP2200 is connected to a controller, normally a PC, which writes instructions to the camera and reads camera output data. A special semi-autonomous video generator can be connected to the shiftregister output so that the camera can generate a video signal. This unit also contains four external vram image memories connected to the shiftregister. MAPP2200 is also available as a *smart camera* which contains a MAPP2200 chip and an embedded 186 PC processor in a compact package. This camera can be used as a stand-alone image processing system. The smart camera communicates via four binary IO channels and a serial PC-port.

MAPP2200 is designed for a maximum clock frequency of 4 MHz, but our experiments have shown that clock frequencies up to 8 MHz is possible on selected chips. The maximum A/D conversion and shiftregister clock frequencies possible are 23 and 10 MHz respectively [27].

MAPP2200 can be used for sheet-of-light range imaging either with the light-sheet projected horizontally or with the light-sheet projected vertically on the sensor. Both ways are illustrated in Figure 7.4. When the sheet-of-light is vertical the PEs are used as one 256-bit processor to find the position of the sheet-of-light impact position of each row in a serial fashion. When the sheet-of-light is horizontal the PEs are used as 256 independent bit-serial processors, where all PEs find their own sheet-of-light impact positions in their respective columns in parallel. Thus, with a vertical sheet-of-light each sensor row processing gives one position value, whereas with a horizontal sheet-of-light each processor is expected to deliver one position datum after processing of all sensor rows.

FIGURE 7.4.



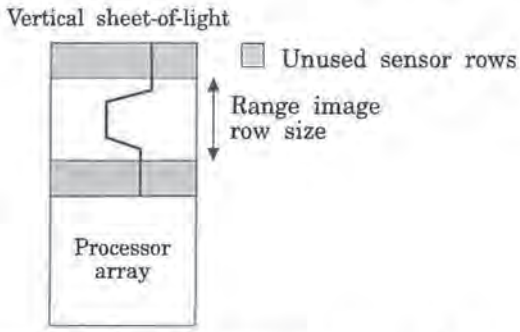
Vertical and horizontal sheet-of-light range imaging on MAPP2200.

In the pseudo code examples given we use syntax similar to the programming language C. All MAPP2200 specific instructions are written in capitals, and MAPP2200 macros start with a capital letter. All other instructions are executed by the controller.

7.2 Trading resolution for speed

The programmability of MAPP2200 makes it possible to generate range images of arbitrary size or range resolution. With a vertical sheet-of-light, range images with less than 256 samples per range image row are obtained by only utilizing a fraction of the sensor rows, see Figure 7.5. This does not change the range pixel frequency, and thus the range image frequency increases. For instance, if we generate 128^2 range images instead of 256^2 ditto the image frequency increases with a factor 4. The readout can also easily be interlaced, so that all sensor rows are utilized, but only every second integration [31].

FIGURE 7.5.

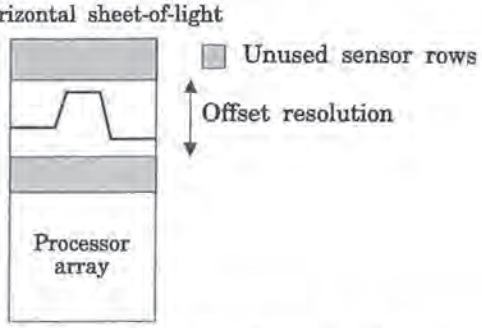


Increasing the range image speed by decreasing the image size.

With a horizontal sheet-of-light it is possible to *increase* the range pixel frequency by limiting the offset position resolution to a fraction of the sensor, see Figure 7.6. This can be made in two ways, either by using maximum accuracy over a smaller range interval, or by using lower accuracy over the maximum range interval. If we only process a fraction of consecutive rows we obtain maximum accuracy over a fraction of the range interval. In MAPP2200 it is

also possible to address several sensor rows concurrently and read out the analog sum of the rows to the A/D converters. If we process, say, the analog sum of pairs of rows we obtain range images with half of the maximum range resolution over the whole range interval.

FIGURE 7.6.



Increasing the range pixel speed by decreasing the range resolution.

7.3 Vertical NSIP algorithm

MAPP2200 can be programmed to emulate the NSIP maximum finding algorithm in the NESSLA architecture (see chapter 6). The performance is of course not as good as in NESSLA since each row of sensor data must be processed in series and not in parallel.

The principle is to integrate each row until the first pixel charge reaches the preset threshold. The pixel which first reaches the threshold is the one which received the highest intensity. Thus, after the maximum finding loop the binarized sensor row vector has a *one* at this PE position. The position of the *one* is then found using the GLU function LFILL. In pseudo code the program can be written

```
   /* NSIP sheet-of-light algorithm on MAPP */
SETV 20                 /* An appropriate threshold */
for (i=x;i<y;i++) {     /* Loop N = (y-x) rows */
   SETR i               /* Set sensor row to read */
   j=100                /* Reset integration counter */
   RESET                /* Reset PD register */
   INTEG                /* Start integration */
   LDI PD               /* Load comparator output */
                        /* Loop until threshold is reached */
   while ((GOR == 0) && (j > 0)) {
       j--
       LDI PD}          /* j = 0 Indicates no range data */
   width[i] = COUNT     /* Width of maxima */
   LD LEDGE Acc         /* Find no of maxima */
   err[i] = COUNT       /* Read error value */
   LFILL Acc            /* Find leftmost pixel pos */
   pos[i] = COUNT       /* Read position values */
   int[i] = j           /* Read intensity */
}
```

The algorithm output consists of four values for each sensor row, the leftmost position of the maxima (*pos*), the maxima width (*width*), the error value (*err*) and the counter value corresponding to the maximum intensity (*int*). The *width* value is used to compensate for the maxima width using the formula

$$pos[i] = pos[i] - \frac{(width[i] - 1)}{2} \tag{7.1}$$

Thus, we reduce the *pos* value with half the maxima width to the mid maxima position value and get half pixel sub-pixel accuracy. If the *err* value is larger than one we know that there were more than one maxima, and that the *pos* datum is uncertain. If we use up to 100 NSIP interrogations for each sensor row the possible range pixel frequency is 15-30 KHz, which corresponds to frame rates of 0.2-0.5 Hz for the image size 256^2.

Unfortunately, the design of MAPP2200 makes it hard to use for NSIP range imaging. This is because the read transistors attached to the column readout bus also act as photodiodes, see Figure 7.2. When we use MAPP2200 for NSIP processing we activate one row of read transistors and activate the integrate transistors in the read-out circuits. This opens a path for charge transport from the capacitor C, via the read bus to the photo diode. However, additional charge is leaking from C via the parasitic photo diodes attached to the bus. This gives the result that if high intensity reaches several sensors in the same column this light might discharge its column capacitor C more than the higher but single intensity can do in another column in the same row. Since this leakage increases the generated intensity levels using the intensity as a confidence measure often fails. As a whole, the leakage effect often leaves errors which are hard to detect and remedy, such as contours getting expanded over occluded areas [31]. The only real remedy would be to redesign the read-out circuitry. The leakage effect is also visible in some of the other algorithms we have implemented, but since the time the analog PD register is connected to the bus without a reset is shorter the effect is smaller. Examples of images obtained using this algorithm can be found in [29] and [31].

7.4 Vertical successive approximation algorithm

This implementation uses a successive approximation algorithm previously described in [29], [37], [35] and [31] to find the maximum intensity in each row. The intensity is converted to an eight bit integer. The successive approximation is done by first setting the threshold to the middle of the interval, i.e the most significant D/A bit set to *one* ($10000000_2 = 128_{10}$). If any PE has a pixel value above this threshold we know that the maximum intensity is higher and we keep that D/A bit set to *one* and then test again with the next D/A bit set to *one* ($11000000_2 = 196_{10}$). If no PE has a pixel above the threshold we know that the maximum intensity is lower, and we reset that D/A bit to *zero* and test again with the next D/A bit set to *one* ($01000000_2 = 64_{10}$). Thus, we half the possible maximum intensity interval at each iteration and after eight iterations we obtain the maximum intensity for that sensor row. If we threshold the row with the found maximum we should, normally, obtain a binary vector with a single *one* at the position of the maximum. The position of the *one* is then found using the GLU function LFILL. In pseudo code the program can be written

```
        /* Vertical Successive approximation */
        /* Sheet-of-light algorithm */
SETR i                              /* Start row */
for(i=x;i<y;i++) {                  /* Loop over N = (y-x) rows */
  INTEG                             /* Read pixel to PD reg */
  HOLD                              /* Hold pixel value */
  volt = 128                        /* Initial approximation */
  for(j=7;j>=(8-b);j--) {           /* Succ. app. loop */
      SETV volt                     /* Set A/D voltage */
      LDI PD                        /* Load binary row */
      if GOR                        /* If "hit" */
       ST PD,R(0)                   /* Save binary vector */
      volt -= ((1-GOR)<<j)          /* Change volt */
      volt += (1<<(j-1))            /* Set next bit */
  }
  LD R(0)                           /* Load best approx vector */
  width[i] = COUNT                  /* Read width value */
  LD LEDGE ACC                      /* Detect left edges */
  err[i] = COUNT                    /* Read number of maxima */
  LFILL ACC                         /* Fill from leftmost one */
  LD GLU                            /* Read fill result */
  pos[i] = COUNT                    /* Read position value */
  RESET,+                           /* Reset PD/pixel, new row */
  int[i] = volt                     /* Read intensity value */
  }
```

The time to acquire one *NxN* range image is approximately

$$T_{im} = \frac{N^2(4b+10)}{f_p} \tag{7.2}$$

where b is the number of steps in the successive approximation loop. Fewer steps in the loop gives higher ranging speed, but the accuracy of the range data might decrease. Also, the resolution of the intensity data decrease. With $b=8$ the maximum range data frequency is 190 KHz, and the 256^2 image frequency is 2.9 Hz. With $b=4$ the corresponding figures are 308 KHz and 4.7 Hz. With a dedicated PAL controller we reached approximately 160 KHz rangel frequency [35], [31]. Without the error and peak width filtering the speed increase around 10%. In chapter 4 peak finding using thresholding was found to be slightly less noise sensitive than maximum finding. It might therefore be worthwhile to threshold the peak with for instance half the peak intensity instead of with the peak intensity. This actually increases the algorithm speed, since the store of the binary vector in the inner loop can be omitted, but in our tests no accuracy improvement was detected. The store of vectors in the inner loop is made to remove errors which occurred when a final thresholding at the maximum found intensity gave only zeros due to analog noise in the readout circuitry [31].

An example of the combined range and intensity output of the successive approximation algorithm can be seen in Figure 7.7.

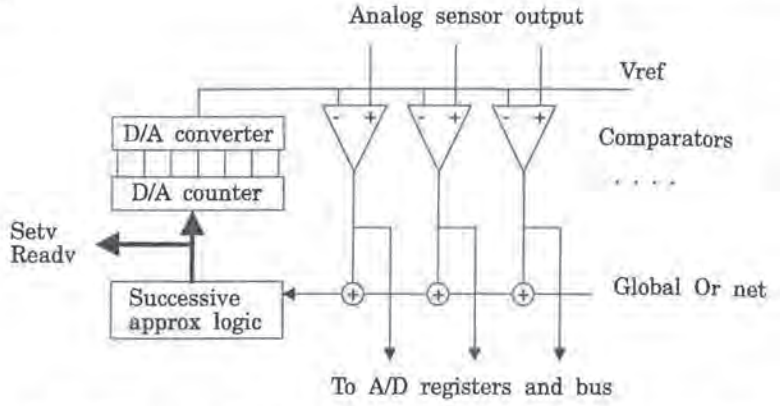
FIGURE 7.7.

A 256x256 range image plotted as stacked height curves and with the intensity information as plot intensity.

Improvements in MAPP2200 architecture

If the successive approximation loop could be performed without transfers to the accumulator the inner loop of the algorithm could be speeded up with a factor of at least three since only one instead of three-four instructions would be needed. This implies that a Global Or net, wired to the comparator outputs and input to a small sequencer to control the A/D converter, is added, see Figure 7.8. To make it possible to read out the D/A counter value (the maximum intensity) another modification is also needed. These modifications should not be too difficult to implement since the existing automatic A/D conversion net is rather similar. This modification should be useful in other MAPP2200 applications such as finding the brightest pixel over the whole sensor.

FIGURE 7.8.



A modified A/D converter design with automatic maximum detection.

7.5 Vertical thresholding algorithm

The pseudo code for an implementation of a vertical thresholding algorithm on MAPP2200 can be written

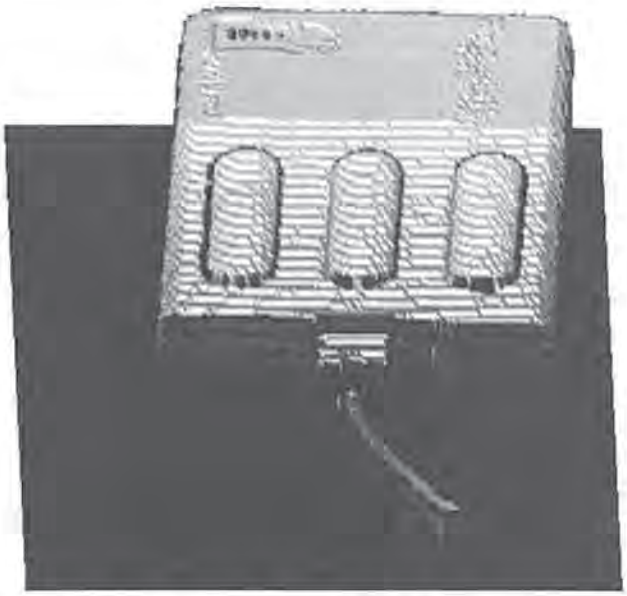
```
        /* Vertical thresholding algorithm */
SETV volt               /* Set A/D threshold voltage */
SETR i
for(i=x;i<y;i++) {      /* Loop over N = (y-x) rows */
   INTEG                /* Read analog sensor row */
   LDI PD               /* Load thresholded row */
   RESET,+              /* Reset PD/pixel, new row */
   width[i] = COUNT     /* Read the peak width */
   LD LEDGE Acc         /* Detect no of peaks, error */
   err[i] = COUNT       /* Read the error val */
   LFILL Acc            /* Fill from leftmost one */
   LD GLU               /* Read fill value */
   pos[i] = COUNT       /* Read position value */
}
```

The time to acquire one *NxN* range image is approximately

$$T_{im} = \frac{9N^2}{f_p} \tag{7.3}$$

If we use a clock frequency of 8 MHz we achieve approximately 880 K rangels per second (an image frequency of 13 Hz) with error/peak width correction and 1.3 MHz without. A complete range image acquired with the thresholding algorithm can be seen in Figure 7.9.

FIGURE 7.9.

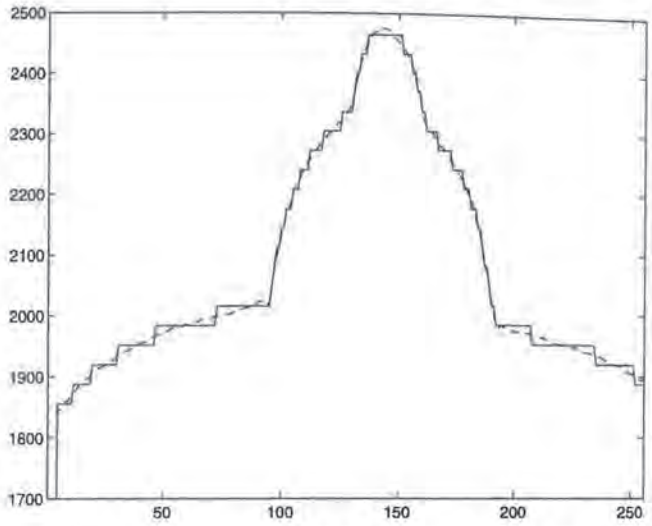


A 3D plot of a complete range image of a computer mouse acquired with the thresholding algorithm.

7.6 High accuracy combination algorithm

By augmenting the successive approximation maximum finding algorithm with a number of thresholdings, the extra thresholds selected uniformly between the found maximum and the supposed noise level, we obtain higher accuracy. Each extra thresholding gives a new range value estimate, and by averaging all estimates we can get higher accuracy than with only a single maximum finding or thresholding. Of course the range imaging speed decreases, but in some applications high accuracy is more important than high speed. We feel that the combination of maximum finding and thresholding is better than to just use several pre-set multiple thresholds since we always use the whole available signal. We applied this technique using 5 extra thresholds, and a resulting combination profile with 14-bit resolution and the maximum-finding profile with 9-bit resolution can be seen in Figure 7.10.

FIGURE 7.10.

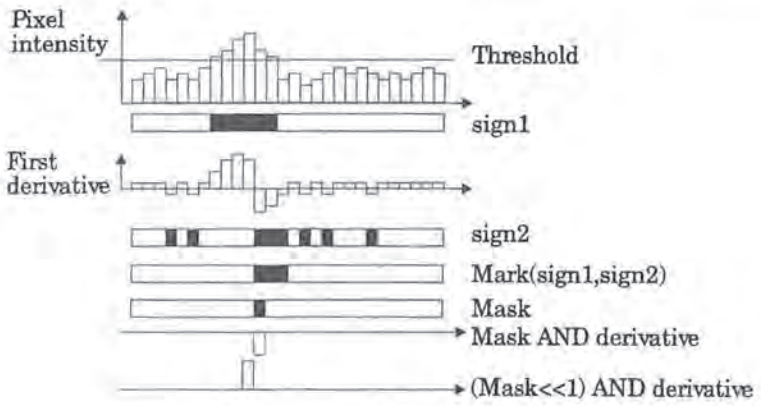


A comparison of maximum finding (solid) and maximum finding combined with multiple thresholds (dashed). The profile depicts a cross-section of a jet-fighter model.

7.7 Vertical first-derivative method

In [7] a peak detector based on the position of the zero-crossing of the first derivative was presented. This is a similar implementation on MAPP2200. We use a thresholding algorithm to find the initial peak position and we use the NLU and GLU to read out the intensity values at the zero crossings of the first derivative, see Figure 7.11. Then we interpolate a sub-pixel value from the first derivative.

FIGURE 7.11.



The peak detection in the vertical first-derivative method using the GLU/NLU.

In our implementation we use a 5x1 Sobel-type binomial filter to approximate a smoothing first derivative operator and in pseudo code the algorithm looks like

```
/* First derivative peak-detection algorithm */

for (i=x;i<y;i++) {                  /* Read N = (y-x) rows */
   pix = ReadAD                      /* Read A/D converter b */
   Sub(pix,thresh,sign1)             /* save only sign, 2b */
                                     /* Derivative filter */
   Add(pix,left_pix,tmp)              /* 1 1 kernel 3b */
   Add(tmp,right_pix,tmp)            /* 1 1 -> 1 2 1 */
   Add(tmp,left_pix,tmp)             /* 1 1 -> 1 3 3 1 */
   Sub(tmp,right_pix,tmp)            /* 1 -1 -> 1 2 0 -2 -1 */
                        /* sign of tmp = sign2, possible pos */
                        /* Find zero crossings within peak */
   MARK(sign1,sign2)                 /* 2 clock */
   LFILL
   pos[i] = COUNT                    /* Read position of peak */
   LEDGE                             /* Mask for positive val */
   ST MASK                           /* Store mask */
   for (j=b;j>0;j--) {               /* Read nval, 3b */
       LD MASK
       AND der(j)                    /* Mask out bit */
       nval += GOR<<j                /* Read bit to controller */
   }
   LD (LSHIFT) MASK                  /* Pos for positive val */
   Read(pval)                        /* Read positive value 3b */
}
```

The sub-pixel peak position is then found as

$$pos[i] = pos[i] - \frac{nval}{pval - nval} \tag{7.4}$$

note here that nval is negative. Since we need time to A/D convert the next row while processing the present one the time for an *NxN* image is approximately

$$T_{im} = N^2\left(Max\left(\frac{2^b}{f_{AD}}, \frac{21b+7}{f_p}\right) + \frac{b}{f_p}\right) \tag{7.5}$$

This gives a rangel speed of approximately 44 KHz if *b* is eight. With a 3x1 derivative filter the corresponding speed is 67 KHz, but it will also be more sensitive to noise. The range frequencies obtained are not very high, but on the other hand this algorithm is, as was shown in section 4, capable of high accuracy even under noisy conditions.

7.8 Horizontal maximum finding algorithm

A horizontal maximum finding algorithm over *a* rows is obtained by the following

```
/* Horizontal maximum finding sheet-of-light
   algorithm over part of the sensor */
/* Position counter is Gray coded */
pos = 0                              /* Reset counter; loga */
Max = 0                              /* Reset max; loga */
for (i=x;i<y;i++) {                  /* Read a = (y-x) rows */
  pix = readAD                       /* Read A/D converted row b */
  if (pix > Max){                    /* Test for maximum 2b */
```

```
        Max = pix                    /* Save new maximum 3b */
        pos = rowno                  /* Save position 3loga */
    }
    rowno++                          /* Increment pos 2 */
}
Convert(pos)                         /* Convert from Gray code */
Move(pos,Sreg)                       /* Readout pos loga */
SAVE                                 /* Shift out position */
Move(Max,Sreg)                       /* Readout Max b */
SAVE                                 /* Shift out intensity */
```

Since we need time to A/D convert the next row while processing the present one the time for an *NxN* image with the range resolution a is approximately

$$T_{im} = NMax\left(\frac{N}{f_s} + \frac{\log a}{f_p}, aMax\{\frac{2^b}{f_{AD}}, \frac{6b + 3\log a + 2}{f_p}\} + \frac{b + 2\log a}{f_p}\right) \tag{7.6}$$

If we choose $a = 64$ and $b = 8$ the A/D conversion sets the speed limit with an image frequency of 4.75 Hz. The range and intensity pixel frequency is approximately 310 KHz. If we evaluate equations (7.3) and (7.6) we note that if the range accuracy needed is more than 6 bits the vertical algorithm is faster, but for 6-bit accuracy and less the horizontal algorithm is preferable. Note that if the intensity as well as the range is to be read out the limit-frequency is halved (~ 5MHz), and that no error and peak width corrections are implemented.

7.9 Horizontal thresholding algorithm

In this implementation, previously described in [34], [32] each PE finds the row position of the first pixel above the threshold (= the start position of the peak) by counting the number of rows before any pixel is above the threshold, and the row position of the last pixel above the threshold (= the end of the peak) by counting the number of rows before the end of the first peak is found. The peak position is then computed with half pixel sub-pixel accuracy as the sum (mean) of the two counters.

In pseudo code the horizontal thresholding algorithm is described as follows

```
        /* Horizontal thresholding sheet-of-light algorithm */
SETV volt                            /* Set threshold voltage */
Count1 = Count2 = 0                  /* Reset counters 2logn */
START = 1                            /* Reset Mask bits */
ISTOP = 0
ERR = 0
SETR x                               /* Set row */
for (i=x;i<y;i++) {                  /* Read a = (x-y) rows */
  INTEG                              /* Read analog row value */
  LD PD                              /* Read inverted row */
  RESET,+                            /* Reset PD, new row
  ST Rx(i)                           /* Store inv. row */
                        /* Peak start counter */
  AND START
  ST START
  XOR Count1(i)                      /* Change counter bit */
  ST Count1(i)                       /* Counter Gray coded */
                        /* Peak end counter mask */
```

```
    LD Rx(i)                        /* Load invert row */
    AND Rx(i-1)                     /* And prev row */
    OR ISTOP                        /* Or with stop */
    ST ISTOP                        /* Save stop */
                        /* Error filtering */
    AND Rx(i)
    OR ERR
    ST ERR
                        /* Increment end counter */
    LDI ISTOP                       /* Read count 2 cond */
    XOR Count2(i)                   /* Change counter bit */
    STORE Count2(i)
}
Convert(Count1)                     /* from Gray 2loga */
Convert(Count2)                     /* from Gray 2loga */
Add(Count1,Count2,Sum)              /* add 3loga */
Move(Sum,Sreg)                      /* Load sreg loga */
sreg(8) = ERR                       /* Load err bit to sreg */
SAVE                                /* Shift out row */
```

The START bit in each PE is used to detect the beginning of a peak. If a peak is found it will be set to *zero* for the duration of the sensor readout. When the START bit is *zero* no start increment of *Count1* is possible in this PE and *Count1* will retain the peak start position value. The ISTOP bit in each PE is used to detect the end of a peak. If the end of a peak is found ISTOP is set to *one* for the duration of the sensor readout. When ISTOP is *one* no increment of *Count2* is possible and *Count2* will retain the peak stop position value. In the code we output the error value, but it is also possible to make the error filtering in MAPP, i.e. use the error bit to AND with the data bits. This requires approximately log*a* cycles.

Since we need time to shift out the result of the previous sensor processing while processing the previous the time for an *NxN* (N = 256) image with the range resolution 2log*a* is approximately

$$T_{im} = N\left(Max\{\left(\frac{N}{f_S} + \frac{1+\log a}{f_p}\right), \left(\frac{18a}{f_p} + \frac{9\log a}{f_p}\right)\} \right) \tag{7.7}$$

For n = 128 (8 bit range accuracy) the range image frequency is 13 Hz and the rangel frequency 860 KHz, i.e. the speed is very similar to the vertical implementation. However, for lower range accuracy than 8 bits the horizontal algorithm has higher speed. For instance, if the range accuracy is 6 bits the rangel frequency is 3.3 MHz. Also, when the accuracy is less than 5 bits it is possible to output 2 rangels during each 8-bit shift, so the rangel frequency can actually increase above the 10 MHz shiftregister transfer frequency.

7.10 Horizontal thresholding with USM

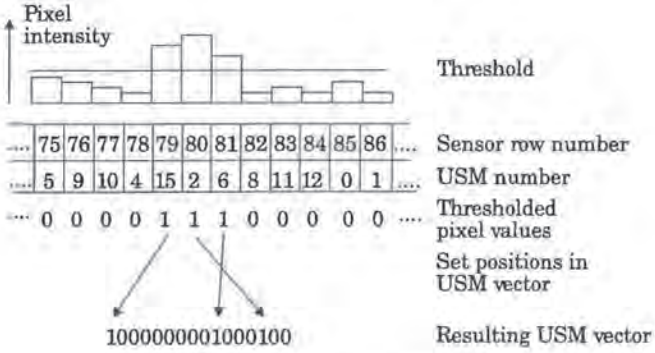
The Unique Sequence Method, USM, was originally described by Åstrand et. al. in [66] and [67]. Although it is technically a horizontal thresholding algorithm, the USM implementation is very different from the previous algorithms, giving radically different performance.

To understand how the USM algorithm works, study Figure 7.12. With each sensor row we have associated a *USM number*, say between 0 and 15. The sequence of USM numbers is called the *USM sequence*. How the USM sequences can be generated will be discussed below.

Now, if we threshold a peak which covers n consecutive pixels we thereby select n USM numbers from the sequence. Due to the characteristics of the USM sequence the combination of USM numbers thus selected is unique, and can be used to determine the peak position.

When implementing USM the USM numbers are represented by bit positions in a 16 bit word, called the *USM vector.* When thresholding each sensor row we thereby obtain a bit indicating if this bit in the USM vector should be set or not, and by OR-ing this bit with the previously stored bit we obtain a new bit value in the USM vector. The peak position indicated by the USM vector after all rows have been thresholded can then be found by a look-up table.

FIGURE 7.12.



The USM peak detection.

USM sequence generation

We have generated USM sequences using a computer program. It iteratively tries to add new USM numbers (i.e. more sensor rows) to the generated sequence while ensuring the uniqueness of the USM vectors generated by all possible peak positions in the sequence. The parameters that control the sequence generation is the length of the USM vector, i.e. the number of bits in the code, and the minimum and maximum peak widths n that should be possible to detect. The program then tries to find the longest possible sequence of USM numbers within the given constraints. There is no guarantee that the generated sequence is optimal, but we believe that no significantly longer sequences will be possible for a given set of input data. The sequence generation is covered in more detail in [66].

Ideally we want to be able to detect peaks with a variety of widths using as few bits in the vector as possible. Using few bits in the vector facilitates the conversion from the USM vector to the peak position. So far we have not implemented this conversion in MAPP2200, instead all USM vectors are output to the controller where a table lookup is performed. Since the MAPP2200 shiftregister is eight bits wide it seems natural to chose USM vector lengths as multiples of eight, and since 8 bits gives very short sequences we use 16 and 24 bit vectors. Data for four generated sequences can be found in Table 7.1. We see that for a 16 bit USM vector we can choose between detecting 3-9 pixels wide peaks using 182 sensor rows or 5-11 pixels wide peaks over the whole sensor. Using a 24 bit USM vector we can detect peak widths down to 2 pixels. The first sequence in the table is the one we have used in our experiments.

TABLE 7.1.

Sequence Length	USM vector length	Minimum peak width	Maximum peak width
182	16	3	9
256	16	5	11
256	24	2	9
232	24	2	11

Since the number of valid USM vectors is around 10% the possible vectors the inversion lookup tables contains a large number of non-valid entries. This gives a limited form of error detection capability. If the peak is thinner than the minimum width it is always guaranteed that no valid USM vector is generated, but if the peak is wider or a noise induced pixel position is set to one the generated USM vector may or may not be part of the valid set. In our experiments we have verified that the probability that a correct USM vector is generated from an incorrect peak is very small.

USM implementation

USM can be implemented on the MAPP2200 sensor in several ways. The most straightforward is to address the sensor rows in sequential order and threshold each row with the pre-specified threshold. The resulting bit is then OR-ed with the data in the USM vector position given by the row number. In pseudo code this will be

```
/* USM Implementation 1 */
/* tab[i] contains the USM number for row i */
Reset(R)                          /* Reset the USM vector, L */
SETR 0                            /* Set start row */
for (i=0;i<a;i++) {
   INTEG                          /* Read pixel value */
   LDI PD                         /* read thresholded bit */
   RESET,+                        /* Reset pixel */
   OR R(tab[i])                   /* Or with USM bit in RAM */
   ST R(tab[i])                   /* Store new bit in RAM */
}
Move(R,Sreg)                      /* move vector to sreg */
SAVE                              /* Shift out USM vector */
```

This implementation requires five MAPP2200 instructions in the inner loop. However, by utilizing the capability of analogue addition of charge from several rows before the thresholding we can implement the algorithm in a much more effective way. If we consider the charge in a pixel as zero if the peak does not illuminate that pixel and above the threshold if it does, then we can add the charge from all pixels who have the same USM number by addressing the corresponding sensor rows, and then threshold the result, getting the logic OR of the thresholded sensor rows as before. Thus, instead of the table with the USM sequence we generate a table which contains the row indices for the rows with USM number zero, the rows with indices one and so on. This table is then used in the implementation which looks like

```
/* USM Implementation 2 */
/* tab[i][0..(J-1)] contains the row numbers for
       the USM number i */
/* The number of rows in each USM bin is J */
Reset(R)                          /* Reset the USM vector, L */
for (i=0;i<L;i++) {    /* for all L USM numbers */
   RESET                          /* Reset PD register */
   SETR(tab[i][0])                /* select first row */
   INTEG                          /* Start integration */
   for (j=1;j<J;j++) {
       SETR(tab[i][j])            /* Select rows */
   }
   ST PD, R(i)                    /* Store thresholded bit */
}
Move(R,Sreg)                      /* move vector to sreg */
SAVE                              /* Shift out USM vector */
```

Using this implementation there is only one MAPP2200 instruction in the inner loop and the time to acquire one *NxN* range image using an *L* bit code word is approximately

$$T_{im} = N\left(Max\left(\left\lceil \frac{L}{8} \right\rceil \cdot \frac{N}{f_s}, \frac{a+5L}{f_p} \right) + \frac{L}{f_p} \right) \tag{7.8}$$

where a is the number of rows in the sequence. If a is 182 and L is 16 the range image frequency is 74 Hz and the range pixel frequency 4.8 MHz.

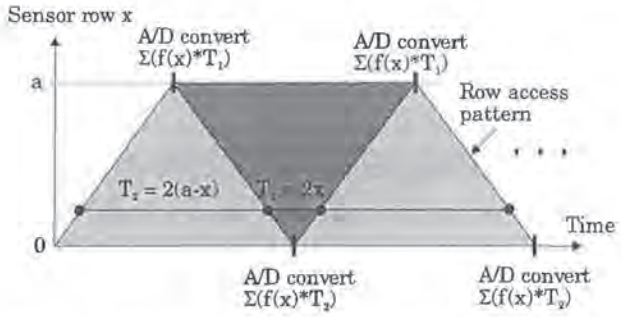
USM drawbacks

A large drawback of the USM algorithm is the limited peak widths that can be detected. This sets a limit in how much the reflectivity of the viewed object can change. Thus, the USM is not suited for applications where the object reflectivity changes very much. Also, the use of analog addition before the thresholding makes the algorithm more sensitive to background noise. In this implementation the shiftregister communication is the bottleneck, not the processing. Thus, if an effective way of translating the USM vector to a peak position can be implemented in MAPP2200 the performance can increase the range pixel speed up to almost 8 MHz.

7.11 Horizontal center-of-gravity algorithm FBM

In [31] we presented a straight-forward implementation of a center-of-gravity peak finding algorithm. The implementation has very low performance, less than 100 KHz rangel frequency, and has never been implemented. The Forward Backward Method, FBM, however is a very effective way to implement a horizontal center-of-gravity algorithm on MAPP2200. FBM was originally described by Åstrand et. al. in [66] and [67]. Just as the USM it utilizes the ability to add charge from several sensor rows before thresholding or A/D conversion. To understand the FBM study Figure 7.13. We scan the sensor rows in a forward-backward pattern, adding analog charge from all accessed rows, and at each turning point the collected intensity sum is A/D converted.

FIGURE 7.13.



The row access pattern and A/D conversion in the FBM.

For each row x the integration time is alternating between $2x$ and $2(a-x)$. The intensity sum achieved after a forward scan written as an integral is

$$S_1 = \int 2xf(x)\,dx \tag{7.9}$$

and correspondingly the intensity sum achieved from a backward scan is

$$S_2 = \int 2(a-x)f(x)\,dx \tag{7.10}$$

Now, if we combine these two sums we get

$$S_1 + S_2 = \int 2xf(x)\,dx + \int 2(a-x)f(x)\,dx = 2a\int f(x)\,dx \tag{7.11}$$

and thus we can compute the center-of-gravity position of $f(x)$ using

$$P = \frac{\int xf(x)\,dx}{\int f(x)\,dx} = a\frac{S_1}{S_1 + S_2} \tag{7.12}$$

In pseudo code the FBM implementation looks like

```
/* Horizontal center-of-gravity using FBM */

RESET
laser(on)
INTEG                                  /* Start integration */
for (i=0;i<a;i++) {                    /* Forward scan */
   SETR i                              /* Access gives integration */
}
laser(off)
HOLD
INITAD                                 /* A/D convert S1 */
WAITAD
Move(AD,Sreg)
SAVE
RESET
laser(on)
INTEG                                  /* Start integration */
for (i=a-1;i>=0;i--) {
   SETR i
}
```

```
laser(off)
HOLD
INITAD                                  /* A/D convert S2 */
WAITAD
Move(AD,Sreg)
SAVE
```

If the position P is to be calculated within MAPP, a bit-serial division is required which takes approximately $6b^2$ cycles where b is the number of bits. To avoid this the two intensity values are shifted out and the range values are then computed in the controller. A feature is that intensity data, i.e. the sum of the two components, is acquired as well as range data. Note that we modulate the laser to avoid integration during the A/D conversion.

The time for an NxN range image is then approximately

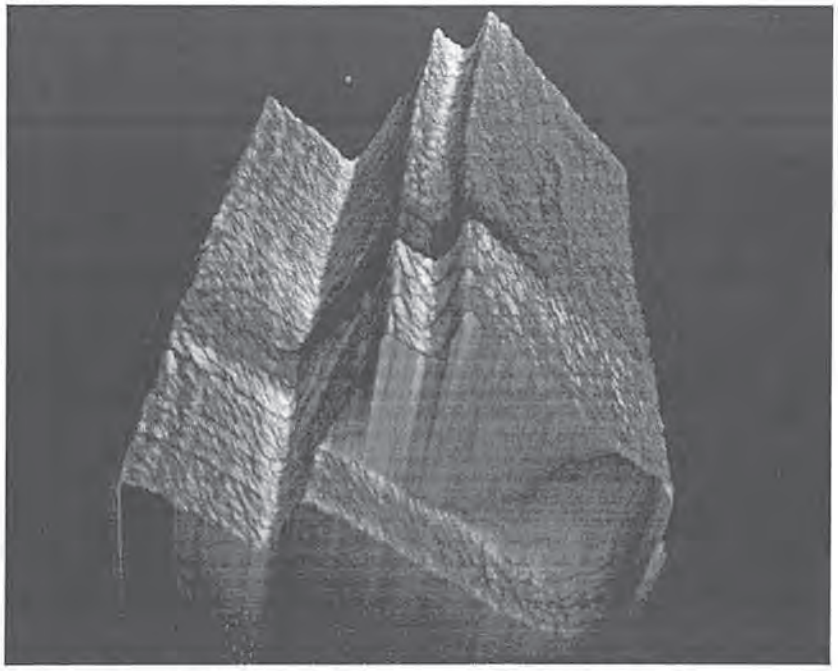
$$T_{im} = N\left(max\left(\frac{2N}{f_s}, \left(\frac{2a+18}{f_p} + \frac{2\times 2^b}{f_{AD}} \right) \right) + \frac{2b}{f_p} \right) \tag{7.13}$$

The data output using the shiftregister limits the rangel frequency to 4.8 MHz if the maximum A/D converter precision is used. Actually, by computing a new range vector for each forward or backward scan readout the rangel frequency can be doubled, giving 9.6 MHz rangel frequency. To reduce the noise sensitivity we can apply an image subtraction using an image acquired without the laser illumination. This requires two additional scans and A/D conversions, and two bit-serial subtractions. Since the computations can be performed concurrently with the A/D conversions and shifts, and the maximum A/D converter frequency is more than double the shift frequency, the performance loss for the image subtraction is very low.

FBM has a large potential for sub-pixel accuracy peak detection. The limiting factor is the A/D conversion, limited to eight bits, which constrains the theoretical position resolution. According to Matlab simulations around 9 bits of accuracy can be attained. However, this resolution is independent of the number of sensor rows actually used. Experiments have shown that around eight bits of resolution can be obtained using as few as eight sensor rows. In Figure 7.14 we give an example of an FBM image with eight bit resolution acquired using FBM on 16 sensor rows.

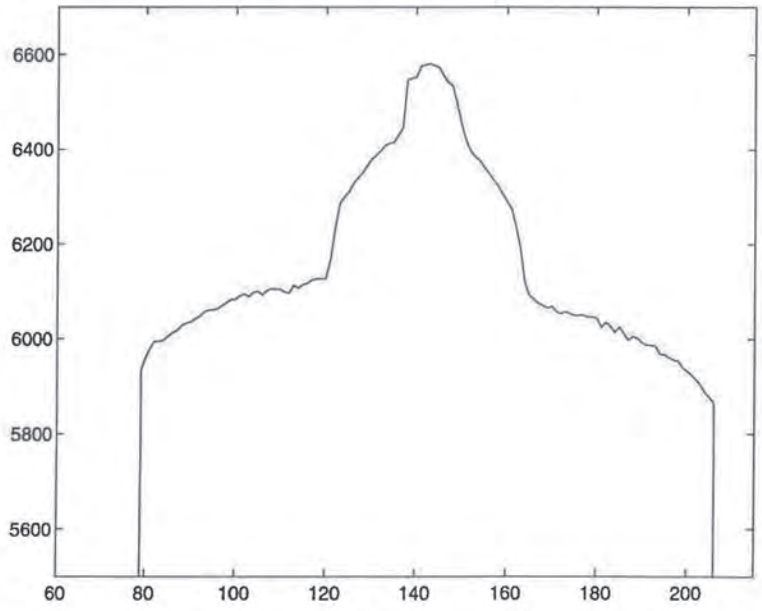
Now, since we can achieve around eight bits of resolution using FBM on a few sensor rows FBM has potential for very high accuracy if several consecutive scans over different parts of the sensor are combined. That is, we obtain a coarse resolution by noting at which row interval the laser illumination was detected, and a fine resolution using the output from this interval. An example where FBM has been used iteratively over the sensor is shown in Figure 7.15. 32 FBM scans over the sensor has been combined into one complete range profile with 5+8 = 13 bit resolution. We observe some noisy areas which occur when the scale factor a does not match between the different iterations. Part of this comes from the fact that a depends on the width of the laser line, and due to the perspective projection this width changes over the sensor. Thus it is very hard to trim the transitions between different scans. Note that the ranging speed is reduced proportionally to the number of scans over the sensor.

FIGURE 7.14.



A 3D plot of an FBM range and intensity image acquired using 16 sensor rows.

FIGURE 7.15.



An FBM profiles with 13 bit resolution. Compare with the profile in Figure 7.10.

7.12 Horizontal integrating algorithm I, TIM

The Total Integration Method, TIM, was originally described by Åstrand et. al. in [66]. Implementation-wise it is very similar to the horizontal thresholding algorithm without peak-width and error computation. However, instead of resetting the readout register after each row access the charge is added and we compare the sum with the threshold. If we consider the rows not illuminated by the laser as containing zero charge we get the same algorithm as with horizontal thresholding. Furthermore, after the scan of all rows the sum of all row charges is found in the registers, and after A/D conversion we have the total integrated column intensities. Thus we get both range and intensity information. In pseudo code the algorithm looks like

```
/* Horizontal total integrating method, TIM */
/* Gray(i) returns the correct counter bit pos */
Count = 0                          /* Reset position counter */
SETV volt                          /* Set threshold voltage */
SETR x                             /* Select start row */
RESET                              /* Reset readout register */
for (i=0;i<(y-x);i++) {            /* Read a = y-x rows */
   INTEG,+                         /* Access row i */
   LD PD
   XOR R(Gray(i))                  /* Change counter bit */
   ST A,R(Gray(i))
}
HOLD
SETV 0                             /* Reset A/D Start value */
INITAD                             /* A/D conversion */
Convert(Gray)                      /* 2loga */
Move(Counter,Sreg)
SAVE
```

The time to acquire one NxN image of range and intensity data is

$$T_{im} = Nmax\left(\left(\frac{b+\log a}{f_p}+\frac{2N}{f_{sh}}\right),\left(\frac{6+3\log a+4a}{f_p}+\frac{2^b}{f_{AD}}\right)\right) \tag{7.14}$$

and if only range data are acquired this is reduced to

$$T_{im} = Nmax\left(\left(\frac{\log a}{f_p}+\frac{N}{f_{sh}}\right),\left(\frac{2+3\log a+4a}{f_p}\right)\right) \tag{7.15}$$

For $b=8$ and $a = 256$ the range pixel frequency is almost 2 MHz (1.8 MHz if the intensity also is read out). If the resolution is reduced to 6 bits or less the shiftregister communication limits the performance and the rangel frequency increases to almost 10 and 5 MHz respectively.

Although TIM is very fast it has several drawbacks. It is very sensitive to background illumination since the total column charge is compared with the threshold, and the acquired peak position depends heavily on the peak width, and therefore on the peak intensity. If the peak intensity is acquired it might be possible to compensate for this, but no accurate compensation scheme has been devised. Despite all this the image quality can be quite good. We have successfully experimented with increasing the threshold incrementally during each sensor scan to reduce the sensitivity to accumulated background illumination noise.

7.13 Horizontal integrating algorithm II, OEM

The Odd Even Method, OEM, was originally described by Åstrand et. al. in [66] and is an extension of the TIM. In OEM we integrate the odd and even lines separately in order to overcome the intensity dependance and to extract some sub-pixel information. The algorithm works in two phases, one for odd rows and one for even rows. Each phase is identical to the TIM algorithm, the difference being that the even rows are accessed in reverse order. By reversing the scan order we get one measurement for each edge of the peak. It is then possible to approximate the peak position with their mean value. Since we still integrate all pixel charges we can find the total intensity by adding the intensities from the two passes. In pseudo code the implementation looks like

```
* Horizontal total integrating method, TIM */
/* Gray(i) returns the correct counter bit pos */
Count_o = Count_e = 0              /* Reset counters */
SETV volt                          /* Set threshold voltage */
RESET                              /* Reset readout register */
INTEG                    /* Start the even integration */
for (i=0;i<(y-x);i=i+2) {          /* a = (y-x) */
   SETR x+i                        /* Access row */
   LD PD
   XOR R(Gray_o(i))
   ST A,R(Gray_o(i))
}
HOLD
SETV 0                             /* Reset A/D Start value */
INITAD                             /* A/D conversion */
WAITAD
Move(AD,Reg1)

SETV volt                          /* Set threshold voltage */
RESET                              /* Reset readout register */
INTEG                    /* Start the odd integration */
for (i=0;i<(y-x);i=i+2) {          /* Read a = y-x rows */
   SETR y-i/* Access row */
   LD PD
   XOR R(Gray_e(i))
   ST A,R(Gray_e(i))
}
HOLD
SETV 0                             /* Reset A/D Start value */
INITAD                             /* A/D conversion */
WAITAD
Move(AD,Reg2)
Convert(Gray_o)                    /* Convert counters */
Convert(Gray_e)
Add(Odd,Even,Sum)                  /* Mean of counters is peak */
Move(Sum,Sreg)                     /* Shift out mean peak pos */
SAVE
Add(Reg1,Reg2,Sum)                 /* Add intensities */
Move(Sum,Sreg)                     /* Shift out intensity sum */
SAVE
```

The time to acquire one range image using OEM is then

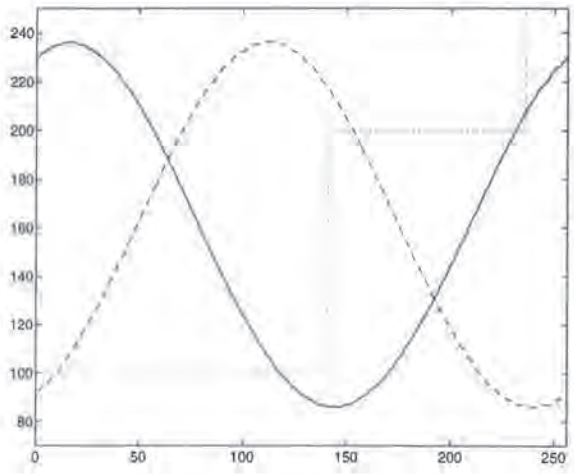
$$T_{im} = Nmax\left(\left(\frac{b+\log a}{f_p}+\frac{2N}{f_{sh}}\right),\left(\frac{14+2b+4a}{f_p}+\frac{2\cdot 2^b}{f_{AD}}\right)\right) \tag{7.16}$$

The OEM performance is slightly lower than for the TIM, and for b=8 and a = 256 the range pixel frequency is 1.8 MHz (1.5 if the intensity is readout as well). For less than 6 bit resolution the shiftregister limits the performance and the rangel speeds are close to 10 and 5 MHz respectively.

In the pseudo code we compute the intensity sum and transfer that out to the controller. However, it might be advantageous to shift out the intensity and peak position components instead as they contain sub-pixel and error information. If the two positions indicated by the counters are dissimilar one of them is probably affected by noise and the absolute difference can be thresholded into an error value.

That the intensity components can be used for sub-pixel information can be understood by noting the following. Assume that the peak is approximately one pixel wide. Then, if the intensity components are equal the peak is centered between two pixels, and if the even sum is much larger than the odd sum the peak is centered around an even pixel. A simulation of the intensity components as a function of the peak position can be found in Figure 7.16. In the simulation we have used MAPP2200 sensor geometry, and the uneven row distance is the cause of the non symmetric intensity curves. Now, if we can recover the "phase" of the two components we should be able to estimate the peak position more accurately.

FIGURE 7.16.

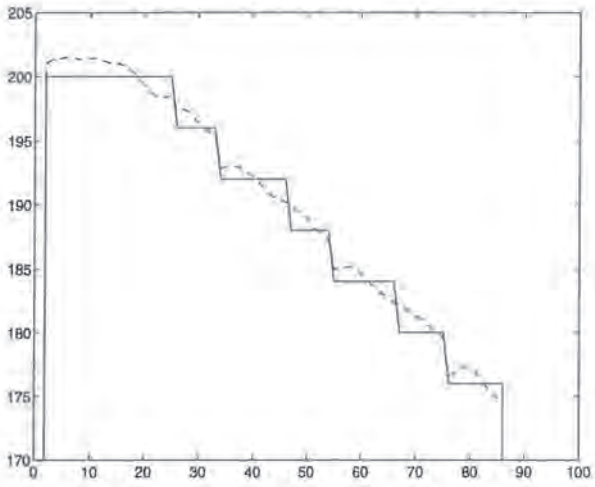


Odd and even pixel intensities for different peak positions. The integer peak position increment is shown amplified.

In Figure 7.17 we show the result when we have used the intensity components to try to recover sub-pixel accuracy. In the figure we see that the method fails at every second integer transition, but otherwise the results looks good. This is because the acquired sub-pixel information and the integer position values are "out-of-phase", i.e. the change in integer position does not occur simultaneously with the change from plus to minus sub-pixel information. In this example we have managed to select the threshold so that one transition is in-phase with the sub-pixel information but the other is not. Thus, it is hard to use the intensity components for

sub-pixel accuracy ranging. However, if a continuous surface is viewed any change in distance may be determined with high sub-pixel accuracy from the intensity components, but not the absolute peak position. Also the amplitude of the sinusoids depends heavily on the peak width, which makes the method unsuitable for use with materials having large intensity variations. Actually, if we only study the intensity components the results are very similar to the FBM results, but in the FBM the unsymmetrical sensor row positioning gives no non-symmetry in the components since the odd and even sensor rows not are processed separately.

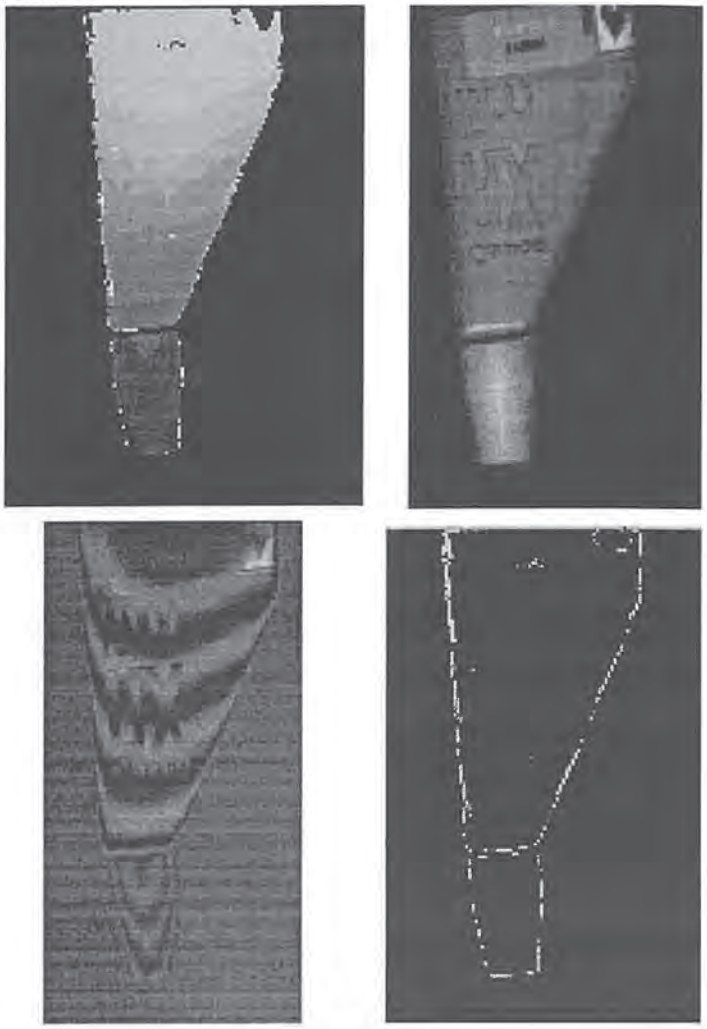
FIGURE 7.17.



The integer and recovered sub-pixel range for a range profile. Note that the predicted non-symmetry of the integer position increments is clearly visible.

Four image results from one OEM range image acquisition is shown in Figure 7.18.

FIGURE 7.18.



Range (top left), intensity (top right), sub-pixel information (bottom left) and error (bottom right). The sub-pixel information is the difference between the intensity components and the error image is the difference between the range components.

7.14 Multiple peak readout

Multiple peak readout can easily be incorporated into the MAPP2200 vertical sheet-of-light programs using the PEs to mask out the peaks from the binary position vector one after the other as in the code example below.

```
   /* Multiple peak redout from MAPP2200,binary peak row in Acc*/
ST Acc, R0              /* Store temporary row */
while (GOR)             /* While any peak left */
   LFILL R0             /* Detect leftmost pixel */
   LD GLU
   pos[i] = COUNT       /* Read position */
   LEDGE                /* Find leftmost pixel */
```

```
    MARK R0              /* Mask out leftmost peak */
    LD GLU
    width[i] = COUNT     /* Read peak width */
    XOR R0               /* Remove peak from temporary row */
    ST Acc, R0           /* New row as temporary */
  endwhile
```

The width and position data is in fact the run-length code of the thresholded row, and the extra execution time is approximately 1.5 µs for each additional peak readout (at a PE clock frequency of 8 MHz).

7.15 RANGER RS2200

The RANGER RS2200 range sensor is a MAPP2200 sensor with sheet-of-light range imaging software and it has been commercially available from IVP since mid-1994. So far systems have been sold in for instance Sweden, USA, Italy and Israel.

The system is based on the results presented in [31] and it incorporates the vertical algorithms in section 7.3 - section 7.6, i.e. NSIP maximum finding, successive approximation maximum finding, and thresholding algorithms. Both the successive approximation and the thresholding algorithm support multiple thresholdings to increase the range resolution. The system also contains tool functions for system accuracy prediction, calibration and range image viewing. With the PC version maximum rangel frequency of between 90-150 KHz can be obtained, and with the Smart Camera version the maximum rangel frequency is approximately 200 KHz.

Chapter 8

Comparisons and conclusions

In this part of the thesis we have presented a total system overview of sheet-of-light range imaging. We have covered geometric design and calibration as well as several algorithms. A novel smart sensor architecture for range imaging called NESSLA has been presented, and several algorithm implementations have been studied using the MAPP2200 sensor. Here we give a comparison between our results and those for other smart sensor implementations presented in the literature.

The *range pixel speed* differences among the presented smart sensor architectures are not very large. Most of them are capable to output position data at around 1-2 MHz, which is 50-100 times faster than what is possible with a standard video sensor. The best predicted performance, up to and above 5-10 MHz, was found for the NESSLA architectures and for certain MAPP2200 implementations.

The NESSLA architecture has digital data transports from the pixel cells which should be less *noise sensitive* than the analog transfers in all other designs. However, the extensive use of logic in the pixels makes it hard to have a large sensor aperture, and thus the light sensitivity might suffer. The NESSLA design automatically adapts for different light intensities and therefore the risk for sensor saturation is very low. Sensor saturation can otherwise be a problem, especially when we get specular reflections of the laser light.

The Analog processing sensor cells architecture is the most *complex* design with the analog signal processing in each sensor cell. The NESSLA design is also quite complex but we believe that it is possible to fabricate a working 256^2 sensor with standard 0.8 µm CMOS VLSI process design. Furthermore, the NESSLA design does not need any analog hardware outside the sensor area which makes it rather VLSI friendly. The PSD designs seems to be easy to implement but they need extra analog processing to obtain the offset (range) data from the sensor outputs. The analog bottleneck in the previously presented designs can be avoided by using the design with on-chip A/D conversion presented here. Both the Row readout successive approximation and Column readout peak detection architectures utilize a standard sensor area and are therefore not very complex to implement.

The NESSLA architecture, the Column read-out peak detection architecture and several of the MAPP2200 implementations can all output grey level *intensity images* concurrently with the range data. Theoretically, the PSD architecture can output reflectance maps, but no reports of this are available. Intensity information is very useful both as image information and as a confidence measurement.

We believe that a sheet-of-light range imaging system using the NESSLA architecture can output range images with range image frequency and accuracy comparable to or better than the architectures found in the literature. NESSLA also has a built-in error correction which does not exist in any of the others except some of the MAPP2200 implementations. The Column readout architecture can also be very effective and it is fairly easy to implement in VLSI.

The MAPP2200 implementations are maybe not as fast as the special purpose architectures for the same accuracy, but the overwhelming advantage of this sensor is that it is commercially available. With MAPP2200 a sheet-of-light range camera system can be built today, whereas

the other designs are prototypes at the best. Also, with MAPP2200 it is possible to trade resolution for speed and obtain low resolution images at high speed. An additional feature of MAPP2200 is its general programmability which makes it possible to implement not only the sheet-of-light range finding algorithm but also subsequent processing of range data as has been shown in a can sorting application example [30]. This alleviates the need for extra hardware and makes the overall system installation easier

References

[1] Andersson (Gökstorp) M.J., *Range from Out-of-Focus Blur.*, Lic. Thesis. LIU-TEK-LIC-1992:17, Linköping 1992.

[2] Araki K.,Shimizu M., Noda T., Chiba Y., Tsuda Y., Ikegaya K., Sannomiya K., Gomi M., *A range finder with PSD's*, Machine Vision and Applications, MVA '90, Tokyo 1990.

[3] Araki K., Tsuda Y., Shimizu M., Ikegaya K., Noda T., Sannomiya K., Chiba Y., Gomi M., *High Speed and Continuous 3-D Measurement System, 11th Int.* Conf. on Pattern Recognition, Vol 4, Hague 1992.

[4] Beeler T.E., *Producing Space Shuttle Tiles with a 3-D Non-Contact Measurement System*, In "Three-Dimensional Machine Vision" ed. by T Kanade, Kluwer Academic Publishers 1987.

[5] Beraldin J.A., Rioux M., Blais F., Cournoyer L., Domey J., *Registered intensity and range imaging at 10 mega-samples per second,* Optical Engineering 31 (1), pp 88-94, 1992.

[6] Besl P. J., *Active Optical Range Imaging Sensors*, Advances in Machine Vision, Ch. 1, Springer, 1988.

[7] Blais F., Rioux M., *Real-time numerical peak detector*, Signal Processing 11, pp 145-155, North Holland, 1986.

[8] Bracewell R.N, *The Fourier transform and its applications,* McGraw-Hill, 1986.

[9] Burke M. et. al., *RST, Road Surface Tester*, Report from OPQ systems, Linköping 1991.

[10] Case S.K. Jalkio J.A. Kim R.C., *3-D Vision System Analysis and Design,* In "Three-Dimensional Machine Vision" ed. by T Kanade, Kluwer Academic Publishers 1987.

[11] Danielsson P.E., *Smart Sensors for Computer Vision*, Report No. LiTH-ISY-I-1096, Linköping 1990.

[12] Eklund J.E., *Oral communication,* Linköping 1995.

[13] Eklund J.E., Svensson C., Åström A., *Near-Sensor Image Processing, a VLSI Realization,* Proc of Transducers95, Eurosensors IX, Stockholm, 1995.

[14] Eklund J.E., Svensson C., Åström A., *Near-Sensor Image Processing, a VLSI Realization,* Proc of 8th Annual IEEE Int. ASIC Conf., Austin, 1995.

[15] Erhardt, *Silicon Cylindrical Lens Arrays for Improved Photoresponse in Focal Plane Arrays,* SPIE Vol. 501, 1984.

[16] Fischer H., Lulé T., Schneider B., Schulte J., Böhm M., *Analog image detector in Thin Film on ASIC (TFA)-technology,* EUROPTO - Sensors and Control for Advanced Automation II, Frankfurt, 1994.

[17] Forchheimer R, Ingelhag P, Jansson C, *MAPP2200, a second generation smart sensor,* SPIE vol 1659, 1992.

[18] Forchheimer R., Åström A., *Near-Sensor Image Processing. A New Paradigm,* IEEE trans. on Image Processing, vol. 3, no. 6, november 1994.

[19] Forchheimer R., Ödmark A., *Single chip linear array processor,* Applications of digital image processing, SPIE, Vol. 397, 1983

[20] Garcia D. F., del Rio M. A., Diaz J. L., Fransisco J. S., *Flatness defect measurement system for steel industry based on a real-time linear-image processor,* Proc of IEEE/SMC'93, Le Touquet 1993.

[21] Gruss A., Carley R., Kanade T., *Integrated Sensor and Range-Finding Analog Signal Processor,* IEEE Journal of Solid State Circuits, Vol 26, pp 184-191, 1991.

[22] Gökstorp M., *Range Computation in Robot Vision,* Ph.D. Thesis No. XX, Linköping, 1995.

[23] Haralick R.M., Shapiro L.G., *Computer Vision, Vol II,* Addison Wesley, 1993.

[24] Horn B. K. P., *Robot Vision,* MIT press 1986.

[25] Ingelhag P., *A contribution to high performance CMOS circuit techniques*, report LiU-TEK-LIC-1992:35, Linköping, november 1992.

[26] Jansson C, *Oral communications*, Linköping september 1993.

[27] Jansson C., *CMOS Signal Conditioned Sensor Arrays and A/D-D/A Conversion Techniques,* Ph.D. thesis No. 377, Linköping, 1995.

[28] Jiang X-Y, Gunke H, *Quantization Errors in Active Range Sensing*, Proc of SCIA-93, pp 913-920, Tromsø 1993.

[29] Johannesson M., *Sheet-of-Light range imaging experiments with MAPP2200,* Report No. LiTH-ISY-I-1401, Linköping 1992.

[30] Johannesson M, *Can Sorting Using Sheet-of-Light Range Imaging and MAPP2200*, Proc of IEEE/SMC'93, Le Touquet 1993.

[31] Johannesson M., *Sheet-of-light Range Imaging,* Lic thesis No. 404, LIU-TEK-LIC-1993:46, Linköping, 1993.

[32] Johannesson M., *Fast, programmable, sheet-of-light range imager using MAPP2200*, Proc. of San Diego SPIE '94, San Diego, 1994.

[33] Johannesson M.,*Calibration of a MAPP2200 sheet-of-light range camera,* Proc of SCIA'95, Uppsala, 1995.

[34] Johannesson M., Åstrand E., Åström A., *A Fast and Programmable Sheet-of-light Range Imager Using MAPP2200*, Proc. of SSAB '94, Halmstad, 1994.

[35] Johannesson M, Åström A, *Sheet-of-Light range imaging with MAPP2200,* Proc of SCIA -93, pp 1283-1290, Tromsø 1993.

[36] Johannesson M., Åström A., Danielsson P.E., *An Image Sensor for Sheet-of-Light Range Imaging*, proc of MVA '92, Tokyo 1992.

[37] Johannesson M, Åström A, Ye Q.Z, Danielsson P.E., *Architectures for Sheet-of-Light Range Imaging,* Report No. LiTH-ISY-I-1335, Linköping 1992.

[38] Kanade T, Gruss A, Carley L.R, *A Fast Lightstripe Rangefinding System with Smart VLSI Sensor,* Machine Vision for Three-Dimensional Scenes, Ed. H Freeman, Academic Press, 1990.

[39] Knutsson H., Westin C.F., *Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data,* Proc of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, New York, 1993.

[40] *MAPP2200 Technical Description*, IVP Integrated Vision Products AB, Linköping,1992.

[41] Melen T., Sommerfelt A., *Calibrating Sheet-of-Light Triangulation Systems,* Proc of SCIA '95, Uppsala 1995.

[42] *Micro-Optics Monolithic lenslet modules,* Prospect from United Technologies Adaptive Optics Associates, 1992.

[43] Moss J.P., McCane M., Fright W. R., Linney A. D., James D. R., *A three-dimensional soft tissue*

analysis of fifteen patients with Class II, Division 1 malocclusions after bimaxillary surgery, Am. Jour. of Orthodontics and Dental Orthopedics, Vol 105, 1994.

[44] Mundy J.L, Porter G.B. III, *A Three-Dimensional Sensor Based on Structured Light,* In "Three-Dimensional Machine Vision" ed. by T Kanade, Kluwer Academic Publishers 1987.

[45] Nakagawa Y, Ninomiya T, *Three-Dimensional Vision Systems Using the Structured-Light Method for Inspection Solder Joints and Assembly Robots,* In "Three-Dimensional Machine Vision" ed. by T Kanade, Kluwer Academic Publishers 1987.

[46] Nygårds J, Wernersson Å, *Gripping Using a Range Camera: Reducing Ambiguities for Specular and Transparent Objects,* Proc from Robotikdagar, Linköping 1993.

[47] Poussart D., Laurendeau D., *3-D Sensing for Industrial Computer Vision,* Advances in Machine Vision, Ch. 3, Springer, 1988.

[48] *RANGER RS2200 Technical Description,* IVP, Integrated Vision Products, Linköping, 1994.

[49] Rioux M, *Laser Range finder based on synchronized scanners,* In "Robot sensors Vol 1 Vision" ed. by A Pugh, Springer Verlag 1986.

[50] Rowa P., *Automatic visual inspection and grading of lumber,* Seminar/Workshop on scanning technology and image processing on wood, Skellefteå, 1992.

[51] Saint-Marc P., Jezouin J.L., Medioni G., *A Versatile PC-Based Range Finding System,* IEEE Trans on Robotics and Automation, Vol 7, No 2, 1991.

[52] Sato K, *Oral communications,* Osaka december 1992.

[53] Stuivinga M., Nobel J., Verbeek P.W., Steenvoorden G.K., Joon M.M., *Range-Finding Camera Based on a Position-Sensitive Device Array,* Sensors and Actuators 17, pp 255-258, 1989.

[54] Technical Arts Corp, *White Scanner 100a User's Manual,* Seattle, USA 1984.

[55] Theodoracatos V, Calkins D, *A 3-D Vision system Model for Automatic Object Surface Sensing,* In "Journal of Computer Vision", 11-1, pp 75-99, 1993.

[56] Toborg S., *A 3-D Wafer Scale Architecture for Early Vision Processing,* Proc. of Int. Conf. on Application Specific Array Processors, pp 247-258, Princeton 1990.

[57] Trucco E., Fischer R. B., *Acquisition of Consistent Range Data Using Local Calibration,* Proc of IEEE Int Conf. on Robotics and Automation, San Diego, 1994.

[58] Veldkamp W., *Binary Optics,* McGraw-Hill Yearbook of Science and Technology 1990, McGraw-Hill 1991.

[59] Verbeek P, *Oral communications,* Linköping april 1993.

[60] Verbeek P.W., Nobel J., Steenvoorden G.K., Stuivinga M., *Video-speed Triangulation Range Imaging,* NATO ASI Series, Vol F 63, pp 181-186, Springer-Verlag Berlin, Heidelberg 1990.

[61] Verbeek P.W., Groen F.C.A., Steenvoorden G.K., Stuivinga M.E.C., *Range cameras at video speed based on a PSD-array or on CCD-interpolation,* Proc. of IASTED Int. Symp. ROBOTICS AND AUTOMATION, pp 67-69, Lugano 1985.

[62] Weste N., Eshraghian K., *Principles of CMOS VLSI Design,* Addison Wesley, 1985.

[63] Westelius C-J, *Preattentive Gaze Control for Robot Vision,* LIU-TEK-LIC-1992:14, Linköping 1992.

[64] Ye Qin-Zhong, *Contributions to the Development of Machine Vision Algorithms,* Ph.D. Thesis No. 201, Linköping University 1989.

[65] Åstrand E.,*Detection and Classification of Surface Defects in Wood A Preliminary Study,* Report No. LiTH-ISY-I-1437, Linköping 1992.

[66] Åstrand E., Johannesson M., Åström A., *Five contributions to the art of sheet-of-light range imaging*, Report No. LiTH-R-ISY-1611, Linköping, 1994.

[67] Åstrand E., Johannesson M., Åström A., *Two novel methods for range imaging using the MAPP2200 smart sensor*, Proc of Machine Vision Applications in Industrial Inspection, SPIE '95, San José, 1995.

[68] Åström A, *A Smart Image Sensor. Description and Evaluation of PASIC.*, Lic. Thesis. LIU-TEK-LIC-1990:57, Linköping 1990.

[69] Åström A, *Smart Image Sensors,* Ph.D. thesis No 319, Linköping 1993.

[70] Åström A, Forchheimer R, *Near-Sensor Image Processing A new approach to low level image processing*, Proc. of 2nd Singapore Int. Conf. on Image Processing, 1992.

[71] Åström A., Forchheimer R., Ingelhag P., *An Integrated Sensor/Processor Architecture Based on Near-Sensor Image Processing,* Proc of CAMP'93, New Orleans, 1993.