
Conference Proceedings
Volume 3

1993, International Conference on Systems, Man and Cybernetics

Systems Engineering in the Service of Humans

Palais de l'Europe
Hôtel Westminster
Le Touquet - France
October 17 - 20, 1993

Hosted by
Ecole Centrale de Lille

With the Support of
Région Nord Pas de Calais

Copyright and Reprint Permission : Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 27 Congress Street, Salem, MA 01970. Instructors are permitted to photocopy isolated articles for non-commercial classroom use without fee. For other copying, reprint or republication permission, write to IEEE Copyright Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. All rights reserved. Copyright © 1993 by the Institute of Electrical and Electronics Engineers, Inc.

IEEE Catalog Number : 93CH3242-5

ISBN : 0-7803-0911-1 (Softbound Edition)

ISBN : 0-7803-0912-X (Casebound Edition)

ISBN : 0-7803-0913-8 (Microfiche Edition)

Library of Congress : 92-55091

Printed in France by Imprimerie Artésienne (Liévin), for GERF EC Lille (Villeneuve d'Ascq)

Can sorting using sheet-of-light range imaging and MAPP2200

Mattias Johannesson
Image Processing Laboratory
Department of Electrical Engineering
Linköping University
S-581 83 Linköping, SWEDEN

Abstract This report describes a real-time can sorting application using a sheet-of-light range imaging algorithm on the MAPP2200 smart sensor camera. The camera outputs range data with a maximum frequency of 800 KHz. The range data are automatically corrected, and uncertain data are removed. We acquire range images of fish conserves passing on a conveyor belt and finds the orientation of up to ten cans per second.

I. INTRODUCTION

Range imaging is an important and powerful tool in many machine vision applications. Sheet-of-light range imaging is one range imaging method which is especially well suited in applications where the scene or the apparatus is moving, for instance conveyor belts [4] or road surveys. Such a sheet-of-light range imaging system can be seen in Fig 1 and Fig 2. The range data are acquired with triangulation. At each sensor exposure we find the reflection of the laser light in each row, and these position values corresponds to range data, see Fig 3. The baseline distance between the optical center of the sensor and the laser together with the angle between the optical axis of the sensor system and the light-sheet are the triangulation parameters determining the possible resolution, see Fig 1. In general a larger baseline gives better resolution, but also more occlusion problems. Occlusion occurs when the camera cannot see the laser sheet, and thus no range data can be found.

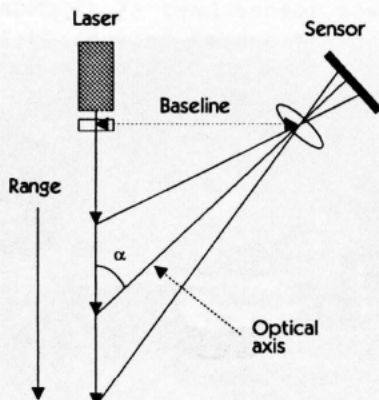


Fig. 1. A sheet-of-light range imaging setup seen with the sheet-of-light perpendicular to the plane of the paper.

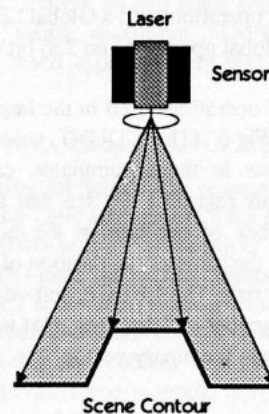


Fig. 2. A sheet-of-light range imaging system seen with the sheet-of-light parallel to the plane of the paper.

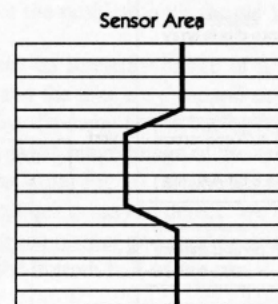


Fig. 3. A snapshot of the sensor area (rotated 90 degrees with respect to Fig 2).

Each sensor integration gives a 1-dimensional range image of a cross section of the scene. Complete 2-dimensional range images can be acquired in at least three ways, by moving the scene, i.e. on a conveyor belt or a rotating table, by moving the apparatus, i.e. in a road survey vehicle, or by scanning the light sheet over the scene synchronous with the integrations. In our application we use a conveyor belt to move the objects through the laser sheet.

II. MAPP2200 AND IMAGING SETUP

MAPP2200 (Matrix Array Picture Processor) [2], [3] 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 a *single* chip, see Fig 4. Each PE consists of an 8 bit A/D output register, an 8 bit parallel shiftregister cell, 96 bits of memory and a bit serial ALU. Each ALU has a Point Logic Unit (PLU) for one to three input boolean operations, a Neighborhood Logic Unit (NLU) for left-right neighbor-PE dependent operations and a Global Logic Unit (GLU) which performs global operations on 256 bit wide words.

The NLU/GLU operations used in the implementation are demonstrated in Fig 5. (The COUNT value, which is the number of set bits in the accumulator, can be read via MAPP2200's status register.) We see that after a LFILL, left-fill, the number of set bits in the accumulator (the COUNT value) is the same as the position of the leftmost set bit in the original row. The LEDGE, left-edge, command is used to find the number of left edges, that is, the number of connective objects in the binary vector.

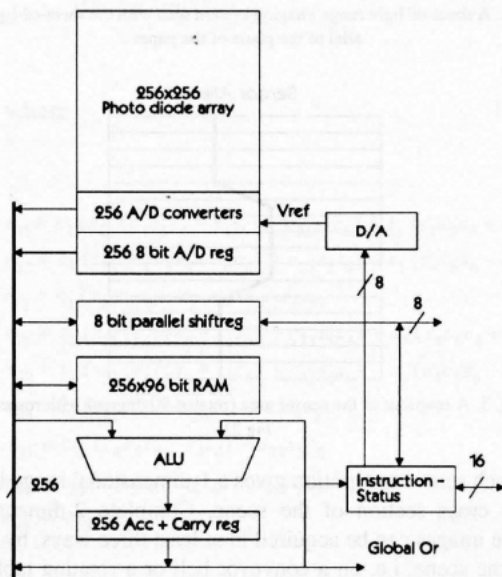


Fig. 4. Block diagram of a MAPP2200 chip.

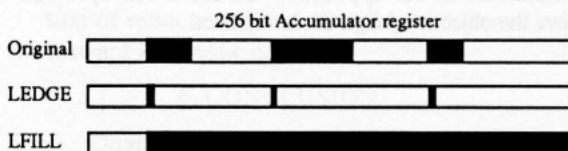


Fig. 5. The NLU and GLU functions used in this application. (Black is one).

A. MAPP2200 smart camera

The industrial version of the system is to be implemented in a MAPP2200 *smart camera* system. The smart camera contains a MAPP2200 sensor and an Intel i186 processor board, and is a stand-alone unit. Thus, in the production system no external PC is needed, and the outputs from the camera processor is connected direct to the mechanism for the can rotation.

B. Used setup

The setup consists of a MAPP2200 *smart sensor* camera controlled by a PC, and a 2.5 mW laser diode with a cylindric lens. The camera takes images of a fixed line projected onto the scene. Therefore the objects have to be moved through the laser curtain to obtain range images of complete 3D objects. To emulate a conveyor belt movement we have used a LEGO® train with cans attached to the wagons, see Fig 6. The train has a maximum speed of approximately 1.3 m/s. The scene width is limited to approximately 20 cm, each pixel corresponding to approximately 0.8 mm. The laser sheet has a width of approximately 2 mm, and the range accuracy is approximately 1 mm at a standoff distance of 0.4 m.

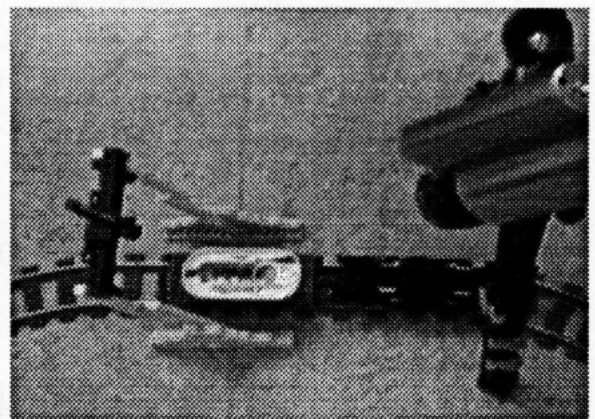
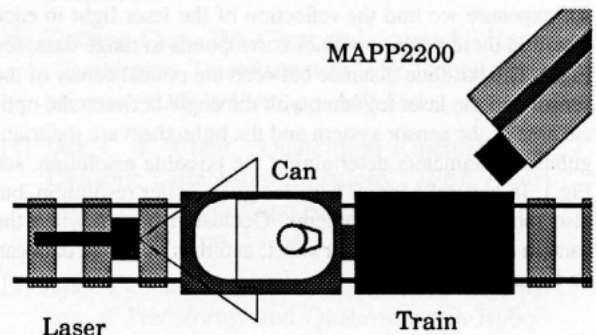


Fig. 6. An overview of the setup. Top: symbolic, Bottom: photo.

III. RANGE IMAGING ALGORITHM

The range (offset) data for a sensor row is the position of the maximum intensity (or the position of the center of gravity). This implementation uses a thresholding algorithm, described in more detail in [6], to find this value. Instead of explicit finding the maximum we threshold the data and use the binary result to find an approximation of the position of the maximum. The relation between finding maximum, center of gravity and thresholding can be seen in Fig 7. The center of gravity technique, which gives very fine sub-pixel accuracy, is very sensitive to background illumination and is cumbersome to compute. Our empirical tests have shown that the threshold algorithm data are almost as accurate as the other methods, and if the background illumination cannot be eliminated the center-of-gravity algorithm cannot be used [6].

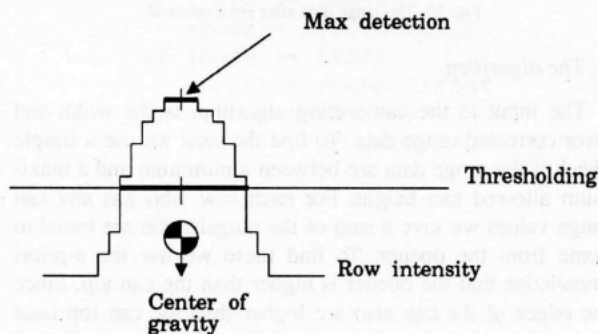


Fig. 7. Different methods to find the range value in a row of data.

In pseudo-code notation the MAPP2200 program for the threshold algorithm can be written:

```
main {
  SETV volt /* set threshold */
  for (i=0;i<256;i++) {
    READPD(i) /* read sensor row */
    LDI PD /* load to acc */
    width[i] = COUNT /* peak width */
    LD LEDGE Acc /* Find #peaks */
    err[i] = COUNT
    LFILL Acc /* Find position */
    LD GLU /* Load result */
    pos[i] = COUNT
  }
}
```

The output consists of three values, the position (range), the width and the error value. The width value is used to compensate the range data for the laser sheet thickness using the formula

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

Using this correction we obtain half pixel sub-pixel resolution. The error value is used to remove ambiguous range values. If the error value is larger than one we know that several distinct peaks were found, and that the range data are uncertain. If the error value is zero we have occlusion since not enough light has reached this sensor row and thus no peaks have been found.

If we control the camera with a PC we can achieve approximately 100 K rangels/seconds. In the smart camera we can use higher clock frequencies and reach approximately 200 K rangels/second. Using maximum MAPP2200 clock frequency we can reach approximately 800 K rangels/second [6].

IV. CAN SORTING APPLICATION

When packaging fabricated fish conserves it is important that they have the same orientation. This is because the can text shall be possible to read by the buyer when looking at a can on a shelf. To obtain the correct orientation the cans pass the range imaging setup. This detects which cans should be rotated 180 degrees, and sends control signals to the rotation mechanism. Solving the problem with ordinary image processing is non-trivial since it would involve some feature extraction, either to find the roughly ring-shaped opener or read the text. This would probably only be possible with considerable amounts of 2D operations. When we use the range data we can solve the problem with simple 1D operations.

In Fig 8 we see an intensity image of a fish can. We see that the opener and the text are dark and that the overall can intensity is almost the same as the background. If we, on the other hand, look at the range image of the same scene, seen in Fig 9 (uncorrected) and Fig 10 (erroneous rangels removed), we see that the opener is easy to detect. We can use the range information from the opener and easily estimate if the opener is on the top or the bottom half of the can.

Another feature when using the range data is that the ring feature is the same for several can types, and that the algorithm therefore can sort different types and sizes without any adjustments.

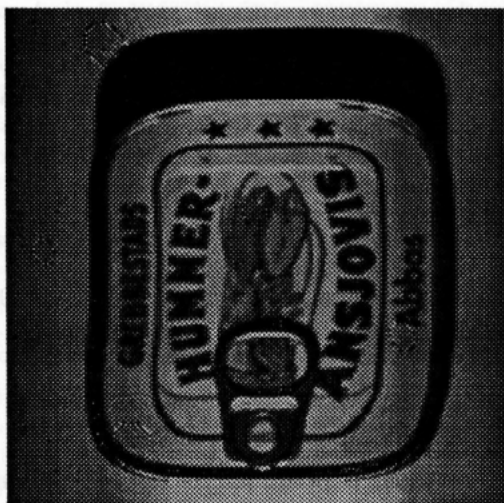


Fig. 8. An intensity image of a fish can.

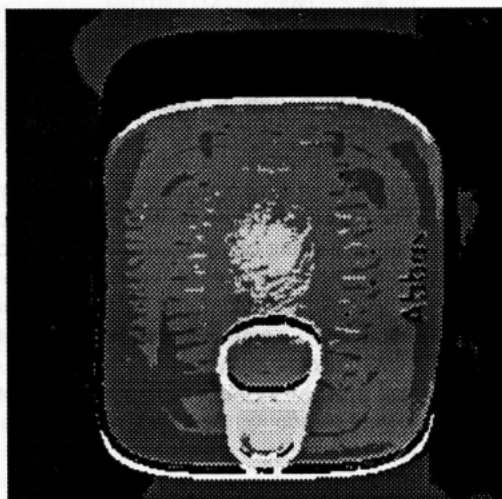


Fig. 10. The range data after error removal.

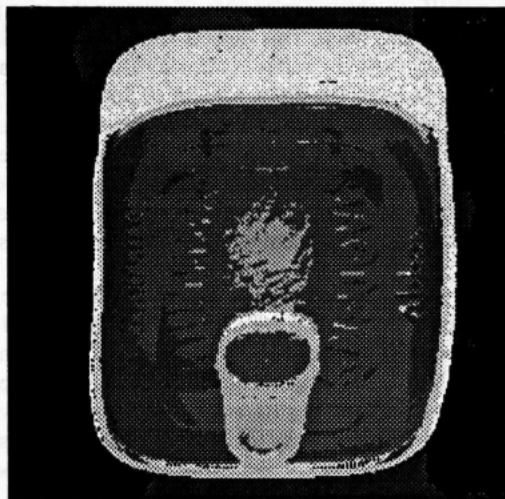


Fig. 9. A range image of a fish can. (Dark is far away and light is near.)

A. The algorithm

The input to the can sorting algorithm is the width and error corrected range data. To find the cans we use a simple check if the range data are between a minimum and a maximum allowed can height. For each row who has any can range values we save a sum of the rangels who are found to come from the opener. To find these we use the a-priori knowledge that the opener is higher than the can top. Since the edges of the can also are higher than the can top (and higher than the opener) we must discard some can data at either edge. The process is visualized in Fig 11.

The skip of the data at the edges of the can is made using a FIFO buffer which is read only when data has been written twice to the whole FIFO. This ensures that as many rangels as the buffer is long are skipped at both edges.

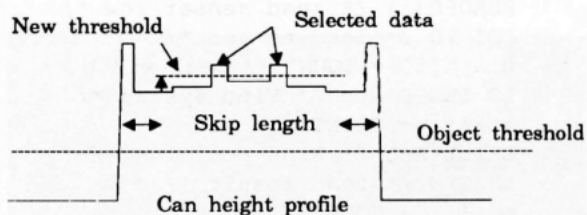


Fig. 11. The can opener detection process. From the first detected can rangels a fixed number is skipped, and then a can top height is found. Then any pixel above the threshold, which is the found can top height plus an offset, are summed to the result of the row. A fixed number of object pixels at the end of the can are also skipped.

When the algorithm detects the end of a can the contributions from the rows are separated into the first and second halves, and the sums of these are compared. The highest sum comes from the half with the opener. In pseudo code the algorithm can be written:

```
main /* LEN = 10, OFFSET = 2 */
do {
  start = 0
  count = 0
  bufc = 0
  for (i=0;i<256;i++) { /* row */
    range = READRANGE(i)
    if (LOW < range < HIGH) { /* can */
      start = 1
      FIFO[bufc%LEN] = range;
      bufc++ /* buffer counter */
      if (bufc >= LEN*2)
        rtemp = FIFO[bufc%LEN]
      if (bufc == LEN*2)
        thresh = rtemp+OFFSET
      if (rtemp > thresh)
        count++ /* ring found */
    }
    if (start == 1) { /* can found */
      data[rowno] = count
      rowno++
    }
    else {
      if (rowno > 0) { /* can passed */
        for (j=0;j<rowno/2;j++)
          front += data[j]
        for (j=rowno/2;j<rowno;j++)
          back += data[j]
        if (front > back)
          signal('OK')
        else
          signal('ROTATE')
        rowno = 0
      }
    }
  }
  until break /* run until stopped */
}
```

In Fig 12 we show the rangels selected in the algorithm to find the orientation of the can in Fig 8 to Fig 10. In Fig 13 and Fig 14 we show the range data and selected rangels for another can type.

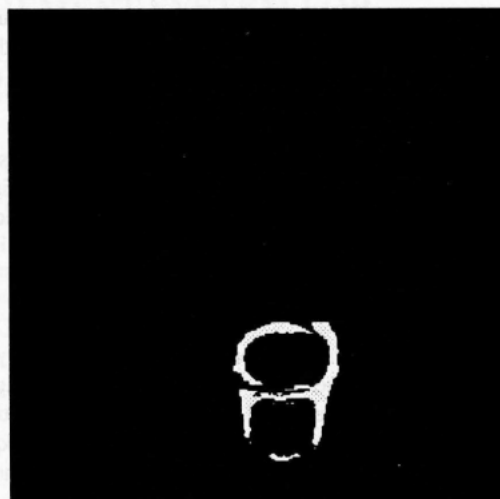


Fig. 12. The range data used to determine the orientation. We see that almost all selected data comes from the opener.

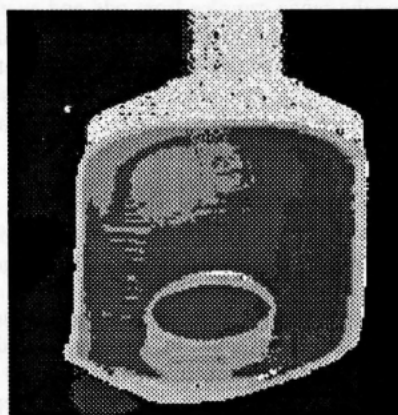


Fig. 13. Another example of a can image.



Fig. 14. The rangels selected as belonging to the opener.

V. RESULTS

The algorithm has been tested on three different can types of two different sizes, and it works without fault with up to ten cans/second. If we sample below 30 lines on each can the output starts to get uncertain.

If higher speeds are required several paths are possible. If the can speed is well defined we can image only the first half of the can and determine if that half contains the opener. This makes it possible to inspect cans with a smaller time interval since the post processing can be done while the last half of the cans pass.

If we know that the cans comes at a specific sideways position we can use fewer than 256 rangels in each image line. This does not alter the rangel frequency, and thus the time for each processed row decrease. If we use this technique the need for the FIFO buffer disappears since we can choose to view only the middle of the cans.

Another possibility to increase the row frequency is to interlace the range samples from odd and even sensor rows. This gives doubled range row frequency and theoretically doubles the scan direction bandwidth whereas the x direction bandwidth remains the same. Since the total sample density remains the same no total bandwidth gain is made. The bandwidth in the diagonal directions decrease.

If we use special hardware which can output instructions at maximum MAPP2200 clock frequency (4-8 MHz) we get at least five times better performance on the image acquisition.

Another possibility is to use one camera to control two adjacent production lines in parallel, but this sets high demands on the program. It is probably hard to implement this without sending the range data to two separate processors who controls one production line each.



VI. CONCLUSIONS

We have shown a real-time application which uses a sheet-of-light range imaging system to find the orientation of cans. The algorithm has so far been adapted for two sizes of cans and gives accurate results with as few as 30 lines of data per can. This allows a speed of approximately 10 cans per second in a smart camera implementation.

REFERENCES

- [1] Johannesson M, Åström A., "Sheet-of-light range imaging with MAPP2200," *Proc. of SCIA 93*, Tromsø 1993, pp 1283-1290.
- [2] Forchheimer R, Ingelhaug P, Jansson C, "MAPP2200, a second generation smart sensor," *SPIE vol 1659*, 1992.
- [3] "MAPP2200 technical description," *IVP Integrated Vision Products*, Linköping 1992.
- [4] Rowa P, "Automatic visual inspection and grading of lumber," *Seminar/Workshop on scanning technology and image processing on wood*, Skellefteå, 1992.
- [5] Johannesson M, "Sheet-of-light range imaging experiments with MAPP2200," *Report No. LITH-ISY-I-1401*, Linköping 1992.
- [6] Johannesson M., "Fast sheet-of-light range imaging with MAPP2200," *Unpublished*, Linköping 1993.