View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by IntechOpen

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

12

Real-Time 3-D Environment Capture Systems

Jens Kaszubiak, Robert Kuhn, Michael Tornow, Bernd Michaelis
Otto-von-Guericke-University of Magdeburg
Germany

1. Introduction

Real-time environment capture systems provide autonomous robotic machines and vehicles with vital information for interacting with their environments. Navigation and obstacle detection are key features of these systems. The systems are designed to capture objects, and establish their locations and approximate dimensions. These system capabilities are described in detail in this chapter. The ability to identify objects to facilitate manipulation or interaction is not the primary subject of investigation.

Environment capture systems can be designed with a number of sensors. However, information provided by many sensors is very limited. Very large amounts of information can be obtained with systems comprising cameras and a few sensors, as camera images can be processed in a great variety of ways. These systems have some issues, namely − very complex algorithms, high memory requirements for images, and limited real-time capabilities. However, rapid advances in micro-electronics are quickly addressing these issues.

In this chapter, we show how measurement methods based on stereophotogrammetry can be adapted and optimized for embedded systems. We will also deal in detail with key challenges and issues encountered in designing the systems. We provide experimental results for a number of typical applications.

2. 3-D Environment Capture Procedures

There is a host of sensor systems available that are suited for environment capture. They can be combined to compensate for different subsystem deficiencies and weaknesses. By combining a number of sensors, extensive information can be obtained from the surroundings.

Among the many environment capture systems available for vehicles are:

- radar systems
- ultrasonic devices
- laser systems
- active and passive optical measurement methods

Some measurement systems are based on the principle of the active propagation delay of an emitted signal. Ultrasonic sensors are suitable for short distance targets. These systems are useful indoors for measurement ranges of a few meters. Distances to objects can be measured very accurately (see Uhler et al., 2003). Ultrasound measurements taken outdoors can be disturbed by bad weather conditions − with the result that the detection range is

Open Access Database www.i-techonline.com

Source: Scene Reconstruction, Pose Estimation and Tracking, Book edited by: Rustam Stolkin,
ISBN 978-3-902613-06-6, pp.530, I-Tech, Vienna, Austria, June 2007

restricted and the reliability is impaired. Due to the narrow aperture angles many sensors are needed for fully comprehensive environment capture.

Radar sensors are deployed for objects at close range (24 GHz), and for objects at long distances (76 GHz). The aperture angle is very small. The surroundings can only be completely scanned and surveyed by panning the radar antenna (as in aeronautics), or by deploying a bank of radar sensors. Devices for ground-based vehicles are offered at this time for distances to objects ranging from a few meters to many hundred meters. However, these systems interfere with other equipment and pose a risk to living organisms. They are also expensive (Venhovens & Naab, 2000).

More recently, laser scanners are being adopted for use in environment capture systems. They also work on the propagation delay principle and scan and range a 3-d point at a particular point in time. These systems are quite accurate. For complete environment capture, a mechanically moveable deflection mirror is needed (Fuerstenberg et al., 2003). Distances to objects are of the order of a few meters and go up to some hundred meters. Object distances are limited in outdoor public areas by laser radiation and the risk of serious damage to biological eyes.

Active optical propagation delay techniques (Lange & Seitz, 2001; Tyrrel, 2004) have been under investigation for some years now. These techniques produce an extensive depth image with a single sensor. The Photonic Mixer Device (PMD) is one such system. In this system, a modulated optical signal is transmitted to a scene and reflections from the scene are captured by the elements of a matrix (preferably a CMOS sensor). This is similar to the way ultrasonic sensors work. The advantage of this system is that only one camera is needed. Quite dense disparity maps are generated, and they can be processed in real-time. One drawback with this technique is the high processing power needed. The maximum object distance is a function of the wavelength of the modulated signal, the optical transmit capacity brought to bear on the scene, and sensor characteristics. This method of measurement is a very recent innovation and has a lot of potential.

Images are processed directly in many optical systems. The texture of the object is overlaid with additional information in active imaging systems. Distance and surface information can be determined to a high degree of accuracy with the help of fringe projection.

Accuracies on the order of µm can be achieved for limited observation spaces with active systems comprising a number of cameras. Numerous applications for these techniques can be found in the fields of automation, quality assurance, medical systems, and, to a lesser extent, security systems. They are not widely deployed for environment capture.

When a number of active systems are deployed together there is always a risk that they will interfere with each other. Systems with large aperture angles, or systems designed to range over great distances, sometimes emit very powerful light. This restricts their use.

3-d data can also be acquired with the help of passive optical systems in systems comprising more than one camera. The simplest case of multi-camera systems, i.e. stereophotogrammetry, comprises two cameras. This dual-camera constellation generates depth images.

Environment capture systems do not need to be very accurate. However, a large observation space has to be scanned and surveyed rapidly. Passive systems, such as stereophotogrammetric measuring systems, are suited for these types of applications (Knoeppel et al., 2000). The images provide information on the entire scene being scanned.

The camera system is the key to adapting a stereophotogrammetric system for different applications. The arrangement of the cameras, the base, aperture angle, and the resolution of the cameras determine the measurement range. The normal case of stereophotogrammetry, whereby the cameras are arranged parallel to each other, is suited for environment capture over large distances.
In the next section we describe what we consider to be the key algorithms and parameters needed for a compact 3-d environment capture solution based on a stereo camera system.

3. Function and Setup of Stereo Camera Systems for Generating 3-D Depth Information

The coordinates of a point in 3-d space are determined using a stereo camera system by mapping the object point in two camera images taken from different angles. When a point has been detected in both camera images, the 3-d coordinates of the point are calculated using basic geometrical functions.

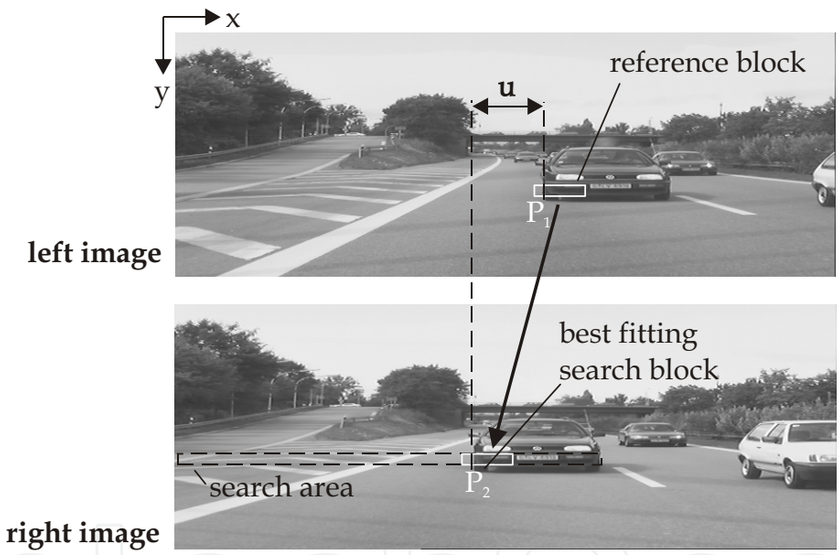


Figure 1. Basic principles of block matching

Typically, a 3-d stereo camera system consists of two calibrated cameras connected together rigidly, whereby the clearance between them is the base.
The detection of the object point in the two images (generally known as the correspondence search) is typically nontrivial. There are a number of detection solutions available.
One option is to attach suitable indicator marks to the target. These marks are detected in the image by segmenting and measuring punctiform patterns. It is normally impossible or impracticable to attach separate markers to targets in the applications at hand. Typically one has to use the object texture instead.

In correspondence analysis, a section of the first image (reference block, position P_1 in fig. 1) and a block of the same size (search block, position P_2 in fig. 1) in the second image are isolated. The similarity between these two blocks is then calculated (see fig. 1). Due to the geometry of the shot (the two projection centers and the object point form a plane that can be intersected by the image plane – also known as the epipolar line), the corresponding point can only be located on a specific line in the search image. The similarity for every point along this line is then determined (see fig. 2). The location of the extremum (= offset u of the features with respect to each other) is the position of the corresponding point.

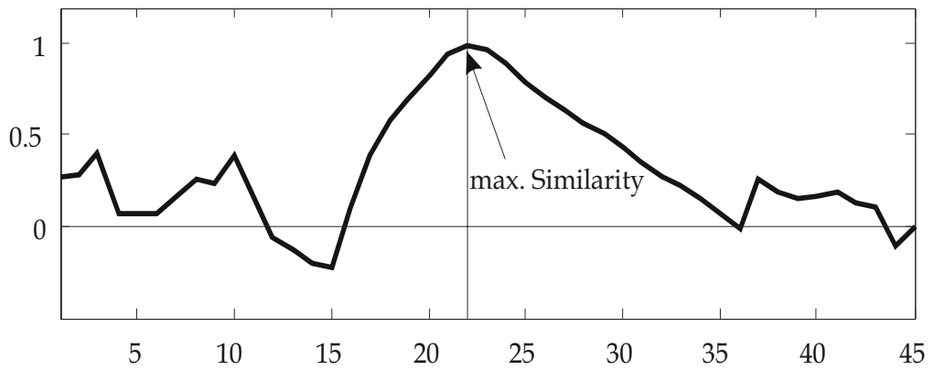


Figure 2. A typical correlation function (NCCF)

Standard correlation functions such as the Sum of Absolute Differences (SAD; eq. 1) or the Normalized Cross Correlation Function (NCCF; eq. 2) can be invoked to calculate the similarity. The quality of the results provided by these two functions differs; the numerical overheads also differ. The SAD is calculated for each pixel in the search window by computing the correlation value using

$$SAD(\xi,\eta) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|F(i,j) - P(\xi+i,\eta+j)\right| \tag{1}$$

$F(j,i)$ - *pixel in the search block*

$P(j,i)$ - *pixel in reference block*

M, n - window size

ξ,η - *displacement in x, y- direction*

F and P represent $m \times n$ large image cut-outs from the reference and search images. The SAD function is very sensitive to brightness variations and does not always give useful results. The NCCF function produces better results but they are numerically more complex to compute :

$$NCCF(\xi,\eta) = \frac{\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\left(\overline{F(i,j)} \cdot \overline{P_r(\xi+i,\eta+j)}\right)}{\sqrt{\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\overline{F(i,j)}^2 \cdot \sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\overline{P_r(\xi+i,\eta+j)}^2}} \tag{2}$$

$\overline{F(j,i)}$ - *zero-mean pixel in search block*

$\overline{P(j,i)}$ - *zero-mean pixel in reference block*

It is difficult to analyze the correspondence for objects such as white walls with little or no texture of their own. In these cases a preliminary search should be made for edges or image sections that have sufficient texture. The correspondence search should only be made at these locations. Other areas are removed or interpolated. This can also be beneficial, because it thins out the 3-d point cloud.

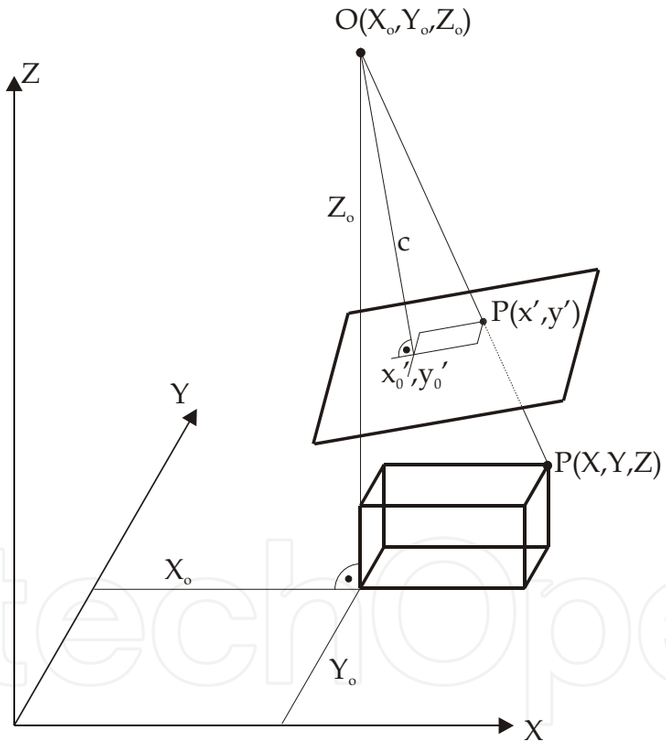


Figure 3. Elements of a camera orientation

When the object point has been detected in both images, the 3-d coordinates of the point are calculated using the collinearity equations (Schenk, 1999) (see also fig. 3):

$$x' = x'_0 - c_k \frac{a_{11}(X - X_0) + a_{21}(Y - Y_0) + a_{31}(Z - Z_0)}{a_{13}(X - X_0) + a_{23}(Y - Y_0) + a_{33}(Z - Z_0)} \tag{3}$$

$$y' = y'_0 - c_k \frac{a_{12}(X - X_0) + a_{22}(Y - Y_0) + a_{32}(Z - Z_0)}{a_{13}(X - X_0) + a_{23}(Y - Y_0) + a_{33}(Z - Z_0)} \tag{4}$$

The coordinates x' and y' are the image coordinates, and X, Y and Z are the object coordinates of the observed point (see fig. 3). The other elements in the equation define the internal and external orientation of the camera. The constant c_k is the calibrated focal length, and x_0' and y_0' the principal point. This point is defined as the point of intersection of the optical axis and the image plane. Terms for the radial and tangential distortion are examples of other elements in the internal orientation. The six elements in the external orientation describe the location of the camera (X_0, Y_0, Z_0) and its viewing angle (a_{xx} are the elements of the rotation matrix). Camera calibration is complete once these values have been determined.

Typically, the calibration data is produced using a bundle block adjustment. A calibration plate with known points is required for this procedure. The nine unknown parameters for eq. 3 and eq. 4 are then determined by adjusting the intermediate observations. The relationship between observations (image coordinates) and unknowns (calibration) is nonlinear. Approximate values need to be assigned to the unknowns. This is often a very difficult and time-consuming exercise! Once the calibration data are known, the mapped object point can be calculated from the image coordinates by re-arranging eq. 3 and eq. 4.

The normal case of stereophotogrammetry is a special case of a stereo system. The cameras are arranged so that both image planes are in the same plane, and the camera axes run parallel to one another. Determining the object coordinates is then simplified to a beam intersection problem and is thus more straightforward than the general case. The formulas for calculating the 3-d coordinates then become:

$$X = x' \cdot \frac{B}{u}; \quad Y = y' \cdot \frac{B}{u}; \quad Z = c_k \cdot \frac{B}{u} \tag{5}$$

The base (i.e. the clearance between the two cameras) is represented by B. The disparity u is the offset of the same object point in the two camera image planes (fig. 1). The image coordinates of the point in the reference image are x' and y'.

Another advantage of the normal case is the simplified correspondence analysis. This can be restricted to the pixel line in the search image, without having to calculate the epipolar line separately (see also fig.1). The systematic errors in the camera system can be compensated by using the standard camera model (eq. 3 and eq. 4). The corrections ΔZ and ΔX can be calculated with $\Delta Z = d \cdot Z^2 + e \cdot Z + f$ and $\Delta X = g \cdot Z + h \cdot X + i$ ($d,...,i \in \Re$). Combining ΔZ *and* ΔX *with* eq. 1 yields

$$Z = \frac{k}{u^2} + \frac{l}{u} + m \quad (k, l, m \in \Re) \tag{6}$$

Coefficients *k*, *l* and *m* only have to be acquired for eq. 6 during calibration. There is no need to determine base and focal length. The derivations for X and Y are similar (see Albertz & Kreiling, 1989).

Rectification has to be performed in order to apply the algorithms from the normal case of stereophotogrammetry to a general camera set-up. The original images taken by the camera in the general camera set-up are transformed so that they are the same as an image in the normal stereo case. A virtual image plane, in the same location as the normal case, is calculated. The original images are then converted to this plane. We can derive the transformation from the calibration parameters. The conversion can be carried out on special image processing hardware during imaging.

The accuracy of a stereo camera system is a function of the geometry of the imaging configuration and the image processing accuracy. System accuracy can often be estimated successfully by applying the laws of error propagation to the formulas for the normal case (eq. 5). The mean error σ_Z in the typically predominant Z direction then becomes

$$\sigma_z = \frac{Z^2}{c \cdot B} \sigma_u \tag{7}$$

Better accuracies are achieved the shorter the distance to the object, the larger the camera constant, the wider the camera base, and the more accurate the image coordinate calculations are.

The resolution of the cameras yields a quasi quantization of the measurable distance. The disparity is less than a pixel width at a distance to the object that is greater than a specific threshold. The resolution can also be improved with a subpixel interpolation function. The optics − and thus system accuracy − are also affected by air. Camera aperture angles determine the system viewing range. However, large aperture angles give rise to distortions, that have to be corrected by non-linear correction terms in eq. 3 and 4 (El-Melegy & Farag, 2003). This is always necessary with precision measurements.

The above image processing algorithms have to be implemented and evaluated. The various implementations are described in the next section.

4. Designing an Embedded Stereo System

4.1 Hardware Architectures for Information Processing

Image processing algorithms may be implemented on standard PCs or PC clusters. This option is very common due to the good availability of state-of-the-art high performance PCs. These computer systems can be deployed universally in a range of applications due to their considerable computing power.

They are not so useful in mobile applications because of the low space and performance requirements of these applications. Specially developed embedded systems are favored in these situations. The embedded hardware is often only suited for specific applications. Often, PCs or embedded processors are not powerful enough for real-time image processing due to their architecture and bus systems. A number of processor elements, connected in parallel or in a pipeline is a more suitable arrangement. Processor elements can be complete processors or special hardware elements. Options for different hardware structures are shown in fig. 4.

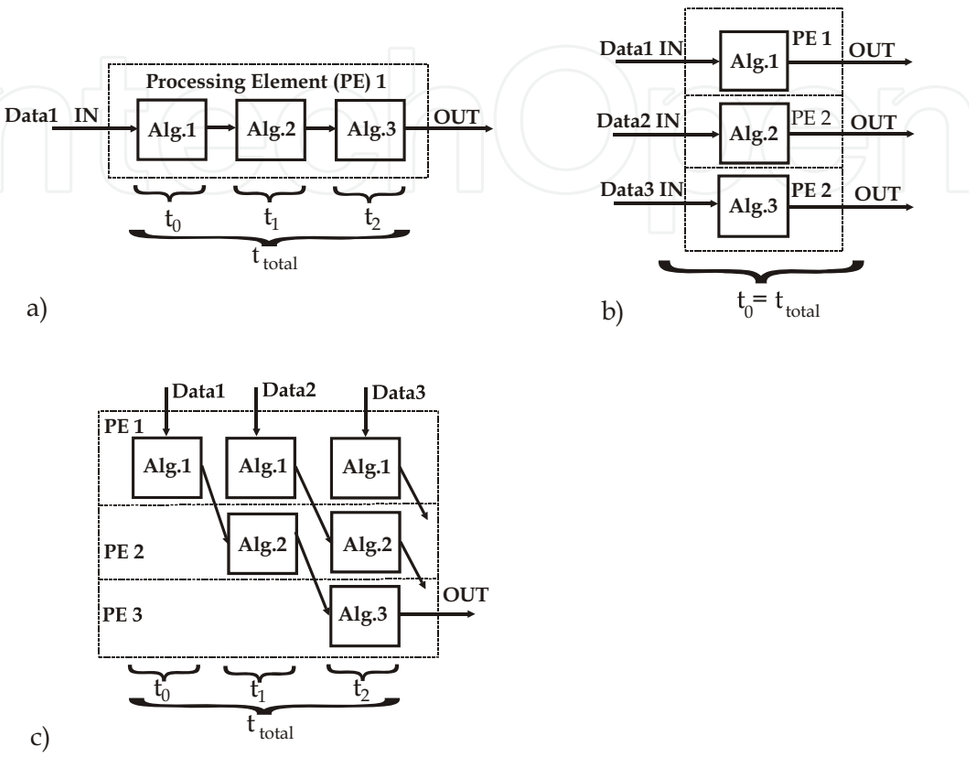


Figure 4. Processing principles a) Serial b) Parallel c) Pipeline

A single processor system is shown in fig 4 a). The algorithms are processed chronologically and in series by a single processor. This processing model is called Computing in Time. The data throughput per unit time is increased in comparison to fig. 4 a) by the parallel processing in fig. 4 b) and c). The processing elements may be fully fledged processors, or specially developed hardware. FPGAs are available for hardware development in low-volume production. FPGAs are programmable logic chips. They may also be described as special processors, where a program may be implemented and executed in hardware.

A number of processing operations may be executed simultaneously in a single processing step in the hardware logic. The algorithm is data-flow oriented, i.e. it is continuously compiled and executed as a single instruction in structured logic. This programming model is distributed spatially and is known as Computing in Space (fig. 4 b) and 4 c)).

Data-flow oriented algorithms are best implemented in hardware, whereas control-flow oriented algorithms run better on a single processor or processor system. Data-flow oriented algorithms can process high data rates. They consist of basic operations and there are few logic branches in the data flow. On the other hand, control-flow oriented algorithms can process small quantities of data only, but with a very complex data flow.

4.2 Data Flow in the Image Processing System

The standard data flow model of an image processing algorithm (Noelle, 1996) is shown in fig. 5. There is a tendency to use data-flow oriented algorithms for preprocessing, and control-flow oriented algorithms for feature extraction, and interpretation & classification.

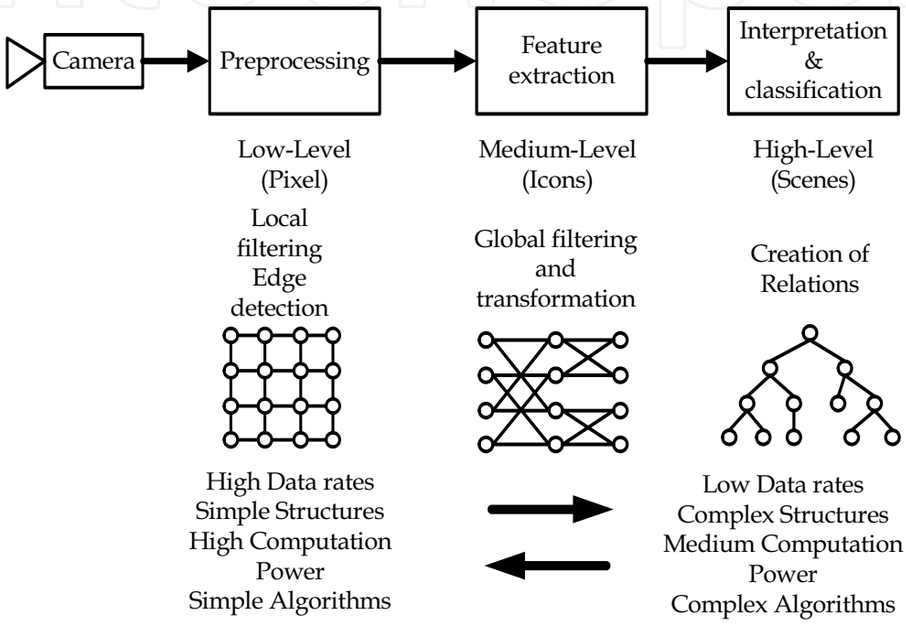


Figure 5. Data flow model of image processing algorithms (Noelle, 1996)

Preprocessing is often implemented in hardware logic. Interpretation & classification is typically run on processors as shown in fig. 5. This system of hardware logic and processors is also referred to as hardware-software co-design (Gupta & Rajesh, 1995). The developer decides which algorithms to implement in hardware and which in software, or the structure of the hardware-software co-design is automatically selected by a separate algorithm.

A typical image processing algorithm implemented as a hardware-software co-design for a stereo system is shown in fig. 6 (Kaszubiak et al., 2005). The preprocessing stage comprising the correlation and edge detection functions, and the subpixel interpolation function are implemented in hardware. The implementation on a single processor is not efficient enough, due to the architecture of the data bus systems which cause bottlenecks when images are read in real time. Knoeppel (Knoeppel et al., 2000) presented this type of system. Implementation in hardware requires that a data-flow oriented solution is developed for real-time data processing.

The object detection step using a depth histogram (see section 4.4) can also be viewed as a data-flow oriented algorithm. It is therefore also implemented in the FPGA hardware. The downstream stages in the processing chain are implemented as a pipeline on three processors, because of the many control-flow oriented structures.

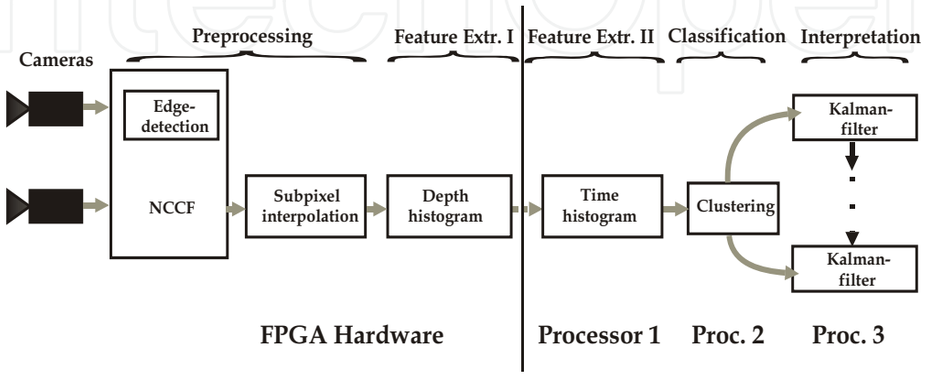


Figure 6. Algorithm as a hardware-software co-design

We can achieve real-time processing conditions for a stereo image processing system as shown in fig. 6 with a 1024x500 pixel resolution and a 25 Hz image rate by implementing the hardware-software co-design in an embedded system. We will now describe how the different parts of the algorithm are adapted and implemented as an embedded system as shown in fig. 6.

To detect objects with the very limited resources of an embedded system, a data reduction stage with a straightforward algorithm is needed. The first step is to acquire the 3-d points/disparity map with the help of stereo image analysis. We will discuss the implementation of these algorithms in hardware in the next section.

4.3 Optimizing and Implementing the Algorithms for Generating the 3-D Disparity Map in Hardware

4.3.1 Steps for Reducing Computation Costs

The input data for generating the 3-d disparity map is a sequence of stereo images containing a lot of information that is not vital for the application.

The strategy for system optimization and thus the reduction in computation costs for determining the disparity map consists of the application of a suitable similarity criterion, the pre-selection of relevant image sections, data organization and flow, and improving the input data.

When the lighting conditions are good, simple similarity criteria are used. The simple structure of the SAD function (eq. 1) and its simple mathematical elements make it the function of choice for many applications. It is also processed at a higher speed than the NCCF (eq. 2). However, since the SAD is not very reliable, a preprocessing step or extra criteria are needed for reliable results. The benefits of speed in some applications

compensates for some of the disadvantages of the SAD function. Nevertheless, the NCCF has proven itself to be a reliable criterion for many cases of environment capture.

By rectifying the images, image data processing could be simplified, but the rectification task has the disadvantage of high computation costs. To save computation costs we can avoid fully rectifying the images to produce the epipolar condition by applying correction equations (eq. 6) as described in section 3, especially if we can align the cameras accurately. The objects being captured are mainly other vehicles, houses, bridges, and people; and they all have long vertical edges, which allows a number of lines to be averaged in the vertical direction − thus keeping system calibration work to a minimum. A slight rotation of the camera can be compensated for by averaging 2-4 lines and combining them to a single new line.

We can perform the correction step and the step for calculating the 3-d coordinates in different parts of the system. We can optimize the computation costs by determining the 3-d coordinates for the center point of scanned objects only. In order to reduce computational overheads the correction step is run for the necessary points only.

The computation power needed to fully compute all blocks in a line is very high. This is, however, necessary for continuous image processing over a longer period.

Environment capture systems can be divided into two major groups according to their detection ranges:

- imaging systems with a wide aperture angle for objects at close range
- systems with a narrow aperture angle for objects at greater distances (as needed for a lane change assistant (see section 5.2))

We have developed an optimized algorithm for this latter case.

4.3.2 Hierarchical Search Algorithm

We used the error characteristic (eq. 7) in determining the location over the detection range in this algorithm. It is desirable to have a constant relative error over the entire detection range in many environment capture applications. We optimize the area correlation algorithm by means of an image pyramid (Tornow et al. 2003) for implementation in hardware. We make use of the fact that the maximum camera resolution is only needed at the greatest object distance, whereas the available resolution at close range is more a hindrance than a benefit due to the large disparity.

We reduce computation costs by producing image layers with different resolutions. The layers in the pyramids are ranked by factor 2 (see fig. 7). We have to adapt the accuracy for locating distant objects, as the error in determining the distance is a function of the square of the distance. This means that there is considerable redundancy available for the measurement accuracy for objects at close range. We generate each layer by replacing two pixels by their mean value. L_0 is the image taken with the original resolution. The individual layers are arranged so that the correlation method described in section 3 can be applied in all resolution layers in the same manner.

We can thus achieve a large detection range with an approximately constant relative error using a very small search block in each layer. Only one specific distance range is represented by each layer in the pyramid. We can cover the entire detection range by summarizing the data from all layers, as shown in fig. 8.

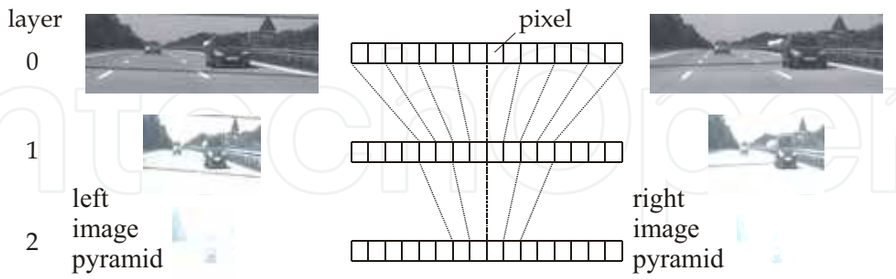


Figure 7. Generating the layers

Calculating the root of NCCF (eq. 2) poses problems in the hardware implementation. The squared NCCF can be used as an alternative, as only the position of the extremum is relevant. All negative values are set to zero.

We now establish the locations of the maxima above a predefined threshold in the resulting search block. Only maxima corresponding with object features (object edges) are processed. The disparity with the greatest weighted correlation value is then selected for the reference block from all layers (see below). Disparities from layers with reduced resolution have to be recalculated to the original resolution.

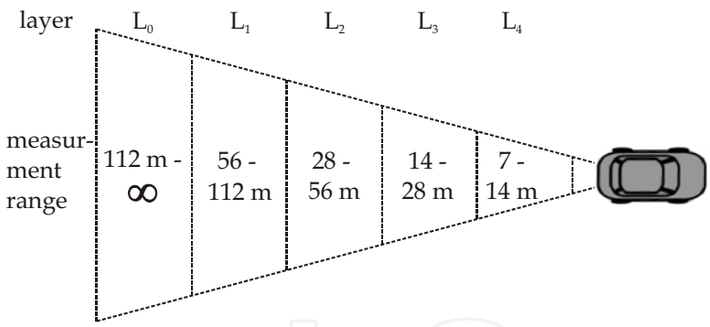


Figure 8. Distribution of the entire range for the application in section 5.2

Having created a disparity map for each layer, we append this information to the resulting disparity map. We now select the significant layer for each 16-pixel block in the original resolution. We search for the extremum of the correlation values in the stacked blocks (fig. 9) in the layers.

Objects that are very close only produce a response that is above the threshold in the layer with the lowest resolution. The further away the objects are, the more layers respond. Different layer resolutions generate results with different accuracies. We have to introduce a penalty to ensure the best result is always selected. Thus, the higher resolution layer always wins.

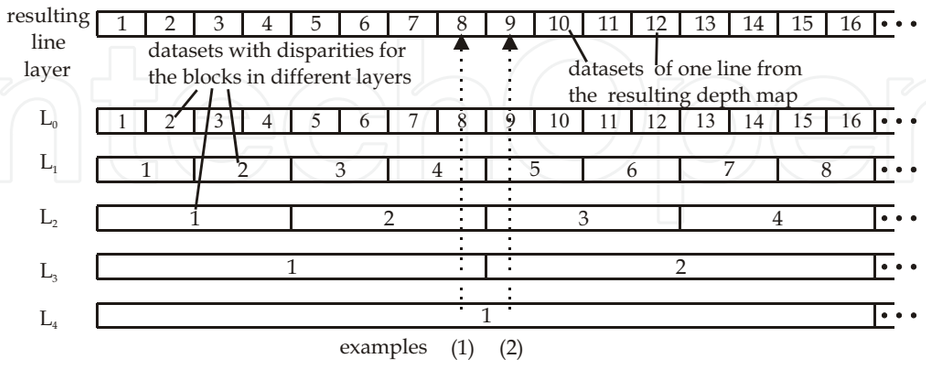


Figure 9. Assembling the layers

Disparities u_{layer} and x-coordinates x_{layer} in the disparity map are only valid for the layer they were calculated in. Disparities u and coordinates x′ in the original resolution in the stereo images are needed for calculating the object coordinates from the disparity and the image coordinates with eq. 5. They can be calculated with eq. 9 and eq. 10. The index or the exponent layer is the associated number. The value s, the difference between the center points of two neighboring reference blocks, and v_{layer}, the position-related number of the reference block in the layer, are needed to calculate x_{layer}. The following relationship applies:

$$x_{layer} = v_{layer} \cdot s \tag{8}$$

The window size for the reference and search blocks is defined in section 3 by m x n. Thus the middle of the reference block is represented by m/2. Thus

$$x' = (x_{layer} + \frac{m}{2}) \cdot 2^{layer} \tag{9}$$

and

$$u = (u_{layer} + u_{\min}) \cdot 2^{layer} \tag{10}$$

u_{min} is an offset in eq. 10 and it represents the minimum disparity. All values except v_{layer} are measured in pixels.

The maximum effect of implementing this procedure in hardware is achieved when the size of the reference block m is equal to the search block. In this case the costly methods for loading the output data for the correspondence analysis are no longer needed, and the maximum clock rate can be lowered significantly. This yields 16 correlation results for the 16x1 pixel block size in our application. We were able to remove disparities smaller than 8 pixels in all layers to reduce the data further, as we are only dealing with objects at distances up to 150 m.

The disparity is then calculated to subpixel accuracy by means of quadratic interpolation (Tornow et al., 2006). We then determine the 3-d coordinates using eq. 5 and add them to the resulting disparity map.
If a standard area correlation based on the epipolar geometry up to a maximum disparity of 256 pixels is executed, then the processed data volume and thus the necessary clock rate are increased sixteen times due to the great number of block combinations. By running the hierarchical algorithm with 5 layers and meeting the same requirements the data volume is only doubled. Thus, the correlator only runs at double the pixel clock for continuous real-time processing. This optimization means that a 3-d disparity map (fig. 10.) is calculated quasi simultaneously with the image acquisition from the stereo image pairs.
We must now extract the necessary information from the disparity map. The information we need to take from the disparity map differs from application to application. We discuss some typical applications in section 5.

4.4 High-Level Processing

4.4.1 Object Detection

Objects with closed contours are detected and an attempt is made to classify them.
We search for points belonging to an object and assign them to an object cluster. We will briefly explain selected clustering algorithms and present a typical application. 3-d points can be clustered in accordance with their spatial relationships (segmenting algorithm). We can also represent in a histogram the relationship between the points in the disparity map (condensation algorithm). There is another algorithm that finds the collinear points (Hough transform) belonging to an object (see section 4.4.3).
Segmenting algorithm (Knoeppel et al., 2000)**:** There is a nonlinear relationship between the disparity map coordinates x′,y′ and u, and the real coordinates X,Y,Z according to eq. 5. All image coordinates have to be converted to real coordinates for the geometrical algorithm. The different 3-d points can then be correlated spatially with one another, thus locating 3-d points that are spatially related. Many different criteria may apply, namely target shape (triangle, square, circle) or distance to an object. Features are detected using a feature space, where the features are entered for every point. 3-d points are clustered in the feature space. These points can be assigned to a specific object. Very good a-priori knowledge is often needed to segment objects. It can be a very difficult task to span a unique feature space. The segmenting algorithms are therefore only suited for a small number of different objects that can be uniquely identified, such as traffic signs (Fang et al., 2003).
Condensation algorithm (Dellaert et al., 1999): The condensation algorithm may be better suited than the segmenting algorithm for detecting raised objects or edges. This is the case when objects and edges that are fixed on a plane have to be located (see also 4.1.1. 4.1.2). However, one cannot differentiate between the objects of a given category. The vehicle category can be detected very well on a road with this algorithm. The different car models cannot be separated into subcategories with the condensation algorithm, as too few features are processed. This algorithm is based on a histogram that allows the vertical edges for the objects contained in the image to be located, as the 3-d points on an edge are located at the same distance away.

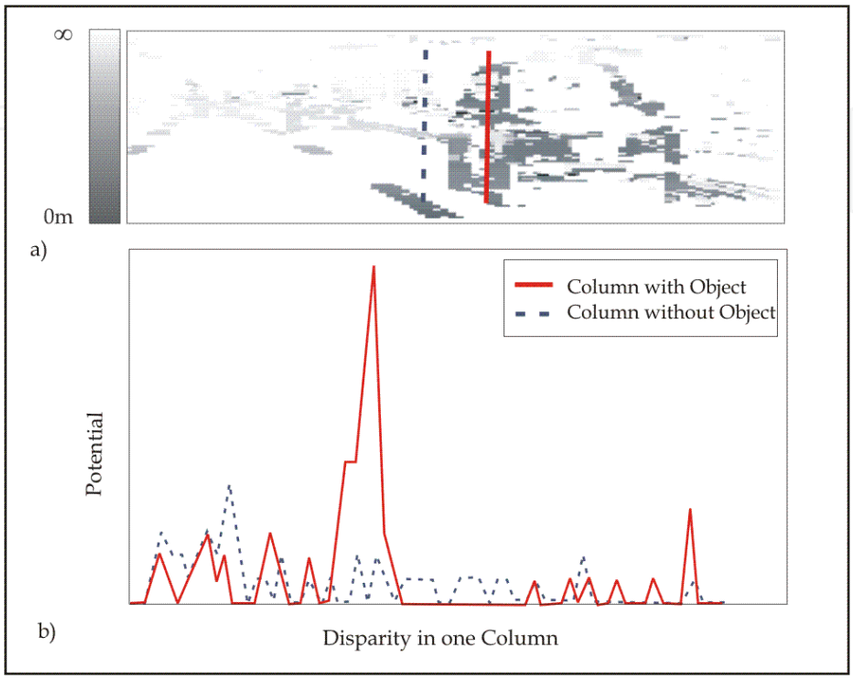


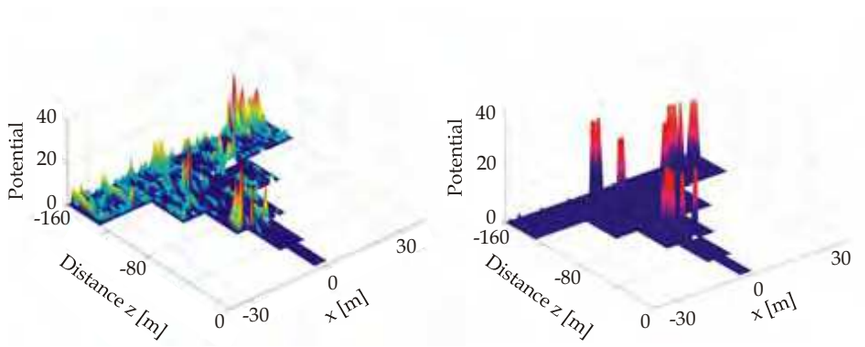
Figure 10. Condensation algorithm a) Depth map b) Potential view along the column

A detected 3-d point produces a potential on the plane. If another 3-d point is found at the same distance (vertical edge), the potential increases accordingly. Fig. 10 a) shows a disparity map, and fig. 10 b) a section view of the generated histogram along the columns in fig. 10 a).

A raised object is detected when the potential reaches a predefined threshold. The potential only indicates the height and location of a detected object, but gives no detailed information on its shape. The advantage of the condensation algorithm is the rapid speed at which it processes the 3-d points (it uses only the distance and potential of a specific position to process the scene).

We will describe a typical condensation algorithm and discuss its benefits and advantages in an embedded system. Approaching objects are detected on a plane and their speed is determined. The camera system itself is also moving. Examples of this type of application are vehicle detection (Kaszubiak et. al., 2005) and the detection of pedestrians (Gavrila, 2004). Complex search operations are executed by the clustering and tracking algorithms − making them control-flow oriented algorithms. They run on processors.

The condensation algorithm is deployed as a clustering algorithm to detect raised objects on a plane. Raised objects on this plane are not only approaching objects, but can also be objects that are not moving or objects that are moving in the opposite direction. These objects are filtered out with the help of two histograms (fig. 11).



a) b)

Figure 11. a) Depth histogram b) Time histogram

The depth histogram is generated from the disparity map. 3-d points at the same distance and lateral position (vertical edge) are accumulated. The non-linear relationship from eq. 5 is not invoked for generating the depth histogram, as we are dealing with vertical edges only. This means that we can generate the depth histogram solely with the image coordinates (x′,y′,u). Thus we can avoid using a large number of calculations to generate the 3-d points. The variables x′ and u serve also as addresses for a memory cell in the histogram. The memory cell is an accumulator that totals the number of accesses to this memory cell per image. Fig. 11 a) shows the depth histogram for the disparity map in fig. 10 a). One depth histogram is generated per image. Raised objects in the depth histogram are found with a threshold.

The time histogram (fig. 11 b) is structurally the same as the depth histogram. Vertical edges found in the depth histogram are tracked in the time histogram. The current depth histogram is compared with the time histogram. A search is made in the time histogram for a maximum at the location of a maximum in the depth histogram, or in a search area in the vicinity of this location in the previous image.

The accumulator entry at this position indicates the age of the edge. If an edge is found in the search block, the accumulator entry is set at the location of the point in the current depth histogram and incremented. Raised objects only are detected with this search box if they are objects that are located at the same distance from the observer's vehicle in a number of images, or if they are objects that are approaching the observer's vehicle. If the point in the previous image belonged to a cluster, the number of this cluster is also stored in the new image. The age of entries in the time histogram that have no corresponding maxima in the current depth histogram is decremented. Thus previously detected objects can disappear from the time histogram.

Clustering is based on the time histogram. Values in the time histogram that are greater than a specific threshold are assigned to different object clusters. The closer the objects are to the cameras, the larger they appear in the image sensor. This means that distances between left and right vehicle edges increase in the image.

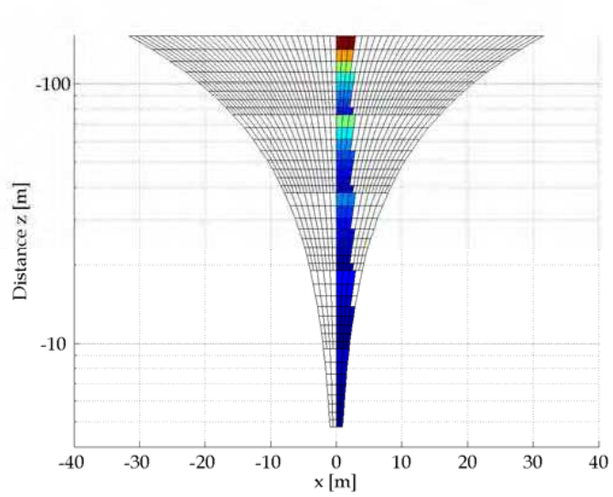


Figure 12. Mapped width of an object in the histogram as a function of the distance

We distribute the depths in a hierarchy and generate the layers as described in section 4.3.2. This allows us to keep the clearances between object edges within an almost constant range. Fig. 12 shows the range for a 1.80 m wide object in the histogram. We can see from the figure that the lateral spread of the object over the entire depth is effectively constant. By reducing the resolution we keep object edges very close together so they can be detected easily.
We do not need to calculate the 3-d edge coordinates because we cluster the objects in the histogram. We calculate one 3-d edge only for every detected and clustered object. This 3-d edge is the center point of the object cluster. Thus we reduce considerably the number of calculations needed. This straightforward clustering technique, based on the hierarchical depth map, also reduces the number of computations. This type of algorithm is very suitable for implementation in an embedded system.

4.4.2 Tracking

Disturbances during image acquisition and quantization of the hierarchical depth (section 4.3.2) cause jumps in the distance measurements. These jumps have to be smoothed in order to determine object speeds. A Kalman filter is used to smooth the jumps (S. Lee & Y. Kay, 1990). The Kalman filter is an ideal, recursive data processing algorithm that allows us to determine object speeds without delay and after an initial settling period. Let us consider an object at a distance of 150m behind the camera system as an example. We wish to detect this object and track it to a distance of 10m. The object is traveling at a uniform speed of 45km/h. It accelerates to 65km/h at a distance of 100m. The Kalman filter algorithm smoothes and estimates the speed as shown in fig. 13. Distance and speed are plotted as negative values, as the object moves toward the camera system from behind.

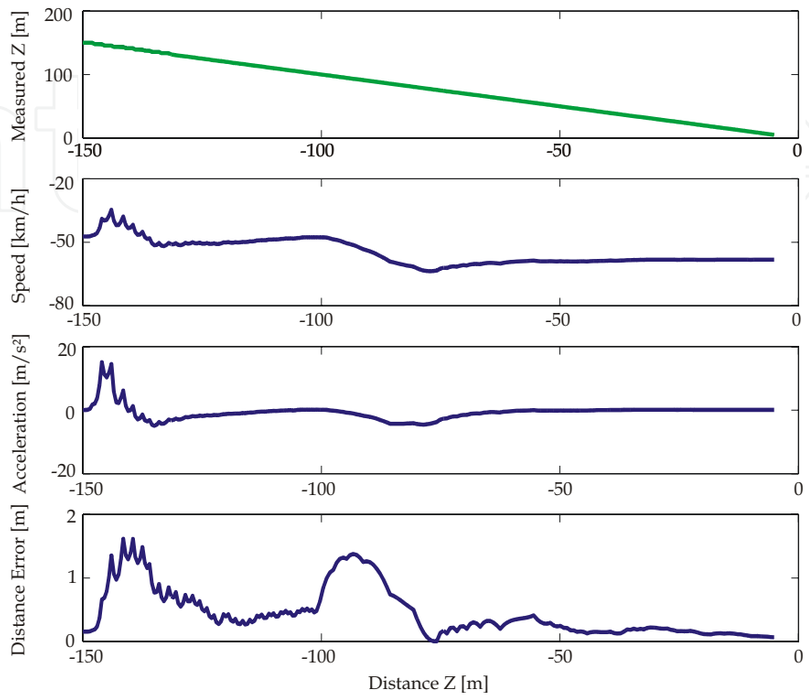


Figure 13. Speed and distance estimation with the Kalman filter

4.4.3 Detecting Boundaries

We often need to detect the boundaries of the planes on which the objects move. We tackle this task using the **Hough transform** (Kluge & Lakshmanan, 1995). It enables us to find straight lines located at different angles in the image, e.g. plane boundaries. Among the items we are looking for are lane markings, skirting boards, markings on playing fields, and the like.

The Hough transform typically executes in the camera image. However, in some applications it can be run inside the disparity map, the depth histogram or the 3-d space. A basic element of the Hough transform is the description of a straight line in the Hessian normal form (eq. 11)

$$r = X \cdot \cos\varphi + Y \cdot \sin\varphi \tag{11}$$

A line bundle is placed on the point (fig. 14). A radius r and an angle φ are associated with each line. The (r,φ)-coordinates points to the accumulator cells in a (r,φ) histogram. All points on the same line in the (X,Y) space have one line in their line bundle with the same (r,φ) coordinates. These lines are then accumulated in the (r,φ) space and a histogram is generated (this procedure is the same as for the condensation algorithm). High histogram entries indicate a line at this location in the image. We can search for the associated points along these lines by executing an inverse transformation in the (X,Y) space.

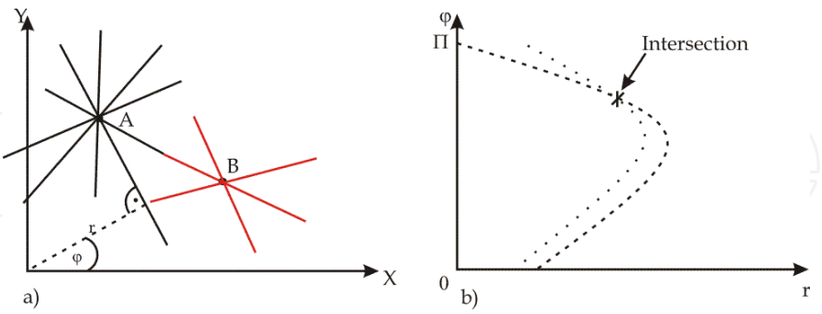


Figure 14. Hough transform a) Line bundle in the X, Y system b) Line bundle in the r,φ system

Again here we can reduce the computation overhead by running the Hough transform in the depth histogram. We only need to generate the line bundle in a specific area because lines that are parallel with the line of viewing only appear in the image.
For example, we could then restrict the angles of the generated line bundle to a 0° to 50° range for lines to the left of the camera system and to a 310° to 360° range for lines to the right of the camera system. This would produce the histogram in the Hough space as shown in fig. 15.

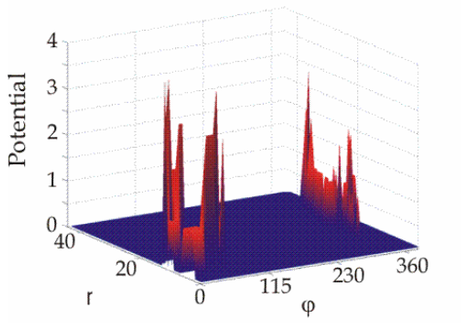


Figure 15. Accumulator in the Hough space

The algorithms outlined above are all suited for numerous applications in robotics, autonomous vehicles, and in road traffic situations. We now present a number of examples.

5. Typical Applications

5.1 Stereo Applications in Road Traffic

Stereo-image analysis is deployed on roads for a variety of purposes. The algorithms encountered here can be divided into two main categories:

- feature-based algorithms (direct triangulation of special features)
- correlation algorithms for determining corresponding points in the images

Combinations of these two categories also exist. There is a wide spectrum of different quality criteria, that differ greatly in computational costs and sensitivity to image

information (see also section 3). Recent research in graph theory is producing very promising techniques for highly structured scenes. The optical flow is also applied in some cases. Great care needs to be taken in selecting the right methods. Many examples are to be found in the literature.

[Saneyoshi, 1996] detects vehicles with a stereo camera system using a SAD criterion implemented in hardware. The maximum object distance is very restricted due to the low camera resolution. The system does not run in real time despite fast processing speeds. Lane and road intersection detection at close range in software is presented.

A stereo system is described for detection and tracking in urban traffic in [Franke et al., 1997]. He deploys detected information on the surroundings for route planning for autonomous vehicles. Franke detects small features (that are restricted to corners and edges) in the image for the correspondence search. Each feature is marked with a typical code word that indicates, for example, whether a feature is a bottom left corner or top right corner etc. Code words only are compared to save CPU time during the correspondence search.

[Stiller et al., 1998] introduces special correspondence hypotheses to improve the recognition capability of vehicles in real road traffic situations. A number of quality scores and road parameters are assigned to each correspondence hypothesis.

[Yoshika et al., 1999] deploys a low resolution stereo camera module (16x46 pixels) for detecting vehicles in the blind spot area. The system has special hardware for the correspondence search. The stereo module produces a 3-d point for every pixel. No details are given as to how the correspondences are determined.

Subaru has offered a stereo-camera-based Adaptive Cruise Control (ACC) system since 2000. A 4x4 correlation is calculated in a special hardware. The size of the control unit is 35x20 cm without cameras. The camera is a 640x256 pixel CCD camera. The cycle time is 100ms.

The Acadia vision processor from a company called Sarnoff is another solution. It has a special integral stereo unit on chip. The system is available as a PC development environment. Camera integration is not part of the offering.

The company called 3-d-IP (www.3d-ip.com) offers an FPGA implementation of a stereo image analysis based on artificial neural networks along with a stereo head.

The Point Grey Research company has very recently launched the Bumblebee system with continuous hardware-supported stereo image analysis on the market. The system consists of a very compact module with an integral camera system with a fixed base. The CCD camera resolution is low. The system is very suited for close-range work indoors. However, it is only of limited use outdoors due to the integrated CCD cameras.

We will now present the experimental results of our investigations, based on our brief summary of global state-of-the-art technology, and the description of our own in-house design and development work in sections two to four.

5.2 Experimental Results for Embedded Solutions

The focus of our investigations is a driver assistance system which is a lane change assistant that observes the area behind the vehicle and determines the speed of, and distance to, observed vehicles with reference to the speed of the observer vehicle. The system also checks whether the drivers of the observer vehicles has sufficient time to change lanes, and alerts him if there is a risk of collision.

The detection range of 10-150m was choosen to match driver reaction times and the high speeds encountered on highways. The field of vision is small. Lenses with a 30° aperture angle and a 25mm focal length are used. It was possible to apply the normal case of stereophotogrammetry.
CMOS cameras are better suited than CCD cameras for outdoor applications – they have a wide dynamic range for brightness, and are not unduly affected by glare. The application requires a very high line resolution. The column resolution may be relatively low. Color is in fact more of a hindrance than a benefit for detecting vehicles, as the results from a simple algorithm can be corrupted by a multitude of color information. More complex algorithms use up more computation power. We had only CMOS cameras with a resolution of 1024x1024 at our disposal. This forced us to increase the base of the measuring system to 70cm. This allowed us to achieve a 1% relative error for the static case. An error of this order of magnitude is satisfactory for the lane change assistant.
High speeds and very short reaction times mean very challenging real-time requirements. The system must be able to detect and locate a number of objects in the range of sight in a fraction of a second, i.e. approximately 5-8 images at 25 images/s. We deployed a hardware solution implemented in Altera FPGAs to calculate the distance. The system is based on the hierarchical algorithm described above and applies the NCCF. The system can cover a large detection range and provides approximately constant relative accuracy over the entire detection range. The calculations are performed during image acquisition and are completed approximately 70µs after image acquisition.
A disparity map is generated and passed to the object detection and tracking stages with the help of embedded software. By skillfully distributing the algorithmic logic on a number of softcore processors (this can also be implemented in FPGAs), this step is also completed within one image acquisition time. The condensation algorithm is applied for object detection, and tracking is by Kalman filter. The Kalman filter needs a few images to settle when new objects enter the viewing range, and when there are periods when data drop-outs occur. If the environment capture time needs to be decreased significantly, faster cameras are needed. We only consider objects moving towards the vehicle or remaining at the same relative distance from the camera system. Finally, object locations, approximate object dimensions, and object speed are transferred to the master system. Experimental results for object imaging on a highway are shown in fig. 16 a.

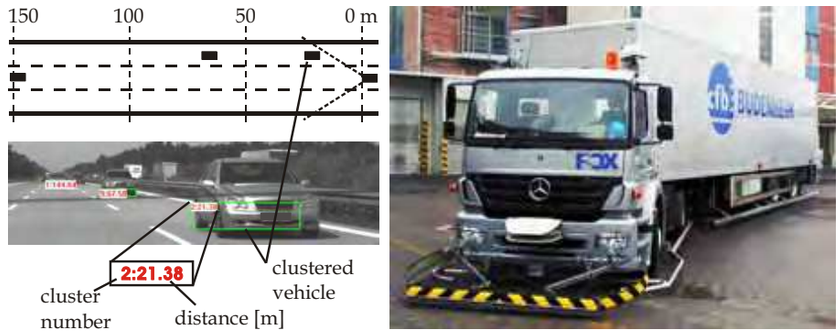
5.3 Robotics and Autonomous Vehicles

The detection and tracking of objects both at a distance and at close range is common in road traffic scenes, environment capture systems for autonomous vehicles or robotic machines deal primarily with close-range objects. Key features of these close-range systems are:

- a large field of vision
- outstanding accuracy
- the ability to capture all objects in the vehicle/robot surroundings

Some systems are designed to detect and track obstacles in a narrow driving lane only, or all objects in the surroundings that are needed to select alternative routes or to establish a free passageway. Detected objects could be identified with the help of image processing algorithms in very complex systems. Examples of such objects are signs and landmarks.

The Götting KG, FOX GmbH companies have joined together to produce autonomous vehicles based on conventional trucks. The speed of the vehicles fitted with standard, traditional detection and tracking systems based on laser scanners is very restricted at this time due to stringent safety requirements on company sites. Magdeburg University is involved in a project to develop a system consisting of a combination of stereophotogrammetric algorithms and a laser scanner that will capture all objects in the vehicle surroundings, and increase the approved speed. The vehicle in question is shown in fig. 16 b.
A large aperture angle is needed for the relatively large field of vision. Should cameras with lenses smaller than 8 mm be deployed, then we recommend a rectification stage. The base of the stereo camera system can be reduced to 5-10 cm as the specified object distances are quite small (20-30m with typically moderate error specifications).
We will not normally need the hierarchical algorithm to calculate the depth values described in section 4 because of the restricted detection range. We may be able to utilize simplified criteria such as the SAD function, as we may be operating under more favorable lighting conditions.



(a) (b)

Figure 16. a) Driver assistance system, b) Autonomous vehicle (FOX GmbH & Götting KG)

Object detection and tracking is more difficult compared to section 5.2, as the objects are significantly more complex and can vary to a greater degree than with the driver assistance system described above. We can still apply the condensation algorithm and the stereo image analysis for gray value cameras for the simple case of obstacle detection. However, we recommend a color image processing system combined with complex signal processing for the capturing and possible identification of all objects.

6. Conclusion

The field of environment capture technology is vast: a wide range of different measured variables are processed. A complete picture of the surroundings can be provided by deploying very many sensors. In other words we can gain comprehensive information on the surroundings. Optical sensors and camera systems can be deployed in a variety of ways, so that massive amounts of varied information can be acquired with few sensors.

Image processing is the key technology applied for processing information from optical image sensors. Typical applications for image sensors include driver assistance systems, autonomous vehicles, and robotic machines. Stereo camera systems supply the necessary 3-d information for environment capture and have crucial advantages. Image processing algorithms can be very complex and with high computation overheads.
Robust solutions are needed for the applications cited. Challenging real-time requirements are often specified for robotic machines and driver assistance systems.
We often have to modify signal processing algorithms very extensively in order to implement them in hardware. This is very challenging for continuous real-time processing at high speed. Thus, we present a hardware-software co-design for an algorithm for locating multiple objects in a variety of applications in section 4.
The measurement technique is based on algorithms from stereophotogrammetry. We implemented an optimized algorithm in conjunction with image pyramids in an FPGA as a parallel structure in hardware that would cover a large detection range as required in driver assistance systems. The algorithm implemented in hardware consists of the NCCF calculation, subpixel interpolation, and depth histogram generation (condensation algorithm).
Other object detection tasks (time histogram, clustering) and tracking system run on three processors, operating in a pipeline configuration in real time. We applied a Kalman filter to track captured objects — this was to smooth out jumps and invalid detections, and to successfully and accurately estimate distance and speed.
There are a number of different approaches out there for generating disparity maps in hardware. Each of these approaches is suited for different applications. Many of these algorithms are established and well known. Much of the global research effort focuses on effective real-time hardware or software implementations.
The standard of global research on higher-level processing techniques for analyzing the disparity map and image information is very high indeed. The development of reliable, robust, and real-time algorithms continues to be the prerequisite for a broad application base.

7. References

Albertz, J. & Kreiling, W. (1989) , Photogrammetric Guide, *Wichmann-Verlag*, ISBN 3-87907-384-8

Dellaert, F.; Burgard, W.; Fox, D. & Thrun, S. (1999), Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, p. 2588

El-Melegy, M.T. & Farag, A.A.; Statistically Robust Approach to Lens Distortion Calibration with Model Selection, *International Conf. on Computer Vision and Pattern Recognition*, CVPR-03, Workshop on Intelligent Learning, Madison, Wisconsin, June 16-22, 2003, pp. 150-156.

Fang, C.Y.; Chen, S.W. & Fuh, C.S. (2003), Road-sign detection and tracking, *IEEE Transactions on Vehicular Technology*, vol. 52, nr. 5, pp. 1329-1341

Franke, U.; Görzig, S.; Lindner, F.; Mehren, D. & Paetzold, F. (1997), Steps towards An Intelligent Vision System For Driver Assistance In Urban Traffic, *Intelligent Transportations Systems*, pp. 601–606

Franke, U.; Gavrila, D.M.; Görzig, S.; Lindner, F.; Paetzhold, F. & Wöhler, C. (1998), "Autonomous Driving Approaches Downtown", *IEEE Intelligent Systems*, vol.13, no.6, pp. 40-48

Fuerstenberg, K.C.; Dietmayer, J. & Lages, U. (2003), Laserscanner Innovations for Detection of Obstacles and Road, *AMAA 7th International Conference on Advanced Microsystems for Automotive Applications*

Gavrila, D.M. (2004), Pedestrian Detection from a Moving Vehicle, *Lecture Notes in Computer Science, Springer* , pp. 37-49, ISBN: 978-3-540-67686-7

Gupta & Rajesh (1995), Co-Synthesis of Hardware and Software for Digital Embedded Systems, *Kluwer Academic Publishers*, ISBN 0-7923-9613-8

Kaszubiak, J.; Tornow, M.; Kuhn, R.W.; Michaelis, B. & Knoeppel, C. (2005), Real-time vehicle and lane detection with embedded hardware, *IEEE Intelligent Vehicle Symposium*, pp. 618–623

Kluge, K. & Lakshmanan, S. (1995), A Deformable Template Approach to Lane Detection, *IEEE Intelligent Vehicle Symposium*, pp. 54–59

Knoeppel, C.; Regensburger, U. & Michaelis, B (2000), Robust Vehicle detection at Large Distance Using Low Resolution Cameras, *IEEE Intelligent Vehicle Symposium*, pp. 267-272

Lange, R. & Seitz, P. (2001), Solid-State Time-of-Flight Range Camera, *IEEE Journal of Quantum Electronics,* vol. 37, no. 3, pp. 390-397

Lee, S. & Kay, Y. (1990), A kalman filter approach for accurate 3d motion estimation from a sequence of stereo images, *10th International Conference on Pattern Recognition*, pp. 104–108

Noelle, M. (1996), Konzepte zur Entwicklung paralleler Algrithmen der digitalen Bildverarbeitung, *VDI-Verlag*, ISBN 3-18-341010-9

Saneyoshi, K. (1996), Drive assist using stereo image recognition, *IEEE Intelligent Vehicle Symposium*, pp. 230-235

Schenk, T. (1999), Digital Photogrammetry - Background, Fundamentals, Automatic Orientation Procedures, *Terra Science*, ISBN 0-203-30595-7

Stiller, C.; Pöschmüller, W. & Hürtgen, B. (1997), Stereo Vision in Driver Assistence Systems, *Intelligent Transportation Systems*

Tornow, M.; Michaelis, B.; Kuhn, R.W.; Calow, R. & Mecke, R. (2003), Hierarchical Method for Stereophotogrammetric Multi-objekt-Postion Measurement. *Pattern Recognition, DAGM Symposium*, pp. 164–171

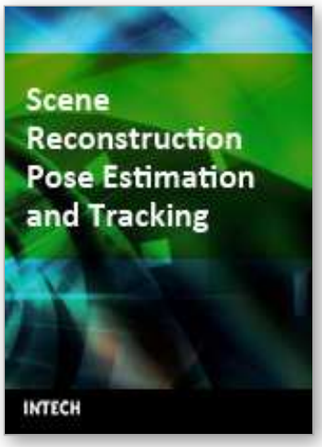
Tornow, M.; Kaszubiak, J.; Schindler, T.; Kuhn, R.W. & Michaelis, B. (2006), Hardware Approach for Real Time Machine Stereo Vision, *Journal of Systemics, Cybernetics and Informatics*, vol. 4, no. 1, ISSN: 1690-4524

Tyrrel, J. (2004), Low-cost sensor puts 3D-cameras in the picture, *Opto and Laser Europe The European magazine for photonics professionals*, no. 123, pp. 20–21

Uhler, W.; Mathony, H.J. & Knoll, P.M. (2003), Driver Assistance Systems for Safety and Comfort / Robert Bosch GmbH, Driver Assistance Systems, *EU-Projekt EDEL im 5. Rahmenprogramm*, edel-eu.org

Venhovens, P. & Naab, K. (2000), Adiprasito, B.: Stop and Go Cruise Control. In: *Int. Journal of Automotive Technology* 1, S. 61–69

Yoshika, T.; Nakaue, H. & Uemura, H. (1999), Development of Detection Algorithm for Vehicles Using Multi-Line CCD Sensor, *Intelligent Transportation Symposium*



Scene Reconstruction Pose Estimation and Tracking

Edited by Rustam Stolkin

ISBN 978-3-902613-06-6

Hard cover, 530 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

This book reports recent advances in the use of pattern recognition techniques for computer and robot vision. The sciences of pattern recognition and computational vision have been inextricably intertwined since their early days, some four decades ago with the emergence of fast digital computing. All computer vision techniques could be regarded as a form of pattern recognition, in the broadest sense of the term. Conversely, if one looks through the contents of a typical international pattern recognition conference proceedings, it appears that the large majority (perhaps 70-80%) of all pattern recognition papers are concerned with the analysis of images. In particular, these sciences overlap in areas of low level vision such as segmentation, edge detection and other kinds of feature extraction and region identification, which are the focus of this book.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jens Kaszubiak, Robert Kuhn, Michael Tornow and Bernd Michaelis (2007). Real-Time 3-D Environment Capture Systems, Scene Reconstruction Pose Estimation and Tracking, Rustam Stolkin (Ed.), ISBN: 978-3-902613-06-6, InTech, Available from: http://www.intechopen.com/books/scene_reconstruction_pose_estimation_and_tracking/real-time_3-d_environment_capture_systems

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen