RESOURCE OPTIMIZED STEREO MATCHING IN RECONFIGURABLE HARDWARE FOR AUTONOMOUS SYSTEMS

Fredrik Ekstrand

2011
Copyright © Fredrik Ekstrand, 2011
ISSN 1651-9256
Printed by Mälardalen University, Västerås, Sweden
Abstract

There is a need for compact, high-speed, and low-power vision systems for enabling real-time mobile autonomous applications. One approach to achieve this is to implement the low- to intermediate-level applications in hardware. Reconfigurable hardware have all these qualities without the limitation of fixed functionality that accompanies application-specific circuits. Resource constraints in reconfigurable hardware calls for resource optimized implementations with maintained performance.

The research group in Robotics at Mälardalens University is moving toward the completion of a reconfigurable hardware-platform for stereo vision, coupled with a compact embedded computer. This system will incorporate hardware-based preprocessing components enabling visual perception for autonomous machines. This thesis covers the reconfigurable hardware section of the vision system concerning the realization of scene depth extraction. It shows the advantages of image preprocessing in hardware and propose a resource optimized approach to stereo matching. The work quantifies the impact of reduced resource utilization and a desire for increased accuracy in disparity estimation. The implemented stereo matching approach performs on par with recent similar implementations in terms of accuracy, but excels in terms of resource utilization and resource sharing, as the external memory requirement is removed for larger images.

Future work aims to further include processes for navigation, and structure and object recognition. Furthermore, the system will be adapted to real world scenarios, both indoors and outdoors.
Acknowledgements

This thesis could not have been done without the support of my supervisors Lars Asplund, Mikael Ekström and Giacomo Spampinato. Thank you for believing in me. Your supervision has brought me a long way toward the realization of an independent researcher.

Many thanks to Carl Ahlberg, Jörgen Lidholm, and Batu Akan for all the deep discussions, provoking ideas, crazy stunts and laughs. A special thank you to Carl and Jörgen for never backing away from assisting and helping, as co-authors and as friends.

A warm thank you to Hüseyin Aysan for all the laughs, all the fika, and for always giving a helping hand and being a great friend.

MDH is a place of work and study, but even more than that, it is a place of interaction with and learning from a body of diverse knowledge and experience, manifested in many joyous conversations. Thank you Nikola, Martin, Marcus, Jimmie, Andreas, Mikael Å, Farhang, Moris, Mirko, Jenny, Carola, Susanne, Malin, Ingrid and all the rest for always answering with a smile. An additional thanks to Martin, Nikola, Mikael, Marcus and Giacomo for never hesitating to assist and help. And a big thanks to Adnan without whom this thesis would have never materialized until this day.

Last, but certainly not least, a huge thank you to my family! My precious sons William and Edward who have supplied me with drawings to enhance my office, questions to occupy my mind, and unconditional love. A huge thank you to my fabulous wife Sara for your support, dedication, devotion and love! I could not have done it without you.

Fredrik Ekstrand
Västerås, September 26, 2011
List of Publications

Papers Included in the Licentiate Thesis

**Paper A**  *Two Camera System for Robot Applications; Navigation*, Jörgen Lidholm, Fredrik Ekstrand and Lars Asplund, In proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Hamburg, Germany, September 2008


---

1The included articles are reformatted to comply with the licentiate thesis specifications
Other relevant publications

Conferences


# Contents

I  Thesis  

1  Introduction  
1.1 Background  
1.1.1 Reconfigurable Hardware  
1.1.2 Feature detectors  
1.1.3 Feature Matching  
1.1.4 Stereo Matching  
1.1.5 Area-based Matching  
1.1.6 Disparity Map Creation  
1.2 Motivation  
1.3 Outline of thesis  

2  Related Work  
2.1 Visual Navigation  
2.2 Stereo Matching  
2.3 Resource Constraint  
2.4 Area Matching  
2.5 Support Window  
2.6 Disparity Map Improvements  

3  Research Summary  
3.1 Paper Overview  
3.1.1 Paper A  
3.1.2 Paper B  
3.1.3 Paper C  
3.2 Research Methodology
Bibliography .................................................................................. 65

7 Paper C: Utilization and Performance Considerations in Resource Optimized Stereo Matching for Real-Time Reconfigurable Hardware 67
7.1 Introduction .............................................................................. 69
7.2 Background ............................................................................. 71
7.3 Related Work ........................................................................... 71
7.4 Improvements .......................................................................... 73
  7.4.1 Support Window .................................................................. 73
  7.4.2 False Matches .................................................................... 73
  7.4.3 Consistency Check .............................................................. 74
  7.4.4 Confidence Evaluation ...................................................... 74
  7.4.5 Textureless Areas ............................................................... 75
  7.4.6 Filtering ............................................................................ 75
  7.4.7 Color .............................................................................. 75
7.5 Our Implementation ................................................................. 76
  7.5.1 Support Window .................................................................. 76
  7.5.2 Consistency Check .............................................................. 76
  7.5.3 Propagation ....................................................................... 77
  7.5.4 Filtering ............................................................................ 78
  7.5.5 Color .............................................................................. 79
7.6 Result Summary ........................................................................ 79
7.7 Conclusions ............................................................................. 81
Bibliography .................................................................................. 83
I

Thesis
Chapter 1

Introduction

Self-parking cars, pedestrian-sensitive self-braking trucks, driver-less mining-machines, and museum guiding-robots are all examples of intelligent and autonomous agents. Autonomous agents are entities that are assigned a task and execute it without further guidance or interference from the task originator. Such an agent senses its environment, adopts an approach accordingly, and executes an action toward the fulfillment of the task. Those are the same fundamentals which form the definition of a robot: sense, plan, and act.

Spatial awareness is elementary in any autonomous mobile machine. There are two fundamentals in the concept of spatial awareness - a knowledge of the environment, and one’s own relation to that environment. (It can be argued that an autonomous agent is really not in an environment, but part of the environment.) Knowledge of the environment requires sensors, and the degree of perception is determined by the properties of the sensors. Regardless of sensor type, the resolution, accuracy and speed of the sensor limit the awareness. There are several types of sensors for sensing the surrounding space, but the two predominant types used in robotics and industry are rangefinders and vision.

Rangefinders (such as radar, sonar and laser) are active systems that emit waves (such as electromagnetic or light) and then measure the reflections of the waves off an object. Vision sensors, or cameras, are passive sensors that measure whatever light that falls onto the sensor, whether direct or reflected. Common for rangefinders is that they are not as fast as passive visual systems as they rely on returned waves, whereas cameras can measure at much shorter intervals as the light flow is one-directional and constant. These passive and
general properties make cameras versatile, but also limited in the application of
range finding as they lack the built-in ranging property of rangefinders. With-
out knowledge of the temporal origin of the measured light, cameras cannot
use the time-of-flight or accumulation methods used by the range-finders, but
need to correlate the sensed data over a spatial difference. This is commonly
performed by triangulation of views from different angles of the same scene,
either by a movement of a single camera or by the use of multiple cameras,
referred to as stereo vision.

All types of rangefinders are well suited for map generation and obstacle
detection, but they are not optimal for object recognition, or tracking, as they
only convey the structure of the surroundings, and nothing about its colors or
patterns. Stereo-vision systems are an approximation of human eyes and can
enable machines to match or relate to our perception of the world. All infor-
mation about the environment exist in the data generated by the cameras. It
simply needs to be extracted. This simple part has occupied the large computer
vision research community for many years, and still do.

In this thesis, we present a stereo vision system for embedded mobile
robotics. The end goal with this research platform is to fit a real-time au-
tonomous system for navigation and object recognition in a compact and power
efficient hardware system. In order to fit all system parts, each component must
be made as compact and efficient as possible. This thesis focuses on reducing
the task of extracting depth information of a scene through matching of view-
separated images.

### 1.1 Background

Computer vision involves digital processing of images. Images are captured
with a sensor measuring the light falling onto the sensor surface. The amount
of light is transformed into a digital representation which is communicated off
the sensor. The quality of an image is contingent on the sensor architecture, the
lens, the converter electronics, the circuit board design, and many more factors.
A great deal of research is dedicated to improving the performance of image
sensors. Our research is focused on the application of the image sensor, and
the process of extracting the information embedded in the sensor data. Many
applications and algorithms exist for image processing, and those concerned
with using the images to enable machines to see are labeled as belonging to
1.1 Background

Machine, or computer, vision.

Computer vision algorithms can generally be characterized by complex and repetitive operations, and large amounts of data, as detailed by Ratha and Jain [1]. Moreover, vision algorithms can be classified as belonging to either low-level, intermediate-level (segmentation), or high-level (higher order structure and matching). Regardless of level, vision algorithms are all preprocessing steps for a main algorithm, such as navigation or object recognition, but the separation is far from distinct. A complete vision system needs to integrate solutions for all levels in order to complete the main application. In this thesis, we are concerned with low-level algorithms.

By definition, the performance of a system is contingent on the performance of its parts. Being the initial node in the chain, the sensors set the performance limit. Image sensors can provide high frame rates, but require the receivers of their pixel stream to match their speed. If a receiver is to receive and process images continuously, it needs to be able to both receive and execute operations on every pixel in time before the next pixel arrives. This implies an operating frequency several times higher than the pixel frequency of the image sensor. Real-time image processing requires reading and operating on millions of pixels per second, putting a hard requirement on the throughput ability of the processing system. The concept of real-time will vary with the topic, and by real-time we mean the execution time of an action or reaction that is adequate to mimic the human counterpart. Concerning cameras, a frame rate of around 30 Hz is sufficient to not appear jerky to the human eye at moderate transitions in the scene. For completely smooth motions an update frequency of above 60 frames per second is required. We use 30 frames per second as the frame rate definition of real-time.

1.1.1 Reconfigurable Hardware

As opposed to standard sequential computer systems, which require a processing frequency several orders of magnitude greater than the pixel frequency, reconfigurable hardware enables pixel-wise image processing at a frequency matching that of the pixels. Reconfigurable hardware, such as FPGAs, is a hardware component where the functionality is loaded at startup. The central processing unit of a typical PC fetches its instructions from memory, executes them, and then stores the result back into memory. In the FPGA, the physical configuration itself is the instructions and there is no setup delay [1]. It is a standalone component that executes like a fixed state machine without an operating system or external components. The big advantage of FPGAs is that
they enable concurrent processing of multiple data by parallelization. This removes the need for the processing unit to operate at a frequency above that of the vision sensor.

An important parameter of reconfigurable hardware is the limited available resources. Being a component of fixed size where the functionality is determined by the physical interconnect of its logic elements, only a certain amount of instructions can be concurrently realized. Moreover, shifting of instruction sets is not possible as resources cannot be reallocated during run-time. In other words, FPGAs can get full [1].

The type of algorithms appropriate for FPGAs are also limited due to the types of operations possible with the internal circuitry. Any type of operation can be realized in theory, but the cost in doing so might render it impractical. Registers, comparators, adders, multipliers, and internal memory are all in finite numbers and realizing complex algorithms might require more than available. Implementations of algorithms thus have to fit both in type and size. The functionality of an FPGA is described with code written in a Hardware Description Language (HDL) such as VHDL or Verilog. FPGAs are easily reconfigured using tools ranging from low-level programming languages, such as HDL, to more general languages, such as variants of C and Python.

1.1.2 Feature detectors

In certain applications, such as navigation and object recognition, limited parts of an image is often of more interest than the rest. These parts are features of an object or a scene, and can be used as descriptors for that object or scene. Algorithms identifying and extracting these defining parts of an image is referred to as Feature Detectors. Different feature detectors are good for different applications, but their common task is to identify salient areas (areas with low similarity in the surrounding area), such as edges, corners, blobs, etc. Their primary function is to reduce the amount of data associated with an object or scene, without sacrificing the important information. One of the most important properties of a feature detector is its repeatability: the ability to repeatedly identify the same feature on any two separate occasions. This ability is crucial when locating features between multiple images, as in matching for tracking, depth, shape, etc. Another important parameter is the information content of a feature detector, a measure of the distinctiveness of a salient point. The more spread out the features are over an object, the higher the information content, and the higher the likelihood of a successful match [2].

A multitude of feature detectors exists, and in Paper A the Stephen and Har-
ris Combined Corner and Edge Detector [3] is used. It has been widely used in computer vision applications for a long time, due to its high repeatability and information content [4]. The Stephen and Harris detector, also common in many other feature detectors, looks at the intensity of each pixel and how it relates to that of its neighbors. A pixel is evaluated based on how well it matches the defined feature types - sharp discontinuity in one direction equals an edge, and in two or more directions equals a corner. The better the match, the higher the absolute cornerness value (positive for corners, negative for edges). The algorithm produces only this definition of a feature, which makes a feature-to-feature correlation challenging. Although similar in fashion, the edges and corners have one small difference: corners are by definition isolated objects not linked to other corners, whereas edges have a stronger relation to other edges and can be formed into lines or curves possible to use for matching [5].

1.1.3 Feature Matching

Matching of individual pixels based solely on their intensity is an almost impossible task. Performing the same operation on corners or edges from the Stephen and Harris detector can be less difficult, but is very scene and parameter dependent as the amount of features impact the matching confidence. Finding a single point from one image in another image of thousands, or even only hundreds, of points with only a single value to compare, is not trivial. An approach is to look at several features and their individual relations, and match them as a point cloud [6]. Such operations are highly iterative, and not suitable for a resource constrained real-time system.

To reduce the challenge of correlation, it is possible to increase the feature uniqueness by including more properties of the feature and its surroundings, such as angle or scale. This property specification adds descriptors to the features, such that it is possible to look at the feature descriptors individually and not simply at their mutual relation. A good example of a feature descriptor is SIFT (Scale Invariant Feature Transform) [7]. However, the added descriptive-ness is computationally intense and of an iterative nature, and the matching process can be very time consuming for extensive feature sets [8].

Matching of non-aligned images, irrespectively of whether based on individual points or areas, require a costly 2-dimensional search across the other image for every element. The remedy is to transform the images into the same coordinate system, a process called rectification. Rectification involves identifying the intrinsic and extrinsic parameters of the image capturing device to determine the relation of the projection planes of respective images. This in-
includes correction of lens distortion, and aligning the images so that image scan-
lines are parallel and aligned between images. The matching problem is thus
reduced to a 1-dimensional search, significantly reducing the complexity, as
long as the geometric distortion is at a minimum [9]. The rectification pro-
cess is computationally heavy, and needs to be performed for every image pair
of unknown relation. For fixed stereo camera systems, the calculation of the
rectification parameters need only be performed once as the parameters of the
capturing devices are static. Rectification is then performed by image transfor-
mation through applying a constant set of parameter-based coordinate shifts.

The concept of extensive feature descriptors, such as SIFT, is to include
more than just the saliency of the point, and also include additional information
on the neighborhood, such as qualities of other salient parts (edges) in the
area, the saliency at different scales, etc. The reason is obvious, identification
is easier the more information available. This notion can be applied to the
underlying pixels directly, without performing an analysis of their properties.
Area-based approaches match an area instead of a point, and they are the most
used approach to stereo matching in computer vision.

1.1.4 Stereo Matching

The area of computer vision contains many branches, and stereo matching, or
stereo correspondence, is one of the widest. It deals with extracting depth in-
formation from 2-dimensional images by way of finding corresponding points
in two, or more, images. The sole purpose of using two cameras is to cap-
ture a scene from two different views at any given time in order to extract
3-dimensional data of the scene.

Any vision approach concerned with depth needs to solve the correspon-
dence problem, that is, which part in one image correlates to which part in
another image. In the machine vision community, the majority of approaches
can be categorized into either of two groups, global or local [10]. In general
terms, the global algorithms are considering the estimation of the separation,
or disparity, of the two view-diverging images as an optimization problem. A
global cost function incorporates both data (matching) and smoothness terms,
which the disparity selection seeks to minimize. Local algorithms, on the other
hand, only consider a limited area surrounding the point under evaluation for
disparity estimation.

Global methods generally outperform local methods in terms of accuracy,
but suffer from a high computational cost. Global methods usually consist
of several, often iterative, steps in their refinement of an initial disparity map.
often attained with a local method [10]. As a consequence, they are not optimal for real-time applications.

Local methods can be further divided into area-based or feature-based correlation. Both are sprung out of the same basic notion - a pixel in itself gives poor correlation data with low confidence in matching, thus a larger view is required. Both approaches use the neighbors, to more or less extent, of the current pixel for more defining data. Area-based methods use them to correlate with another same-size area, whereas feature-based methods use them to determine the interest level of the pixel and use that rather than the underlying image data.

Area-based matching techniques usually create a dense map with depth information for every pixel. Feature-based techniques can only create a sparse map as information is removed from the images. However, it is argued that the confidence in the match is higher with feature-based techniques as they are only matching on individual pixels, rather than a set of pixels [9]. Nevertheless, which technique to use should be based on the application.

Feature-based matching techniques are more concerned with finding a relation to the scene or image as a whole than to get a complete 3-dimensional reconstruction of the scene. They can be used, for example, to determine the ego-motion of the agent, or to correctly identify the rotation and translation of an object. Working with feature images also significantly reduces the amount of data in the system, leaving room for additional calculations or an increased frequency. Thus, for applications not in need of depth information in the whole scene but rather high speed, such as certain object recognition [11], the feature-based approach is a good candidate. An additional advantage is that a crystallization of the important information in the lower-level can both reduce the amount of data as well as its rate. The data rate reduction is advantageous for higher level processes, but only if the data is sufficient.

### 1.1.5 Area-based Matching

Area-based methods correlate the entire pixel neighborhood, element by element, through the use of a support window. The support window is compared with same size support windows in the other image, and is usually in the form of a square. To evaluate the similarities of two windows, a correlation measure is required. Several exist, but one of the simplest and most straightforward to implement, and thus widely used, is the SAD (Sum of Absolute Differences). With the SAD, the matching cost for two points residing in two different images is calculated through an aggregation of the element-wise absolute differ-
ences of the support windows for respective point.

One of the fundamental problems of window-matching is the selection of the window-size [12]. A small window achieves higher precision in the disparity estimation, but exhibits more noise. Large windows reduce noise by an increase of the matching data, but reduce the precision, especially at depth discontinuities. Thus, the optimal window size will vary from scene to scene, but also within a scene.

Several approaches have been proposed to solve the size selection problem. Variable-size windows, as proposed by Kanade and Okutomi [12], are adapting to the conditions of the underlying image and have been shown to significantly improve the matching, but lack in terms of speed. This idea have been refined to variable window shapes, as presented by Mei et al. [13] and weighting of the support window, as proposed by Yoon et al. [14], to only consider information on similar data, such as color. All these approaches strive to improve the outcome of the matching algorithm, the generated disparity map.

### 1.1.6 Disparity Map Creation

The role of the disparity map is to convey the depth in an image represented as the distance of the index of a certain point between two images. The matching algorithm will approximate the real-world depth relation for the entire image, but hard-to-match areas of the image, such as those of low texture or low signal-to-noise ratio, will generate false matches. Additionally, foreground objects occlude background objects, and due to the different perspectives in the two images, the parts that are occluded will differ in the two images. This causes pixels adjacent to object borders, or depth discontinuities, to be estimated at the depth of the foreground object as the edge is a very prominent feature. This causes the disparity maps to extend outside of the foreground object, and is called foreground fattening. The inadequacies of the area-based approaches limit the possible quality of the disparity map, and several approaches have been proposed to deal with this.

Approaches seeking to create dense disparity maps try to remedy the deficiencies, whereas those aiming for a sparse but highly confident disparity map simply remove them. Regardless of the approach, the initial step is to identify the erroneous values, which can be done using a set of assumptions about the underlying image. They act as constraints on the disparity map, and can be used to determine the validity of a match, as explained by Ozanian and Tak-ouhi [15].

The surface continuity constraint states that a scene is made up of solid sur-
faces which vary smoothly. As a consequence, adjacent pixels are most likely at the same depth. The uniqueness constraint states that a point in one image can only have one corresponding point in the other image, which is natural as the images depict physical objects. The ordering constraint states that the order of pixels in one image must be fulfilled in the disparity map. Violations of these constraints occur, for instance, at depth discontinuities, heavily slanted surfaces, and occlusion. However, for the most part they can be used to validate the estimated disparity of a pixel in rectified images.

One of the common ways of finding these violations is to perform a left-right consistency check (LRC) [16]. A regular matching procedure uses one of the images as the base and then tries to find corresponding pixels in the other image. Pixels that have no corresponding mate, as they are not visible in the other image, will generate false matches. The LRC also performs matching with the other image as the base and then checks to see that a pixel indicated as the match in one image is referring back to the indicating pixel in the other image, that is, that they select each other as the best match. This is a very robust method that identifies the majority of false matches due to perspective distortion [17].

After false matches are identified, sparse approaches just discard them and leave the pixels void of disparity. Dense approaches need to assign a value though, and the constraints mentioned earlier can also be utilized for this purpose. Instead of estimating the disparity by correlation, similarity in adjacent pixels, which are assumed to be of same surface according to above constraints, can approximate the disparity. A popular method is to use median-filtering to remove noise and smooth the disparity map. As surfaces are more likely to be smooth than bumpy, this increases the quality of the map. Another approach is to interpolate or propagate values from surrounding pixels to fill in empty areas.

The quality, or correctness, of a disparity map is assessed through comparison with the scene ground truth. A set of stereo image pairs were proposed by Scharstein and Szeliski [10] and they are used as the benchmark of correspondence approaches today, with tools available online [18].

1.2 Motivation

Reducing the workload in a visual perception application can be achieved in two ways: reduce the amount of data by only sending data of interest to the application, or extract necessary information so that the receiver only needs
to consume, not process. The first scenario is realized with a feature detector, and the second with any of several transforms: depth extraction, segmentation, object identification, etc.

The initial project with our stereo camera system was to produce a fast navigation application capable of simultaneous localization and mapping (SLAM) through the use of vision [6],[19]. In short, SLAM is a process where an agent enters an unknown environment, picks out identifying landmarks or geometries so that it can move around and always find its way back to the starting point with the help of the identified visual cues. As the agent traverses the environment, it continuously builds a map of the environment which it later uses for navigation.

Common approaches are to use the SIFT [7] or SURF [20] feature descriptors for landmark matching. The biggest challenge of SLAM is to identify salient areas with high confidence in the estimated depth. The SIFT approaches rely on unique identifiers which is slow and/or large in implementation. Simpler feature detectors can be made faster, but lose in matching confidence. However, a lack of accuracy might be compensated with higher frequency. We thus opted for a fast but less accurate approach in an attempt to reduce the computational complexity.

To improve the accuracy of the initial approach, we then propose a concurrent simple correspondence approach for an increase of the disparity estimation confidence. A stereo matching component running concurrently with the simple feature detector, delivering depth information for the features. This approach needs to be resource optimized to not hinder the application processes.

Disparity map estimation, however, is a non-trivial problem which the community is only now starting to find a complete solution to. However, these solutions either require bulky systems or extended computation time. For mobile autonomous systems, real-time operation is required. Extracting depth from two images of half a million pixels at this rate is no small feat. Additionally, a complete vision system residing in an FPGA requires several processing components just for preprocessing the image data, such as, image rectification, motion artifact compensation, and depth estimation. Furthermore, higher-level applications, such as tracking, object recognition, or navigation, should also fit. Fitting all these parts of an autonomous agent onto a compact and power-constrained embedded mobile system is a real challenge.

It is necessary to adopt an approach that is capable of meeting the requirements for the low-level processing to enable high-level processing, but that can also fit the high-level processes concurrently. Thus, all building blocks need to be reduced. Enabling more computations in the FPGA, by reducing
the components for preprocessing, will improve the capability and flexibility of the system. Furthermore, it is not important to achieve maximum accuracy in the algorithms. With a high-speed system, correction or filtering can be used to compensate. A high sample rate allows for more simple algorithms. Then, rather than trying to develop a new feature detector or correspondence algorithm, our focus is on utilizing "good enough" algorithms by combining and optimizing them for reconfigurable hardware. The end goal is a small and high-speed hardware system working as the eyes and visual cortex of any type of autonomous vehicle or robot.

1.3 Outline of thesis

The continuation of this thesis consists of two main parts. The first part consists of 3 chapters: Chapter 2 presents the related work; Chapter 3 provides an overview of the included research papers; Chapter 4 presents overall contributions and conclusions together with possible future work. The second part of this thesis consists of Chapters 5 through 7 and is a collection of the research publications which form the basis of this thesis.
Chapter 2

Related Work

The concept of using reconfigurable hardware for image processing is not new. Several competent approaches exist, but most have one or more tradeoffs: quality, resource utilization, or limitation in image size. Which is the most important parameter is an application specific question, but for our purpose, resource utilization is important as we seek to fit an entire autonomous agent in our system.

2.1 Visual Navigation

Several SLAM approaches have been presented, such as Barfoot [21], Bertolli et al. [6], and Montemerlo et al. [22]. However, the approaches are not suitable for FPGA implementation. An FPGA implementation of SURF is presented by Svab et al. in [23]. However, they only implement part of the algorithm as the complexity and time-consuming nature of the algorithm makes it difficult to realize on the FPGA. The descriptor generation is handled in software on a Power-PC, and the complete navigation system is residing on a laptop. Hence, another approach is required to fit a complete navigation system in an embedded system.

2.2 Stereo Matching

Performance measurements of correspondence algorithms, such as presented by Hirschmüller and Scharstein [24], mostly focus on the accuracy of the dis-
parity map, whereas real-time implementations rank the throughput, or frame rate, higher.

2.3 Resource Constraint

Since the aim of our work is to achieve an acceptable performance at a low resource usage, we need to specify what low resource usage is. Resource utilization in an FPGA is normally expressed in slices and LUTs (LookUp-Table which realize boolean operations). In our previous work, our system produced an acceptable disparity map at 1221 slices when implemented in a Spartan-3 FPGA. This is just above 4% of the available slices on the chip.

Several stereo matching approaches with low resource usage have been proposed, such as by Arias-Estrada et al. [25]. Their utilization is only 4.2K slices on a Virtex-II, but with a fair disparity map. The implementation presented by Lee et al. [26] comes in at a resource usage below 10K slices. The produced disparity map is moderate showing extensive blurring of edges and noise.

For higher quality disparity maps, the resource usage inevitably go up. Very good results are presented by Zhang et al. [27], but the utilization is 95K slices plus a large amount of ALUTs and DSP blocks, leaving little room for concurrent processing. A collection of proposed FPGA implementations is presented by Lazaros et al. in [28].

2.4 Area Matching

Very accurate results have been presented for area-based approaches [18], but the high quality of these implementations mostly come at the expense of computational power and, hence, processing time.

Recently a number of non-global near real-time implementations have been presented. They are not truly local as they are akin to global methods such as Dynamic Programming [29], but operate on a limited area [30]. The near real-time software implementations tend to utilize special purpose hardware, such as GPUs [31],[13], to accelerate the processing. Although impressive in their performance, they are not really suitable for mobile and embedded systems, considering the cost, size and power requirements. Transferring these approaches to an FPGA is not optimal, as they resort to iterative approaches with computational and memory requirements that are hard to realize for the limited resources of an FPGA [31]. Large memory can be included when constructing
an FPGA-system, but the memory speeds required are above the capacity of standard FPGAs.

2.5 Support Window

There are numerous proposals to overcome the static window issue, as discussed before. Adaptive window approaches suggested by Kanade and Okutomi [12] or the one by Boykov et al. [32] are an ill match for our system, as they exhibit the same problems as we do with noise and sensitivity to low-texture areas. Additionally, they rely on models with empirically derived parameters unique to every scene. This might not be much different from empirical selection of window size for our standard approach, but it is not an improvement either.

Hirschmüller et al. [33] suggests an approach using multiple windows for good depth discontinuity performance. Although based on SAD, it requires a large memory. Another multiple window approach proposed by Chonghun et al. [34] seems promising at first, but their reason for multiple windows is the refinement of an overly-smoothed noise-free first estimation, the inverse from our approach.

Adaptive support-weight approaches, as suggested by Yoon et al. [14] and Gu et al. [35], produce good disparity maps but at a low frame rate.

Yi et al. [36] found that the effect of the shape of the support window has less impact than the number of pixels in the window. This together with the result from Lee in [26] that square matching windows can be reduced to half the height without substantial reduction in quality, leads to a question of to what extent a window height reduction can be compensated with an increased width. Ambrosch showed that for window widths beyond the commonly used sizes (up to 21 pixels) the accuracy actually degrades [37].

The ultimate reduction in window height is the 1-dimensional window. It is not extensively found in literature, possibly because its produced disparity map is noisy. However, a few implementations can be found.

Ambrosch [37] uses a 1x1 SAD, for weighting the comparison of a Census matching approach in advantage of the center pixel. Calin et al. use a 1-dimensional SSD [38] implementation. It runs at 30 fps producing dense disparity maps of 160x120 pixels on an FPGA. The objects of the disparity map are excessively bloated, as to be expected when using a wide correlation window, and the depth resolution is limited, partially due to the small image size. Lefebvre et al. [39] presents an approach for 1-dimensional matching
paired with a confidence estimation. The work produces semi-dense disparity maps with associated match confidence map. However, the matching is made through multiple 1-dimensional windows of different sizes and not in real-time. An interesting conclusion of theirs is that the basic 1-dimension approach yields better results than the 2-dimensional in areas of texture and near depth discontinuities [40]. The difference is actually quite substantial for larger window sizes, with the advantage of the 2-dimensional in other areas being marginal. The matching algorithm is SSD, but any correlation technique may be used to construct the correlation volume from which the estimate the disparity and confidence. They show that 1-dimensional windows contain sufficient information for estimating semi-dense disparity maps with good confidence. The approach is far from real-time with a calculation time of 7 seconds for the Tsukuba image pair.

2.6 Disparity Map Improvements

For completing hollow disparity maps, common approaches are to interpolate or propagate disparity values from nearby matched pixels. Yoon et al. [41] perform a spatial interpolation by the use of median filtering. In propagation, the approach is that a window of estimated disparity values completes the non-valid elements with the least value available in the window to limit the foreground fattening, Fusiello et al. [42]. However, a propagation of background disparity values will thin out and often break thin foreground objects. The propagation window can instead be weighted to include disparity information only from same object neighbors. Sun et al. [30] restrict the selection to pixels of similar color, supported by the color-disparity constraint. Although producing good results, propagation methods rely on a fairly accurate first disparity estimation. Moreover, it is common with streaking artifacts in methods of propagation [30].
Chapter 3

Research Summary

The research group in Robotics at Mälardalens Högskola is focused on visual pre-processing for robots and autonomous machines. This initial and crucial stage of autonomy deals with information gathering and environment perception - such as navigation based on visual cues, and object recognition. The work presented has been performed within this group, and the focus has been on electronics, hardware, and looking at computer vision from an electronics perspective.

This chapter presents a short overview of the underlying papers of this thesis.

3.1 Paper Overview

3.1.1 Paper A

Two Camera System for Robot Applications; Navigation, Jörgen Lidholm, Fredrik Ekstrand and Lars Asplund, In proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Hamburg, Germany, September 2008

Summary  We present a hardware-based stereo vision system for navigation. The objective is to create a system for simultaneous location and mapping through the use of vision on an embedded reconfigurable hardware system.

SLAM is a complex task with a lot of data to process and many parameters to consider. Our approach is to see if it is possible to use only a limited feature
descriptor, instead of SIFT or SURF, at high speed to identify landmarks. Although only concerned with a limited number of reasonably separated features, the confidence in a straightforward matching technique (SAD on cornerness) is too low as the corner descriptors are too simple for matching on individual basis. This led to the alternative approach suggested here, which is a combination of traditional stereo matching, back-projection [43] and tracking.

We propose to remove the problem of outlier detection and removal by matching of 3D coordinates. The approach is similar to that of area-based matching. For every feature in one image we match with all possible features in the other image, constrained by the rectified image condition limiting the search area to 1-dimensional. There is no selection performed, all the possible matches are stored (similar to the Disparity Space Image in left-right consistency check implementations). Within this set there can be only one valid match.

This landmark set is stored and the robot is moved slightly. By tracking the motion using wheel-based odometry, we have a notion on how the correct features should have moved in 3D space, and by back-projecting this onto the stored landmark set coordinates, we get their expected new coordinates. Correlating these with the newly acquired landmark set, only those representing the correct landmark should match. The confidence of the landmark increases with the uniqueness and stability (number of correlations). Of course, wheel-based odometry is not reliable over longer paths, so as soon as a sufficient set of landmarks with good confidence is generated, it is superseded by visual odometry.

An FPGA implementation of Stephen and Harris combined edge and corner detector is used to reduce the data amount in the main application. A novel approach focused on a high frame rate to reduce the problem of matching and tracking is proposed. The approach, however, was not fully developed and a modified approach was presented in [44] by the use of clustering.

My contribution I am the second author of this paper contributing with electronics design and implementation, co-implementation of VHDL-components, co-developing the idea, and formulating sections of the text.

3.1.2 Paper B

Resource Limited Hardware-based Stereo Matching for High-Speed Vision System, Fredrik Ekstrand, Carl Ahlberg, Mikael Ekström, Lars Asplund and Giacomo Spampinato, In proceedings of the 5th International Conference on Au-
Summary  The depth assessment in Paper A was not satisfactory. An alternative approach is to work with features with a good initial 3D coordinate guess. A matching component providing valid disparity information in the salient parts of the image only, will allow for depth information without feature matching (by superposition). This concurrent matching component must use only a limited set of resources, in order not to restrict the other processes.

The task is to find a stereo matching approach suitable for resource constrained implementation. An important issue is also the memory requirement of the matching component when handling large images, as the higher level processes may not be blocked from memory access by the correspondence component.

A constrained implementation of two popular correlation approaches specifically suited for hardware implementation, SAD and Census, showed that the basic approach performed best with significant limitation of the matching area. A 1D SAD implementation resulted in a resource optimized disparity component suitable for the task, fulfilling the prerequisites of no limitations in terms of external memory or image size.

My contribution  I am the main author of this paper contributing with the idea, literature survey, algorithm and hardware implementation, and verification. The second author provided relevant insights, data for the publication, software-based validation of findings, and paper revision. The other authors have contributed by giving feedback on the theory and actively participating in paper revisions.

3.1.3 Paper C


Summary  As a direct result of the findings in Paper B, we formulated an extension of the approach into a matching component producing a dense disparity map with retained low resource utilization. Established methods for improving area-based matching methods are implemented from a hardware perspective.
The approach significantly improves the performance of the implementation from Paper B and performs on par with recently published real-time dense disparity map components. The resource utilization is kept low and the memory and image size restrictions are maintained.

**My contribution** I am the main author of the paper, contributing with the state of the art and formulating the approach, as well as performing the hardware implementation and verification. The second author contributed with problem identification, initial testing, development of the approach, and software-based validation. The third author contributed with relevant feedback and insights together with paper revisions. The other authors have contributed by giving feedback on the theory and actively participating in paper revisions.

### 3.2 Research Methodology

The research is based on literature surveys to perceive the state of the art. Approaches are evaluated based on suitability of implementation through empirical methodologies including analysis of quantitative data by community practice.
Chapter 4

Conclusions and Future Work

This thesis gives a quick overview and introduction to image-processing in reconfigurable hardware. Important aspects for implementing in hardware is the suitability of the algorithm in terms of speed, complexity and resource utilization. We have looked at minimizing the system impact to enable concurrent processing of traditionally computationally expensive operations. The key aspect is to focus on speed and process on the go without retaining data in low-level processing.

4.1 Contributions

The work presented in this thesis enables different levels of depth extraction. For the minimized approach of running next to a feature-based navigation system, the approach can supply 3D data in salient areas in high speed and at very low resource usage. Salient regions are important in a wide range of applications, and feature detectors use these regions to enable everything from autonomous navigation to face-detection. Combining feature-based matching with a compact, fast and potent disparity estimator can relieve some of the need for expensive feature descriptors. The benefits would be higher speed and lower resource usage, enabling higher system integration.

We have shown in this thesis that it is possible to retain the quality of one of the most widely used stereo matching algorithm while removing a few of
its downsides. For approaches with a demand for more dense 3D data, the
improved versions can produce semi-dense disparity maps at a high speed, and
without a limitation on the size of the images processed.

The removal of matching data introduces noise, which can be removed by
filtering, especially in area-based matching. The median-filtered 1-dimensional
stereo matching component effectively reduce the resource utilization, but with
retained accuracy. Moreover, the median filter does not improve the 2-dimensional
approach with any significance, which is why the 1-dimensional implementa-
tion in certain aspects actually outperforms its larger counterpart.

4.2 Future Work

Future work includes integration of the feature detector and the disparity esti-
mator to provide feature matching and tracking with high confidence. Another
interesting question is if an advanced confidence measurement can invalidate
false matches at an early stage, and thereby keep the noise from ever entering
the disparity domain. For this to have any relevance, an extended propaga-
tion function is required. As is evident in this thesis, removal of data requires
compensation.

The next step is to run the autonomous system performing navigation in-
doors. Coming future work is to adopt the system for outdoors. A whole new
range of parameters will then need to be considered, such as motion compen-
sation, radiometric distortion, visual noise, etc.
Bibliography


