Review of Stereo Vision Algorithms: From Software to Hardware

Article · November 2008

DOI: 10.1080/15599610802438680

Lazaros Nalpantidis
Technical University of Denmark

Antonios Gasteratos
Democritus University of Thrace

International Journal of Optomechatronics, 2: 435–462, 2008
Copyright © Taylor & Francis Group, LLC
ISSN: 1559-9612 print/1559-9620 online
DOI: 10.1080/15599610802438680

REVIEW OF STEREO VISION ALGORITHMS: FROM SOFTWARE TO HARDWARE

Nalpantidis Lazaros,[1] Georgios Christou Sirakoulis,[2] and Antonios Gasteratos[1]

[1]*Democritus University of Thrace, Department of Production and Management Engineering, Xanthi, Greece*
[2]*Democritus University of Thrace, Department of Electrical and Computer Engineering, Xanthi, Greece*

Stereo vision, resulting in the knowledge of deep information in a scene, is of great importance in the field of machine vision, robotics and image analysis. In this article, an explicit analysis of the existing stereo matching methods, up to date, is presented. The presented algorithms are discussed in terms of speed, accuracy, coverage, time consumption, and disparity range. Towards the direction of real-time operation, the development of stereo matching algorithms, suitable for efficient hardware implementation is highly desirable. Implementations of stereo matching algorithms in hardware for real-time applications are also discussed in details.

1. INTRODUCTION

Stereo vision is a flourishing field, attracting the attention of many researchers (Forsyth and Ponce 2002; Hartley and Zisserman 2004). New approaches are presented every year. Such an expanding volume of work makes it difficult for those interested to keep up with it. An up-to-date survey of the stereo vision matching algorithms would be useful for those already engaged to the field, giving them a brief overview of the advances accomplished, as well as for the newly interested ones, allowing for a quick introduction to the state-of-the-art.

Since the excellent taxonomy presented by Scharstein and Szeliski (2002) and the interesting work of Sunyoto, Mark and Gavrila (2004) many new *stereo matching*, i.e., *stereo correspondence*, methods have been proposed (Yoon and Kweon 2006a; Klaus et al. 2006). Latest trends in the field mainly pursue real-time execution speeds, as well as decent accuracy. As indicated by this survey, the algorithms' theoretical matching cores are quite well established leading the researchers towards innovations resulting in more efficient hardware implementations.

Detecting conjugate pairs in stereo images is a challenging research problem known as the correspondence problem, i.e., to find for each point in the left image, the corresponding point in the right one (Barnard and Thompson 1980). To determine these two points from a conjugate pair, it is necessary to measure the similarity

Address correspondence to Antonios Gasteratos, Democritus University of Thrace, Department of Production and Management Engineering, GR-67 100 Xanthi, Greece. E-mail: agaster@pme.duth.gr

NOMENCLATURE

SAD	Sum of absolute differences	E	Energy function
SSD	Sum of squared differences	I_i	Image intensity ($i = l$, r for the left and the right image, respectively)
NCC	Normalized cross-correlation		
D	Disparity		

of the points. The point to be matched without any ambiguity should be distinctly different from its surrounding pixels. Several algorithms have been proposed in order to address this problem. However, every algorithm makes use of a *matching cost function* so as to establish correspondence between two pixels. The most common ones are absolute intensity differences (AD), the squared intensity differences (SD) and the normalized cross correlation (NCC). Evaluation of various matching costs can be found in (Scharstein and Szeliski 2002; Mayoral et al. 2004; Hirschmuller and Scharstein 2007). Usually, the matching costs are aggregated over *support regions*. Those support regions, often referred to as support or *aggregating windows*, could be square or rectangular, fix-sized or adaptive ones. The aggregation of the aforementioned cost functions, leads to the core of most of the stereo vision methods, which can be mathematically expressed as follows, for the case of the sum of absolute differences (SAD)

$$SAD(x,y,d) = \sum_{x,y \in W} |I_l(x,y) - I_r(x,y-d)| \tag{1}$$

for the case of the sum of squared differences (SSD)

$$SSD(x,y,d) = \sum_{x,y \in W} (I_l(x,y) - I_r(x,y-d))^2 \tag{2}$$

and for the case of the NCC

$$NCC(x,y,d) = \frac{\sum_{x,y \in W} I_l(x,y) \cdot I_r(x,y-d)}{\sqrt{\sum_{x,y \in W} I_l^2(x,y) \cdot \sum_{x,y \in W} I_r^2(x,y-d)}} \tag{3}$$

where I_l, I_r are the intensity values in left and right image, (x, y) are the pixel's coordinates, d is the disparity value under consideration and W is the aggregated support region. Despite of sophisticated aggregation being traditionally considered as time-consuming, Gong and his colleagues (2007) study the performance of various aggregation approaches suitable for real-time methods.

The selection of the appropriate disparity value for each pixel is performed afterwards. The simpler algorithms make use of winner-takes-all (WTA) method of disparity selection

$$D(x,y) = \arg\min SAD(x,y,d) \tag{4}$$

i.e., for every pixel (x, y) and for constant value of disparity d the minimum cost is selected. Equation 4 refers to the SAD method but any other could be used instead. However, in many cases disparity selection is an iterative process, since each pixel's disparity is depending on its neighboring pixels' disparity. As a result, more than one iterations are needed in order to find the best set of disparities. This stage differentiates the local from the global algorithms, which will be analyzed later in this section. An additional disparity refinement step is frequently used. The general structure of the majority of stereo correspondence algorithms (Scharstein and Szeliski 2002) is shown in Figure 1.

Stereo correspondence algorithms can be grouped into those producing *sparse* output and those giving a *dense* result. Feature based methods stem from human vision studies and are based on matching segments or edges between two images, thus resulting in a sparse output. This disadvantage, dreadful for many purposes, is counterbalanced by the accuracy and speed obtained. However, contemporary applications demand more and more dense output. This is the reason why this work is mainly focused on stereo correspondence algorithms that produce dense output. In order to categorize and evaluate them a context has been proposed (Scharstein and Szeliski 2002). According to this, dense matching algorithms are classified in local and global ones. *Local methods* (area-based) trade accuracy for speed. They are also referred to as window-based methods because disparity computation at a given point depends only on intensity values within a finite support window. *Global methods* (energy-based) on the other hand are time consuming but very accurate. Their goal is to minimize a global cost function, which combines data and smoothness terms, taking into account the whole image. Of course, there are many other methods (Liu et al. 2006) that are not strictly included in either of these two broad classes. The issue of stereo matching has recruited a variation of computation tools. Advanced computational intelligence techniques are not uncommon and present interesting and promiscuous results (Binaghi et al. 2004; Kotoulas et al. 2005).

While the aforementioned categorization involves stereo matching algorithms in general, in practice it is valuable for software implemented algorithms only. Software implementations make use of general purpose personal computers (PC) and usually result in considerably long running times. However, this is not an option when the objective is the development of autonomous robotic platforms, simultaneous localization and mapping (SLAM) or virtual reality (VR) systems. Such tasks require real-time, efficient performance and demand dedicated hardware and consequently specially developed and optimized algorithms. Only a small subset of the already proposed algorithms is suitable for hardware implementation. Hardware implemented algorithms are characterized from their theoretical algorithm as well as the implementation itself. There are two broad classes of hardware implementations: the field-programmable gate arrays (FPGA) and the application-specific integrated circuits (ASIC) based ones. Figure 2 depicts an ASIC chip (*a*) and a FPGA development board (*b*). Each one can execute stereo vision algorithms without the necessity of a PC, saving volume, weight and consumed energy. However, the evolution of

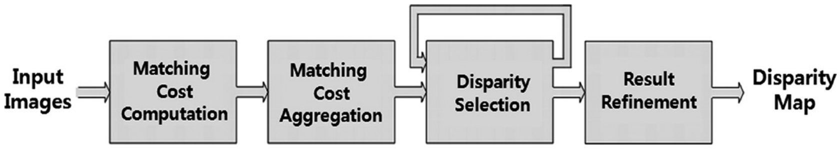

Figure 1. Generalized block diagram of a stereo correspondence algorithm.

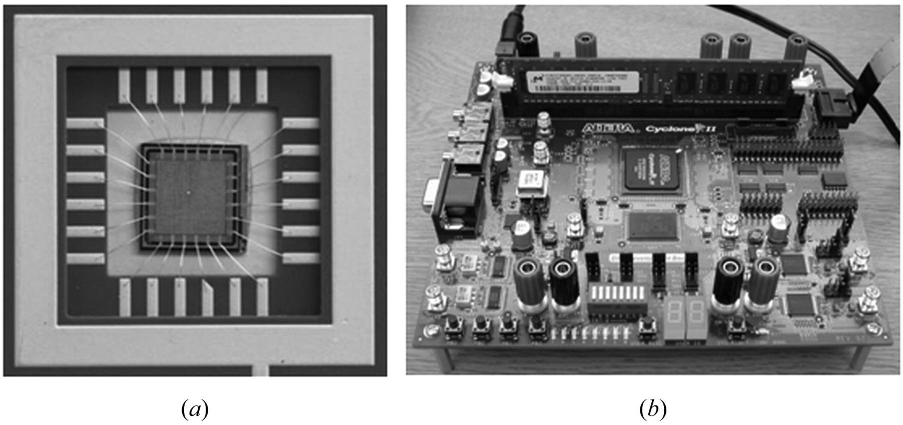

Figure 2. An ASIC chip (*a*) and a FPGA development board (*b*).

FPGA has made them an appealing choice due to the small prototyping times, their flexibility and their good performance.

According to the structure of this paper, recent stereo matching algorithms are presented in section 2. Algorithms resulting in both dense and sparse output are discussed, focusing, however, on the first category. In particular, dense disparity algorithms utilizing local, global or other kinds of methods are considered. Section 3 deals with hardware implementations which are able to achieve real-time performance. FPGA implementations that use SAD, DP or phase-based methods are presented. ASIC implementations are also covered. Finally, the conclusions and the outcomes from the previously deployed bibliographic survey are evaluated in section 4. The taxonomy deployed in this paper, in accordance with the aforementioned structure is schematically presented in Figure 3.

2. RECENT STEREO MATCHING ALGORITHMS

The issue of stereo correspondence is of great importance in the field of machine vision (Jain et al. 1995), computer vision (Forsyth and Ponce 2002), virtual reality, robot navigation (Metta, Gasteratos, and Sandini 2004), depth measurements (Manzotti et al. 2001) and environment reconstruction as well as in many other aspects of production, security, defense, exploration, and entertainment. Calculating the distance of various points or any other primitive in a scene relative to the position of a camera is one of the important tasks of a computer vision system. The most common method for extracting depth information from intensity images is by means of a pair of synchronized camera-signals, acquired by a stereo rig. The point-by-point matching between the two images from the stereo setup derives the depth images, or the so called disparity maps, (Faugeras 1993). This matching can be done as a one dimensional search if accurately rectified stereo pairs in which horizontal scan lines reside on the same epipolar line are assumed, as shown in Figure 4. A point P_1 in one image plane may have arisen from any of points in the line C_1P_1, and may appear in the alternate image plane at any point on the so-called epipolar line E_2 (Jain et al. 1995). Thus, the

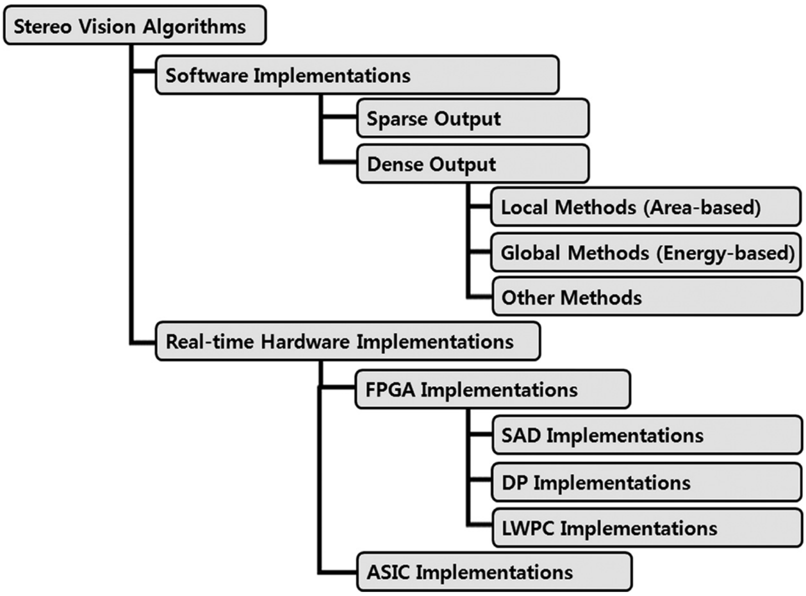

Figure 3. Categorization of stereo vision algorithms.

search is theoretically reduced within a scan line, since corresponding pair points reside on the same epipolar line. The difference on the horizontal coordinates of these points is the disparity. The disparity map consists of all disparity values of the image. Having extracted the disparity map, problems such as 3D reconstruction, positioning, mobile robot navigation, obstacle avoidance, etc., can be dealt with in a more efficient way (Murray and Jennings 1997; Murray and Little 2000).

Contemporary research in stereo matching algorithms, in accordance with the ideas of Maimone and Shafer (1996), is reigned by the test bench available

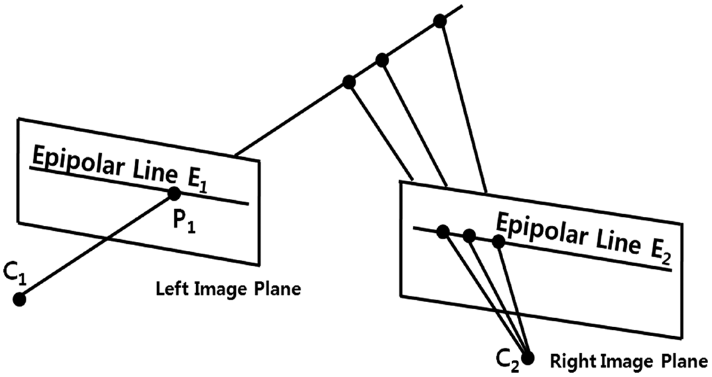

Figure 4. Geometry of epipolar lines, where C_1 and C_2 are the left and right camera lens centers, respectively. Point P_1 in one image plane may have arisen from any of points in the line C_1P_1, and may appear in the alternate image plane at any point on the epipolar line E_2.

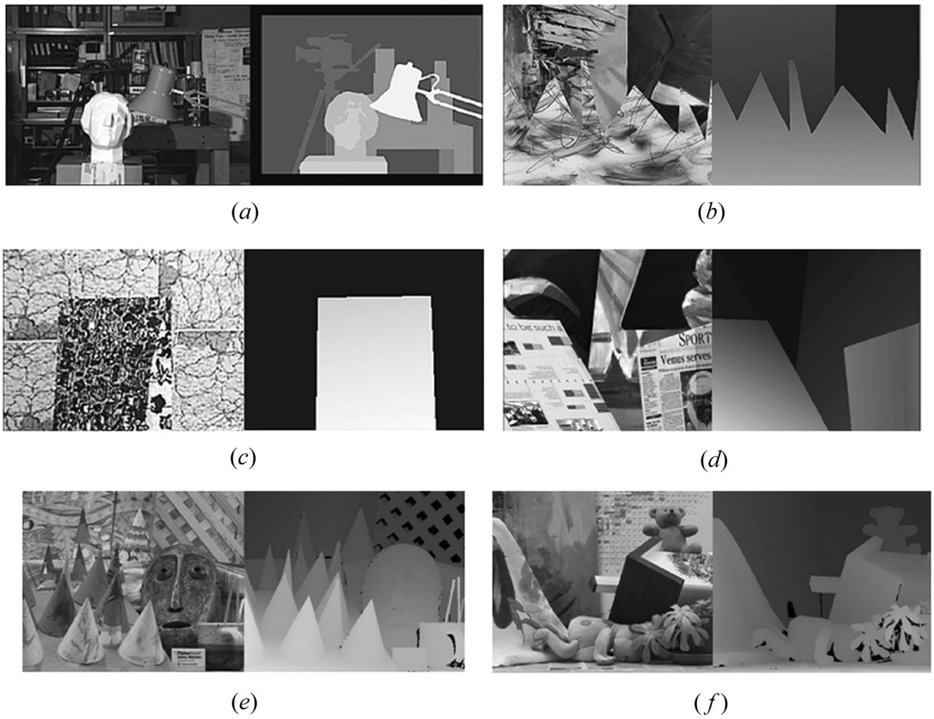

Figure 5. Left image of the stereo pair (left) and ground truth (right) for the Tsukuba (*a*), Sawtooth (*b*), Map (*c*), Venus (*d*), Cones (*e*) and Teddy (*f*) stereo pair.

at the site maintained by Scharstein and Szeliski (*http://vision.middlebury.edu/stereo/*). As numerous methods have been proposed since then, this section aspires to review the most recent ones, i.e., mainly those published during and after 2004. Most of the results presented in the rest of this paper are based on the image sets (Scharstein and Szeliski 2002; Scharstein and Szeliski 2003) and test provided there. The most common image sets are presented in Figure 5. Table 1 summarizes their size as well the number of disparity levels. Experimental results based on these image sets are given, where available. The preferred metric adopted by in this paper, in order to depict the quality of the resulting disparity maps, is the percentage of pixels whose absolute disparity error is greater than 1 in the unoccluded areas of the image. This metric, considered the most representative of the result's quality, was used so as to make comparison easier. Other metrics, like error rate and root mean square error are also employed. The speed

Table 1. Characteristics of the most common image sets

	Tsukuba	Map	Sawtooth	Venus	Cone	Teddy
Size in pixels	384 × 288	284 × 216	434 × 380	434 × 383	450 × 375	450 × 375
Disparity levels	16	30	20	20	60	60

with which the algorithms process input image pairs is expressed in frames per second (fps). This metric has of course a lot to do with the used computational platform and the kind of the implementation. Inevitably, speed results are not directly comparable.

2.1. Dense Disparity Algorithms

Methods that produce dense disparity maps gain popularity as the computational power grows. Moreover, contemporary applications are benefited by, and consequently demand dense depth information. Therefore, during the latest years efforts towards this direction are being reported much more frequently than towards the direction of sparse results.

Dense disparity stereo matching algorithms can be divided in two general classes, according to the way they assign disparities to pixels. Firstly, there are algorithms that decide the disparity of each pixel according to the information provided by its local, neighboring pixels. There are, however, other algorithms which assign disparity values to each pixel depending on information derived from the whole image. Consequently, the former ones are called local methods while the latter ones global.

2.1.1. Local methods. Local methods are usually fast and can at the same time produce descent results. Several new methods have been presented. In Figure 6*a* Venn diagram presents the main characteristics of the below presented local methods. Under the term color usage we have grouped the methods that take advantage of the chromatic information of the image pair. Any algorithm can process color images but not everyone can use it in a more beneficial way. Furthermore, in Figure 6 NCC stands for the use of normalized cross correlation and SAD for the use of sum of absolute differences as the matching cost function. As expected, the use of SAD as matching cost is far more widespread than any other.

Muhlmann and his colleagues (2002) describe a method that uses the sum of absolute differences (SAD) correlation measure for RGB color images. It achieves high speed and reasonable quality. It makes use of the left to right consistency and uniqueness constraints and applies a fast median filter to the results. It can

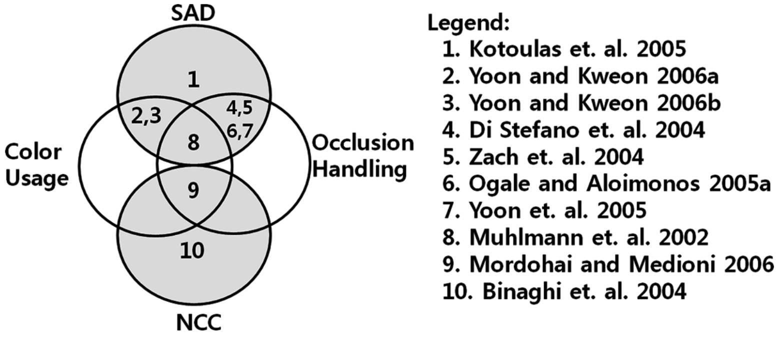

Figure 6. Diagrammatic representation of the local methods' categorization.

achieve 20 fps for 160 × 120 pixels image size, making this method suitable for real-time applications. The PC platform is Linux (kernel 2.4.6) on a dual processor 800 MHz Pentium III system with 512 MB of RAM.

Another fast area-based stereo matching algorithm, which uses the SAD as error function, is presented in (Di Stefano et al. 2004). Based on the uniqueness constraint, it rejects previous matches as soon as better ones are detected. In contrast to bidirectional matching algorithms this one performs only one matching phase, having though similar results. The results obtained are tested for reliability and sub-pixel refined. It produces dense disparity maps in real-time using an Intel Pentium III processor running at 800 MHz. The algorithm achieves 39.59 fps speed for 320 × 240 pixels and 16 disparity levels and the root mean square error for the standard Tsukuba pair is 5.77%.

On the contrary, Ogale and Aloimonos (2005a) take into consideration the shape of the objects depicted and demonstrate the importance of the vertical and horizontal slanted surfaces. The authors propose the replacement of the standard uniqueness constraint referred to pixels with a uniqueness constraint referred to line segments along a scanline. So the method performs interval matching instead of pixel matching. The slants of the surfaces are computed along a scanline, a stretching factor is then obtained and the matching is performed based on the absolute intensity difference. The object is to achieve minimum segmentation. The experimental results indicate 1.77%, 0.61%, 3.00%, and 7.63% error percentages for the Tsukuba, Sawtooth, Venus and Map stereo pairs, respectively. The execution speed of the algorithm varies from 1 to 0.2 fps on a 2.4 GHz processor.

Another method that presents almost real-time performance is reported in (Yoon et al. 2005). It makes use of a refined implementation of the SAD method and a left-right consistency check. The errors in the problematic regions are reduced using different sized correlation windows. Finally, a median filter is used in order to interpolate the results. The algorithm is able to process 7 fps for 320 × 240 pixels images and 32 disparity levels. These results are obtained using an Intel Pentium 4 at 2.66 GHz Processor.

A window-based method for correspondence search is presented in (Yoon and Kweon 2006a) that uses varying support-weights. The support-weights of the pixels in a given support window are adjusted based on color similarity and geometric proximity to reduce the image ambiguity. The difference between pixel colors is measured in the CIELab color space because the distance of two points in this space is analogous to the stimulus perceived by the human eye. The running time for the Tsukuba image pair with a 35 × 35 pixels support window is about 0.016 fps on an AMD 2700+ processor. The error ratio is 1.29%, 0.97%, 0.99%, and 1.13% for the Tsukuba, Sawtooth, Venus and Map image sets, respectively. These figures can be further improved through a left-right consistency check.

The same authors propose a pre-processing step for correspondence search in the presence of specular highlights in (Yoon and Kweon 2006b). For given input images, specular-free two-band images are generated. The similarity between pixels of these input-image representations can be measured using various correspondence search methods such as the simple SAD-based method, the adaptive support-weights method (Yoon and Kweon 2006c) and the dynamic programming (DP) method. This pre-processing step can be performed in real time and compensates satisfactory for specular reflections.

Binaghi (Binaghi et al. 2004) on the other hand, have chosen to use the zero mean normalized cross correlation (ZNCC) as matching cost. This method integrates a neural network (NN) model, which uses the least-mean-square delta rule for training. The NN decides on the proper window shape and size for each support region. The results obtained are satisfactory but the 0.024 fps running speed reported for the common image sets, on a Windows platform with a 300 MHz processor, renders this method as not suitable for real-time applications.

Based on the same matching cost function a more complex area-based method is proposed in (Mordohai and Medioni 2006). A perceptual organization framework, considering both binocular and monocular cues is utilized. An initial matching is performed by a combination of normalized cross correlation techniques. The correct matches are selected for each pixel using tensor voting. Matches are then grouped into smooth surfaces. Disparities for the unmatched pixels are assigned so as to ensure smoothness in terms of both surface orientation and color. The percentage of unoccluded pixels whose absolute disparity error is greater than 1 is 3.79, 1.23, 9.76, and 4.38 for the Tsukuba, Venus, Teddy and Cones image sets. The execution speed reported is about 0.002 fps for the Tsukuba image pair with 20 disparity levels running on an Intel Pentium 4 processor at 2.8 MHz.

There are, of course, more hardware-oriented proposals as well. Many of them take advantage of the contemporary powerful graphics machines to achieve enhanced results in terms of processing time and data volume. A hierarchical disparity estimation algorithm implemented on programmable 3D graphics processing unit (GPU) is reported in (Zach et al. 2004). This method can process either rectified or uncalibrated image pairs. Bidirectional matching is utilized in conjunction with a locally aggregated sum of absolute intensity differences. This implementation, on an ATI Radeon 9700 Pro, can achieve up to 50 fps for 256 × 256 pixel input images.

Moreover, the use of Cellular Automata (CA) is exploited in (Kotoulas, Gasteratos et al. 2005). This work presents an architecture for real-time extraction of disparity maps. It is capable of processing 1Mpixels image pairs at more than 40 fps. The core of the algorithm relies on matching pixels of each scan-line using a one-dimensional window and the SAD matching cost as described in (Kotoulas, Georgoulas et al. 2005). This method involves a pre-processing mean filtering step and a post-processing CA based filtering one. CA are models of physical systems, where space and time are discrete and interactions are local. They can easily handle complicated boundary and initial conditions. In CA analysis, physical processes and systems are described by a cell array and a local rule, which defines the new state of a cell depending on the states of its neighbors. All cells can work in parallel due to the fact that each cell can independently update each own state. Therefore the proposed CA algorithm is massively parallel and is an ideal candidate to be implemented in hardware (Nalpantidis, Sirakoulis, and Gasteratos 2007).

The main features of the discussed local algorithms are summarized in Table 2.

Table 2. Characteristics of local algorithms

Author	Method	Features	Speed (fps)	Image size	Disparity levels	Computational platform
Muhlmann et al. 2002	SAD	Color usage Occlusion handling Left to right consistency Uniqueness constraints	20	160 × 120		Intel Pentium III 800 MHz with 512 MB RAM
Di Stefano, Marchionni and Mattoccia 2004	SAD	Occlusion handling Uniqueness constraint	39.59	320 × 240	16	Intel Pentium III 800 MHz
Ogale and Aloimonos 2005a	SAD	Occlusion handling Interval uniqueness constraint	1	384 × 288	16	2.4 GHz
Yoon et al. 2005	SAD	Occlusion handling Left-right consistency check Variable windows	7	320 × 240	32	Intel Pentium 4 2.66 GHz
Yoon and Kweon 2006a	SAD	Color usage Varying support-weights	0.016	384 × 288	16	AMD 2700 +
Yoon and Kweon 2006b	SAD	Color usage Varying support weights Specular reflection compensation	0.016	384 × 288	16	AMD 2700 +
Binaghi et al. 2004	ZNCC	Varying windows based on neural networks	0.024	284 × 216	30	300 MHz
Mordohai and Medioni 2006	NCC	Color usage Occlusion handling Tensor voting	0.002	384 × 288	20	Intel Pentium 2.8 MHz
Zach, Karner and Bischof 2004	SAD	Occlusion handling Implemented on GPU Bidirectional matching	50	256 × 256	88	ATI Radeon 9700 Pro
Kotoulas et al. 2005	SAD	Cellular automata	40	1000 × 1000		

2.1.2. Global methods.

Contrary to local methods, global ones produce very accurate results. Their goal is to find the optimum disparity function $d = d(x, y)$ which minimizes a global cost function E, which combines data and smoothness terms.

$$E(d) = E_{data}(d) + \lambda \cdot E_{smooth}(d). \tag{5}$$

where E_{data} takes into consideration the (x, y) pixel's value throughout the image, E_{smooth} provides the algorithm's smoothening assumptions and λ is a weight factor.

The main disadvantage of the global methods is that they are more time consuming and computational demanding. The source of these characteristics is the iterative refinement approaches that they employ. They can be roughly divided in those performing a global energy minimization and those pursuing the minimum for independent scanlines using DP.

In Figure 7 the main characteristics of the below discussed global algorithms are presented. It is clear that the recently published works utilizes global optimization preferably rather than DP. This observation is not a surprising one, taking into consideration the fact that under the term global optimization there are actually quite a few different methods. Additionally, DP tends to produce inferior, thus less impressive, results. Therefore, applications that don't have running speed constraints, preferably utilize global optimization methods.

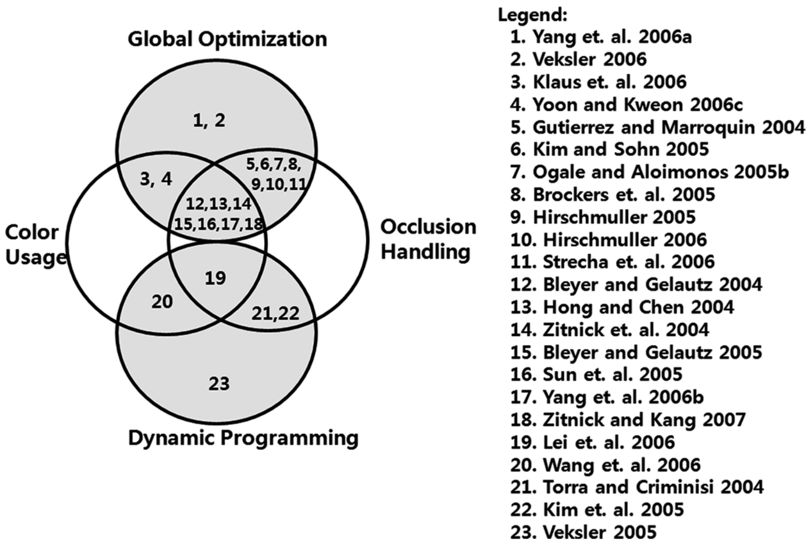

Figure 7. Diagrammatic representation of the global methods' categorization.

2.1.2.1. Global optimization. The algorithms that perform global optimization take into consideration the whole image in order to determine the disparity of every single pixel. An increasing portion of the global optimization methodologies involves segmentation of the input images according to their colors.

The algorithm presented in (Bleyer and Gelautz 2004) uses color segmentation. Each segment is described by a planar model and assigned to a layer using a mean shift based clustering algorithm. A global cost function is used that takes into account the summed up absolute differences, the discontinuities between segments and the occlusions. The assignment of segments to layers is iteratively updated until the cost function improves no more. The experimental results indicate that the percentage of unoccluded pixels whose absolute disparity error is greater than 1 is 1.53, 0.16, and 0.22 for the Tsukuba, Venus and Sawtooth image sets, respectively.

The stereo matching algorithm proposed in (Hong and Chen 2004) makes use of color segmentation in conjunction with the graph cuts method. The reference image is divided in non-overlapping segments using the mean shift color segmentation algorithm. Thus, a set of planes in the disparity space is generated. The goal of minimizing an energy function is faced in the segment rather than the pixel domain. A disparity plane is fitted to each segment using the graph cuts method. This algorithm presents good performance in the textureless and occluded regions as well as at disparity discontinuities. The running speed reported is 0.33 fps for a 384 × 288 pixel image pair when tested on a 2.4 GHz Pentium 4 PC. The percentage of bad matched pixels for the Tsukuba, Sawtooth, Venus, and Map image sets is found to be 1.23, 0.30, 0.08, and 1.49, respectively.

The ultimate goal of the work described in (Zitnick et al. 2004) is to render dynamic scenes with interactive viewpoint control produced by a few cameras. A suitable color segmentation-based algorithm is developed and implemented on a programmable ATI 9800 PRO GPU. Disparities within segments must vary smoothly, each image is treated equally, occlusions are modeled explicitly and consistency between disparity maps is enforced resulting in higher quality depth maps. The results for each pixel are refined in conjunction with the others.

Another method that uses the concept of image color segmentation is reported in (Bleyer and Gelautz 2005). An initial disparity map is calculated using an adapting window technique. The segments are combined in larger layers iteratively. The assignment of segments to layers is optimized using a global cost function. The quality of the disparity map is measured by warping the reference image to the second view and comparing it with the real image and calculating the color dissimilarity. For the 384 × 288 pixel Tsukuba and the 434 × 383 pixel Venus test set, the algorithm produces results at 0.05 fps rate. For the 450 × 375 pixel Teddy image pair, the running speed decreased to 0.01 fps due to the increased scene complexity. Running speeds refer to an Intel Pentium 4 2.0 GHz processor. The root mean square error obtained is 0.73 for the Tsukuba, 0.31 for the Venus and 1.07 for the Teddy image pair.

Moreover, Sun and his colleagues (2005) presented a method which treats the two images of a stereo pair symmetrically within an energy minimization framework that can also embody color segmentation as a soft constraint. This method enforces that the occlusions in the reference image are consistent with the disparities found for the other image. Belief propagation iteratively refines the results. Moreover, results for the version of the algorithm that incorporates segmentation are better. The percentage of pixels with disparity error larger than 1 is 0.97, 0.19, 0.16, and 0.16 for the Tsukuba, Sawtooth, Venus and Map image sets, respectively. The running speed for the aforementioned data sets is about 0.02 fps tested on a 2.8 GHz Pentium 4 processor.

Color segmentation is utilized in (Klaus et al. 2006) as well. The matching cost used here is a self-adapting dissimilarity measure that takes into account the sum of absolute intensity differences as well as a gradient based measure. Disparity planes are extracted using an insensitive to outliers technique. Disparity plane labeling is performed using belief propagation. Execution speed varies between 0.07 and 0.04 fps on a 2.21 GHz AMD Athlon 64 processor. The results indicate 1.13, 0.10, 4.22, and 2.48 percent of bad matched pixels in non-occluded areas for the Tsukuba, Venus, Teddy and Cones image sets, respectively.

Finally, one more algorithm that utilizes energy minimization, color segmentation, plane fitting and repeated application of hierarchical belief propagation is presented in (Yang et al. 2006b). This algorithm takes into account a color-weighted correlation measure. Discontinuities and occlusions are properly handled. The percentage of pixels with disparity error larger than 1 is 0.88, 0.14, 3.55, and 2.90 for the Tsukuba, Venus, Teddy and Cones image sets, respectively.

In (Yoon and Kweon 2006c) two new symmetric cost functions for global stereo methods are proposed. A symmetric data cost function for the likelihood, as well as a symmetric discontinuity cost function for the prior in the MRF model for stereo is presented. Both the reference image and the target image are taken into account to improve performance without modeling half-occluded pixels explicitly and without using color segmentation. The use of both of the two proposed symmetric cost functions in conjunction with a belief propagation based stereo method is evaluated. Experimental results for standard test bed images show that the performance of the belief propagation based stereo method is greatly improved by the combined use of the proposed symmetric cost functions. The percentage of pixels badly matched for the non-occluded areas was found 1.07, 0.69, 0.64, and 1.06 for the Tsukuba, Sawtooth, Venus and Map image sets, respectively.

The incorporation of Markov random fields (MRF) as a computational tool is also a popular approach.

A method based on the Bayesian estimation theory with a prior MRF model for the assigned disparities is described in (Gutierrez and Marroquin 2004). The continuity, coherence and occlusion constraints as well as the adjacency principal are taken into account. The optimal estimator is computed using a Gauss-Markov random field model for the corresponding posterior marginals, which results in a diffusion process in the probability space. The results are accurate but the algorithm is not suitable for real-time applications, since it needs a few minutes to process a 256×255 stereo pair with up to 32 disparity levels, on an Intel Pentium III running at 450 MHz.

On the other hand, Strecha and his colleagues (2006) treat every pixel of the input images as generated either by a process, responsible for the pixels visible from the reference camera and which obey to the constant brightness assumption, or by an outlier process, responsible for the pixels that cannot be corresponded. Depth and visibility are jointly modeled as a hidden MRF, and the spatial correlations of both are explicitly accounted for by defining a suitable Gibbs prior distribution. An expectation maximization (EM) algorithm keeps track of which points of the scene are visible in which images, and accounts for visibility configurations. The percentages of pixels with disparity error larger than 1 are 2.57, 1.72, 6.86, and 4.64 for the Tsukuba, Venus, Teddy and Cones image sets, respectively.

Moreover, a stereo method specifically designed for image-based rendering is described in (Zitnick and Kang 2007). This algorithm uses over-segmentation of the input images and computes matching values over entire segments rather than single pixels. Color-based segmentation preserves object boundaries. The depths of the segments for each image are computed using loopy belief propagation within a MRF framework. Occlusions are also considered. The percentage of bad matched pixels in the unoccluded regions is 1.69, 0.50, 6.74, and 3.19 for the Tsukuba, Venus, Teddy and Cones image sets, respectively. The aforementioned results refer to a 2.8 GHz PC platform.

In (Hirschmuller 2005) an algorithm based on a hierarchical calculation of a mutual information based matching cost is proposed. Its goal is to minimize a proper global energy function, not by iterative refinements but by aggregating matching costs for each pixel from all directions. The final disparity map is sub-pixel accurate and occlusions are detected. The processing speed for the Teddy image set is 0.77 fps. The error in unoccluded regions is found less than 3% for all the standard image sets. Calculations are made on an Intel Xeon processor running at 2.8 GHz.

An enhanced version of the previous method is proposed by the same author in (Hirschmuller 2006). Mutual information is once again used as cost function. The extensions applied in it result in intensity consistent disparity selection for untextured areas and discontinuity preserving interpolation for filling holes in the disparity maps. It treats successfully complex shapes and uses planar models for untextured areas. Bidirectional consistency check, sub-pixel estimation as well as invalid-disparities interpolation are performed. The experimental results indicate that the percentages of bad matching pixels in unoccluded regions are 2.61, 0.25, 5.14, and 2.77 for the Tsukuba, Venus, Teddy and Cones image sets, respectively, with 64 disparity levels searched each time. However, the reported running speed on a 2.8 GHz PC is less than 1 fps.

The work done by Kim and Sohn (2005) introduces a two-stage algorithm consisting of hierarchical dense disparity estimation and vector field regularization. The dense disparity estimation is accomplished by a region dividing technique that uses a Canny edge detector and a simple SAD function. The results are refined by regularizing the vector fields by means of minimizing an energy function. The root mean square error obtained from this method is 0.9278 and 0.9094 for the Tsukuba and Sawtooth image pairs. The running speed is 0.15 fps and 0.105 fps respectively on a Pentium 4 PC running Windows XP.

An uncommon measure is used in (Ogale and Aloimonos 2005b). This work describes an algorithm which is focused on achieving contrast invariant stereo matching. It relies on multiple spatial frequency channels for local matching. The measure for this stage is the deviation of phase difference from zero. The global solution is found by a fast non-iterative left right diffusion process. Occlusions are found by enforcing the uniqueness constraint. The algorithm is able to handle significant changes in contrast between the two images and can handle noise in one of the frequency channels. The Matlab implementation of the algorithm is capable of processing the Middlebury image pairs at 0.5 to 0.25 fps rate, on a 2 GHz computer platform.

The method described in (Brockers et al. 2005) uses a cost relaxation approach. A similarity measurement is obtained as a preliminary stage of the relaxation process. Relaxation is an iterative process that minimizes a global cost function while taking into account the continuity constraint and the neighbor-pixel expected similarity. The support regions are 3D within the disparity space volume and have Gaussian weights. The disparity is available at any time of the iteratively refinement phase, having of course diminished accuracy for little iteration cycles. This feature makes this method suitable for time-critical applications. The percentages of bad matching pixels in unoccluded regions are found to be 4.76, 1.41, 8.18, and 3.91 for the Tsukuba, Venus, Teddy and Cones image sets, respectively.

Another algorithm that generates high quality results in real time is reported in (Yang et al. 2006a) It is based on the minimization of a global energy function comprising of a data and a smoothness term. The hierarchical belief propagation iteratively optimizes the smoothness term but it achieves fast convergence by removing redundant computations involved. In order to accomplish real-time operation authors take advantage of the parallelism of graphics hardware (GPU). Experimental results indicate 16 fps processing speed for 320 × 240 pixel self-recorded images with 16 disparity levels. The percentages of bad matching pixels in unoccluded regions for the Tsukuba, Venus, Teddy and Cones image sets are found to be 1.49, 0.77, 8.72, and 4.61. The computer used is a 3 GHz PC and the GPU is an NVIDIA GeForce 7900 GTX graphics card with 512 M video memory.

The work of Veksler (2006) indicates that computational cost of the graph cuts stereo correspondence technique can be efficiently decreased using the results of a simple local stereo algorithm to limit the disparity search range. The idea is to analyze and exploit the failures of local correspondence algorithms. This method can accelerate the processing by a factor of 2.8, compared to the sole use of graph cuts, while the resulting energy is worse only by an average of 1.7%. These results proceed from an analysis done on a large dataset of 32 stereo pairs using a Pentium 4 at 2.6 GHz PC. This is a considerable improvement in efficiency gained for a small price in accuracy, and it moves the graph-cuts based algorithms closer to real-time implementation. The running speeds are 0.77, 0.38, 0.16, 0.17, 0.53, and 1.04 fps for the Tsukuba, Venus, Teddy, Cones, Sawtooth and Map image sets, respectively while the corresponding error percentages are found 2.22, 1.39, 12.8, 8.87, 1.18, and 0.51.

The main features of the discussed global algorithms that utilize global optimization are summarized in Table 3.

2.1.2.2. Dynamic programming. Many researchers develop stereo correspondence algorithms based on DP. This methodology is a fair trade-off between the complexity of the computations needed and the quality of the results obtained. In every aspect, DP stands between the local algorithms and the global optimization ones. However, its computational complexity still renders it as a less preferable option for hardware implementation.

The work of Torra and Criminisi (2004) presents a unified framework that allows the fusion of any partial knowledge about disparities, such as matched features and known surfaces within the scene. It combines the results from corner, edge and dense stereo matching algorithms to impose constraints that act as guide points to the standard DP method. The result is a fully automatic dense stereo system with up to four times faster running speed and greater accuracy compared to results obtained by the sole use of DP.

Moreover, a generalized ground control points (GGCPs) scheme is introduced in (Kim et al. 2005). One or more disparity candidates for the true disparity of each pixel are assigned by local matching using oriented spatial filters. Afterwards, a two-pass DP technique that performs optimization both along and between the scanlines is performed. The result is the reduction of false matches as well as of the typical inter-scanline inconsistency problem. The percentage of bad matched pixels in unoccluded regions is 1.53, 0.61, 0.94, and 0.706 for the Tsukuba, Sawtooth, Venus and Map image sets. The running speeds, tested on a Pentium 4 at 2.4 GHz PC, vary from 0.23 fps for the Tsukuba set with 15 disparity levels down to 0.08 fps for the Sawtooth set with 21 disparity levels.

Wang et al. (2006) present a stereo algorithm that combines high quality results with real-time performance. DP is used in conjunction with an adaptive aggregation step. The per-pixel matching costs are aggregated in the vertical direction only resulting in improved inter-scanline consistency and sharp object boundaries. This work exploits the color and distance proximity based weight assignment for the pixels inside a fixed support window as reported in (Yoon and Kweon 2006a). The real-time performance is achieved due to the parallel use of the CPU and the GPU of a computer. This implementation can process 320 × 240 pixel images with 16 disparity levels at 43.5 fps and 640 × 480 pixel images with 16 disparity levels at 9.9 fps. The test system is a 3.0 GHz PC with an ATI Radeon XL1800 GPU.

Table 3. Characteristics of global algorithms that use global optimization

Author	Method	Features	Speed (fps)	Image size	Disparity levels	Computational platform
Bleyer and Gelautz 2004	Global cost function	Color segmentation Occlusion handling				
Hong and Chen 2004	graph cuts	Color segmentation Occlusion handling	0.33	384 × 288	16	Intel Pentium 4 2.4 GHz
Zitnick et al. 2004	Global cost function	Color segmentation Occlusion handling GPU utilization				ATI 9800 PRO GPU
Bleyer and Gelautz 2005	Global cost function	Color segmentation Occlusion handling	0.05	384 × 288	16	Intel Pentium 4 2.0 GHz
Sun et al. 2005	Belief propagation	Color segmentation Occlusion handling Symmetricaly treatment	0.02	384 × 288	16	Intel Pentium 4 2.8 GHz
Klaus et al. 2006	Belief propagation	Color segmentation	0.07	384 × 288	16	AMD Athlon 64 2.21 GHz
Yang et al. 2006b	Hierarchical belief propagation	Color segmentation Occlusion handling				
Yoon and Kweon 2006c	Belief propagation	Color segmentation Symmetrical cost functions				
Gutierrez and Marroquin 2004	Gauss-Markov random field	Occlusion handling Continuity Coherence Occlusion Adjacency	<0.017	256 × 255	32	Intel Pentium III 450 MHz
Strecha et al. 2006	Hidden Markov random field	Occlusion handling Expectation maximization algorithm				
Zitnick and Kang 2007	Belief propagation within a MRF framework	Color segmentation Occlusion handling				2.8 GHz
Hirschmuller 2005	Global cost function	Occlusion handling Mutual information	0.77	450 × 375	60	Intel Xeon 2.8 GHz
Hirschmuller 2006	Global cost function	Occlusion handling Mutual information Bidirectional match	<1	450 × 375	64	2.8 GHz
Kim and Sohn 2005	Vector field regularization	Occlusion handling Canny edge detector	0.15	384 × 288	16	Intel Pentium 4
Ogale and Aloimonos 2005b	Left-right diffusion process	Occlusion handling Phase-based matching	0.5	384 × 288	16	2 GHz
Brockers et al. 2005	Cost relaxation	Occlusion handling 3D support regions				
Yang et al. 2006a	Hierarchical belief propagation	GPU utilization	16	320 × 240	16	-3 GHz CPU -NVIDIA GeForce 7900 GTX GPU
Veksler 2006	graph cuts	Prior local stereo algorithm	1.04	434 × 383	20	Intel Pentium 4 2.6 GHz

Table 4. Characteristics of global algorithms that use DP

Author	Method	Features	Speed (fps)	Image size	Disparity levels	Computational platform
Torra and Criminisi 2004	DP	Occlusion handling Prior feature matching				
Kim et al. 2005	DP	Occlusion handling Prior disparity candidate assignment Two-pass inter-scanline optimization	0.23	384 × 288	16	Intel Pentium 4 2.4 GHz
Wang et al. 2006	DP	Color usage Interscanline consistency Adaptive aggregation Parallel usage of CPU and GPU	43.5	320 × 240	16	3.0 GHz CPU -ATI Radeon XL1800 GPU
Veksler 2005	DP	Applied to pixel-tree structure	~2			
Lei et al. 2006	DP	Occlusion handling Color usage Applied to region-tree structure	0.1	384 × 288	16	1.4 GHz Intel Pentium M

On the contrary, the algorithm proposed in (Veksler 2005) applies the DP method not across individual scanlines but to a tree structure. Thus the minimization procedure accounts for all the pixels of the image, compensating the known streaking effect without being an iterative one. Reported running speed is a couple of frames per second for the tested image pairs. So, real-time implementations are feasible. However, the results obtained are comparable to those of the time-consuming global methods. The reported results of bad matched pixels percentages are 1.77, 1.44, 1.21, and 1.45 for the tested Tsukuba, Sawtooth, Venus and Map image sets, respectively.

In (Lei et al. 2006) the pixel-tree approach of the previous work is replaced by a region-tree one. First of all, the image is color-segmented using the mean-shift algorithm. During the stereo matching, a corresponding energy function defined on such a region-tree structure is optimized using the DP technique. Occlusions are handled by compensating for border occlusions and by applying cross checking. The obtained results indicate that the percentage of the bad matched pixels in unoccluded regions is 1.39, 0.22, 7.42, and 6.31 for the Tsukuba, Venus, Teddy and Cones image sets. The running speed, on a 1.4 GHz Intel Pentium M processor, ranges from 0.1 fps for the Tsukuba dataset with 16 disparity levels to 0.04 fps for the Cones dataset with 60 disparity levels.

The main features of the discussed global algorithms that utilize DP are summarized in Table 4.

2.1.3. Other methods. There are of course other methods, producing dense disparity maps, which can be placed in neither of previous categories. The below discussed methods use either wavelet-based techniques or combinations of various techniques.

Such a method, based on the continuous wavelet transform (CWT) is found in (Huang and Dubois 2004). It makes use of the redundant information that results from the CWT. Using 1D orthogonal and biorthogonal wavelets as well as 2D orthogonal wavelet the maximum matching rate obtained is 88.22% for the Tsukuba

Table 5. Characteristics of the algorithms that cannot be clearly assigned to any category

Author	Method	Features
Huang and Dubois 2004	Continuous wavelet transform	1D orthogonal and biorthogonal wavelets 2D orthogonal wavelet
Liu et al. 2006	Wavelet-based	Non-uniform rational B-splines curves
De Cubber et al. 2008	Intensity based	Stereo-vision and structure-from-motion fusion

pair. Upsampling the pixels in the horizontal direction by a factor of two, through zero insertion, further decreases the noise and the matching rate is increased to 84.91%

Another work (Liu et al. 2006) presents an algorithm based on non-uniform rational B-splines (NURBS) curves. The curves replace the edges extracted with a wavelet based method. The NURBS are projective invariant and so they reduce false matches due to distortion and image noise. Stereo matching is then obtained by estimating the similarity between projections of curves of an image and curves of another image. A 96.5% matching rate for a self recorded image pair is reported for this method.

Finally, a different way of confronting the stereo matching issue is proposed in (De Cubber et al. 2008). The authors, comprehending that there is no all-satisfying method, investigate the possibility of fusing the results from spatially differentiated (stereo vision) scenery images with those from temporally differentiated (structure from motion) ones. This method takes advantage of both method's merits improving the performance.

The main features of the discussed algorithms that cannot be clearly assigned to any of the aforementioned categories are summarized in Table 5.

2.2. Sparse Disparity Algorithms

Algorithms resulting in sparse, or semi-dense, disparity maps tend to be less attractive as most of the contemporary applications require dense disparity information. Though, they are very useful when fast depth estimation is required and at the same time detail, in the whole picture, is not so important. This type of algorithms tends to focus on the main features of the images leaving occluded and poorly textured areas unmatched. Consequently high processing speeds, accurate results but with limited density are achieved. Very interesting ideas flourish in this direction but since contemporary interest is directed towards dense disparity maps, only a few indicatory algorithms are discussed here.

Veksler (2002) presents an algorithm that detects and matches dense features between the left and right images of a stereo pair, producing a semi-dense disparity map. A dense feature is a connected set of pixels in the left image and a corresponding set of pixels in the right image such that the intensity edges on the boundary of these sets are stronger than their matching error. All these are computed during the stereo matching process. The algorithm computes 1 fps with 14 disparity levels for the Tsukuba pair producing 66% density and 0.06% average error in the non occluded regions.

Another method developed by Veksler (2003) is based on the same basic concepts as the former one. The main difference is that this one uses the graph cuts algorithm for the dense feature extraction. As a consequence this algorithm produces semi-dense results with significant accuracy in areas where features are detected. The results are significantly better considering density and error percentage but require longer running times. For the Tsukuba pair it achieves a density up to 75%, the total error in the non occluded regions is 0.36% and the running speed is 0.17 fps. For the Sawtooth pair the corresponding results are 87%, 0.54% and 0.08 fps. All the results are obtained from a Pentium III PC running at 600 MHz.

On the other hand, Gong and Yang (2005a) in their paper propose a DP algorithm, called reliability-based dynamic programming (RDP) that uses a different measure to evaluate the reliabilities of matches. According to this the reliability of a proposed match is the cost difference between the globally best disparity assignment that includes the match and the one that does not include it. The interscanline consistency problem, common to the DP algorithms, is reduced through a reliability thresholding process. The result is a semi-dense unambiguous disparity map with 76% density, 0.32% error rate and 16 fps for the Tsukuba and 72% density, 0.23% error rate and 7 fps for the Sawtooth image pair. Accordingly, the results for Venus and Map pairs are 73%, 0.18%, 6.4 fps and 86%, 0.7%, 12.8 fps. As it can be seen the reported execution speeds, tested on a 2 GHz Pentium 4 PC, are encouraging for real-time operation if a semi-dense disparity map is acceptable.

A similar to the previous one near-real-time stereo matching technique is presented in (Gong and Yang 2005b) by the same authors, which is also based on the RDP algorithm. This algorithm can generate semi-dense disparity maps. Two orthogonal RDP passes are used to search for reliable disparities along both horizontal and vertical scanlines. Hence, the interscanline consistency is explicitly enforced. It takes advantage of the computational power of programmable graphics hardware, which further improves speed. The algorithm is tested on an Intel Pentium 4 computer running at 3 GHz with a programmable ATI Radeon 9800 XT GPU equipped with 256 MB video memory. It results in 85% dense disparity map with 0.3% error rate at 23.8 fps for the Tsukuba pair, 93% density, 0.24% error rate at 12.3 fps for the Sawtooth pair, 86% density, 0.21% error rate at 9.2 fps for the Venus pair and 88% density, 0.05% error rate at 20.8 fps for the Map image pair. If needed, the method can also be used to generate more dense disparity maps deteriorating the execution speed.

The main features of the discussed algorithms that produce sparse output are summarized in Table 6.

Table 6. Characteristics of the algorithms that produce sparse output

Author	Method	Density	Speed (fps)	Image size	Disparity levels	Computational platform
Veksler 2002	Local	66	1	384 × 288	14	Intel Pentium III 600 MHz
Veksler 2003	Graph cuts	75	0.17	384 × 288	16	Intel Pentium 4 2 GHz
Gong and Yang 2005a	RDP	76	16	384 × 288	16	Intel Pentium 4 3 GHz CPU
Gong and Yang 2005b	RDP	85	23.8	384 × 288	16	ATI Radeon 9800 XT GPU

3. REAL-TIME HARDWARE IMPLEMENTATIONS

There are many applications, as mentioned above, that demand extraction of the disparity map from image pairs in real-time. Moreover, most of these applications demand dense output. PCs due to their serial-processing architecture find it difficult to meet these requirements. This problem can be efficiently confronted by the use of dedicated hardware. In addition, the need for dedicated hardware is more evident in the case of autonomous units, where the existence of a PC is not a convenient solution. Hardware implementations can accelerate the performance of the stereo vision systems. They are able to provide the parallelism that is commonly useful in image processing and vision algorithms. In particular, regular and simple structures such as CA or basic filtering modules can be easily and efficiently implemented in hardware. By processing several parts of the data in parallel and performing specific calculations, their overall performance is considerably better compared to software solutions running on serial general purpose processors.

The hardware implementation of global algorithms is neither an appealing nor an easy option. As stated above, global methods are time and computational demanding because of their iterative nature. This is also the reason that prevents them from being implemented with parallel structures. On the contrary, global algorithms require odd, rather than simple and straightforward, implementations. DP though is inherently the simplest of all the other global approximations.

In contrast, local methods could be greatly benefited by the use of such parallel and straightforward structures. Parallelism and simplicity are key factors, available in dedicated hardware implementations, that can reduce the required running times. There are several works that describe local methods implemented on hardware. What most of them have in common is that they implement a rather simple algorithm and make extensively use of computation concurrency. Performance is refined by custom choices during the hardware architecture development phase. A generalized block diagram of a hardware implementable stereo correspondence algorithm is shown in Figure 8.

Hardware implementation involves using either FPGA or ASIC. DSP based solutions have also been reported in the past (Faugeras et al. 1993), however they are not reported as frequently, due to their inhibited difficulty in parallel processing. A survey of the recent bibliography confirms that FPGA implementations are preferable. That is because the time required for fabrication and test of ASIC implementations is considerably long and its cost is high. Moreover there is almost no flexibility for future improvements and modifications. On the other hand FPGA provide rapid prototyping time, are far less expensive and can be easily adapted to new specifications. In this way FPGA combine the best parts of hardware solutions with those of the software ones.

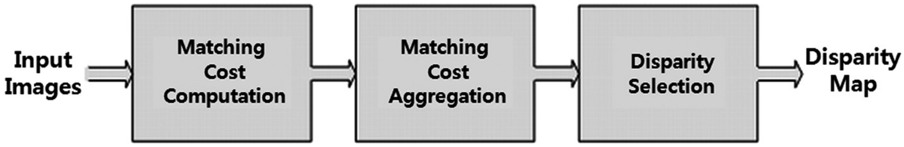

Figure 8. Generalized block diagram of a hardware implementable stereo correspondence algorithm.

3.1. FPGA Implementations

All the hardware implementations examined in this paper can achieve real-time operation. However, the use of FPGAs is now the most convenient and reasonable choice for hardware development. They are cheap and perform remarkably well. The available resources of the devices are constantly growing, allowing for more complex algorithms to be implemented. The variety of available electronic design automation (EDA) tools and the absence of fabrication stage make the prototyping times very short. Another advantage is that the resulting hardware implementation is open for further upgrades. Thus, FPGA implementations are very flexible and fault tolerant.

In the rest of this subsection FPGA implemented methods based on SAD, DP and Local Weighted Phase-Correlation (LWPC) are presented. Table 7 demonstrates the main characteristics of the below discussed works. This table is populated according to the available data. It is evident that the simplest and most straightforward method of all, i.e., SAD, is the most preferable one.

3.1.1. FPGA implementations based on SAD. As expected, when it comes to hardware implementations SAD-based methods are the most preferred ones. SAD calculation requires simple computational modules, as it involves only summations and absolute values' calculations.

The FPGA based architecture presented in (Arias-Estrada and Xicotencatl 2001) is able to produce dense disparity maps in real time. The architecture implements a local algorithm based on the SAD aggregated in fixed windows. Input data are processed in parallel on a single chip. An extension to the basic architecture is

Table 7. FPGA implementations' characteristics

Author	Matching cost	Aggregation	Image size	Disparity levels	Window size	Speed (fps)	Device
Arias-Estrada and Xicotencatl 2001	SAD	fixed window	320 × 240	16	7 × 7	71	Xilinx Virtex XCV800HQ240-6
Jia et al. 2003	SAD	fixed window	640 × 480	64	9 × 9	30	
Miyajima and Maruyama 2003	SAD	fixed window	640 × 480	200	7 × 7	20	Xilinx Virtex-II
Chonghun et al. 2004	SAD	adaptive window	1024 × 1024	32	16 × 16 (max)	47	Xilinx Virtex-II 6000
Yi et al. 2004	SAD	fixed window	270 × 270	27	9 × 9	30	Xilinx Virtex-II XC2V8000
Lee et al. 2005	SAD	fixed window	640 × 480	64	32 × 32	30	Xilinx Virtex-II XC2V8000
Hariyama et al. 2005a	SAD	adaptive window	64 × 64	64	8 × 8 (max)	30	Altera APEX20KE
Georgoulas et al. in press	SAD	adaptive window	640 × 480	80	7 × 7 (max)	275	Altera Stratix II EP2S180F1020C3
Jeong and Park 2004	DP	single line	1280 × 1000	208		15	Xilinx Virtex-II XC2V8000
Park and Jeong 2007	DP	2 lines	320 × 240	128		30	Xilinx Virtex-II pro-100
Darabiha et al. 2006	LWPC		256 × 360	20		30	4x Xilinx Virtex2000E
Masrani and MacLean 2006	LWPC		480 × 640	128		30	4x Altera Stratix S80

also proposed in order to compute disparity maps on more than 2 images. This method can process 320 × 240 pixel images with 16 disparity levels at speeds higher than 71 fps. The devise utilization is 4.2 K slices equivalent to 69 K gates.

The system developed by Jia et al. (2003) is able to compute dense disparity maps in real time using the SAD method over fixed windows. The whole algorithm, including radial distortion correction, Laplacian of Gaussian (LoG) filtering, correspondence finding, and disparity map computation is implemented in a single FPGA. The system can process 640 × 480 pixel images with 64 disparity levels and 8 bit depth precision at 30 fps speed, and 320 × 240 pixel images at 50 fps.

The SAD algorithm aggregated over fixed windows is the option utilized in (Miyajima and Maruyama 2003) as well. This stereo vision system is implemented on a single FPGA with plenty of external memory. It supports camera calibration and left-right consistency check. The performance is 20 fps for 640 × 480 pixel images and 80 fps for 320 × 240. The number of disparity levels for these results are 200 and the device utilization is 54%. Changing the number of disparity levels results only in changing the circuit size and not the performance.

One more simplified version of the adaptive windows aggregation method in conjunction with SAD is used in (Chonghun et al. 2004). It can process images of size up to 1024 × 1024 pixels with 32 disparity levels at 47 fps speed. The resources needed are 3.4 K slices i.e., 10% of the utilized FPGA area.

Another simple implementation of the SAD method with fixed windows is proposed in (Yi et al. 2004). The effect of various window shapes is investigated. The results indicate that 270 × 270 pixel images with 27 disparity levels can be processed at 30 fps speed achieving 90% of correct matches. The utilization of the FPGA reported is in any case less than 46 K slices equivalent to 8 M gates.

The same core algorithm as in (Yi et al. 2004) is used in the work reported in (Lee et al. 2005). The aggregating window shape is found to play a significant role in this implementation. Using rectangular windows instead of square ones reduces the resource usage to 50% i.e., less than 10 K slices and at the same time, preserves the same output quality. The proposed system can process 640 × 480 pixel images with 64 disparity levels at 30 fps rate and 320 × 240 pixel images with 64 disparity levels at 115 fps.

On the other hand, a slightly more complex implementation than the previous ones is proposed in (Hariyama et al. 2005a). It is based on the SAD using adaptive sized windows. The proposed method iteratively refines the matching results by hierarchically reducing the window size. The results obtained by the proposed method are 10% better than that of the fixed-window method. The architecture is fully parallel and as a result all the pixels and all the windows are processed simultaneously. The speed for 64 × 64 pixel images with 8 bit grayscale precision and 64 disparity levels is 30 fps. The resources consumption is 42.5 K logic elements, i.e., 82% of the utilized device.

Finally, SAD aggregated using adaptive windows is the core of the work presented in (Georgoulas et al. in press). A hardware based CA parallel-pipelined design is realized on a single FPGA device. The achieved speed is nearly 275 fps, for 640 × 480 pixel image pairs with a disparity range of 80 pixels. The presented hardware-based algorithm provides very good processing speed at the expense of accuracy. The device's utilization is 149 K gates, that is 83% of the available resources.

3.1.2. FPGA implementations based on DP. The use of DP is an alternative as well. The implementation presented in (Jeong and Park 2004) uses the DP search method on a trellis solution space. It copes with the vergence cameras case, i.e., cameras with optical axes that intersect. The images received from a pair of cameras are rectified using linear interpolation and then the disparity is calculated. The architecture has the form of a linear systolic array using simple processing elements. The design is canonical and simple to be implemented in parallel. The implementation requires 208 processing elements. The resulting system can process 1280 × 1000 pixel images with up to 208 disparity levels at 15 fps.

An extension of the previous method is presented in (Park and Jeong 2007). The main difference is that data from the previous line are incorporated so as to enforce better inter-scanline inconsistency. The running speed is 30 fps for 320 × 240 pixel images with 128 disparity levels. The number of utilized processing elements is 128. The percentage of pixels with disparity error larger than 1 in the unoccluded areas is 2.63, 0.91, 3.44, and 1.88 for the Tsukuba, Map, Venus and Sawtooth image sets, respectively.

3.1.3. FPGA implementations based on phase-based methods. Moreover phase-based techniques can be implemented on hardware as well. The algorithm implemented in (Darabiha et al. 2006) is called Local Weighted Phase-Correlation (LWPC). Hardware implementation of the algorithm turns out to be more than 300 times faster than the software one. The platform used is the Transmogrifier-3A (TM-3A) containing four (4) Xilinx Virtex2000E FPGAs connected via a 98 bit bus. A description of the programmable hardware platform, the base stereo vision algorithm and the design of the hardware can be found in the paper. 66.6 K look-up tables (LUT) and 83 K flip-flops (FF) are required. This implementation can produce dense disparity maps of 256 × 360 pixel image pairs with 20 disparity levels and 8 bit sub-pixel accuracy at the rate of 30 fps.

The same LWPC method is used in (Masrani and MacLean 2006). The platform used is the Transmogrifer-4 containing four (4) Altera Stratix S80 FPGAs. The system performs rectification and left-right consistency check to improve the accuracy of the results. The speed for 640 × 480 pixel images with 128 disparity levels is 30 fps. The hardware resources demanded are roughly the same as in (Darabiha et al. 2006) due to the reuse of the available temporal information of the input video sequence.

3.2. ASIC Implementations

On the other hand ASIC implementation is an option as well, but it is more expensive, except of the case of massive production. The prototyping times are considerable longer and the result is highly process-dependent. Any further changes are difficult and additionally time and money consuming. Their performance supremacy does in most cases not justify choosing ASICs. These are the main reasons that make recent ASIC implementation publications rare in contrast to the FPGA-based ones.

Published works concerning ASIC implementations (Hariyama et al. 2000; Hariyama et al. 2005b) of stereo matching algorithms are restricted to the use of SAD. The reported architectures make extensive use of parallelism and seem promising. Though, they lack undisputed experimental results.

4. CONCLUSIONS

The stereo correspondence problem remains an active area for research. More and more modern applications demand not only accuracy but real-time operation as well. Getting to know the state-of-the-art or even keeping up with it is a time consuming task. This work examines the latest breakthroughs in the area of stereo vision. Software as well as hardware implementations have been considered and presented. In every case a categorization was attempted according to their major attributes. In order to help the reader among the numerous works presented, graphic diagrams as well as summarizing tables are utilized.

It seems that both area and energy based methods walk towards this objective with satisfactory results. There are many new remarkable methods. Attributes such as color-information usage and occlusion handling are common place for many of those methods. The majority, though, of these algorithms are software implemented and not suitable for real-time operation.

Hardware VLSI implementation of these algorithms could be a radical solution towards higher running speeds. That is because hardware implementations can easily embody parallel processing structures, in contrast to the serial-processing PCs. However, it is not only the sequential data processing available at general purpose PCs that is to blame. PCs are generally heavy, occupy considerable volume and consume much energy. An algorithm to be hardware implementable has to be a simple enough and not an iterative one. But when it comes for hardware implementation there are two options, ASICs and FPGAs. The advantages and disadvantages of each one have been explained. The use of FPGAs facilitates the development and is preferred by most of the active researchers. Many algorithms especially developed for and implemented in FPGAs are presented.

ACKNOWLEDGEMENTS

The work is supported by the E.U. funded project View-Finder, FP6-IST-2005-045541. Finally, we would like to thank the anonymous reviewers for their constructive criticism that allowed us to improve our paper.

REFERENCES

Arias-Estrada, Miguel and Juan M. Xicotencatl. 2001. Multiple stereo matching using an extended architecture. *Proceedings of 11th International Conference on Field-Programmable Logic and Applications* 2147:203–212.

Barnard, Stephen T. and William B. Thompson. 1980. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2(4):333–340.

Binaghi, Elisabetta, Ignazio Gallo, Giuseppe Marino, and Mario Raspanti. 2004. Neural adaptive stereo matching. *Pattern Recognition Letters* 25(15):1743–1758.

Bleyer, Michael and Margrit Gelautz. 2004. A layered stereo algorithm using image segmentation and global visibility constraints. *Proceedings of International Conference on Image Processing* 5:2997–3000.

Bleyer, Michael and Margrit Gelautz. 2005. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing* 59(3):128–150.

Brockers, Roland, Marcus Hund, and Barbel Mertsching. 2005. Stereo vision using cost-relaxation with 3D support regions. *Proceedings of Image and Vision Computing New Zealand* :96–101.

Chonghun, Roh, Ha Taehyun, Kim Sungsik, and Kim Jaeseok. 2004. Symmetrical dense disparity estimation: algorithms and FPGAs implementation. *Proceedings of IEEE International Symposium on Consumer Electronics* :452–456.

Darabiha, Ahmad, James W. Maclean, and Jonathan Rose. 2006. Reconfigurable hardware implementation of a phase-correlation stereo algorithm. *Machine Vision and Applications* 17(2):116–132.

De Cubber, Geert, Lazaros Nalpantidis, Georgios Ch. Sirakoulis, and Antonios Gasteratos. 2008. Intelligent robots need intelligent vision: visual 3D perception. *Proceedings of International workshop on Robotics for risky interventions and Environmental Surveillance.*

Di Stefano, Luigi, Massimiliano Marchionni, and Stefano Mattoccia. 2004. A fast area-based stereo matching algorithm. *Image and Vision Computing* 22(12):983–1005.

Faugeras, O. 1993. *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, MA, USA: MIT Press.

Faugeras, Olivier, Bernard Hotz, Herve Mathieu, Thierry Vieville, Zhengyou Zhang, Pascal Fua, Eric Theron, Laurent Moll, Gerard Berry, Jean Vuillemin, Patrice Bertin, and Catherine Proy. 1993. Real time correlation based stereo: algorithm implementations and applications. INRIA Technical Report RR-2013.

Forsyth, D. A. and J. Ponce. 2002. *Computer Vision: A modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall.

Georgoulas, Christos, Leonidas Kotoulas, Georgios Ch. Sirakoulis, Ioannis Andreadis, and Antonios Gasteratos. Real-Time Disparity Map Computation Module. *Journal of Microprocessors and Microsystems* 32(3):159–170.

Gong, Minglun, Ruigang Yang, Liang Wang, and Mingwei Gong. 2007. A performance study on different cost aggregation approaches used in real-time stereo matching. *International Journal of Computer Vision* 75(2):283–296.

Gong, Minglun and Yee-Hong Yang. 2005a. Fast unambiguous stereo matching using reliability-based dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(6):998–1003.

Gong, Minglun and Yee-Hong Yang. 2005b. Near real-time reliable stereo matching using programmable graphics hardware. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1:924–931.

Gutierrez, Salvador and Jose Luis Marroquin. 2004. Robust approach for disparity estimation in stereo vision. *Image and Vision Computing* 22(3):183–195.

Hariyama, Masanori, Yasuhiro Kobayashi, Haruka Sasaki, and Michitaka Kameyama. 2005a. FPGA implementation of a stereo matching processor based on window-parallel-and-pixel-parallel architecture. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science* 88(12):3516–3522.

Hariyama, Masanori, Haruka Sasaki, and Michitaka Kameyama. 2005b. Architecture of a stereo matching VLSI processor based on hierarchically parallel memory access. *IEICE – Transactions on Information and Systems* E88-D(7):1486–1491.

Hariyama, Masanori, Toshiki Takeuchi, and Michitaka Kameyama. 2000. Reliable stereo matching for highly-safe intelligent vehicles and its VLSI implementation. *Proceedings of IEEE Intelligent Vehicles Symposium* :128–133.

Hartley, R. and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision*. Second ed. Cambridge, UK: Cambridge University Press.

Hirschmuller, Heiko. 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:807–814.

Hirschmuller, Heiko. 2006. Stereo vision in structured environments by consistent semi-global matching. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:2386–2393.

Hirschmuller, Heiko and Daniel Scharstein. 2007. Evaluation of cost functions for stereo matching. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* :1–8.

Hong, Li and George Chen. 2004. Segment-based stereo matching using graph cuts. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1:74–81.

Huang, Xiaodong and Eric Dubois. 2004. Dense disparity estimation based on the continuous wavelet transform. *Proceedings of Canadian Conference on Electrical and Computer Engineering* 1:465–468.

Jain, R., R. Kasturi, and B. G. Schunck. 1995. *Machine vision*. New York, USA: McGraw-Hill.

Jeong, H. and S. Park. 2004. Generalized Trellis Stereo Matching with Systolic Array. In *Parallel and Distributed Processing and Applications*, Berlin-Heidelberg: Springer Verlag.

Jia, Yunde, Yihua Xu, Wanchun Liu, Cong Yang, Yuwen Zhu, Xiaoxun Zhang, and Luping An. 2003. A Miniature stereo vision machine for real-time dense depth mapping. *Proceedings of Third International Conference on Computer Vision Systems* 2626:268–277.

Kim, Hansung and Kwanghoon Sohn. 2005. 3D reconstruction from stereo images for interactions between real and virtual objects. *Signal Processing: Image Communication* 20(1):61–75.

Kim, Jae Chul, Kyoung Mu Lee, Byoung Tae Choi, and Sang Uk Lee. 2005. A dense stereo matching using two-pass dynamic programming with generalized ground control points. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:1075–1082.

Klaus, Andreas, Mario Sormann, and Konrad Karner. 2006. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *Proceedings of 18th International Conference on Pattern Recognition* 3:15–18.

Kotoulas, Leonidas, Antonios Gasteratos, Georgios Ch. Sirakoulis, Christos Georgoulas, and Ioannis Andreadis. 2005. Enhancement of fast acquired disparity maps using a 1-D cellular automation filter. *Proceedings of IASTED International Conference on Visualization, Imaging and Image Processing* :355–359.

Kotoulas, Leonidas, Christos Georgoulas, Antonios Gasteratos, Georgios Ch. Sirakoulis, and Ioannis Andreadis. 2005. A novel three stage technique for accurate disparity maps. *Proceedings of EOS Conference on Industrial Imaging and Machine Vision at the World of Photonics Congress* :13–14.

Lee, SungHwan, Jongsu Yi, and JunSeong Kim. 2005. Real-time stereo vision on a reconfigurable system. *Proceedings of 5th International Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation* 3553:299–307.

Lei, Cheng, Jason Selzer, and Yee-Hong Yang. 2006. Region-tree based stereo using dynamic programming optimization. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:2378–2385.

Liu, Chong, Wei Pei, Salvator Niyokindi, Jincai Song, and Liding Wang. 2006. Micro stereo matching based on wavelet transform and projective invariance. *Measurement Science and Technology* 17(3):565–571.

Maimone, Mark W. and Steven A. Shafer. 1996. A taxonomy for stereo computer vision experiments. *Proceedings of ECCV Workshop on Performance Characteristics of Vision Algorithms* :59–79.

Manzotti, Ricardo, Antonios Gasteratos, Giorgio Metta, and Giulio Sandini. 2001. Disparity estimation on log-polar images and vergence control. *Computer Vision and Image Understanding* 83(2):97–117.

Masrani, Divyang K. and W. James MacLean. 2006. A real-time large disparity range stereo-system using FPGAs. *Proceedings of 7th Asian Conference on Computer Vision* 3852:42–51.

Mayoral, Rafael, Gabriel Lera, and Maria Jose Perez-Ilzarbe. 2006. Evaluation of correspondence errors for stereo. *Image and Vision Computing* 24(12):1288–1300.

Metta, Giorgio, Antonios Gasteratos, and Giulio Sandini. 2004. Learning to track colored objects with log-polar vision. *Mechatronics* 14(9):989–1006.

Miyajima, Yosuke and Tsutomu Maruyama. 2003. A real-time stereo vision system with FPGA. *Proceedings of International conference on field-programmable logic and applications* 2778:448–457.

Mordohai, Philippos and Gerard G. Medioni. 2006. Stereo using monocular cues within the tensor voting framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(6):968–982.

Muhlmann, Karsten, Dennis Maier, Jurgen Hesser, and Reinhard Manner. 2002. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision* 47(1–3):79–88.

Murray, Don and Cullen Jennings. 1997. Stereo vision based mapping and navigation for mobile robots. *Proceedings of IEEE International Conference on Robotics and Automation* 2:1694–1699.

Murray, Don and James J. Little. 2000. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots* 8(2):161–171.

Nalpantidis, Lazaros, Georgios Ch. Sirakoulis, and Antonios Gasteratos. 2007. Review of stereo matching algorithms for 3D vision. *Proceedings of 16th International Symposium on Measurement and Control in Robotics* :116–124.

Ogale, Abhijit S. and Yiannis Aloimonos. 2005a. Shape and the stereo correspondence problem. *International Journal of Computer Vision* 65(3):147–162.

Ogale, Abhijit S. and Yiannis Aloimonos. 2005b. Robust Contrast Invariant Stereo Correspondence. *Proceedings of IEEE International Conference on Robotics and Automation*: 819–824.

Park, Sungchan and Hong Jeong. 2007. Real-time stereo vision FPGA chip with low error rate. *Proceedings of International Conference on Multimedia and Ubiquitous Engineering*: 751–756.

Scharstein, Daniel and Richard Szeliski. *http://vision.middlebury.edu/stereo/* (accessed February 20, 2008).

Scharstein, Daniel and Richard Szeliski. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision* 47(1–3):7–42.

Scharstein, Daniel and Richard Szeliski. 2003. High-accuracy stereo depth maps using structured light. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1:195–202.

Strecha, Christoph, Rik Fransens, and Luc J. Van Gool. 2006. Combined depth and outlier estimation in multi-view stereo. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:2394–2401.

Sun, Jian, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. 2005. Symmetric stereo matching for occlusion handling. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:399–406.

Sunyoto, Harris, Wannes van der Mark, and Dariu M. Gavrila. 2004. A comparative study of fast dense stereo vision algorithms. *Proceedings of IEEE Intelligent Vehicles Symposium*: 319–324.

Torra, Philip H. S. and Antonio Criminisi. 2004. Dense stereo using pivoted dynamic programming. *Image and Vision Computing* 22(10):795–806.

Veksler, Olga. 2002. Dense features for semi-dense stereo correspondence. *International Journal of Computer Vision* 47(1–3):247–260.

Veksler, Olga. 2003. Extracting dense features for visual correspondence with graph cuts. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1:689–694.

Veksler, Olga. 2005. Stereo correspondence by dynamic programming on a tree. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:384–390.

Veksler, Olga. 2006. Reducing search space for stereo correspondence with graph cuts. *Proceedings of British Machine Vision Conference* 2:709–718.

Wang, Liang, Miao Liao, Minglun Gong, Ruigang Yang, and David Nister. 2006. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. *Proceedings of Third International Symposium on 3D Data Processing, Visualization, and Transmission* :798–805.

Yang, Qyngxiong, Liang Wang, and Ruigang Yang. 2006a. Real-time Global Stereo Matching Using Hierarchical Belief Propagation. *Proceedings of British Machine Vision Conference* 3:989–998.

Yang, Qyngxiong, Liang Wang, Ruigang Yang, Henrik Stewenius, and David Nister. 2006b. Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation and Occlusion Handling. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:2347–2354.

Yi, Jongsu, Junseong Kim, Liping Li, John Morris, Gareth Lee, and Philippe Leclercq. 2004. Real-time three dimensional vision. *Proceedings of Advances in Computer Systems Architecture* 3189:309–320.

Yoon, Kuk-Jin and In So Kweon. 2006a. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(4):650–656.

Yoon, Kuk-Jin and In So Kweon. 2006b. Correspondence search in the presence of specular highlights using specular-free two-band images. *Proceedings of 7th Asian Conference on Computer Vision* 3852:761–770.

Yoon, Kuk-Jin and In So Kweon. 2006c. Stereo matching with symmetric cost functions. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2:2371–2377.

Yoon, Sukjune, Sung-Kee Park, Sungchul Kang, and Yoon Keun Kwak. 2005. Fast correlation-based stereo matching with the reduction of systematic errors. *Pattern Recognition Letters* 26(14):2221–2231.

Zach, Christopher, Konrad Karner, and Horst Bischof. 2004. Hierarchical disparity estimation with programmable 3D Hardware. *Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* :275–282.

Zitnick, C. Lawrence and Sing Kang. 2007. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision* 75(1):49–65.

Zitnick, C. Lawrence, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. 2004. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics* 23(3):600–608.