Real-time Stereo Vision Applications

Christos Georgoulas, Georgios Ch. Sirakoulis and Ioannis Andreadis

Laboratory of Electronics, Democritus University of Thrace
Xanthi, Greece

1. Introduction

Depth perception is one of the important tasks of a computer vision system. Stereo correspondence by calculating the distance of various points in a scene relative to the position of a camera allows the performance of complex tasks, such as depth measurements and environment reconstruction (Jain et al., 1995). The most common approach for extracting depth information from intensity images is by means of a stereo camera setup. 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). The computational demanding task of matching can be reduced to a one dimensional search, only by accurately rectified stereo pairs in which horizontal scan lines reside on the same epipolar plane, as shown in Figure 1. By definition, the epipolar plane is defined by the point \( P \) and the two camera optical centers \( O_L \) and \( O_R \). This plane \( PO_LO_R \) intersects the two image planes at lines \( EP_1 \) and \( EP_2 \), which are called epipolar lines. Line \( EP_1 \) is passing through two points: \( E_L \) and \( P_L \), and line \( EP_2 \) is passing through \( E_R \) and \( P_R \) respectively. \( E_L \) and \( E_R \) are called epipolar points and are the intersection points of the baseline \( O_LO_R \) with each of the image planes. The computational significance for matching different views is that for a point in the first image, its corresponding point in the second image must lie on the epipolar line, and thus the search space for a correspondence is reduced from 2 dimensions to 1 dimension. This is called the epipolar constraint. The difference on the horizontal coordinates of points \( P_L \) and \( P_R \) 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 & Jennings, 1997; Murray & Little, 2000).

![Fig. 1. Geometry of epipolar plane](attachment:image.png)
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 & Thompson, 1980). To determine a conjugate pair, it is necessary to measure the similarity of the points. The point to be matched should be distinctly different from its surrounding pixels. In order to minimise the number of false correspondences in the image pair, several constraints have been imposed. The uniqueness constraint (Marr & Poggio, 1979) requires that a given pixel from one image cannot correspond to more than one pixel on the other image. In the presence of occluded regions within the scene, it may be impossible at all to find a corresponding point. The ordering constraint (Baker & Binford, 1981) requires that if a pixel is located to the left of another pixel in image, i.e. left image, the corresponding pixels in right image must be ordered in the same manner, and vice versa, i.e. ordering of pixels is preserved across the images. The ordering constraint may be violated if an object in the scene is located much closer to the camera than the background, and one pixel corresponds to a point on the object while the other pixel corresponds to a point in the background. Finally, the continuity constraint (Marr & Poggio, 1979), which is valid only for scenarios in which smooth surfaces are reconstructed, requires that the disparity map should vary smoothly almost everywhere in the image. This constraint may be violated at depth discontinuities in the scene.

Three broad classes of techniques have been used for stereo matching: area-based (Di Stefano et al., 2004; Scharstein & Szelinski, 2002), feature-based (Venkateswar & Chellappa, 1995; Dhond & Aggarwal, 1989), and phase-based (Fleet et al., 1991; Fleet, 1994). Area-based algorithms use local pixel intensities as a distance measure and they produce dense disparity maps, i.e. process the whole area of the images. An important drawback of area-based techniques is the fact that uniform depth across a correlation window is assumed, which leads to false correspondences near object edges, especially when dealing with large windows. A compact framework was introduced by (Hirschmuller, 2001), where instead of using a large window several smaller neighbouring windows are used. Only the ones that contribute to the overall similarity measure in a consistent manner are taken into account. A left-right consistency check is imposed to invalidate uncertain correspondences. Accurate depth values at object borders are determined by splitting the corresponding correlation windows into two parts, and separately searching on both sides of the object border for the optimum similarity measure. These improvements of the classical area-based approach are demonstrated by (Hirschmuller, 2001) and in more detail by (Hirschmuller et al., 2002) to significantly improve the overall performance of three-dimensional reconstruction.

On the other hand, feature-based algorithms rely on certain points of interest. These points are selected according to appropriate feature detectors. They are more stable towards changes in contrast and ambient lighting, since they represent geometric properties of a scene. Feature-based stereo techniques allow for simple comparisons between attributes of the features being matched, and are hence faster than area-based matching methods. The major limitation of all feature-based techniques is that they cannot generate dense disparity maps, and hence they often need to be used in conjunction with other techniques. Because of the sparse and irregularly distributed nature of the features, the matching results should be augmented by an interpolation step if a dense disparity map of the scene is desired. Additionally, an extra stage for extensive feature detection in the two images is needed, which will increase the computational cost. Thus feature-based methods are not suitable for real-time applications.
In phase-based techniques the disparity is defined as the shift necessary to align the phase value of band-pass filtered versions of two images. In (Fleet et al., 1991) it is shown that phase-based methods are robust when there are smooth lighting variations between stereo images. It also shows that phase is predominantly linear, and hence reliable approximations to disparity can be extracted from phase displacement.

Real-time stereo vision techniques capable of addressing the stereo vision matching problem, producing disparity maps in real-time speeds, are presented in this chapter. While these techniques are based on many different approaches to detect similarities between image regions, all of them present real-time characteristics along with increased accuracy on the computed disparity map.

2. Real-Time Stereo Vision Implementations

Numerous applications require real-time extraction of 3D information. Many researchers have been focused in finding the optimum selection of tools and algorithms to obtain efficient results. The main characteristics of a real-time stereo vision implementation are the produced accuracy of the extracted disparity map, versus the frame rate throughput of such a system. There is always a trade off between disparity map accuracy and speed. Most of the applications require also a dense output. Software-based techniques cannot easily handle such requirements due to the serial behaviour. Real-time dense disparity output requires a significant amount of computational resources. Software-based techniques cannot easily handle such requirements due to the serial operation. Increasing the image size of the stereo pair, or the disparity levels range, can result in a dramatic reduction on the operating throughput. Thus, most of the recent research on real-time stereo vision techniques is oriented towards the use of a dedicated hardware platform. Hardware devices offer the use of parallelism, pipelining and many more design techniques, which result in efficient overall operation in image processing, presenting considerably better results compared to serial software-based solutions.

2.1 SAD-based Implementations

SAD-based implementations are the most favourable area-based techniques in real-time stereo vision, since they can be straightforwardly implemented in hardware. The calculations required in terms of design units are simple, since only summations and absolute values are performed. Parallel design units can be utilized in order to handle various disparity ranges, in order to reduce the computational time required. Area-based methods techniques involve window based operation, where small image windows are directly compared along corresponding epipolar lines according to a pixel-based similarity measure. Common similarity measures are the cross-correlation coefficient, the sum of absolute differences, or the sum of squared differences (Franke & Joos, 2000). Evaluations of various techniques using similarity measures are given by (Scharstein & Szelinski 2002; Hirscmuller & Scharstein, 2007). The mathematical formula of the SAD similarity measures is presented below:

\[ SAD(i, j, d) = \sum_{\mu=-W}^{W} \sum_{\nu=-W}^{W} |I_r(i + \mu, j + \nu) - I_l(i + \mu, j - d + \nu)| \]  

(1)
where $I_l$ and $I_r$ denote the left and right image pixel grayscale values, $d$ is the disparity range, $w$ is the window size and $i, j$ are the coordinates (rows, columns) of the center pixel of the working window for which the similarity measures are computed. Once the SAD is computed for all pixels and for all disparity values, a similarity accumulator has been constructed for each pixel, which indicates the most likely disparity. In order to compute the disparity map a search in the SAD for all disparity values, $(d_{\text{min}} \text{ up to } d_{\text{max}})$, is performed for every pixel. At the disparity range, $(d_{\text{min}} \text{ up to } d_{\text{max}})$, where the SAD is minimum for a pixel, this value is given as the corresponding pixel value for disparity map:

$$D(i, j) = \arg \min_{d \in [d_{\text{min}}, d_{\text{max}}]} \ SAD(i, j, d) \quad (2)$$

The FPGA based architecture along with an off-the-self PCI board by (Niitsuma & Maruyama, 2005), uses an SAD-based technique to efficiently calculate optical flow. Dense vector maps can be generated by the proposed system at 840 frames per second for a 320x240, and at 30 frames per second for a 640x480 pixels stereo image pair correspondingly. A matching window of 7x7 pixels is used by the area-based technique, along with a maximum disparity range of 121 levels.

The stereo matching architecture presented by (Ambrosch et al., 2009) presents a cost-efficient hardware pipelined implementation of a real-time stereo vision using an optimised technique of the SAD computation. Disparity maps are calculated using 450x375 input images and a disparity range of up to 100 pixels at a rate of nearly 600 frames per second. Their results show that the device resource usage increases exponentially when increasing the desired frame rate. On the other hand, increasing the block size leads to a more linear increase of consumed logic elements due to their SAD optimized implementation.

Another implementation that uses a modified version of the SAD computation is the one presented by (Lee et al., 2005). Various versions of SAD algorithms are synthesized by the authors to determine resource requirements and performance. By decomposing a SAD correlator into column and row SAD calculator using buffers, a saving of around 50% is obtained in terms of resource usage of the FPGA device. Additionally, by using different shapes of matching windows, rather than rectangular ones, they reduced storage requirements without the expense of quality. Disparity maps at the rate of 122 frames per second are produced, for an image pair of 320x240 pixels spatial resolution, with 64 levels of disparity.

The FPGA based architecture presented in (Arias-Estrada & 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. Parallel processing of the input data is performed by the proposed design architecture. An extension to the basic architecture is also proposed in order to compute disparity maps on more than 2 images. This method can process 320x240 pixels image pairs with 16 disparity levels at speeds reaching 71 frames per second.

A technique based on adaptive window aggregation method in conjunction with SAD is used in (Roh et al., 2004). It can process images of size up to 1024x1024 pixels with 32 disparity levels at 47 frames per second. The implemented window-based algorithms present low FPGA resource usage, along with noticeable performance in disparity map quality.

In (Niitsuma & Maruyama, 2004), a compact system for real-time detection of moving objects is proposed. Realization of optical flow computation and stereo vision by area-based matching on a single FPGA is addressed. By combining those features, moving objects as well as distances to the objects, can be efficiently detected. Disparity map computation at a
rate of 30 frames per second, for 640x480 pixels images, with 27 disparity levels, is achieved by the proposed system. Finally, a slightly more complex implementation than the previous ones is proposed in (Hariyama et al., 2005). 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 64x64 pixel images with 8 bit grayscale precision and 64 disparity levels is 30 frames per second. The SAD-based hardware implemented stereo vision implementations discussed above are summarized in Table 1 below.

<table>
<thead>
<tr>
<th>Author</th>
<th>Frame rate (fps)</th>
<th>Image Size (pixels)</th>
<th>Disparity Range</th>
<th>Window Size (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Niitsuma &amp; Maruyama, 2005</td>
<td>840</td>
<td>320×240</td>
<td>121</td>
<td>7×7</td>
</tr>
<tr>
<td>Ambrosch et al., 2009</td>
<td>599</td>
<td>450×375</td>
<td>100</td>
<td>9×9</td>
</tr>
<tr>
<td>Lee et al., 2005</td>
<td>122</td>
<td>320×240</td>
<td>64</td>
<td>16×16</td>
</tr>
<tr>
<td>Arias-Estrada &amp; Xicotencatl, 2001</td>
<td>71</td>
<td>320×240</td>
<td>16</td>
<td>7×7</td>
</tr>
<tr>
<td>Roh et al., 2004</td>
<td>47</td>
<td>1024×1024</td>
<td>32</td>
<td>16×16</td>
</tr>
<tr>
<td>Niitsuma &amp; Maruyama, 2004</td>
<td>30</td>
<td>640×480</td>
<td>27</td>
<td>7×7</td>
</tr>
<tr>
<td>Hariyama et al., 2005</td>
<td>30</td>
<td>64×64</td>
<td>64</td>
<td>8×8</td>
</tr>
</tbody>
</table>

Table 1. SAD-based Hardware Implementations

### 2.2 Phase-based Implementations

New techniques for stereo disparity estimation have been exhibited, in which disparity is expressed in terms of phase differences in the output of local, band-pass filters applied to the left and right views (Jenkin & Jepson, 1988; Sanger, 1988; Langley et al., 1990). The main advantage of such approaches is that the disparity estimates are obtained with sub-pixel accuracy, without requiring explicit sub-pixel signal reconstruction or sub-pixel feature detection and localization. The measurements may be used directly, or iteratively as predictions for further, more accurate, estimates. Because there are no restrictions to specific values of phase (i.e. zeros) that must first be detected and localized, the density of measurements is also expected to be high. Additionally the computations may be implemented efficiently in parallel (Fleet et al. 1991). Hardware implementation of such algorithms turned out to be much faster than software-based ones. The PARTS reconfigurable computer (Woodfill & Herzen, 1991), consists of a 4x4 array of mesh-connected FPGAs. A phase algorithm based on the census transform, which mainly consists of bitwise comparisons and additions, is proposed. The algorithm reaches 42 frames per second for 320x240 pixels image pair, with 24 levels of disparity. The method in (Masrani & MacClean, 2006), uses the so-called Local Weighted Phase-Correlation (LWPC), which combines the robustness of wavelet-based phase-difference methods with the basic control strategy of phase-correlation methods. Four FPGAs are used to perform image rectification and left-right consistency check to improve the quality of the produced disparity map. Real-time speeds reaching 30 frames per second for an image pair with 640x480 pixels with 128 levels of disparity. LWPC is also used in (Darabiha et al., 2006).
Again four FPGAs are used for the hardware implementation of the algorithm, reaching 30 frames per second for a 256x360 pixels image pair with 20 disparity levels. The phase-based hardware implementations are presented in Table 2 below.

<table>
<thead>
<tr>
<th>Author</th>
<th>Frame rate (fps)</th>
<th>Image Size (pixels)</th>
<th>Disparity Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodfill &amp; Herzen, 1991</td>
<td>42</td>
<td>320x240</td>
<td>24</td>
</tr>
<tr>
<td>Masrani &amp; MacClean, 2006</td>
<td>30</td>
<td>640x480</td>
<td>128</td>
</tr>
<tr>
<td>Darabiha et al., 2006</td>
<td>30</td>
<td>256x360</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2. Phase-based Hardware Implementations

2.3 Disparity Map Refinement

The resulting disparity images are usually heavily corrupted. This type of random noise is introduced during the disparity value assignment stage. The disparity value assigned to some pixels does not correspond to the appropriate value. Hence, in a given window, some pixels might have been assigned with the correct disparity value and some others not. This can be considered as a type of random noise in the given window. Various standard filtering techniques, such as mean, median, Gaussian can not provide efficient refinement (Murino et al., 2001). Typical low-pass filters result in loss of detail and do not present adequate false matchings removal. Adaptive filtering is also unsuccessful, presenting similar results.

2.3.1 CA Filtering

Filtering using a cellular automata (CA) approach presents better noise removal with detail preservation and extremely easy, simple and parallel hardware implementation (Popovici & Popovici, 2002; Rosin, 2005).

Regarding CA, these are dynamical systems, where space and time are discrete and interactions are local and they can easily handle complicated boundary and initial conditions (Von Neumann, 1966; Wolfram, 1983). Following, a more formal definition of a CA will be presented (Chopard & Droz, 1998). In general, a CA requires:

1. a regular lattice of cells covering a portion of a d-dimensional space;
2. a set \( C(\hat{r}, t) = \{ C_1(\hat{r}, t), C_2(\hat{r}, t), \ldots, C_m(\hat{r}, t) \} \) of variables attached to each site \( \hat{r} \) of the lattice giving the local state of each cell at the time \( t = 0, 1, 2, \ldots \);
3. a Rule \( R = \{ R_1, R_2, \ldots, R_m \} \) which specifies the time evolution of the states \( C(\hat{r}, t) \) in the following way:

\[
C_j(\hat{r}, t + 1) = R_j\left( C(\hat{r}, t), C(\hat{r} + \hat{\delta}_1, t), C(\hat{r} + \hat{\delta}_2, t), \ldots, C(\hat{r} + \hat{\delta}_q, t) \right)
\]

where \( \hat{r} + \hat{\delta}_k \) designate the cells belonging to a given neighborhood of cell \( \hat{r} \).

In the above definition, the Rule \( R \) is identical for all sites, and it is applied simultaneously to each of them, leading to a synchronous dynamics.

CA have been applied successfully to several image processing applications (Alvarez et al., 2005; Rosin, 2006; Lafe, 2000). CA are one of the computational structures best suited for a VLSI realization (Pries et al., 1986; Sirakoulis, 2004; Sirakoulis et al., 2003). Furthermore, the
CA approach is consistent with the modern notion of unified space-time. In computer science, space corresponds to memory and time to processing unit. In CA, memory (CA cell state) and processing unit (CA local Rule) are inseparably related to a CA cell (Toffoli & Margolus, 1987).

According to the disparity value range, every disparity map image is decomposed into a set of \( d \) images, where \( d \) is the range of the disparity values, a technique similar to the so-called ‘threshold decomposition’. Hence for a given image pair with i.e. 16 levels of disparity, 16 binary images are created, where \( C_1 \) image has logic ones on every pixel that has value 1 in the disparity map, and logic zeros elsewhere. \( C_2 \) image has ones on every pixel that has value 2 in the disparity map, and zeros elsewhere, and so on. The CA rules are applied separately on each \( C_d \) binary image and the resulting disparity map is further recomposed by the following formula:

\[
D(i, j) = \sum C_d(i, j) \cdot d, \quad d \in [d_{\min}, d_{\max}]
\]  

(4)

The CA rules can be selected in such way that they produce the maximum possible performance within the given operating windows. The main effect of this filtering is the rejection of a great portion of incorrect matches.

2.3.2 Occlusion and false matching detection

Occluded areas can also introduce false matches in the disparity map computation. There are three main classes of algorithms for handling occlusions: 1) methods that detect occlusions (Chang et al, 1991; Fua, 1993), 2) methods that reduce sensitivity to occlusions (Bhat & Nayar, 1998; Sara & Bajcsy, 1997), and 3) methods that model the occlusion geometry (Bellumeur, 1996; Birchfield & Tomasi, 1998). Considering the first class, left-right consistency checking may also be used to detect occlusion boundaries. Computing two disparity maps, one based on the correspondence from the left image to the right image, and the other based on the correspondence from the right image to the left image, inconsistent disparities are assumed to represent occluded regions in the scene. Left-right consistency checking is also known as the “two-views constraint”. This technique is well suited to remove false correspondences caused by occluded areas within a scene (Fua, 1993). Due to its simplicity and overall good performance, this technique was implemented in many real-time stereo vision systems (Faugeras et al., 1993; Konolige, 1997; Matthies et al., 1995).

Using the left-right consistency checking, valid disparity values are considered, only those that are consistent in both disparity maps, i.e. those that do not lie within occluded areas. A pixel that lies within an occluded area will have different disparity value in the left disparity map, from its consistent pixel in the right disparity map. For example, a non-occluded pixel in the left disparity image must have a unique pixel with equally assigned disparity value in the right disparity map according to the following equations:

\[
D_{\text{left-right}}(i, j) = D_{\text{right-left}}(i, j-d), \quad (d= D_{\text{left-right}}(i, j))
\]  

(5)

\[
D_{\text{right-left}}(i, j) = D_{\text{left-right}}(i, j+d), \quad (d= D_{\text{right-left}}(i, j))
\]  

(6)

The same applies, for false matched points not exclusively due to occlusions, but due to textureless areas or sensor parameter variations. These points are assigned with a false
disparity value during the disparity map assignment stage described by equation (2), since there might be more than one minimum SAD value for a given pixel, which leads to false disparity value assignment for that pixel. Thus, the disparity value assigned to some pixels does not correspond to the appropriate correct value. Performing this consistency check, the occluded pixel along with the false matched points within the scene can be derived.

3. Hardware Realization

Most of the real-time stereo vision techniques implementation relies on the use of an FPGA device. FPGAs provide with high processing rates, which is ideal for speed demanding applications. On the other hand, they offer high density designs with low cost demands, shorter time-to-market benefits, which enable them in many hardware-based system realizations. Compared to an ASIC device, their main advantage is the much lower level of NRE (Non-Recurring Engineering) costs, typically associated with ASIC design. Additionally, FPGAs provide extensive reconfigurability, since they can be rewired in the field to fix bugs, and much simpler design methodology compared to ASIC devices. Compared to a processor, their main advantage is the higher processing rate. This is due to the fact that FPGAs can customize the resources’ allocation to meet the needs of a specific application, whereas processors have fixed functional units.

3.1 SAD-based Disparity Computation with CA post-filtering

The work by (Georgoulas et al., 2008), presents a hardware-efficient real-time disparity map computation system. A modified version of the SAD-based technique is imposed, using an adaptive window size for the disparity map computation. A CA filter is introduced to refine false correspondences, while preserving the quality and detail of the disparity map. The presented hardware provides very good processing speed at the expense of accuracy, with very good scalability in terms of disparity levels.

CA are discrete dynamical systems that can deal efficiently with image enhancement operations such as noise filtering (Haykin, 2001). More specifically, the reasons why CA filter can be ideally implemented by VLSI techniques are: (1) the CA generating rules have the property of native parallel processing; (2) the proposed 2-D CA cell structure with programmable additive rules is easily implemented by using AND/OR gates.

In area-based algorithms the search is performed over a window centered on a pixel. However, a major issue is that small windows produce very noisy results, especially for low textured areas, whereas large windows fail to preserve image edges and fine detail. Thus, it is beneficial to estimate a measure of local variation, in terms of pixel grayscale value, over the image using variable sized windows, in order to obtain more efficient disparity map evaluation. The measure of a pixel local variation in a support window is a simple statistic of the intensity differences between neighboring pixels in the window.

This first step consists of calculating the local variation of image windows for the reference (left) image. Local variation (LV) is calculated according to the following formula:

\[
LV(p) = \sum_{i=1}^{N} \sum_{j=1}^{N} |f(i, j) - \mu| , \text{where } \mu = \text{average grayscale value of image window (7)}
\]
where the local variation for a given window central pixel \( p \) is calculated according to the neighboring pixel grayscale values. \( N \) is the selected square window size, in this case, 2 or 5. In the case of a 2x2 window the local variation is calculated for the upper left pixel. Initially the local variation over a window of 2x2 pixels is calculated and points with smaller local variation than a certain threshold value are marked for further processing. The local variation over a 5x5 range is computed for the marked points and is then compared to a second threshold. Windows presenting smaller variation than the second threshold are marked for larger area processing. To obtain optimum results various thresholds configurations can be manually selected.

The overall architecture is realised on a single FPGA device of the Stratix II family of Altera devices, with a maximum operating frequency of 256 MHz. Real time disparity maps are extracted at a rate of 275 frames per second for a 640x480 pixels resolution image pair with 80 levels of disparity. The hardware architecture is depicted in Figure 2. The module operates in a parallel-pipelined manner. The serpentine memory block is used to temporarily store the pixel grayscale values during the processing of the image. The serpentine memory block is used to increase processing speed. As the working windows move over the image, overlapping pixels exist between adjacent windows. The serpentine memory architecture is used to temporarily store overlapping pixels in order to reduce the clock cycles needed to load image pixels into the module (Gasteratos et al., 2006). CA filtering design is most efficient when implemented in hardware, due to the highly parallel independent processing. CA can be designed in a parallel structure, which results in real-time processing speeds.

For a disparity range of 80 and a maximum working window of 7x7, on the first scanline of the image, after an initial latency period of 602 clock cycles, where the set of registers for the right image requires to store 80 overlapping 7x7 working windows, \( 49 + 7 \times 79 = 602 \), output is given every 7 clock cycles. Every time the working window moves to the next scanline, after an initial latency of 7 clock cycles which are the only new pixels due to window overlapping with the previous scanline, output is given once every clock cycle. By using an FPGA device operating at 256MHz for the CA-based approach, a 1Mpixel disparity map can be extracted in 11.77 msec, i.e. 85 frames per second. The relationship between the number of frames processed per second and the processed image width, assuming square images and a disparity range of 80 is presented in Figure 3.
3.2 Occlusion-aware Disparity Computation
In (Georgoulas & Andreadis, 2009) a SAD window based technique using full color RGB images as well as an occlusion detection approach to remove false matchings are employed. The architecture is based on fully parallel-pipelined blocks in order to achieve maximum processing speed. Depending on the required operating disparity range the module can be parameterized, to adapt to the given configuration, in order to obtain efficient throughput rate. Both from qualitative and quantitative terms, concerning the quality of the produced disparity map and the frame rate output of the module, a highly efficient method dealing with the stereo correspondence problem is presented.
The overall architecture is realised on a single FPGA device of the Stratix IV family of Altera devices, with a maximum operating frequency of 511 MHz. Real-time speeds rated up to 768 frames per second for a 640x480 pixel resolution image pair with 80 disparity levels, are achieved, which enable the proposed module for real stereo vision applications. The relationship between the number of frames processed per second and the processed image size assuming square images, for an operating range of 80 disparity levels, is presented in Figure 3. The hardware architecture is shown in Figure 4.

![Diagram of FPGA Design](image)

**Fig. 4. FPGA Design (Georgoulas & Andreadis, 2009)**

### 3.3 FPGA device specifications

The architectures by (Georgoulas et al., 2008; Georgoulas & Andreadis, 2009) have been implemented using Quartus II schematic editor by Altera. Both approaches have been then simulated to prove functionality, and once tested, finally mapped on FPGA devices. The analytical specifications of the target devices are given in Table 3. As it can be found out, space efficiency while maintaining high operating frequencies, is achieved.

<table>
<thead>
<tr>
<th>Author</th>
<th>Device</th>
<th>Total Registers</th>
<th>Total ALUTs (%)</th>
<th>Total LABs (%)</th>
<th>Total Pins (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Georgoulas et al., 2008</td>
<td>Altera EP2S180F1 020C3</td>
<td>5,208</td>
<td>59 (84,307/143,520)</td>
<td>83 (7,484/8,970)</td>
<td>3 (25/743)</td>
</tr>
<tr>
<td>Georgoulas &amp; Andreadis, 2009</td>
<td>Altera EP4SGX290 HF35C2</td>
<td>15,442</td>
<td>59 (143,653/244,160)</td>
<td>74 (9,036/12,208)</td>
<td>10 (70/660)</td>
</tr>
</tbody>
</table>

Table 3. Specifications of target devices
4. Experimental Results

In (Georgoulas et al., 2008) the disparity map is computed using an adaptive technique where the support window for each pixel is selected according to the local variation over it. This technique enables less false correspondences during the matching process while preserving high image detail in regions with low texture and among edges. The post filtering step comprising the CA filter enables satisfactory filtering of any false reconstructions in the image, while preserving all the necessary details that comprise the disparity map depth values. The resulting disparity maps are presented in Figure 5.

Fig. 5. Resulting disparity map for (a) Corridor (b) Cones image pair, respectively

Considering the occlusion-based approach satisfactory improvement in the accuracy of the resulting disparity maps is obtained, while preserving all the necessary details of the disparity map depth values. The resulting disparity maps are presented in Figure 6 along with original image pairs for (a) Tsukuba and (b) Cones.

Fig. 6. Resulting disparity map for (a) Tsukuba (b) Cones image pair, respectively
Quantitative results under various configurations can be seen in Table 4. The Cov (coverage) term, shown in Table 4, states the percentage of the image total pixels, for which a disparity value has been assigned. The Acc (accuracy) term states the ratio of the pixels given a correct disparity value (as compared with the ground truth) to the total assigned pixels.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Tsukuba</th>
<th>Cones</th>
<th>Teddy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc(%)</td>
<td>Cov(%)</td>
<td>Acc(%)</td>
</tr>
<tr>
<td>Georgoulas et al., 2008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Disparity Map</td>
<td>55</td>
<td>88</td>
<td>48</td>
</tr>
<tr>
<td>Refined Disparity Map</td>
<td>88</td>
<td>51</td>
<td>72</td>
</tr>
<tr>
<td>Georgoulas &amp; Andreadis, 2009</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Disparity Map</td>
<td>94</td>
<td>77</td>
<td>99</td>
</tr>
<tr>
<td>Refined Disparity Map</td>
<td>95</td>
<td>91</td>
<td>94</td>
</tr>
</tbody>
</table>

Table 4. Quantitative results of the proposed module under various configurations

5. Conclusions

The stereo correspondence problem comprises an active wide range of research. Many efforts have been made towards efficient solutions to address the various issues of stereo matching. As the improvements in computational resources steadily increase, the demand for real-time applications is getting compulsory. This chapter focuses on the latest improvements in the area of real-time stereo vision.

Area-based techniques prove to be more appropriate, handling the stereo correspondence problem aiming at real-time speeds. Their straightforward implementation in hardware enables them suitable in numerous applications such as high-speed tracking and mobile robots, object recognition and navigation, biometrics, vision-guided robotics, three-dimensional modelling and many more. Phase-based techniques also allow for efficient realization of such systems, requiring though slightly more complex design methodologies. Additionally, it must be noted that there are many other stereo vision techniques that were not covered by this work, due to the fact that they are mainly targeted in software-based platforms presenting higher processing times, not suitable for real-time operations.

FPGA implementations handling the stereo matching problem can be a promising alternative towards real-time speeds. Their uniqueness relies on their architecture and the design methodologies available. Parallel-pipelined processing is able to present great computational capabilities, providing with proper scalability opposed to the serial behaviour of most software-based techniques. On the other hand considering their significantly small volume, low cost, and extensive reconfigurability, they can be oriented towards embedded applications where space and power are significant concerns.
6. References


