Review Article

Literature Survey on Stereo Vision Disparity Map Algorithms

Rostam Affendi Hamzah[1,2] and Haidi Ibrahim[1]

[1]*School of Electrical & Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia*
[2]*Fakulti Teknologi Kejuruteraan (FTK), Universiti Teknikal Malaysia Melaka, Kampus Teknologi, Hang Tuah Jaya,*
76100 Durian Tunggal, Melaka, Malaysia

Correspondence should be addressed to Haidi Ibrahim; haidi_ibrahim@ieee.org

This paper presents a literature survey on existing disparity map algorithms. It focuses on four main stages of processing as proposed by Scharstein and Szeliski in a taxonomy and evaluation of dense two-frame stereo correspondence algorithms performed in 2002. To assist future researchers in developing their own stereo matching algorithms, a summary of the existing algorithms developed for every stage of processing is also provided. The survey also notes the implementation of previous software-based and hardware-based algorithms. Generally, the main processing module for a software-based implementation uses only a central processing unit. By contrast, a hardware-based implementation requires one or more additional processors for its processing module, such as graphical processing unit or a field programmable gate array. This literature survey also presents a method of qualitative measurement that is widely used by researchers in the area of stereo vision disparity mappings.

1. Introduction

Computer vision is currently an important field of research. It includes methods such as image acquisition, processing, analysis, and understanding [1]. Computer vision techniques attempt to model a complex visual environment using various mathematical methods. One of the purposes of computer vision is to define the world that we see based on one or more images and to restructure its properties, such as its illumination, shape, and color distributions. Stereo vision is an area within the field of computer vision that addresses an important research problem: which is the reconstruction of the three-dimensional coordinates of points for depth estimation. A system of stereo vision system consists of a stereo camera, namely, two cameras placed horizontally (i.e., one on the left and the other on the right). The two images captured simultaneously by these cameras are then processed for the recovery of visual depth information [2]. The challenge is to determine the best method of approximating the differences between the views shown in the two images to map (i.e., plot) the correspondence (i.e., disparity) of the environment.

Intuitively, a disparity map represents corresponding pixels that are horizontally shifted between the left image and right image. New methods and techniques for solving this problem are developed every year and exhibit a trend toward improvement in accuracy and time consumption.

Another device that is used to acquire depth information is a time-of-flight (ToF) or structured light sensor. Such a device is a type of active sensor, unlike a classic stereo vision camera. Devices of this type such as the Microsoft Kinect are cheap and have led to increased interest in computer vision applications. However, these active sensors suffer from certain characteristic problems [3]. First, they are subject to systematic errors such as noise and ambiguity, which are related to the particular sensor that is used. Second, they are subject to nonsystematic errors such as scattering and motion blur. According to the comparative analyses performed by Foix et al. [4], Kim et al. [5], and Zhang et al. [6], ToF devices perform satisfactorily only up to a maximum distance of approximately 5–7 meters and are too sensitive to be used in outdoor environments, especially in very bright areas. Because of these limitations of ToF sensors, stereo

Quantity of original papers from ScienceDirect

| Journal/book | 22 2005 | 23 2006 | 39 2007 | 32 2008 | 33 2009 | 43 2010 | 50 2011 | 58 2012 | 41 2013 | 47 2014 |

...... Trendline

(a)

Quantity of original papers from IEEE Xplore

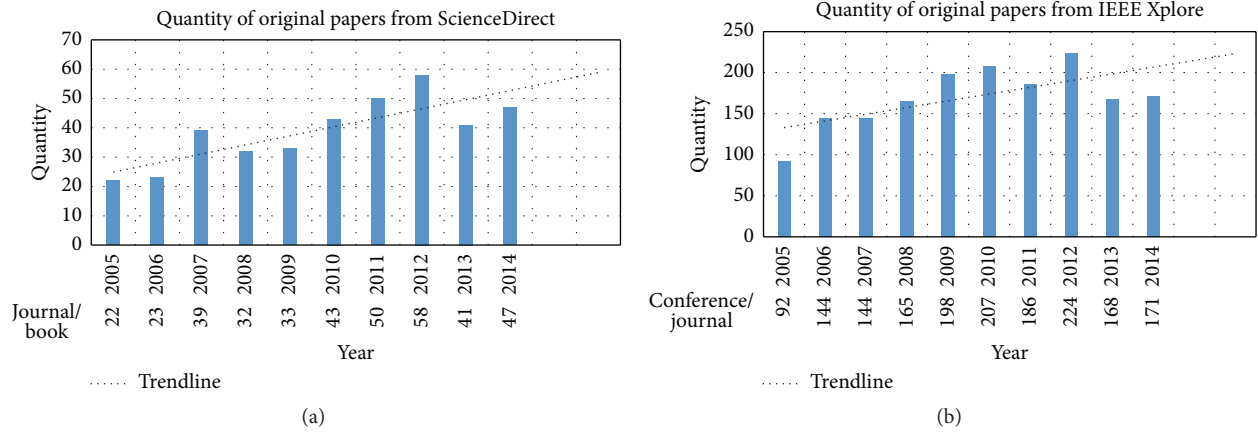| Conference/journal | 92 2005 | 144 2006 | 144 2007 | 165 2008 | 198 2009 | 207 2010 | 186 2011 | 224 2012 | 168 2013 | 171 2014 |

...... Trendline

(b)

FIGURE 1: The yearly distributions of the number of papers published from 2005 to 2014 in the area of stereo vision from the database of (a) ScienceDirect and (b) IEEE Xplore.

vision sensors (i.e., passive sensors) are more reliable and robust; they are able to produce high-resolution disparity maps and are suitable for both indoor and outdoor environments [7].

In stereo vision disparity map processing, the number of calculations required increases with an increasing number of pixels per image. This phenomenon causes the matching problem to be computationally complex [8]. The improvements to and reduction in computational complexity that have been achieved with recent advances in hardware technology have been beneficial for the advancement of research in the stereo vision field. Thus, the main motivation for hardware-based implementation is to achieve real time processing [9]. In real time stereo vision applications, such as autonomous driving, 3D gaming, and autonomous robotic navigation, fast but accurate depth estimations are required [10]. Additional processing hardware is therefore necessary to improve the processing speed.

An updated survey on stereo vision disparity map algorithms would be valuable to those who are interested in this research area. Figures 1(a) and 1(b) illustrate the quantity of original contributions published in this area over the past ten years (i.e., 2005–2014) from the databases of ScienceDirect and IEEE Xplore. The keywords used were stereo vision/stereo vision algorithm, and the components that were searched were the title, abstract, and keywords/index terms of the papers in the databases. All of these papers may represent contributions to fundamental algorithm development, analysis, or application of stereo vision algorithms. In both figures, the trendlines are increasing indicating that the field of stereo vision remains active in research and development and has become an interesting and challenging area of research. This paper provides a brief introduction to the state-of-the-art developments accomplished in the context of such algorithms. This work reviews the latest published stereo vision algorithms and categorizes them into different stages of processing, which are based on the taxonomy proposed by Scharstein and Szeliski [11]. This paper also discusses

two types of implementation platforms for these algorithms (i.e., software-based and hardware-based). In software-based platforms, the techniques are implemented only on a standard CPU, without any other additional processing hardware. In contrast to hardware-based platforms, the algorithms are executed on a CPU, with a GPU or FPGA as a standalone system.

The remainder of the paper is organized as follows. Previous review papers related to stereo vision disparity map algorithms are discussed in Section 2. Then, the taxonomy for the stages of processing performed in stereo vision disparity map algorithms is presented in Section 3. It consists of four subsections (i.e., matching cost computation, cost aggregation, disparity selection and optimization, and disparity refinement). Section 4 presents a review of algorithms implemented through software-based platforms, and Section 5 discusses real time stereo vision disparity map algorithms based on additional hardware (i.e., FPGAs and GPUs). A method of measuring the accuracy of stereo vision algorithms is explained in Section 6, and the conclusion is presented in Section 7.

2. Previous Reviews of Stereo Vision Disparity Map Algorithms

Numerous methods of implementation for stereo vision disparity mapping have been established in the past few years. This can be observed from the review papers listed in Table 1. The contents of these review papers are also summarized in this table. Among these review papers, the main focus was to summarize and compare the accuracy level and execution time of each cited algorithm. However, none of these reviews provided a detailed discussion of the stages of implementation based on the taxonomy proposed by Scharstein and Szeliski [11], as does the survey presented in this paper. Furthermore, this paper also reviews the latest algorithms implemented using two different types of platforms (i.e., software-based and hardware-based).

TABLE 1: Previous review papers on stereo vision disparity map algorithms.

| Year	Author	Focus
2002	Scharstein and Szeliski [11]	Proposed a taxonomy for vision algorithms and provided a quality metric to compare and evaluate multiple blocks of algorithms as shown in Figure 1. They have also provided a test bed for measurable evaluation of stereo depth map algorithms. The test bed or benchmarking dataset consists of four images (Tsukuba, Venus, Teddy, and Cones) which are available at http://www.middlebury.edu/stereo.
2003	Brown et al. [19]	Reviewed advances in stereo vision disparity map algorithms regarding correspondence methods and occlusion handling methods for real time implementations.
2008	Tombari et al. [77]	Presented a survey and compared the different methods of cost aggregation for stereo correspondence through accuracy and computational requirements.
2008	Lazaros et al. [12]	Reviewed developments in stereo vision algorithms implemented via software and hardware categorized in terms of their major attributes. The comparison of local and global methods provided by previously developed algorithms implemented on software and hardware based platforms was presented in this work.
2011	Tombari et al. [101]	Contributed an evaluation of stereo vision depth map algorithms in terms of their 3D object recognition ability.
2013	Tippetts et al. [8]	Reviewed stereo vision algorithms and their suitability for resource-limited systems. They have compiled and presented an accuracy and runtime performance data for all stereo vision disparity map algorithms in the past decade with an emphasis on real time performance.
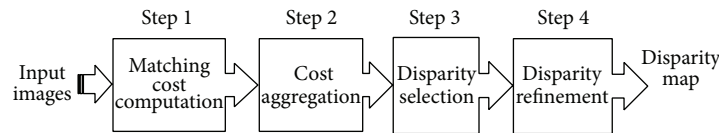
FIGURE 2: A framework for the development of a stereo vision algorithm.

3. A Taxonomy for the Processing Stages of Stereo Vision Disparity Map Algorithms

Most stereo vision disparity map algorithms have been implemented using multistage techniques. These techniques, as codified by Scharstein and Szeliski, consist of four main steps as shown in Figure 2 [11]. In this figure, the input images are obtained from stereo vision sensors (i.e., from at least two cameras). Commonly, these cameras are arranged horizontally and set up which produce two or more corresponding images. For the explanation or the process as described by adopted taxonomy, these input images are assumed to be rectified images. Next, the image pair to be analysed will pass through all of the blocks, in sequence, beginning with Step 1 and ending with Step 4. The output of this process should be a smooth disparity map. In essence, each block represents one or more algorithms whose performance can be measured based on the expected output. This taxonomy has been widely used by many current developers of stereo vision disparity map algorithms [8, 12].

In general, stereo vision disparity map algorithms can be classified into local or global approaches. A local approach is also known as area based or window based approach. This is because the disparity computation at a given point (or pixel) depends only on the intensity values within a predefined support window. Thus, such method considers only local information and therefore has a low computational complexity and a short run time. Local methods include all

four steps of the taxonomy. Examples of implementation of such methods are provided by the work of Mattoccia et al. [13], Arranz et al. [14], and Xu et al. [15]. The disparity map value assignment is achieved through winner take all (WTA) optimization. For each pixel, the corresponding disparity value with the minimum cost is assigned to that pixel. The matching cost is aggregated via a sum or an average over the support window.

By contrast, a global method treats disparity assignment as a problem of minimizing a global energy function for all disparity values. Such a method is formulated as an energy minimization process with two terms in the objective function (i.e., a data term, which penalizes solutions that are inconsistent with the target data and a smoothness term, which enforces the piecewise smoothing assumption with neighboring pixels). The smoothness term is designed to retain smoothness in disparity among pixels in the same region. The disparity map is produced by assigning similar depth values to neighboring pixels. Global methods produce good results but are computationally expensive. Therefore, they are impractical for use in real time systems. Global methods typically skip Step 2 of the taxonomy depicted in Figure 2 (i.e., they do not perform cost aggregation and therefore contain only three steps) [16–18]. Markov random field (MRF) modelling is the approach that is most common approach used in global methods. This type of modelling uses an iterative framework to ensure smooth disparity maps and high similarity between matching pixels.
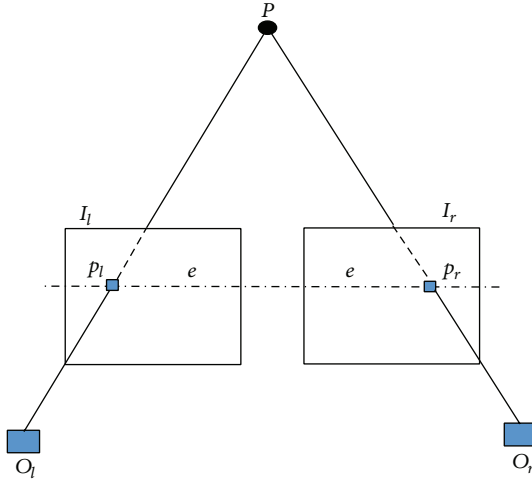
FIGURE 3: Epipolar geometry. The 3D image of target scene at point P.

3.1. Matching Cost Computation. All stereo matching algorithms require a cost criterion to measure the extent of matching between two pixels. The matching cost computation is the stage in which whether the values of two pixels correspond to the same point in a scene is determined. Therefore, the stereo matching cost computation can be defined as a method of determining the parallax values of each point between the left and right images [19]. The matching cost is computed at each pixel for all pixels under consideration. The difference in pixel intensity between a pair of the matching pixels in two images is called the disparity and can be associated with depth values through three-dimensional (3D) projection.

The matching points must lie on epipolar lines e, as shown in Figure 3. This matching can be performed via a one-dimensional horizontal search if the stereo pairs are accurately calibrated [20]. A target point P is viewed from the optical centers O_l and O_r of the two cameras. It produces one left image plane I_l and one corresponding right image plane I_r one from each of the two cameras. The points p_l and p_r represent the matching pixel intensities in the left and right image planes, respectively, for the same scene at point P. Therefore, the targeted left and right matching points must be located at the same coordinates on the horizontal line e. In a stereo vision system, this requirement is imposed in the form of an epipolar constraint [21]. This epipolar constraint plays a significant role in stereo matching. This is because the search for correspondences can be limited to a line instead of the entire image space, thereby reducing the required time and search range. Early stereo vision disparity map algorithm for the matching cost computation task uses pixel based technique [12]. These algorithms are the methods of absolute differences (AD), squared differences (SD); adaptations of the former include the methods of sampling-insensitive absolute differences and truncated absolute differences. These algorithms can be applied to grayscale or color images.

Area based or window based techniques are capable of offering richer data than matching techniques based on individual pixels or features. Such techniques can be more accurate because the matching process considers the entire set of pixels associated with image regions. Common algorithms for window based techniques include the sum of absolute differences (SAD), the sum of squared differences (SSD), normalized cross correlation (NCC), rank transforms (RT), and census transforms (CT) [12]. The matching cost is calculated over a support region. This support region, which is commonly referred to as support or aggregating window, may be square or rectangular and may be fixed or adaptive in size. The major shortcoming of window based technique is that these approaches commonly assume that all pixels within a support window have similar disparity values. This is not necessarily true for pixels near depth discontinuities or edges. Hence, an improper selection of the size and shape of the matching window can lead to poor depth estimations.

3.1.1. Absolute Differences (AD). The AD algorithm aggregates the differences in luminance (or intensity values) between the pixel in the left image I_l and the corresponding pixels in the right image I_r as given by

$$\text{AD}(x, y, d) = \left| I_l(x, y) - I_r(x - d, y) \right|. \tag{1}$$

In this equation, (x, y, d) represents the disparity map coordinates, where (x, y) are the coordinates of the pixel of interest and d is the disparity (or depth) value. Typically, in the matching process, I_l is used as the reference image and the right image I_r represents the target (or candidate) image. The AD algorithm is the simplest among matching cost algorithms. Because of its low complexity, Wang et al. [22] used this algorithm for real time stereo matching using graphics hardware (GPU). The AD algorithm functions satisfactorily in regions with little texture, but, for highly textured images, this algorithm is not capable of producing a smooth disparity map. To overcome this difficulty, the truncated version of the AD algorithm was developed. The truncated absolute difference (TAD) algorithm, as implemented by Min et al. [23] and Pham and Jeon [24], is able to minimize the errors in disparity maps. Furthermore, the TAD algorithm uses the colors and gradients at matching pixels to improve its robustness against variations in illumination.

3.1.2. Squared Differences (SD). The SD algorithm aggregates the squared differences between the reference pixels in I_l and the candidate pixels in I_r as described in

$$\text{SD}(x, y, d) = \left| I_l(x, y) - I_r(x - d, y) \right|^2. \tag{2}$$

Yang et al. [25] implemented the SD algorithm for their matching cost computation in a subpixel estimation method for disparity mapping. Considerable noise was generated at the boundaries in their initial disparity maps. However, Yang et al. applied a bilateral filter (i.e., a type of edge preserving filter) to improve the flattening of edges and to smooth areas near depth discontinuities. Recently, Miron et al. [26] tested various matching cost functions in their stereo disparity map algorithms for intelligent vehicle applications. They concluded that the SD algorithm produced the largest error. The errors that occur due to the SD algorithm are highly sensitive to brightness and noise, especially in real time environment.

3.1.3. Feature Based Techniques. The matching cost function can also be constructed using feature based techniques. This approach attempts to establish correspondences only for similar feature points that can be unambiguously matched. Common methods of feature extraction include those based on visual features (e.g., edges, shapes, textures, segmentation, and gradient peaks), statistical characteristics (e.g., minima, medians, and histograms), and transformation features (e.g., Hough transforms, wavelet transforms, and Gabor transforms) [27]. As an example, Sharma et al. [28] developed a new disparity map algorithm using features derived from the scale invariant feature transform (SIFT) for autonomous vehicle navigation. In their implementation, they modified the SIFT algorithm to use self-organizing map to achieve more efficient performance in feature matching process. Their results indicate that the computation time of their method is reduced compared with the conventional SIFT algorithm. Because only feature points are correlated, although the computational cost is significantly reduced, the complete disparity map cannot be obtained. The feature matching accuracy is still low. Sparse disparity maps are produced because only matching points derived from the targeted object features are used [29]. Furthermore, feature based matching techniques exhibit low accuracy and are rather insensitive to occlusion and textureless areas. Liu et al. [30] used a combination of image segmentation and edge detection for the matching cost function. The time implementation was fast but the accuracy level also remained low in regions of discontinuity. Ekstrand et al. [31] used a segmentation process that created one-dimensional segments containing information on color and edge coordinates. A segment correspondence estimation was performed for every row of the image to reduce the possibility of mismatched segments. However, errors still occurred as a result of using limited vertical support segments during the matching process. Thus, feature based techniques are not preferred, and their usage by researchers for the development of disparity map algorithms remains low.

3.1.4. Sum of Absolute Differences (SAD). The SAD algorithm, defined in (3), considers the absolute difference between the intensity of each pixel in the reference block and that of the corresponding pixel in the target block

$$\text{SAD}(x, y, d) = \sum_{(x,y)\in w} \left| I_l(x, y) - I_r(x - d, y) \right|. \quad (3)$$

The differences are summed over the aggregated support window w to generate a simple metric of block similarity known as a disparity map. The SAD algorithm is a well-known algorithm for matching cost computation. The SAD algorithm is able to function in real time implementation because of its low computational complexity. This was proven by Tippetts et al. [32], who calculated and evaluated SAD performances for real time human pose images in a resource limited system. Lee and Sharma [33] implemented real time disparity map algorithm estimations using the sliding window technique to calculate matching costs using the SAD algorithm. Their algorithm uses parallel processing via a graphical processing unit (GPU). By virtue of applying this new technique at the matching cost stage, the accuracy of stereo vision processing can be increased while simultaneously improving the speed.

Gupta and Cho [34] implemented a new technique using two different sizes of correlation windows in the SAD algorithm. At the first level, the initial cost aggregation is determined, and, at the second level, the object boundaries are improved using a smaller window size compared with the first-level implementation. The results produced are more accurate in terms of pixel matching but still require the implementation of multiple loops at different window sizes. The SAD algorithm is fast, but the quality of the initial disparity map that is produced is low because of the noise at object boundaries and in textureless regions.

3.1.5. Sum of Squared Differences (SSD). Equation (4) presents the SSD algorithm, in which the summation is performed over the squared differences in pixel intensity values between two corresponding pixels in the aggregated support window w

$$\text{SSD}(x, y, d) = \sum_{(x,y)\in w} \left| I_l(x, y) - I_r(x - d, y) \right|^2. \quad (4)$$

An early implementation of the SSD algorithm for the matching cost calculation stage was achieved by Fusiello et al. [36]. They tested the SSD algorithm on multiple fixed window blocks to reduce the incidence of occlusion errors. The purpose of using multiple blocks of windows is to search for the smallest error, to select an appropriate pixel of interest in the disparity map. Yang and Pollefeys [37] used technique similar to that presented in [36], but their algorithm was implemented on platform with a GPU. They achieved good results in terms of speed compared with Fusiello's work. Currently, there is still relatively little research on the use of the SSD algorithm in for stereo vision disparity map algorithms compared with that on other matching cost algorithms. This is evident from the previous review papers [8, 12] on stereo vision disparity map algorithms and is also shown in Table 2.

3.1.6. Normalized Cross Correlation (NCC). The NCC algorithm is another method of determining the correspondence between two windows around a pixel of interest. The normalization within the window compensates for differences in gain and bias [38]. Equation (5) specifies the formula for the NCC technique:

$$\text{NCC}(x, y, d)$$
$$= \frac{\sum_{(x,y)\in w} I_l(x, y) \cdot I_r(x - d, y)}{\sqrt{\sum_{(x,y)\in w} I_l^2(x, y) \cdot \sum_{(x,y)\in w} I_r^2(x - d, y)}}. \quad (5)$$

However, the NCC algorithm tends to blur regions of discontinuity more than other matching cost algorithms [38]. This is because any outliers lead to large errors in the NCC calculations. A new method for low-dimensional image features matching using NCC has been proposed by Satoh [39].

Table 2: Summary of stereo vision disparity map algorithms implemented on different computational platforms.

Year	Author	Method	Step 1	Step 2	Taxonomy stage Step 3	Step 4	Algorithm	Computational platform
2004	Park et al. [69]	Local + global	Edge based	FW	MRF based	Box filtering	Segmentation based	—
2006	Wang et al. [22]	Local	AD	MW	WTA	Median filter	Multiple cost aggregation	CPU Intel P4 and GPU ATI 800
2006	Yoon and Kweon [102]	Local	TAD	ASW	WTA	Left-right consistency check	Adaptive support weight	—
2007	Cigla et al. [70]	Local	Edge based	Arbitrary shaped based AW	WTA	Angle sweep	Multiview region-based	—
2008	Hirschmüller [71]	Global	Semiglobal matching pixel based	—	DP	Left-right consistency check + median filtering	Semiglobal matching	CPU Opteron 2.2 GHz
2009	Yang et al. [63]	Global	Color weighted correlation	—	BP	Left-right consistency check	Hierarchical loopy belief propagation	—
2009	Salmen et al. [16]	Global	SAD	—	DP	Left-right consistency check	Optimized dynamic programming	CPU with 1.8 GHz
2010	Mattoccia et al. [13]	Local	TAD	Joint bilateral filtering based ASW	WTA	Left-right consistency check	Joint bilateral stereo	CPU Intel Core Duo 2.14 GHz
2010	Nalpantidis and Gasteratos [59, 83]	Local	AD	ASW	WTA	—	LCDM	—
2010	Kalarot and Morris [87]	Global	Pixel based	—	DP	—	Symmetric dynamic programming	CPU Intel Quad Core 2.3 GHz and GPU GeForce GTX280, FPGA Altera Stratix III
2011	Lee and Sharma [33]	Local	SAD + RT	Multiresolution based MW	WTA	—	Rank transform + multiresolution	Personal Computer and GPU (detail n/a)
2011	Mei et al. [45]	Local	AD + CT	Cross based AW	WTA	Cross-checking	AD census + dynamic cross	CPU Intel Core 2 Duo 2.20 GHz and GPU GeForce GTX480
2011	Min et al. [23]	Local	TAD	Joint histogram-MW based aggregation	WTA	Box filtering	Histogram-based aggregation	CPU (detail n/a)
2011	Pérez and Sánchez [17]	Global	Pixel based	—	BP	Left-right consistency check	Memory-efficient belief propagation	—
2011	Sharma et al. [28]	Local	Feature based	—	WTA	—	SIFT	Autonomous vehicle navigation
2011	Psota et al. [78]	Local	SAD	ASW	WTA	Median filter	Iterative refinement adaptive support weight	—

TABLE 2: Continued.

Year	Author	Method	Step 1	Step 2	Taxonomy stage Step 3	Step 4	Algorithm	Computational platform
2011	Tippetts et al. [32]	Local	Gaussian filter + profile shape	MW	WTA	Median filter	Profile shape matching	CPU AMD Phenom II 2.8 GHz
2011	Banno and Ikeuchi [68]	Global	Pixel based	—	BP	Directed anisotropic diffusion	3D directed anisotropic diffusion	—
2011	Zhao and Taubin [103]	Local	SAD	Multiresolution based AW	WTA	Cross-checking	Multiresolution adaptive windows	CPU (detail n/a) and GPU GeForce GTX280
2011	Nalpantidis et al. [20]	Local	AD	Gaussian weighted + cellular automata	WTA	—	Efficient hierarchical	Stratix II GX FPGA
2012	Arranz et al. [14]	Global	Pixel based	—	MRF based optimization	—	Multiresolution energy minimization	CPU Intel i7 3.4 GHz with 7 GB RAM
2012	Xiang et al. [9]	Global	Color weighted pixel based	—	Hierarchical belief propagation	Uniqueness constraint + similarity constraint	Hierarchical belief propagation	CPU Intel 2.7 GHz and GPU Geforce GTX260
2012	Yang [79]	Local	Edge based	Minimum spanning tree	WTA	Guided filter	Minimum spanning tree based	CPU Intel i7 1.8 GHz with 4 GB RAM
2013	Chen and Lai [73]	Global	Pixel based	—	GC	—	Parallel GC	CPU Intel i7 (detail n/a)
2013	Chen and Su [54]	Local	Area based	Shape adaptive low complexity AW	WTA	Cross-checking	Shape adaptive low complexity	CPU Intel i5 2.6 GHz
2013	Wang et al. [18]	Global	Pixel based	—	GC	—	Hierarchical bilateral disparity structure (HBDS)	CPU Intel Core 2 Duo 2.33 GHz with 2 GB RAM
2014	Cheng et al. [40]	Local + global	ZNCC	Match-edge based MW	WTA	Global edge constraint	Global edge constraint	n/a
2013	Choi and Park [96]	Global	SAD	—	GC	Truncated linear function	Efficient GPU-based GC	CPU (detail n/a) and GPU GeForce GTX580
2013	Wang et al. [62]	Global	TAD	—	GC	Median filtering + interpolation and depth discontinuity adjustment	Reliable points based GC	CPU Intel Core 2 Duo 2.0 GHz with 2 GB RAM
2013	Michael et al. [66]	Local + global	AD	—	WTA	Left-right consistency check	Real time semiglobal	CPU Intel Xeon and GPU GeForce GTX480
2013	Cigla and Alatan [46]	Local	SAD + CT	Successive weighted summation based ASW	WTA	Permeability based filtering	Separable successive weighted summation	CPU Intel i7 3 GHz with 6 GB RAM, and GPU GeForce GTX480
2013	Gupta and Cho [34]	Local	SAD	MW	WTA	Left-right consistency check	Double window based	CPU (detail n/a)
2013	Hosni et al. [58]	Local	TAD (color + gradient)	ASW	WTA	Weighted bilateral median filtering	ASW + patch match stereo	CPU Intel Core 2 Quad 2.4 GHZ and GPU GeForce GTX480

Table 2: Continued.

Year	Author	Method	Step 1	Step 2	Step 3	Step 4	Algorithm	Computational platform
					Taxonomy stage			
2013	Zhang et al. [57]	Local	TAD (color + gradient)	ASW	WTA	—	Real time GPU	CPU Intel i7 and GPU GeForce GT540M
2013	Kowalczuk et al. [104]	Local	SAD	Iterative ASW + temporal ASW	WTA	Median filtering	Iterative refinement ASW	CPU AMD Phenom II X6 3.0 GHz and GPU GeForce GTX580
2013	Lee et al. [60]	Local	Three-mode cross CT	ASW	WTA	Left-right consistency check	ASW + three-mode cross census	—
2013	Lin et al. [105]	Local	TAD (color + gradient)	Hierarchical cluster based ASW	WTA	Left-right consistency check	Hierarchical clustering algorithm	CPU Intel Core 2 Duo 2.99 GHz and GPU GeForce 9500GT
2013	Zhang et al. [47]	Local	SAD + ALD	Cross based AW	WTA	Left-right consistency check	SAD + arm-length differences (ALD)	CPU Intel Core 2 Duo 3.00 GHz and GPU GeForce GTX570
2013	Pham and Jeon [24]	Local	TAD	MW	WTA	Weighted median filtering	Domain transformation	CPU AMD Athlon 64 X2 Dual Core 2 GHz and GPU GeForce GTX460
2014	Vijayanagar et al. [65]	Local	Segment based	MW	WTA	Multi resolution anisotropic diffusion	Multiresolution anisotropic diffusion	CPU running at 3.2 GHz with 8 GB RAM and GPU AMD FirePro V4900
2013	Xu et al. [15, 80]	Local	TAD	ASW	WTA	—	Adaptive weight selection	—
2014	Wang et al. [75]	Global	Pixel based	—	MRF based optimization	Joint bilateral filtering	MRF based optimization	CPU running at 3 GHz with 3 GB RAM
2013	Mattoccia [99]	Local	Segment based	ASW	WTA	Guided filter	Multiple algorithms (FW, ASW)	FPGA Xilinx Spartan 6
2013	Zhang et al. [6]	Global	BT + AD	—	BP	Color image depth matting + 2D polynomial regression	Multiscale BP	CPU Intel i7 with 4 GB RAM and GPU GeForce GTX295
2013	Fowers et al. [91]	Local	SAD	MW	WTA	—	SAD	CPU Intel Quad Core 3.00 GHz and GPU GeForce GTX295, FPGA Altera Stratix III E260
2013	Wang et al. [94]	Local	AD + CT	AW	WTA	—	Real time HQ-FPGA	FPGA Altera Stratix IV
2013	Ma et al. [44, 67]	Local	TAD	ASW	WTA	Weighted median filtering + left-right consistency check	Constant time weighted median filtering	CPU Intel Xeon 2.83 GHz and GPU GeForce GTX580
2014	Miron et al. [26]	Local	CT	Cross based AW	WTA	Anisotropic local polynomial approximation	Cross-comparison census	—

TABLE 2: Continued.

Year	Author	Method	Step 1	Step 2	Taxonomy stage Step 3	Step 4	Algorithm	Computational platform
2014	Yang et al. [49]	Local	TAD	AW	WTA	Adaptive guided filter	Fast adaptive guided filter	CPU Intel Core Duo 3.16 GHz with 2 GB RAM and GPU Nvidia Tesla C2050
2014	Colodro-Conde et al. [100]	Local	SAD + CT	MW	WTA	Median filtering	SAD + CT	Xilinx FPGA
2014	Jin and Maruyama [92]	Local	CEN + AD	AW	WTA	—	CA + FLC	Xilinx XC6VLX240T
2004	Park et al. [69]	Local + global	Edge based	FW	MRF based	Box filtering	Segmentation based	—
2014	Mota et al. [95]	Global	Symmetry	—	MRF based	—	SymStereo	CPU Intel i7 3 GHz and Geforce GTX 680
2014	Tippetts et al. [106]	Local	Edge	MW	WTA	Vertical filtering	Profile shape matching	Xilinx Virtex 4 FX60
2014	Xu et al. [89]	Local	Pixel based	FW	WTA	—	Pyramidal	AMD Opteron processor 6366HE, Intel Xeon processor E5-2620, Nvidia GeForce GTX 770
2014	Werner et al. [84]	Local	NCC	—	WTA	Cross bilateral filter	Real time full HD	FPGA
2015	Ekstrand et al. [31]	Local	Segmentation	MW	WTA	Median filter	Real time segmentation	Xilinx FPGA-Spartan 3A-1800A
2015	Matsuo et al. [82]	Local	AD + edge	FW	WTA	Weighted joint bilateral filter	In-Loop Feedback Matching	—
2015	Yao et al. [97]	Global	Feature-based	—	MRF	—	Real time adaptive meshing	CPU Xeon and Nvidia Tesla K2
2015	Ploumpis et al. [76]	Global	Feature + particle filter	—	MRF based	Histogram + segmentation	Particle filter + scattered control landmarks	CPU Intel Core 2 Duo
2015	Chen et al. [56]	Local	Median filter + AD	ASW	WTA	—	Trilateral	CPU Intel i7 3 GHz with 8 G RAM

The NCC algorithm was chosen because of its robustness to intensity offsets and changes in contrast. The results achieved in Satoh's work exhibit high accuracy, but considerable computational resources are required. Additionally, Cheng et al. [40] implemented their matching cost calculation using a zero mean normalized cross correlation (ZNCC) in which the pixels at which edges are located are manipulated via a multiple-window strategy. This method relies on a neutral network model. Furthermore, the least-mean-square delta rule is used for training and for the determination of the proper window shape and size for each support region. These techniques offer improved accuracy, but their computational requirements are still high.

3.1.7. Rank Transform (RT).

The matching cost for RT is given by (6) and is calculated based on the absolute difference between two ranks (i.e., Rank_{ref} from the reference image and Rank_{tar} from the target image):

$$
\begin{aligned}
&\text{RT}(x, y, d) \\
&= \sum_{(x,y) \in w} \left| \text{Rank}_{\text{ref}}(x, y) - \text{Rank}_{\text{tar}}(x - d, y) \right|.
\end{aligned}
\tag{6}
$$

In this equation, Rank_{ref} and Rank_{tar} are calculated as shown in

$$
\text{Rank}(x, y) = \sum_{(i,j)(x,y)} L(i, j)
\tag{7}
$$

with $L(i, j)$ as defined in

$$
L(i, j) = \begin{cases} 0: & I(i, j) < I(x, y) \\ 1: & \text{otherwise}, \end{cases}
\tag{8}
$$

where (i, j) are the coordinates of a neighboring pixel and (x, y) are the coordinates of the pixel of interest. Equation (8) computes the number of neighboring pixels $I(i, j)$ that have values larger than that of the central pixel $I(x, y)$. A new model of a disparity map algorithm that uses the RT algorithm to achieve improved accuracy has been proposed by Gac et al. [41]. Through careful selection of the window sizes, a reliable initial disparity map is efficiently obtained. At the time of its publication, this method demonstrated the highest correct initial matching rate of any matching cost algorithm in date. The RT algorithm is typically effective for coping with brightness differences and image distortions. Sometimes when the RT algorithm is used, a matching pixel may look extremely similar to a neighboring pixel, leading to matching ambiguity. In [42], a new extension of the RT approach was developed to reduce this matching ambiguity using a Bayesian model. This model considers not only the similarities between the left and right image pixels similarities but also the level of ambiguity within each image independently. The results of experiments on images exhibiting variations in intensity and brightness differences, as reported by authors of that study, indicate a reduction in matching ambiguities.

3.1.8. Census Transform (CT).

The CT algorithm translates the results of comparisons between a center pixel and its neighboring pixels within a window into a bit string as shown

$$
\text{Census}(x, y) = \text{Bitstring}_{(i,j) \in w} \left(I(i, j) \geq I(x, y) \right).
\tag{9}
$$

This algorithm is calculated using the Hamming distances between the census bit strings of on the corresponding match candidates, as given by

$$
\begin{aligned}
\text{CT}(x, y, d) = &\sum_{(x,y) \in w} \text{Hamming}\left(\text{Census}_{\text{ref}}(x, y) \right. \\
&\left. - \text{Census}_{\text{tar}}(x - d, y) \right),
\end{aligned}
\tag{10}
$$

where $\text{Census}_{\text{ref}}$ represents the census bit string from the reference image and $\text{Census}_{\text{tar}}$ represents the census bit string from the target image. The CT algorithm is rather robust to the disparity discontinuities because of its good outlier tolerance, as described by Humenberger et al. [43]. This claim was proven by performance comparisons between the CT algorithm and the SAD algorithm. The disparity maps produced by the CT algorithm exhibited higher matching quality at object borders than those produced by the SAD algorithm. The disadvantage of the CT algorithm is its tendency to produce incorrect matches in regions with repetitive structures. This shortcoming was mitigated by Ma et al. [44] through their modifications to the CT algorithm. They implemented additional bits to represent the differences between the pixel of interest and the neighborhood pixels. According to their results, the accuracy of the disparity map was improved and the incorrect matching problem was alleviated by this modification. In addition, the proposed algorithm demonstrated greater robustness when applied to a noisy image compared with the conventional CT algorithm.

Several researchers have also developed matching cost methods based on a combination of two algorithms. A combination of the AD and CT algorithms as shown by Mei et al. [45] successfully reduces the occurrence of errors. The reason for combining these two methods is to compensate for their respective limitations. The CT algorithm tends to produce incorrect matches in regions with repetitive local structures, whereas the AD algorithm does not performance well on large, textureless regions. Similarly, a combination of the SAD and CT algorithms will also lead to higher performance but will incur an increase in computational complexity [46]. The SAD and CT cost measures are obtained individually, and the final cost function is constructed as a linear combination of both cost measures based on a weighting factor. The accuracy improvement achieved by Zhang et al. [47] was accomplished by means of a cost measure combining the SAD approach and arm length differences (ALD). The use of ALD was inspired by the similarity of the matching pixels support regions in the vertical direction as a result of the pixels being located on the same horizontal line. This combination is able to reduce errors in most regions, especially those containing repeated color and shapes. Lee et al. [48] combined the CT and gradient difference approaches to achieve a higher matching cost quality. However, according to them, matching ambiguities can occur in certain regions as a result of similar or repetitive texture patterns.
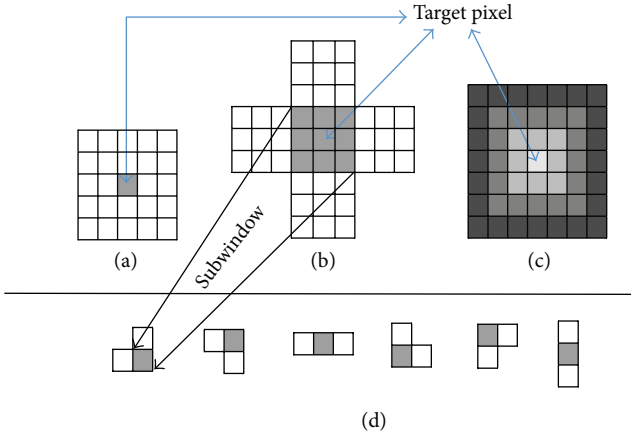
FIGURE 4: Cost aggregation windows. (a) 5×5 pixel square window, (b) adaptive window, (c) window with adaptive support weights, and (d) all six possible resulting shapes of adaptive windows.

3.2. Cost Aggregation. Cost aggregation is the most important stage for determining the general performance of a stereo vision disparity map algorithm, especially for local methods. The purpose of cost aggregation is to minimize matching uncertainties. Cost aggregation is needed because the information obtained for a single pixel upon calculating the matching cost is not sufficient for precise matching. Local methods aggregate the matching cost by summing them over a support region [11]. This support region is typically defined by a square window centered on the current pixel of interest, as shown in Figure 4(a). The most straightforward aggregation method is to apply a simple low-pass filter in the square support window. The fixed-size window (FW) technique (e.g., binomial or Gaussian, uniform (box filters)) suffers an increased error rate when the size of the support window is increased over a certain threshold. Moreover, this method requires the parameters to be set to values suitable for the particular input dataset. Otherwise, it tends to blur object boundaries [49]. To avoid fattening artifacts near depth discontinuities, methods using shifting window or multiple windows (MW) as well as methods using adaptive windows (AW), windows with adaptive sizes, or adaptive support weights (ASW) have been developed.

In the MW technique, multiple windows are selected from among a number of candidates based on the support windows that produce smaller matching costs. This method was implemented by Hirschmüller et al. [50] and Veksler [51] in their previous studies of real time stereo vision disparity map algorithms. However, their experimental results reveal difficulties in preserving dedicated pixel arrangements in disparity maps, especially at object boundaries. This occurs because of the shape of the support windows. This approach is imperfect for a small number of candidates. To resolve this problem, the AW technique was developed to reduce the errors in the disparity map caused by boundary problems. In this method, the support regions are constructed as approximations to the local image structures. Figure 4(b) illustrates

the application of this method with five subwindows with dimensions of 3×3. These subwindows must be located near the target pixel as shown in Figure 4. The cost aggregation with the minimum matching cost value for this pixel is calculated. For example, the cost can be calculated as the summation over the target pixel subwindow and any two other adjacent subwindows. The chosen shape of the valid matching windows for aggregation can therefore be any of the shapes shown in Figure 4(d). In practice, the shape of the adaptive window is adaptively varied to reflect the local image content, such as corners and edges.

The AW technique was implemented by Lu et al. [52] who achieved high quality results both near depth discontinuities and in homogenous regions. Lu's work was improved upon by Zhang et al. [53] through a modification to the concept of adaptive support regions. They developed support regions with arbitrarily adaptive shapes and implemented the algorithm on a GPU for real time applications. The shapes of these support regions are more flexible and are not restricted to be rectangles. These authors achieved high matching accuracy with real time implementation. In this AW technique, the algorithm attempts to find support windows that fit the shape or size of each region, while preventing them from crossing object boundaries. Furthermore, this technique is able to reduce computational costs as discussed by Chen and Su [54]. These authors proposed a shape adaptive low complexity technique for eliminating computational redundancy between stereo image pairs for pixels matching. They grouped pixels with the same depth value to reduce the number of computations.

A comparative study of the use of different support region techniques in the cost aggregation stage was performed by Fang et al. [55]. This study addressed the FW, AW, and ASW approaches. The authors concluded that the most advantageous technique for cost aggregation is the ASW approach. In this technique, each pixel in the support region is assigned a support weight, which depends on its intensity dissimilarity and spatial distance from the anchor pixel as shown in Figure 4(c). The target pixel which is located at the center is assigned different weight depending on distance as indicated by the different tone of colors. Generally, for typical ASW techniques, (11) is used to aggregate the matching costs $C'(p, d)$ at pixel p and disparity d

$$C'(p, d) = \sum_{q \in w_p} W(p, q) C(q, d), \quad (11)$$

where w_p is a square support window centered on pixel p. The window size is a user defined parameter. The value of the $W(p, q)$ function represents the possibility that a pixel q will possess a disparity value similar to that of the window's center pixel p. $C(q, d)$ represents a target pixel q with a disparity value d. Ideally, $W(p, q)$ should return a value of "1" if pixels p and q have equal disparity values and "0" otherwise. Chen et al. [56] developed a trilateral filter based on the ASW approach with using a bilateral filter. They also added a new weighted term to increase the robustness against object boundaries.

Essentially, in ASW application, a higher weight will be allocated to a pixel if its intensity is more similar to that of the anchor pixel and if it is located at a smaller distance from the anchor pixel, as implemented by Zhang et al. [57]. This method is able to produce a disparity map in which the object boundaries are well preserved and the accuracy is very high compared with the previous methods reported in their literature. Hosni et al. [58] presented an extensive evaluation of ASW regions. They performed their test on a GPU to evaluate whether the speed and computational efficiency were sufficient for real time responses. Their evaluations indicated that the ASW approach produces outstanding results in terms of both computational efficiency and the quality of the generated disparity maps. Nalpantidis and Gasteratos [59] developed a new approach based on the ASW technique. They combined it with the quantified gestalt law to calculate a weighting factor. In general, a correlation weight reflects the proximity, similarity, and continuity between both input images (i.e., left and right images).

3.3. Disparity Computation and Optimization. Generally, a stereo matching algorithm represents one of the two major optimizations approaches: the local approach or the global approach. In the local approach, when the final disparities are computed, the disparity for each pixel is essentially selected using a local winner takes all (WTA) strategy as define by

$$d_p = \arg\min_{d \in D} C'(p, d). \tag{12}$$

The disparity associated with the minimum aggregated cost d_p at each pixel is chosen. $C'(p, d)$ represents the aggregate cost obtained after the matching cost calculation, and D denotes the set of all allowed discrete disparities. The WTA strategy is utilized in this stage in local algorithms such as those implemented by Cigla and Alatan [46], Zhang et al. [53], and Lee et al. [60]. According to their findings, the disparity maps obtained at this stage still contain errors in the form of unmatched pixels or occluded regions. Because the aggregation in local methods is performed through summation or averaging over support regions, their accuracy is sensitive to noise and unclear regions. This occurs because only local information from a small number of pixels surrounding the pixel of interest is utilized to make each decision. Therefore, the accuracy of a local method at this stage depends on the matching cost computation and cost aggregation stages. Subsequently, in the disparity refinement stage, the errors will be reduced using several filtering techniques.

By contrast, in a global approach, certain assumptions are made about the depth of field of the scene, which are usually expressed in an energy minimization framework. The bulk of the effort in a global method is expended during the disparity computation phase, and the aggregation step is often skipped [11]. The most commonly adopted assumption is that the scene is locally smooth except for object boundaries, and thus neighboring pixels should have very similar disparities. This constraint is referred to as a smoothness constraint in the stereo vision literature. In the typical global stereo

vision formulation, the objective is to find an optimal energy disparity assignment function $d = d(x, y)$ that minimizes

$$E(d) = E_{\text{data}}(d) + \beta E_{\text{smooth}}(d), \tag{13}$$

where $E_{\text{data}}(d)$ represents the matching costs at the coordinates (x, y); the smoothness energy $E_{\text{smooth}}(d)$ encourages neighboring pixels to have similar disparities based on the previous stated assumptions and β is a weighting factor.

A global method such as the belief propagation (BP) approach requires large amounts of computational resources and memory for the storage of the image data and the execution of the algorithm. For the improvement to the BP technique achieved implemented by Liang et al. [61] which was implemented on the GPU, the time required for processing is still large compared with that required by the local method strategy. Wang et al. [62] implemented global approach using a graph cut (GC) algorithm to optimize the energy function. Their method selects disparity values with a lower energy value. Another well-known global technique is a dynamic programming (DP). DP is executed for each scan line (row) independently, resulting in polynomial complexity. The assumption adopted DP is that of an ordering constraint between neighboring pixels of the same row. Recently, the multiresolution energy minimization framework introduced by Arranz et al. [14] achieved real time performance while maintaining the resolution of producing disparity maps. The advantage of this framework is the reduction in computational complexity that is achieved through the multiresolution technique. However, for images of higher resolution and with many more different levels of disparity, the framework is unable to perform at a real time frame rate (30 frames per second).

3.4. Disparity Map Refinement. The purpose of the disparity refinement stage is to reduce noise and improve the disparity maps. Typically, the refinement step consists of regularization and occlusion filling or interpolation. The regularization process will reduce the overall noise through the filtering of inconsistent pixels and small variations among pixels on disparity map. The occlusion filling or interpolation process is responsible for approximating the disparity values in areas in which the disparity is unclear. Typically, occluded regions are filled with disparities similar to those of the background or textureless areas. Usually, the occlusion regions are detected by using left-right consistency check, such as those implemented by Yang et al. [63] and Heo et al. [64]. If the matching algorithm rejects disparities with low confidence, then the interpolation algorithm will estimate approximations to the correct disparities based on the local neighborhoods. The disparity refinement step normally combines local information from the local neighborhood near each measurement with a confidence metric.

Two classic and common techniques for local disparity refinement are Gaussian convolution and the median filter. In Gaussian convolution, disparities are estimated in combination with those of neighboring pixels in accordance with weights defined by a Gaussian distribution. The primary purpose of this method is to reduce the noise in
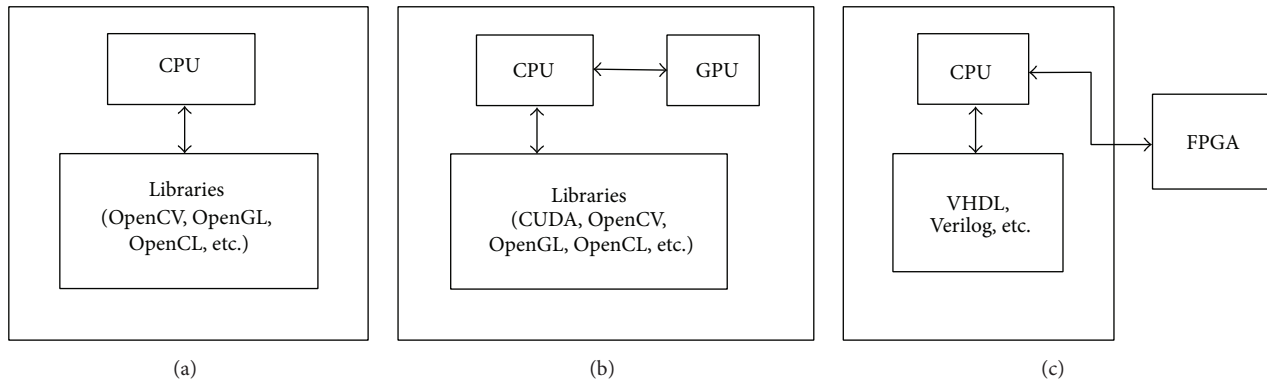
FIGURE 5: Block diagrams of the implementation of stereo vision disparity map algorithms. (a) Software-based one (e.g., only a computer or a laptop), (b) hardware-based onewith a GPU implementation (e.g., a computer or a laptop built with a GPU unit), and (c) hardware-based one with FPGA implementation connected to a computer to burn-in the codes and become a standalone system.

the disparity map, but the Gaussian filter also reduces the amount of fine detail present in the final disparity map. A technique developed by Vijayanagar et al. [65] uses the weights defined by a Gaussian filter to improve disparity maps by approximating missing disparity values based nearby high confidence disparity pixels as a guide to prevent filtering across object boundaries. Meanwhile, the median filter is able to remove small, isolated mismatches in disparity by virtue of its edge preserving property and it is suitable for real time implementation because of its low computational complexity. This filter selects the median value within window of pixels as the final result for the central pixel. In a study by Michael et al. [66], a disparity map refinement approach using median filtering was developed for a real time stereo vision algorithm. Furthermore, the median filter was modified by Ma et al. [67] using the constant time weighted technique. Their modification achieves high accuracy in removing noise and error while maintaining the edges in disparity maps.

The diffusion technique performs a function similar to that of Gaussian convolution. Moreover, there exists an adaptation of this approach called anisotropic diffusion. Unlike Gaussian convolution, which destroys edges and fine details, anisotropic diffusion applies smoothing without crossing any edges as implemented by Banno and Ikeuchi [68] in the disparity map refinement stage of their disparity map algorithm. Their approach was improved upon by Vijayanagar et al. [65], yielding a method called multiresolution anisotropic diffusion. In this method, the disparity map is downsampled using three different resolution factors. At each resolution, 35 iterations of the anisotropic diffusion process are performed. The result of the proposed algorithm is free of occlusion errors and the edges in the disparity map have been refined. Another approach, which was developed by Zhang et al. [6], employs a two-step process to further refine the estimated disparity map. The authors presented the results they achieved through a color image guided depth matting process in a framework based on Bayesian matting and 2D polynomial regression smoothing techniques. This technique was found to be used to be effectively preserving the discontinuities at object boundaries while achieving smoothing in flat regions.

4. Software-Based Stereo Vision Disparity Map Algorithms

This section reviews several software-based implementation processes of global and local methods for the generation of disparity maps. These algorithms were developed and tested using only a CPU as the processing hardware as shown in Figure 5. A software-based implementation is designed to use the CPU to interface with API software. The API software provides a set of libraries, such as Open Computer Vision (OpenCV), Open Computing Language (OpenCL) libraries, and Open Graphic Language (OpenGL). A previous summary of software-based stereo vision disparity map algorithm and their performances was presented by Brown et al. [19]. The discussion also addressed the corresponding methods and occlusion handling techniques. In essence, the differences in these algorithms lie in the cost aggregation stage and in the optimization of the building blocks which determine the main characteristics of the developed algorithms. Several researchers, such as Park et al. [69] and Cigla et al. [70], have developed new algorithms by taking advantages of local and global methods for handling occlusions, object boundaries, and untextured regions. These techniques follow an iterative process for allocating disparities that spread into certain segments by applying pixel similarity, constraints, considering overlapping regions, enforcing smoothness between similarly colored neighboring segments, and penalizing occlusions. Several other researchers have applied other global techniques with various modifications, resulting in a semiglobal approach. This method involves dynamic programming optimization, such as that implemented by Hirschmüller [71] and Salmen et al. [16]. The results reported by these authors indicate low computational complexity.

4.1. Global Approaches. The best known approaches among the global methods are belief propagation (BP) and graph cut (GC) algorithms. Pérez and Sánchez [17] used the BP approach to develop a real time, high-definition algorithm that outperformed classical BP by implementing two BP algorithms in their 3D telepresence systems. The first instance of

BP performs a classification of the pixels into areas designated as reliable, containing occlusion errors and textureless to reduce the numbers of memory accesses required for these three groups of pixels. The second BP process is used to decrease memory traffic by generating the final disparity map with a reduced number of iterations due to information from previous BP iterations. The experimental results demonstrated improved performance. The authors compared this approach with classical BP and observed a 90% improvement in efficiency. Wang and Yang [72] implemented ground control points (GCP) influenced by the MRF model. In their method, GCP-based regularization for the optimization framework is performed using a Bayesian rule. Meanwhile, the energy minimization technique for finding an optimal solution to the inference problem is implemented using the GC approach. Evaluations of this method demonstrated its effectiveness at improving the disparity map reconstruction to regularize problems of incorrect stereo matching. The approaches developed by Pérez and Sánchez [17] and Wang and Yang [72] both produce accurate disparity maps, but their computational costs remain high. However, Wang et al. [18] proposed a hierarchical bilateral disparity structure (HBDS) algorithm based on a GC technique to reduce computational complexity and improve the accuracy of the generated disparity maps. These authors divide all disparity levels hierarchically into a series of bilateral disparity structures to increase the fineness of the disparity map. During the refinement stage, any fattened regions are recalibrated based on the disparity values of all nearby pixels. The evaluation results indicated good performance with reduced processing time and improved disparity map accuracy.

A new technique was proposed by Chen and Lai [73] based on augmenting paths and the adoption of a push-relabelling scheme. The augmenting path algorithm functions by using multiple threads to calculate each block individually and to select another completed nearby block with which to merge. The proposed method identifies the independent processing loops in the GC approach and isolates the computation of each loop. Each image is sliced into smaller image segments, which will then be processed in parallel. The proposed method enables a remarkable decrease in execution time by a factor of 4.7 compared with the original GC approach, but considerable computational programming efforts are required. Kolmogorov et al. [74] developed four different smoothness terms (i.e., data, smoothness, occlusion. and uniqueness) to improve the accuracy of their results. The objectives of their method are to reduce the errors in occluded areas and increase the efficiency during postprocessing. Wang et al. [75] developed an algorithm using the MRF framework to eliminate holes and misaligned pixels. Their work produced high quality disparity maps but also required complex computational programming. Ploumpis et al. [76] developed a new stereo matching approach based on particle filters and scattered control landmarks. The proposed method consists of three steps. First multiple disparity maps are used to acquire a set of features or landmarks and then segment the images. Afterward, to estimate the best disparity values, scan line particle filtering is applied. In the last step, a Markov chain model is employed to reduce the computational

redundancy of the particle filtering process. Using this method, high quality disparity maps can be produced.

4.2. Local Approaches. In local methods, pixel correspondences are generated by measuring the correspondence and similarity between image regions and very effective implementation can be produced using this approach [77]. The assignment of disparity values is achieved by applying WTA strategy after calculating each candidate disparity value individually. The matching cost function is aggregated via a summation or an averaging over a support region. The disparity value with the minimum cost for each pixel is assigned to that pixel. An algorithm based on an efficient cost aggregation strategy was proposed by Mattoccia et al. [13]. These authors used joint bilateral filtering and expanded the calculation structures that allow for the efficient and accurate generation of disparity maps. The idea behind adopting the selected bilateral filtering approach in the developed algorithm is to combine a geometric constraint (i.e., a spatial filter) with color proximity constraint (i.e., a range filter). The performance of the proposed approach was tested for noise and accuracy using the Middlebury dataset. Another refinement technique was proposed by Psota et al. [78] which does not use image segmentation or plane fitting. Instead, the algorithm performs iterative refinement of the results of adaptive weight stereo matching. In each iteration of disparity refinement, the algorithm uses the ASW approach to penalize disparity differences in local windows. A total of eight iterations on the Middlebury dataset were performed by Psota et al. and the correspondence error percentage was observed to decrease from 1.46% to 0.83%. A new technique for local cost aggregation for stereo matching was proposed by Yang [79]. In this technique, the matching cost values are aggregated adaptively based on a tree structure. The nodes of this tree consist of all of the image pixels and the tree contains all edges between nearest neighboring pixels. A spanning tree can be computed by removing unwanted edges. Edge with high weights will be removed during spanning tree construction. Then, the minimum spanning tree MST approach is applied to obtain the sum of the minimum values of all spanning trees. Yang's method offers a low computational complexity and high accuracy but has not been tested for use in real time implementation.

Xu et al. [80] proposed an algorithm that calculates the aggregation cost via the join optimization of both the left and right matching costs. The authors assign reasonable weighting coefficient and exclude occluded pixels, while preserving sufficient support windows for accurate matching. The result is the ability to reduce unwanted pixels in the foreground and increase accuracy in highly textured regions. Furthermore, Lee et al. [60] developed an algorithm based on local approach with no iteration using three-mode cross CT with noise buffering to increase the robustness against image noise in textureless areas. This technique also provides two bits of cross CT within three modes of implementation to increase the reliability of the census measure. Most disparity methods encounter difficulties when confronted with fast-moving objects, but Lee's algorithm addresses this problem by using the concept of optical flow to support weight

TABLE 3: Previous evaluation papers on CPU, FPGA, and GPU.

Year	Author	Focus
2008	Gac et al. [41]	Presented a performance evaluation of different target architectures which are FPGA (Xilinx Virtex 2 Pro), GPU (Nvidia Geforce 8880) and CPU (Xeon dual core 3 GHz) performance based on back projection technique with several ways to speed up. They have reduced the computational time for these three different architectures through memory parallelized architecture.
2010	Kalarot and Morris [87]	Reviewed performance on FPGA (Altera Stratix III) and GPU (Nvidia Geforce GTX 280) performance based on strengths and limitations. The same DP algorithm has been applied to FPGA and GPU. The evaluations have been made on internal clocks, memory space, and disparity range of 128 and 256.
2012	Pauwels et al. [88]	Reviewed performance on FPGA (Xilinx Virtex 4) and GPU (Nvidia Geforce GTX 7900) performance based on optical flow, stereo vision, and local image features including energy, orientation, and phase.
2012	Russo et al. [90]	Presented performance comparison on FPGA (cyclone II) and GPU (Nvidia Geforce GTX 295) for image convolution processing. The performance is measured through clock cycle ratio and execution time.
2013	Fowers et al. [91]	Presented performance comparison on FPGA (Altera Stratix III), GPU (Nvidia Geforce GTX 295) and CPU (Xeon Quad Core W3520) for sliding window applications. They have used SAD algorithm for all three architectures. The performance comparison is based on energy efficiency and time consuming for image processing. Multiple windows sizes (4×4, 9×9, 16×16, 25×25, and 45×45) have been used to evaluate the best performance architecture.
2014	Xu et al. [89]	Presented the speed performance of CPU (AMD Opteron processor 6366HE, Intel Xeon processor E5-2620) and GPU (Nvidia GeForce GTX 770) on the pyramidal stereo algorithm.

computations within a localized window. Koo et al. [81] used a gradient based matching technique to reduce the radiometric errors and improve the matching cost function by using a Gaussian-based weighting function. The reference image is divided into two difference images corresponding to low and high frequencies. Then, the Difference of Gaussian (DoG) function is employed to reduce the errors that arise during the matching process. The authors demonstrated a reduction of errors on a sample set of images acquired in an outdoor environment. Matsuo et al. [82] used a local approach based on the AD algorithm and the Sobel operator in the matching cost calculation stage and box filtering with the WTA optimization in the cost aggregation stage. They used a weighted joint bilateral filter (JBF) in the refinement stage. They produced accurate disparity maps using several iterations and a fixed window size for the JBF. Nalpantidis and Gasteratos [83] developed a new stereo matching algorithm that employs the AD algorithm and performs aggregation by considering the gestalt laws of proximity, similarity, and continuity within a psychophysically based weight assignment framework. Their proposed algorithm yielded accurate results when applied to the Middlebury dataset.

5. Real Time Stereo Vision Disparity Map Algorithms Using Additional Hardware

The ability to implement stereo matching algorithms in real time represents a new research area in the field of computer vision. The results of the online Middlebury benchmarking system established by Scharstein and Szeliski [35] indicate that algorithms developed using a parallel processing approach or additional hardware are able to deliver processing times among the best ones achieved on a standard benchmarking dataset. Real time stereo vision algorithms

are able to achieve rates of greater than 30 frames per second in their disparity mapping output. In this section, the discussion is limited to platforms that use FPGAs and GPUs for real time implementation of stereo vision algorithms. Figure 5 shows a basic block diagram for such hardware-based implementation on FPGA or GPU. The FPGA consists of a CPU, a multi-input/output port and a large set of configurable logic blocks (CLB) that can be configured according to the developer's preferred design. The features of FPGA include versatility and the flexibility to operate either as standalone systems or as coprocessors on expansion cards for computer. FPGAs are most often programmed using hardware description language (HDL) [84] and Bacon et al. [85]. By contrast, a GPU is a dedicated coprocessor with a fixed architecture that enables the acceleration of the rendering of 2D and 3D graphics by offloading the related processes from the CPU. Recent GPU designs have evolved from being dedicated graphics rendering processors to more general parallel processors. A GPU consists of multiple processors with more than hundred cores, depending on the model. A GPU is able to operate in combination with CPU and open-source libraries such as the OpenCL, OpenCV, and Compute Unified Device Architecture (CUDA) libraries [86].

5.1. Comparative Studies of FPGAs and GPUs. Several papers related to performance evaluations of FPGAs, GPUs, and CPUs were reviewed for this literature survey as specified in Table 3. An early comparison among the FPGA, GPU, and CPU was conducted by Gac et al. [41]. The results indicated that the GPU achieved the highest absolute performance in terms of reconstruction time. The authors applied the back projection technique in their global algorithm via 3D tomography image reconstruction. Kalarot and Morris [87] compared the performance of the DP algorithm implemented

on the FPGA and GPU when applied to their own rectified images for different disparity range. Their results indicated that the FPGA offered faster processing than the GPU for a disparity range below 128 but that the FPGA was unable to handle a disparity range of greater than 256, unlike the GPU. This finding can be attributed to the memory limitations of standalone FPGA systems, which prevent their use for processing large images. Another structured evaluation and comparison of the FPGA and GPU was performed by Pauwels et al. [88] in the context of a real time analysis of optical flow, local image features, and stereo vision applications. The authors applied their method to the Middlebury dataset. The comparison was performed based on the hardware architecture, speed, data dependency, accuracy, and time required to design the structure of the algorithms. The presented results demonstrated that the GPU implementation was superior in all respects and yielded more accurate and faster results when implemented as real time stereo vision systems compared with the FPGA implementation. Xu et al. [89] compared the speed performances achieved using CPU and a GPU for their pyramidal stereo algorithm. Their results indicated that the GPU was 12x faster than the CPU.

Russo et al. [90] performed a performance comparison of FPGA and a GPU for image convolution processing. They reported that the GPU exhibited the better performance in terms of execution time and number of clock cycles. This finding was attributed to the fact that the GPU tended to be better able to explore the extremely large amounts of data contained within the high-resolution images. Moreover, the characteristic features of a GPU, such as multiple pipelines and high bandwidth, assist in enhancing its performance. Fowers et al. [91] conducted a performance and energy comparison of a GPU, FPGA, and a CPU with a multicore architecture. They used the SAD algorithm with the sliding window technique as the algorithm implemented for the comparison. It was found that the FPGA provided the best energy efficiency, whereas the GPU delivered the best performance. Jin and Maruyama [92] compared implementation of their algorithm on the FPGA and a GPU based on speed and accuracy. They also proposed a method of improving the circuit design for FPGAs to reduce the required memory resources while maintaining accuracy. However, the results reported in all of the reviewed papers are subject to processing performance, which depend on the available hardware resources and the computational requirements of the considered task. A recent study of FPGAs was conducted by Lentaris et al. [93] for their ongoing projects SPARTAN, SEXTANT, and COMPASS to improve the behaviour of autonomous planetary exploration rovers. The study focused on the potential use of FPGAs for implementing a variety of stereo correspondence, feature extraction, and visual odometry algorithms.

5.2. Global Approaches. Several global methods have been implemented on FPGAs and GPUs for the development of real time stereo vision disparity map algorithms. A global optimization algorithm for stereo matching based on improvement to the BP approach implemented on a GPU was presented by Xiang et al. [9]. Their technique involves the integration of color-weighted correlations to improve hierarchical BP. Occlusion problems are resolved by combining a uniqueness constraint and a similarity constraint for the detection of occluded regions. The approach of Xiang et al. outperforms other BP methods with regard to their real time implementation on GPUs. However, its results in discontinuous regions of the disparity maps are somewhat poor and it requires more complex computations. The approach also suffers an increased time delay when the algorithm is attempting to generate accurate results for such discontinuous regions. An improvement to the poor quality of the disparity maps was achieved by Wang et al. [94]. They used the AD-CT algorithm in the matching cost calculation stage with a semiglobal optimization framework on FPGA board. Semiglobal optimization involves optimizing the smoothness of the disparity map along different directions separately. The designated directions are along lines traveling to the right, bottom, bottom right, and bottom left. This system was found to be able to adjust the image resolution and degree of parallelism to achieve maximum efficiency. The result was the ability to produce high quality disparity maps from high-definition images. In [95], a new cost function was developed for the matching of corresponding pixels. The authors proposed a parallel approach to a variant of a global matching cost calculation method implemented on a GPU for symmetric stereo images. A bank of log-Gabor wavelets was developed for the analysis of such symmetric images in the spectral domain. Using a GPU, the authors achieved real time disparity estimations for high-resolution images.

Implementation of a new algorithm using the GC method on a GPU was presented by Choi and Park [96]. They built their algorithm to operate in three stages using the graph construction method to accelerate the convergence of the GC calculation. A reordering heuristic and initialization method were employed to further increase the execution speed based on the proposed graph construction method. Then, a repetitive block based push and relabel method was used to increase the data transfer efficiency. Finally, they used low-overhead global relabelling algorithm to increase the GPU occupancy. They achieved an improved execution time compared with typical global methods at the cost of considerable programming efforts. Yao et al. [97] treated image warping as an energy minimization problem. First, they developed a sparse disparity map by means of stereo matching process. Then, the map was warped an energy minimization function with three independent terms (i.e., a disparity constraint, a structure constraint, and a temporal constraint). They applied their method to test images with different resolutions and evaluated the results that are based on the execution times required for the GPU and CPU implementation. The GPU runtime was 24x faster than the CPU processing time, satisfying the requirements of real time operation.

5.3. Local Approach. Current FPGA technology offers thousands of small logic blocks embedded in the connection matrix. This allows arbitrary computation blocks to be constructed from basic computing blocks through parallel circuit connections. Detailed summary of information regarding

the advantages and disadvantages of real time implementation of stereo vision algorithms on FPGAs has been provided by Samarawickrama [98]. Kalarot and Morris [87] implemented an algorithm on FPGA using a fast and simple approach by combining the distortion removal and alignment correction tasks in a single step by means of lookup tables. However, a problem was encountered in the case of images of more than 1 megapixel in size, at which the FPGA was unable to process because of its very limited onboard memory. This memory limitation makes FPGAs unsuitable for the processing of high-definition images unless external memory is used to support it. Mattoccia [99] performed a comparison of three different algorithms, namely, a fixed window algorithm, an ASW algorithm, and a semiglobal algorithm implemented on FPGAs. Mattoccia's results demonstrated that the output disparity maps were fairly accurate for all tested algorithms. Recently, Colodro-Conde et al. [100] implemented an area based stereo matching algorithm on FPGA board and tested its performance using the Middlebury dataset. They developed the algorithm to use the SAD approach for the matching cost calculation and the median filter to refine the disparity maps. Their architecture design involved multiple buffers for temporary memory storage. In this design, when the window size is increased, the buffers also need to be increased for the parallel processing of the allocated memory. However, the memory size and inherent frequency of FPGA limit its suitability for such tasks and applications, especially for real time applications. Nalpantidis et al. [20] used FPGA to prepare an efficient implementation of their hierarchical matching algorithm on uncalibrated stereo vision images. In their approach, two-dimensional correspondence search is performed using a hierarchical technique. Then, the intermediate results are refined by three-dimensional cellular automata (CA). The final disparity value is defined in terms of the distance between the matching positions. This proposed algorithm is able to process uncalibrated and nonrectified stereo images when implemented on the FPGA.

Excessive time consumption is the main challenge facing real time algorithm implementation because of their computational complexity. The reasons that real time vision algorithms are generally suitable for implementation on GPUs have been explained and discussed by Kim et al. [107]. A GPU unit is able to run the same instructions on multiple sets of data simultaneously. Based on the functionality, Mei et al. [45] developed a stereo matching algorithm for implementation on a GPU with good performance in terms of both accuracy and speed. The matching cost value was initialized using the AD measure and CT. The cost was aggregated in dynamic cross based regions and updated in a multidirection scan line optimization. Several researchers have developed algorithms based on domain transformations. This technique was previously initiated by Gastal and Oliveira [108], who used a transformation technique that enables the aggregation of 2D cost data using a sequence of 1D filters. This technique was improved upon by Pham and Jeon [24] by means of dimensionality reduction technique. The advantage of this technique is that it reduces the complexity of the computational requirements compared with a 2D cost aggregation calculation. A multiresolution anisotropic

diffusion approach based on a disparity refinement algorithm that can be executed in a real time environment was proposed by Vijayanagar et al. [65]. This algorithm exploits the image pyramid concept to gradually enhance the disparity map at different levels of resolution and to align the object boundaries in color images. This technique allows smoothing to be achieved without loss of edges, making it a useful tool for improving image segmentation.

A novel local method for stereo matching using a GPU was presented by Kowalczuk et al. [104]. The algorithm begins with an approximation based on ASW aggregation and a low-complexity iterative disparity refinement technique. The probabilistic framework combines the summation term into a matching cost minimization via a series of approximations and facilitates interactive processing to improve the accuracy of the disparity map. The refinement algorithm operates by calculating the estimated disparity value of each pixel during the current iteration using nearby pixel disparities from previous iterations. The implementation of this method of cost aggregation and iterative disparity refinement was performed by Yoon and Kweon [102]. Instead of searching for the matching window with the optimal size and shape, it is possible to aggregate costs after local smoothing within a corresponding window to reduce matching noise. Usually noise can be effectively reduced by applying a linear filter such as a Gaussian filter, but the resulting disparity map always exhibits edge fattening. Therefore, to address this problem of mismatching pixels or noise around regions of discontinuity disparity maps, Lin et al. [105] proposed a new algorithm based on an edge preserving filter for the ASW method computed using a hierarchical clustering algorithm. This algorithm used a novel cost aggregation block to compute corresponding response for all the corresponding pixels in a set of sampling points. Tippetts et al. [106] used an intensity profile shape-matching algorithm implemented on an FPGA to achieve real time estimations for microscale unmanned vehicles (i.e., helicopter). The algorithm consists of three steps. In the first stage, filtering is performed using a Gaussian kernel. Then, the shapes of target objects are identified on a row-by-row basis. In the last stage, after the entire image has been processed, a vertical smoothing filter is applied to reduce the remaining noise. The authors also presented designs for FPGA blocks for each stage of implementation.

5.4. The Challenges of Implementing Stereo Vision Algorithms on GPUs and FPGAs. Over the past decade, developments in computing architectures have exhibited a clear trend toward increased heterogeneity and parallelism, with most mainstream microprocessors now possessing multiple cores and robust system architectures [109]. At the same time, the increasing number of accelerator options has considerably increased the complexity of application design because of the need to perform an extensive exploration of the available design space when attempting to choose a suitable device. Although GPUs with CUDA have come into common use as accelerators because of their low cost, ready availability, and simple programming model comparable to that of FPGAs, Ekstrand et al. [110], Perez-Patricio et al. [111], Stein [112], and Long et al. [113] have all presented results that different

Left image   Right image   Ground-truth image
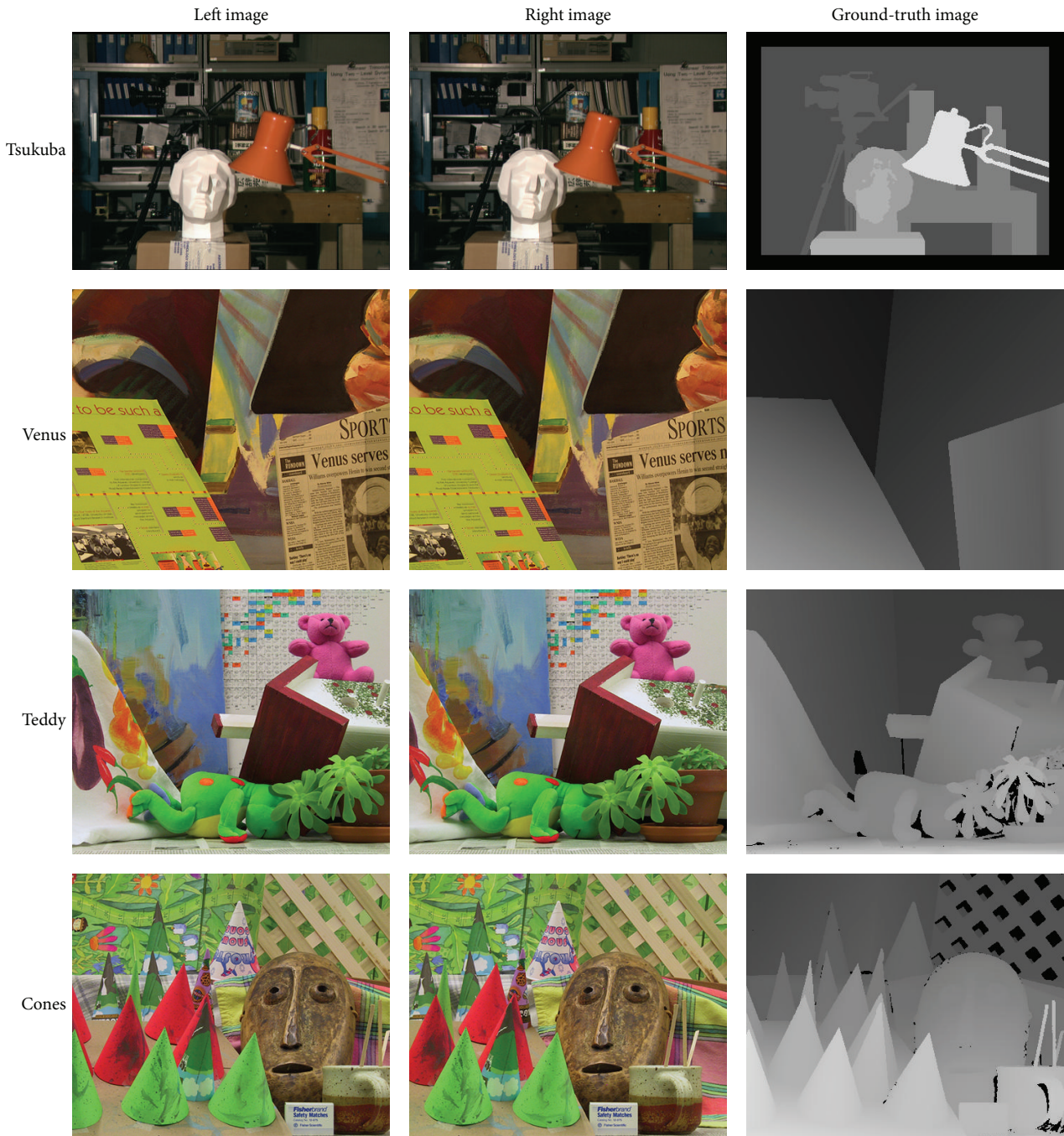
Tsukuba

Venus

Teddy

Cones

FIGURE 6: Standard benchmarking images provided in the Middlebury Stereo Vision datasets in [35].

devices are better suited for different applications. Therefore, sufficient exploration of the different available devices for different applications is critical to prevent researchers from selecting unsuitable devices during the design phase. In this survey, for the summary of performance comparisons between GPUs and FPGAs, the results vary among different implementation and application domains. None of the platforms appears to be universally superior. The preferred design depends on the specifications of the target platform. However, the use of GPUs and FPGAs can facilitate increased

speeds and reduced execution times. The challenge for a new researcher in this field is to determine how to develop an algorithm that is appropriate to a specific application and the most suitable platform.

6. Accuracy Measurement

There are several academic research centers that provide qualitative accuracy assessments of disparity maps through online submissions. The datasets used for these assessments
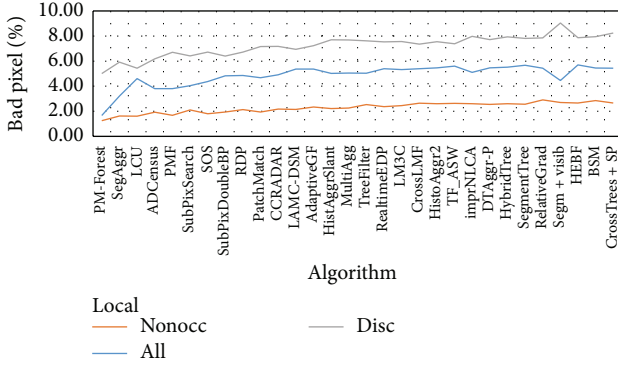
FIGURE 7: The performance of local algorithms in terms of bad pixel percentages for the nonocc, all, and disc which are cited from [35].

FIGURE 8: The performance of global algorithms in terms of bad pixel percentages for the nonocc, all, and disc which are cited from [35].

can also be downloaded from the associated web pages, for example, the Middlebury Computer Vision pages [35], KITTI Vision Benchmark [114], and DIBRIS [115]. These datasets include both static and dynamic scenes. Most of the articles reviewed in this survey use the qualitative accuracy measurements provided by [35]. Thus, this paper uses the same resource to report the accuracy performance of a stereo vision algorithm in terms of the percentage of bad pixels. According to Scharstein and Szeliski [11], the evaluation of the accuracy level for each image is based on three attributes (i.e., the percentages of bad pixels among all pixels in nonoccluded regions (nonocc), all pixels detected as valid pixels (all), and pixels in regions near depth discontinuities and occluded regions (disc)). Four standard benchmarking images are used in this evaluation; these images are Tsukuba, Venus, Teddy, and Cones and the original images and ground-truth images for each are shown in Figure 6. Figures 7 and 8 show the nonocc, all, and disc results obtained on these images for approximately 60 algorithms selected based on both local and global optimizations methods. The bad pixel percentages of these algorithms are among the lowest values represented in the database of [35].

Because this section discusses only accuracy measurements cited from [35] which are based on online submissions, the implementation of the algorithms in Figures 7 and 8 is not specified as either software-based or hardware-based. There is some possibility that accuracy improvement can be achieved through implementation on additional hardware as shown by Pauwels et al. [88]. However, as reported by Kalarot and Morris [87], the primary advantage of hardware-based (i.e., FPGA and GPU) implementation is that the speed or execution time can be tremendously improved compared with implementation using only a CPU. Figure 9 shows the average errors of local and global methods. Here, the algorithms are represented by numbers, which correspond to the algorithms represented at the same x-axis positions in Figure 7 (i.e., for local methods) and 8 (i.e., for global methods). This figure shows only the accuracy performances of existing methods and is intended as a guidance or reference for those who wish to develop their own algorithms.
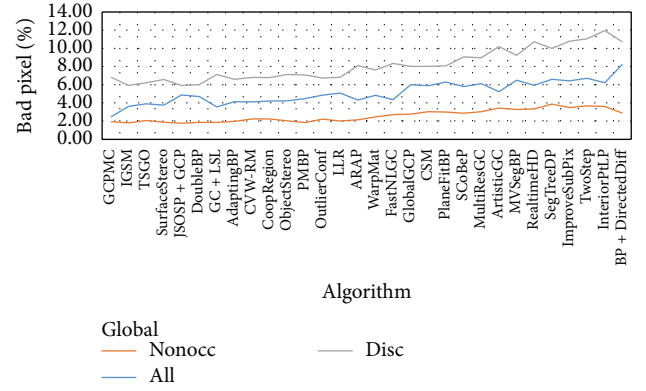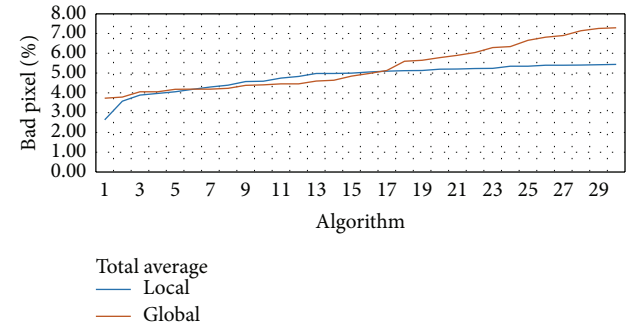

FIGURE 9: The average errors in terms of bad pixel percentage for local and global methods.

7. Conclusion

The stereo matching problem remains a challenge for computer vision researchers. A literature survey of the latest stereo vision disparity map algorithms is provided here and all cited algorithms are categorized according to the processing steps with which they are associated in the taxonomy of Scharstein and Szeliski. Becoming familiar with the state-of-the-art algorithms for stereo vision disparity mapping is a time consuming task. In this survey of the latest developments in the area of stereo matching algorithms, the processing steps composing such an algorithm and their software-based as well as hardware-based implementation was therefore performed and presented to assist in this task. The qualitative measurement of the accuracy of such algorithms was also discussed. To assist the reader in navigating the numerous works presented, Table 2 is presented as a summary. It specifies the steps and computational platforms used in each approach as a reference for the development of new algorithms.

Nomenclature

I_l: Left image
I_r: Right image
d: Disparity value
(x, y): Pixel coordinates

w: Support window

ϵ: Element of

(i, j):   Neighboring pixel coordinates

w_p: Support window centered on pixel p

p: Pixel of interest

d_p: Smallest pixel value

D: Set of disparity values

β: Weighting factor

E: Energy function

q: Neighboring pixels in the support window.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

[2] B. H. Bodkin, *Real-time mobile stereo vision [M.S. thesis]*, University of Tennessee, 2012.

[3] M. Hansard, S. Lee, O. Choi, and R. P. Horaud, *Time-of-Flight Cameras: Principles, Methods and Applications*, Springer, 2012.

[4] S. Foix, G. Alenyà, and C. Torras, "Lock-in time-of-flight (ToF) cameras: a survey," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, 2011.

[5] M. Y. Kim, S. M. Ayaz, J. Park, and Y. Roh, "Adaptive 3D sensing system based on variable magnification using stereo vision and structured light," *Optics and Lasers in Engineering*, vol. 55, pp. 113–127, 2014.

[6] S. Zhang, C. Wang, and S. C. Chan, "A new high resolution depth map estimation system using stereo vision and depth sensing device," in *Proceedings of the IEEE 9th International Colloquium on Signal Processing and Its Applications (CSPA '13)*, pp. 49–53, IEEE, Kuala Lumpur, Malaysia, March 2013.

[7] W. Kazmi, S. Foix, G. Alenyà, and H. J. Andersen, "Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: analysis and comparison," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 128–146, 2014.

[8] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *Journal of Real-Time Image Processing*, pp. 1–21, 2013.

[9] X. Xiang, M. Zhang, G. Li, Y. He, and Z. Pan, "Real-time stereo matching based on fast belief propagation," *Machine Vision and Applications*, vol. 23, no. 6, pp. 1219–1227, 2012.

[10] L. Nalpantidis and A. Gasteratos, "Stereovision-based fuzzy obstacle avoidance method," *International Journal of Humanoid Robotics*, vol. 8, no. 1, pp. 169–183, 2011.

[11] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 7–42, 2002.

[12] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: from software to hardware," *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[13] S. Mattoccia, S. Giardino, and A. Gambini, "Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering," in *Proceedings of the Asian Conference on Computer Vision (ACCV '10)*, vol. 38, pp. 371–380, Xi'an, China, September 2010.

[14] Á. Arranz, Á. Sánchez, and M. Alvar, "Multiresolution energy minimisation framework for stereo matching," *IET Computer Vision*, vol. 6, no. 5, pp. 425–434, 2012.

[15] L. Xu, O. C. Au, W. Sun et al., "Stereo matching by adaptive weighting selection based cost aggregation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '13)*, pp. 1420–1423, Beijing, China, May 2013.

[16] J. Salmen, M. Schlipsing, J. Edelbrunner, S. Hegemann, and S. Luke, "Real-time stereo vision: making more out of dynamic programming," in *Computer Analysis of Images and Patterns*, pp. 1096–1103, Springer, Berlin, Germany, 2009.

[17] J. M. Pérez and P. Sánchez, "Real-time stereo matching using memory-efficient Belief Propagation for high-definition 3D telepresence systems," *Pattern Recognition Letters*, vol. 32, no. 16, pp. 2250–2253, 2011.

[18] Y.-C. Wang, C.-P. Tung, and P.-C. Chung, "Efficient disparity estimation using hierarchical bilateral disparity structure based graph cut algorithm with a foreground boundary refinement mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 5, pp. 784–801, 2013.

[19] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, 2003.

[20] L. Nalpantidis, A. Amanatiadis, G. C. Sirakoulis, and A. Gasteratos, "Efficient hierarchical matching algorithm for processing uncalibrated stereo vision images and its hardware architecture," *IET Image Processing*, vol. 5, no. 5, pp. 481–492, 2011.

[21] T. Song, B. Tang, M. Zhao, and L. Deng, "An accurate 3-D fire location method based on sub-pixel edge detection and non-parametric stereo matching," *Measurement*, vol. 50, no. 1, pp. 160–171, 2014.

[22] L. Wang, M. Gong, M. Gong, and R. Yang, "How far can we go with local optimization in real-time stereo matching," in *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 129–136, Chapel Hill, NC, USA, June 2006.

[23] D. Min, J. Lu, and M. N. Do, "A revisit to cost aggregation in stereo matching: how far can we reduce its computational redundancy?" in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1567–1574, Barcelona, Spain, November 2011.

[24] C. C. Pham and J. W. Jeon, "Domain transformation-based efficient cost aggregation for local stereo matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1119–1130, 2013.

[25] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[26] A. Miron, S. Ainouz, A. Rogozan, and A. Bensrhair, "A robust cost function for stereo matching of road scenes," *Pattern Recognition Letters*, vol. 38, no. 1, pp. 70–77, 2014.

[27] H. Huang and Q. Wang, "A region and feature-based matching algorithm for dynamic object recognition," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '10)*, pp. 735–739, Xiamen, China, October 2010.

[28] K. Sharma, K. Jeong, and S.-G. Kim, "Vision based autonomous vehicle navigation with self-organizing map feature matching technique," in *Proceedings of the International Conference on Control, Automation and Systems*, pp. 946–949, Gyeonggi-do, Republic of Korea, October 2011.

[29] L. Wang, *Novel dense stereo algorithms for high-quality depth estimation from images [Theses and Dissertations—Computer Science]*, University of Kentucky, 2012.

[30] J. Liu, X. Sang, C. Jia, N. Guo, Y. Liu, and G. Shi, "Efficient stereo matching algorithm with edge-detecting," in *Optoelectronic Imaging and Multimedia Technology III*, vol. 9273 of *Proceedings of SPIE*, p. 7, November 2014.

[31] F. Ekstrand, C. Ahlberg, M. Ekström, and G. Spampinato, "High-speed segmentation-driven high-resolution matching," in *Seventh International Conference on Machine Vision (ICMV 2014)*, p. 94451Y, International Society for Optics and Photonics, 2015.

[32] B. J. Tippetts, D.-J. Lee, J. K. Archibald, and K. D. Lillywhite, "Dense disparity real-time stereo vision algorithm for resource-limited systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1547–1555, 2011.

[33] S. H. Lee and S. Sharma, "Real-time disparity estimation algorithm for stereo camera systems," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1018–1026, 2011.

[34] R. K. Gupta and S.-Y. Cho, "Window-based approach for fast stereo correspondence," *IET Computer Vision*, vol. 7, no. 2, pp. 123–134, 2013.

[35] D. Scharstein and R. Szeliski, *Middlebury Stereo Evaluation—Version 2*, 2015.

[36] A. Fusiello, U. Castellani, and V. Murino, "Relaxing symmetric multiple windows stereo using Markov Random Fields," in *Computer Vision and Pattern Recognition*, vol. 2134 of *Lecture Notes in Computer Science*, pp. 91–105, Springer, 2001.

[37] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I/211–I/217, Madison, Wis, USA., June 2003.

[38] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

[39] S. Satoh, "Simple low-dimensional features approximating NCC-based image matching," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1902–1911, 2011.

[40] F. Cheng, H. Zhang, D. Yuan, and M. Sun, "Stereo matching by using the global edge constraint," *Neurocomputing*, vol. 131, pp. 217–226, 2014.

[41] N. Gac, S. Mancini, M. Desvignes, and D. Houzet, "High speed 3D tomography on CPU, GPU, and FPGA," *EURASIP Journal on Embedded Systems*, vol. 2008, Article ID 930250, 12 pages, 2008.

[42] G. Zhao, Y. K. Du, and Y. D. Tang, "A new extension of the rank transform for stereo matching," *Advanced Engineering Forum*, vol. 2, pp. 182–187, 2011.

[43] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1180–1202, 2010.

[44] L. Ma, J. Li, J. Ma, and H. Zhang, "A modified census transform based on the neighborhood information for stereo matching algorithm," in *Proceedings of the 7th International Conference on Image and Graphics (ICIG '13)*, pp. 533–538, IEEE, Qingdao, China, July 2013.

[45] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, pp. 467–474, Barcelona, Spain, November 2011.

[46] C. Cigla and A. A. Alatan, "Information permeability for stereo matching," *Signal Processing: Image Communication*, vol. 28, no. 9, pp. 1072–1088, 2013.

[47] N. Zhang, H. Wang, and J. Cr, "A near real-time color stereo matching method for GPU," in *Proceedings of the 3rd International Conference on Advanced Communications and Computation*, pp. 27–32, Lisbon, Portugal, November 2013.

[48] S. Lee, J. H. Lee, J. Lim, and I. H. Suh, "Robust stereo matching using adaptive random walk with restart algorithm," *Image and Vision Computing*, vol. 37, pp. 1–11, 2015.

[49] Q. Yang, P. Ji, D. Li, S. Yao, and M. Zhang, "Fast stereo matching using adaptive guided filtering," *Image and Vision Computing*, vol. 32, no. 3, pp. 202–211, 2014.

[50] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 229–246, 2002.

[51] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. I-556–I-561, IEEE, Madison, Wis, USA, June 2003.

[52] J. Lu, G. Lafruit, and F. Catthoor, "Anisotropic local high-confidence voting for accurate stereo correspondence," in *Image Processing: Algorithms and Systems VI*, vol. 6812 of *Proceedings of SPIE*, pp. 1–12, San Jose, Calif, USA, January 2008.

[53] K. Zhang, J. Lu, Q. Yang, G. Lafruit, R. Lauwereins, and L. Van Gool, "Real-time and accurate stereo: a scalable approach with bitwise fast voting on CUDA," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 7, pp. 867–878, 2011.

[54] K.-H. Chen and C.-L. Su, "Reducing computation complexity for disparity matching," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '13)*, pp. 2916–2919, Beijing, China, May 2013.

[55] J. Fang, A. L. Varbanescu, J. Shen, H. Sips, G. Saygili, and L. Van Der Maaten, "Accelerating cost aggregation for real-time stereo matching," in *Proceedings of the 18th IEEE International Conference on Parallel and Distributed Systems (ICPADS '12)*, pp. 472–481, IEEE, Singapore, December 2012.

[56] D. Chen, M. Ardabilian, and L. Chen, "A fast trilateral filter-based adaptive support weight method for stereo matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 730–743, 2015.

[57] J. Zhang, J.-F. Nezan, M. Pelcat, and J.-G. Cousin, "Real-time GPU-based local stereo matching method," in *Proceedings of the 7th Conference on Design and Architectures for Signal and Image Processing (DASIP '13)*, pp. 209–214, Cagliari, Italy, October 2013.

[58] A. Hosni, M. Bleyer, and M. Gelautz, "Secrets of adaptive support weight techniques for local stereo matching," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 620–632, 2013.

[59] L. Nalpantidis and A. Gasteratos, "Stereo vision for robotic applications in the presence of non-ideal lighting conditions," *Image and Vision Computing*, vol. 28, no. 6, pp. 940–951, 2010.

[60] Z. Lee, J. Juang, and T. Q. Nguyen, "Local disparity estimation with three-moded cross census and advanced support weight," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1855–1864, 2013.

[61] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, L.-G. Chen, and H. H. Chen, "Hardware-efficient belief propagation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 525–537, 2011.

[62] H.-Q. Wang, M. Wu, Y.-B. Zhang, and L. Zhang, "Effective stereo matching using reliable points based graph cut," in *Proceedings of the IEEE International Conference on Visual Communications and Image Processing (VCIP '13)*, pp. 1–6, Kuching, Malaysia, November 2013.

[63] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 492–504, 2009.

[64] Y. S. Heo, K. M. Lee, and S. U. Lee, "Joint depth map and color consistency estimation for stereo images with different illuminations and cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1094–1106, 2013.

[65] K. R. Vijayanagar, M. Loghman, and J. Kim, "Real-time refinement of kinect depth maps using multi-resolution anisotropic diffusion," *Mobile Networks and Applications*, vol. 19, no. 3, pp. 414–425, 2014.

[66] M. Michael, J. Salmen, J. Stallkamp, and M. Schlipsing, "Real-time stereo vision: optimizing semi-global matching," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '13)*, pp. 1197–1202, Gold Coast, Australia, June 2013.

[67] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 49–56, IEEE, Sydney, Australia, December 2013.

[68] A. Banno and K. Ikeuchi, "Disparity map refinement and 3D surface smoothing via directed anisotropic diffusion," *Computer Vision and Image Understanding*, vol. 115, no. 5, pp. 611–619, 2011.

[69] S. Y. Park, S. H. Lee, and N. I. Cho, "Segmentation based disparity estimation using color and depth information," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3275–3278, Singapore, October 2004.

[70] C. Cigla, X. Zabulis, and A. Alatan, "Region-based dense depth extraction from multi-view video," in *Proceedings of the 14th IEEE International Conference on Image Processing (ICIP '07)*, vol. 5, pp. V213–V216, IEEE, San Antonio, Tex, USA, September 2007.

[71] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[72] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground control points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3033–3040, Providence, RI, USA, June 2011.

[73] B.-Y. Chen and B.-C. Lai, "A high performance parallel graph cut optimization for depth estimation," in *Advances in Intelligent Systems and Applications*, vol. 2, pp. 311–320, Springer, 2013.

[74] V. Kolmogorov, P. Monasse, and P. Tan, "Kolmogorov and Zabih's graph cuts stereo matching algorithm," *Image Processing On Line*, vol. 4, pp. 220–251, 2014.

[75] Y. Wang, F. Zhong, Q. Peng, and X. Qin, "Depth map enhancement based on color and depth consistency," *Visual Computer*, vol. 30, no. 10, pp. 1157–1168, 2014.

[76] S. Ploumpis, A. Amanatiadis, and A. Gasteratos, "A stereo matching approach based on particle filters and scattered control landmarks," *Image and Vision Computing*, vol. 38, pp. 13–23, 2015.

[77] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[78] E. T. Psota, J. Kowalczuk, J. Carlson, and L. C. Pérez, "A local iterative refinement method for adaptive support-weight stereo matching," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition*, pp. 271–277, Las Vegas, Nev, USA, July 2011.

[79] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1402–1409, Providence, RI, USA, June 2012.

[80] L. Xu, O. C. Au, W. Sun et al., "Stereo matching by adaptive weighting selection based cost aggregation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '13)*, pp. 1420–1423, IEEE, Beijing, China, May 2013.

[81] J. Koo, Y.-H. Kim, and S. Lee, "Robust stereo matching algorithm for advanced driver assistance system," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '15)*, pp. 268–269, Las Vegas, Nev, USA, January 2015.

[82] T. Matsuo, S. Fujita, N. Fukushima, and Y. Ishibashi, "Efficient edge-awareness propagation via single-map filtering for edge-preserving stereo matching," in *Three-Dimensional Image Processing, Measurement (3DIPM), and Applications*, vol. 9393 of *Proceedings of SPIE*, International Society for Optics and Photonics, San Francisco, Calif, USA, March 2015.

[83] L. Nalpantidis and A. Gasteratos, "Biologically and psychophysically inspired adaptive support weights algorithm for stereo correspondence," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 457–464, 2010.

[84] M. Werner, B. Stabernack, and C. Riechert, "Hardware implementation of a full HD real-time disparity estimation algorithm," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 66–73, 2014.

[85] D. F. Bacon, R. Rabah, and S. Shukla, "FPGA programming for the masses," *Communications of the ACM*, vol. 56, no. 4, pp. 56–63, 2013.

[86] N. Wilt, *The Cuda Handbook: A Comprehensive Guide to Gpu Programming*, Pearson Education, 2013.

[87] R. Kalarot and J. Morris, "Comparison of FPGA and GPU implementations of real-time stereo vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9–15, IEEE, San Francisco, Calif, USA, June 2010.

[88] K. Pauwels, M. Tomasi, J. Díaz Alonso, E. Ros, and M. M. Van Hulle, "A Comparison of FPGA and GPU for real-time phase-based optical flow, stereo, and local image features," *IEEE Transactions on Computers*, vol. 61, no. 7, pp. 999–1012, 2012.

[89] T. Xu, P. Cockshott, and S. Oehler, "Acceleration of stereo-matching on multi-core CPU and GPU," in *Proceedings of the IEEE International Conference on High Performance Computing and Communications*, pp. 108–115, IEEE, Paris, France, August 2014.

[90] L. M. Russo, E. C. Pedrino, E. Kato, and V. O. Roda, "Image convolution processing: a GPU versus FPGA comparison," in *Proceedings of the 8th Southern Programmable Logic Conference (SPL '12)*, pp. 1–6, Bento Goncalves, Brazil, March 2012.

[91] J. Fowers, G. Brown, J. Wernsing, and G. Stitt, "A performance and energy comparison of convolution on GPUs, FPGAs, and multicore processors," *ACM Transactions on Architecture and Code Optimization*, vol. 9, no. 4, article 25, 2013.

[92] M. Jin and T. Maruyama, "Fast and accurate stereo vision system on FPGA," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 7, no. 1, article 3, 2014.

[93] G. Lentaris, I. Stamoulias, D. Diamantopoulos et al., "SPARTAN/SEXTANT/COMPASS: advancing space rover vision via reconfigurable platforms," in *Applied Reconfigurable Computing*, pp. 475–486, Springer, 2015.

[94] W. Wang, J. Yan, N. Xu, Y. Wang, and F.-H. Hsu, "Real-time high-quality stereo vision system in FPGA," in *Proceedings of the 12th International Conference on Field-Programmable Technology (FPT '13)*, pp. 358–361, Kyoto, Japan, December 2013.

[95] V. Mota, G. Falcao, M. Antunes, J. Barreto, and U. Nunes, "Using the GPU for fast symmetry-based dense stereo matching in high resolution images," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '14)*, pp. 7520–7524, Florence, Italy, May 2014.

[96] Y.-K. Choi and I. K. Park, "Efficient GPU-based graph cuts for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '13)*, pp. 642–648, IEEE, Portland, Ore, USA, June 2013.

[97] S.-J. Yao, L.-H. Wang, C.-L. Lin, and M. Zhang, "Real-time stereo to multi-view conversion system based on adaptive meshing," *Journal of Real-Time Image Processing*, pp. 1–19, 2015.

[98] M. G. Samarawickrama, *Performance Evaluation of Vision Algorithms on FPGA*, Universal-Publishers, 2010.

[99] S. Mattoccia, "Stereo vision algorithms for FPGAs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '13)*, pp. 636–641, Portland, Ore, USA, June 2013.

[100] C. Colodro-Conde, F. J. Toledo-Moreo, R. Toledo-Moreo, J. J. Martínez-Álvarez, J. Garrigós Guerrero, and J. M. Ferrández-Vicente, "Evaluation of stereo correspondence algorithms and their implementation on FPGA," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 22–31, 2014.

[101] F. Tombari, F. Gori, and L. Di Stefano, "Evaluation of stereo algorithms for 3D object recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 990–997, IEEE, Barcelona, Spain, November 2011.

[102] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.

[103] Y. Zhao and G. Taubin, "Real-time stereo on GPGPU using progressive multi-resolution adaptive windows," *Image and Vision Computing*, vol. 29, no. 6, pp. 420–432, 2011.

[104] J. Kowalczuk, E. T. Psota, and L. C. Perez, "Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 94–104, 2013.

[105] Y. Lin, N. Lu, X. Lou, F. Zou, Y. Yao, and Z. Du, "Matching cost filtering for dense stereo correspondence," *Mathematical Problems in Engineering*, vol. 2013, Article ID 654139, 11 pages, 2013.

[106] B. Tippetts, D. J. Lee, K. Lillywhite, and J. K. Archibald, "Hardware-efficient design of real-time profile shape matching stereo vision algorithm on FPGA," *International Journal of Reconfigurable Computing*, vol. 2014, Article ID 945926, 12 pages, 2014.

[107] J.-S. Kim, M. Hwangbo, and T. Kanade, "Parallel algorithms to a parallel hardware: designing vision algorithms for a GPU," in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV '09)*, pp. 862–869, Kyoto, Japan, October 2009.

[108] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Transactions on Graphics*, vol. 30, no. 4, article 69, 2011.

[109] J. Fowers, G. Brown, P. Cooke, and G. Stitt, "A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '12)*, pp. 47–56, ACM, Monterey, Calif, USA, February 2012.

[110] F. Ekstrand, C. Ahlberg, M. Ekström, and G. Spampinato, "Towards an embedded real-time high resolution vision system," in *Advances in Visual Computing*, pp. 541–550, Springer, Berlin, Germany, 2014.

[111] M. Perez-Patricio, A. Aguilar-Gonzalez, J. L. Camas-Anzueto, and M. Arias-Estrada, "A fuzzy logic approach for stereo matching suited for real-time processing," *International Journal of Computer Applications*, vol. 113, no. 2, pp. 1–8, 2015.

[112] F. Stein, "The challenge of putting vision algorithms into a car," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '12)*, pp. 89–94, IEEE, Providence, RI, USA, June 2012.

[113] Q. Long, Q. Xie, S. Mita, H. Tehrani, K. Ishimaru, and C. Guo, "Real-time dense disparity estimation based on multi-path viterbi for intelligent vehicle applications," in *Proceedings of the British Machine Vision Conference*, pp. 1–13, BMVA Press, 2014.

[114] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "The KITTI Vision Benchmark Suite," 2015.

[115] M. Chessa, F. Solari, and S. P. Sabatini, "Datasets for disparity and optic flow evaluation," 2015, http://www.pspc.unige.it/.