International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.4 (2014), pp.37-56
http://dx.doi.org/10.14257/ijsip.2014.7.4.04

Software and Hardware Implementations of Stereo Matching

[1]Li Zhou, [2]Tao Sun3, Yuanzhi Zhan and [3]Jia Wang

[1]School of Information Science and Engineering,
Shandong University,
P. R. China
[2]Shandong Provincial Key Laboratory of Network based Intelligent Computing,
University of Jinan,
P. R. China
[3]School of Information Science and Engineering,
Shandong University
P. R. China
[1]e-mail: lillyzju@gmail.com, [2]e-mail: sunnytiger@msn.com

Abstract

Stereo matching is one of the key technologies in stereo vision system due to its ultra high data bandwidth requirement, heavy memory accessing and algorithm complexity. To speed up stereo matching, various algorithms are implemented by different software and hardware processing methods. This paper presents a survey of stereo matching software and hardware implementation research status based on local and global algorithm analysis. Based on different processing platforms, including CPU, DSP, GPU, FPGA and ASIC, analysis are made on software or hardware realization performance, which is represented by frame rate, efficiency represented by MDES, and processing quality represented by error rate. Among them, GPU, FPGA and ASIC implementations are suitable for real-time embedded stereo matching applications, because they are low power consumption, low cost, and have high performance. Finally, further stereo matching optimization technologies are pointed out, including both algorithm and parallelism optimization for data bandwidth reduction and memory storage strategy.

Keywords: *Stereo Matching, GPU, FPGA, ASIC*

1. Introduction

The common ground of stereo vision systems is to model three-dimensional (3D) space and to render 3D objects, using depth information that is the most important element of stereo vision systems [1]. Stereo matching is one of the most active research topics concerning on the depth information processing capability. It is an important stereo vision technique to extract depth or disparity information from stereo images obtained from slightly different viewpoints, by calculating every pixel's depth information from stereoscopic images. It is widely used in different applications, such as stereo & feature tracking [2], industrial informatics [3], free-viewpoint video synthesis [4-5], three-dimensional video processing [6], multi-view video coding [7], intelligent robots [8], autonomous vehicles [9] and medicinal image processing [10]. It is forecasted that in 2015 more than 30% of all High Definition (HD) panels at home will be equipped with 3D capabilities [11].

Stereo matching quality is restricted by real-time processing capability, high computation and algorithm complexity, high processing bandwidth requirement and high algorithm accuracy. Especially for embedded systems, low power consumption, high processing performance, high resolution, and high flexibility are all required, as well as various duration, frequency, viewing distance, screen size, ambient light, *etc.*, Although the stereo matching problems have been extensively studied during the past decades, it is still difficult to automatically predict high quality depth map because of image noise, textureless regions, consistency and occlusions that are inherent in the captured images or video frames.

Because of the reasons mentioned above, stereo matching is still in developing stage. To keep up with consumer electronic development trends, there are two research directions: software optimization and hardware acceleration. In this paper, we present a general comparison survey about software and hardware processing algorithms based on algorithm inherent characteristic, implementations, and architectures. Section II presents an algorithm overview. Section III gives out the software and hardware implementation analysis based on Central Processing Units (CPU), Digital Signal Processors (DSP), Graphic Processing Unit (GPU), Field Programmable Gate Array (FPGA) and Application-Specific Integrated Circuits (ASIC) accelerators. Section IV illustrates the optimization methods. By comparing software and hardware processing method results, section V and VI points out future prospects of stereo matching implementation research direction and conclusion.

2. Stereo Matching Algorithms Overview

In the past two decades, many stereo matching algorithms have been proposed [12]. Categorizes all methods into sparse stereo and dense stereo matching [13]. Categorizes all methods into explicit matching, hand-designed filters and network learning models. The most popular classification till now is global & local method [14].

Global approach defines constrained energy models to resolve disparity maps uncertainties. It can be formulated as an energy minimization problem of a Markov Random Field (MRF), simultaneously considering labeling smoothness. Graphic Cut (GC) and Belief Propagation (BP) are two well-known methods. Five of up-to-date top-10-ranked algorithms (with default error threshold equals to 1) are based on the optimized global energy function [15]. Although global methods can reach a high quality level with VGA@30 frames per second (fps) performance [16-17], it is still hard for real-time and high resolution application cases because of its computation complexity. Local approach is based on color or intensity patterns within a finite window to determine the disparity. It has less computational complexity and acceptable processing quality, and is more preferred by real-time implementations. That is why the up-to-date real-time stereo applications still largely rely on local methods. The main disadvantages of local methods are noisy results in large un-textured regions and foreground fatting issues at object borders.

Besides global and local methods, Semi-Global Matching (SGM) [18] is based on pixel wise matching of Mutual Information (MI). A 2D global smoothness constraint is deduced by combining many 1D constraints. SGM performs an energy minimization in a Dynamic Programming (DP) fashion on multiple 1D path crossing each pixel. Its quality and execution time are in the middle of global and local methods.

2.1. BP Method

BP defines a message passing update process to iteratively refine the belief labels for every pixel. A message sent from one pixel to another is updated according to the neighboring messages and energy functions using the simple arithmetic operations. BP algorithm has calculation matrix regularities, but requires a great amount of memory to store messages. The total message size scale is on the order of O(n×L^2×T), where n is the number of pixels in the image, L is the number of possible labels (disparity range), and T is the iteration count. Basically it takes O(L^2) time to compute each message and there are O(n) messages per iteration. Besides, since each message is processed hundreds of times, the load/store of messages consumes huge memory bandwidth. For example, 1920×1080 image @ 30fps, 300 disparity levels, 8bits every pixel, the BP message storage requirement is about 2.3 GBytes, the processing bandwidth requirement is around 33.9 TB/s for 100 iterations. Even by reducing disparity level to 32, the message passing computation still needs about 8100 billion operations per second. Therefor, message storage, data bandwidth, and computation complexity is the bottleneck of BP algorithm [19]. That is why BP is hard to work on embedded electronic devices, which have limited memory and calculation capability. Different methods are proposed to reduce BP computation complexity and improve processing quality as shown in Table 1. Tree-Re-Weighted message passing (TRW) algorithms [20] is also a message-passing algorithm similar to LBP. The difference is that it computes a lower bound on the energy, and can be implemented using half memory compared with BP.

Table 1. BP Algorithm Overview

Method	Optimization Technique
Traditional (Loopy) BP [32] (2003)	Formulates depth estimation as an energy-minimization problem in NP-hard MRF to obtain MAP estimation
Hierarchical BP (HBP) [33] (2004)	Proposes a min-convolution algorithm and a hierarchical structure reducing computation complexity from O(L^2) to O(L)
Iterative BP [34] (2005)	Achieves an optimal parameter configuration by iteratively estimate disparity map and corresponding MRF parameters
Fast-converging HBP [35] (2006)	Proposes fast-converging BP algorithm to adaptively update pixel cost to reduce iteration cost
EPT BP [36] (2007)	Envelope Point Transform (EPT) BP supports flexible choice of both message compression ratio and smoothness cost
Blocks BP [37] (2007)	Simply splits image into small blocks and performing optimization for each block independently
Max-product BP [38] (2008)	Proposes Iterated Conditional Modes (ICM) method to find a local minimum by a deterministic ”greedy” strategy
Tile-based BP [16] (2009)	Splits MRF into tiles, only stores messages across neighboring tiles, reducing internal message processing traffic
Node-plane BP [17] (2010)	Proposes spinning-message and sliding-bipartite node plane methods to reduce memory storage cost to O(4HWL)
Constant-Space HBP [39] (2010)	Hierarchically reduces disparity search range with coarse-to-fine manner
Generalized BP (GBP) [40] (2012)	Proposes a min-sum messaging scheme, a caching technique and a direction set method to improve message accessibility

2.2. GC Method

GC minimizes pair wise MRF energies by solving min-cut/max-flow problems on graphic constructions. A graph G = (V; E) is given by a set of vertices V and a set of edges E (sometimes, V is called nodes, E is called links). V usually corresponds to pixels or features extracted from an image, while E encodes spatial relationships. GC also has heavy computational complexity and memory requirements. It takes O(L^3) calculation iterations, which increases fast with the number of labels increases. That is why even with the state-of-the-art numerical optimizers, GC is hard to produce in an acceptable real-time processing manner. GC method yields competitive results, and is proved to be able to handle occlusion reasoning well, but its min-cut technique is used to minimize sub-modular energy that is prone to be a partial labeling problem. Some GC based methods are summarized in Table 2.

Table 2. GC Algorithm Overview

Method	Optimization Technique
Traditional GC [41] (1998)	Finds a minimum multi-way cut on a graph, gives out a greedy method for computing a multi-way cut
Local-minimum GC [42] (2001)	Finds a local minimum with respect to expansion moves and swap moves
Plane based GC [43] (2004)	Scene structure is represented as a set of planes in the disparity space
Smoothness GC [44] (2007)	Develops optimal expansion and swap algorithms for truncated convex priors, only two labels for each pixel
ST-mincut GC [45] (2007)	Introduces a st-mincut approach to speed up the inference process with slightly different energy terms
Template based GC [46] (2011)	Non-overlapping templates are derived from reference image to represent current scene with assigned disparity value

2.3. DP Method

DP decomposes a problem into a set of sub-problems, then efficiently solves them recursively. It lies in the middle of the spectrum with reasonably matching performance at the expense of relatively large storage memory. Algorithm complexity is $O(K \times D^2)$, where K is the number of pixels per scan-line, and D is the disparity range. The major problem of DP is that inter-scanline consistency cannot be well enforced, leading to the well-known “streaking” artifacts. Table 3 gives out an overview of various DP based stereo matching methods.

Table 3. DP Algorithm Overview

Method	Optimization Technique
Traditional DP [47] (1985)	Employs inter-scanline search for possible connected edges and intra-scanline search for edge-delimited intervals
Tree-based DP [48] (2005)	Proposes a minimum spanning tree on the adjacency-graph of an over-segmented image instead of individual scan-lines
Two-pass DP [49] (2005)	Employs a two-pass DP combined with generalized ground control scheme, optimizing along and across the scan-lines
Reliability DP (RDP)[50] (2005)	Introduces a reliability DP based on the cost difference between the best alternate path and the path under use
ORDP [51] (2005)	Orthogonal Reliability DP (ORDP) generates semi-dense disparity maps using only two DP passes based on [50]
Adaptive DP [52] (2006)	Introduces an adaptive aggregation step in the vertical direction based on [50]
Tree-based DP (TDP)[53] (2006)	Proposes a tree structure DP, one pixel disparity estimate depends on other tree pixels
Liner Model DP [54] (2007)	Apples truncated linear model to both horizontal and vertical line dependence function
TDP [55] (2009)	Employs geodesic distance transformation for multiple tree construction according to image geodesic distance
Combined DP [56] (2009)	Combines vertical aggregation and DP scheme to produce disparity maps
Multi-resolution DP [57] (2010)	Uses inter scan-line consistency and scene constraints directly into disparity calculation
Rank Transform DP [1] (2011)	Uses rank transform based matching function, and uses adaptive interaction among neighboring disparities
Hybrid DP [58] (2012)	Combines cross-based adaptive window aggregation and basic dynamic programming

2.4. Window Based Matching

Window based stereo matching algorithm belongs to local matching category. It aggregates matching cost over a given support window. Local window should be large enough to include sufficient intensity variation for matching operation, and be small enough to avoid disparity variation inside the window. Properly designed cost function and selected window type are fundamentals of window based stereo matching method. Fixed window, rectangular window [21], multiple windows [22], adaptive weight (AW) [23], and epipolar geometry-based window [24] and adaptive shape window [25] are proposed in the stereo algorithm.

Cost functions designs, such as SAD, rank [26], census [27] transform, are all valid for cost calculation. Rank and census transforms are two nonparametric transforms, depending on the relative order of pixel values rather than the pixel values them. The differences between rank and census method are that rank method counts the number of pixels in a window which is less than the center pixel, while census maps a pixel window to a bit string. Both methods have an algorithmic structure suitable for hardware implementation, and are invariance to certain types of image distortion and noise. Combined with other optimization technique, window based matching algorithms can reach to reasonable quality and performance [23]. computes pixel-wise adaptive support-weights based on proximity and color distances to center pixel. [28] Uses spatio-temporal correlation and temporal variation of the disparity field [29]. Limits search range around the basic line for fast search [30]. Replaces disparity estimation with planes in texture less regions [31]. Introduces partial sum method based on AW to reduce pixels information in a large window.

2.5. Affine Transformation Method

Scale Invariant Feature Transform (SIFT) method [59] can extract distinctive invariant features from images that are invariant to image scale, rotation, 3D viewpoint, noise, illumination change, and match densely pixel-wise SIFT features between two images while preserving spatial discontinuities [60]. Affine-SIFT (ASIFT) [61] can identify features that undergo very large affine distortions. However, the huge amount of computations required by multiple cascaded transformations makes SIFT difficult to achieve real-time performance [62]. Presents an overview of SIFT approaches based on general purpose multi-core processors, customized multi-core processor and FPGA implementation.

Phase Singularity (PS) represents to a point where a complex signal equals to zero. In stereo matching, PS is estimated by convolving images with complex filters. Compared with SIFT method, PS-based approaches have the advantage of robustness against variations in luminance and imbalance between stereo image sensors at the cost of higher computing resource requirements [63, 64]. Combines PS with SIFT, which can get higher repeatability rates.

2.6. Other Intelligence Method

Neural based method can also reach a reasonable stereo matching quality [15]. Its most distinct characteristic is that it does not require matching between left and right elements. Instead, the binocular stimuli with a specific disparity are matched with binocular neurons in the form of neuronal responses. Different neurons with different preferred weights patterns can indicate the spatial left and right receptive fields. Thus, the response of a neuron indicates a matching degree of two receptive fields [13]. Firstly presents a neurotrophic and spatio-temporal regression model of the laminar cortex architecture for stereo that is able to perform stereo disparity detection competitively with sub-pixel precision. In [65], a multi-layer in-place learning network was used to detect binocular disparities. Fuzzy set theory also can be used to deal with stereo matching [66]. Proposes a threshold-based segmentation to build interval-valued fuzzy sets [67]. Adapts biologically Disparity Energy Model to separate stereo populations, then trains these populations, and extracts disparity values in entire scenes.

3. Software and Hardware Processing of Stereo Matching Methods

In the past decades, with stereo consumer application market flourishing, researchers are devoting more efforts to real-time software and hardware system design for stereo matching. Although High Performance Computing (HPC) can provide considerable computational power, there are still many implementation challenges of stereo matching:

- Ultra high computation complexity: searching all pixel candidacies within an area space needs tens of iterations to find the best matching point, calculates all candidacies matching cost with matrix multiplication or addition. Sub-pixel or higher dynamic disparity range case is even worse. These factors lead to ultra high computation complexity cost.
- Ultra high data bandwidth and on-chip SRAM size requirement: are caused by massive temporary data exchanging required by algorithms. However, memory is always the bottleneck for all systems. Optimized data reuse, parallelization, or hierarchical schemes should be researched for memory related issues.
- Real time processing and low power consumption requirement: needs advanced accelerator processing schemes with both power consumption and high processing performance. The real-time and power issue will be existing for a long time because processing requirement is increasingly faster than processing capability.
- Irregular algorithm parameter selection and additional pre-/post-processing steps: needs to be carefully considered to resolve occlusion, inconsistent, or irregular issues in stereo matching algorithm.

To resolve the above requirement and bottlenecks, strives needed in three directions: algorithms optimization, software acceleration and hardware acceleration. Software accelerator evolves in parallel optimization on CPU, DSP, and GPU. Hardware accelerators are based on FPGA or ASIC.

3.1. CPU Implementation

CPU has the highest flexibility for stereo matching algorithms, but it has a limited acceleration for real-time calculation of dense disparity map because CPU has less specific acceleration processing unit. Early research work [50] only achieves non-video rate performance due to limited computing power [68]. Firstly implements software processing with low bit depth motion estimation algorithms for outdoor robot navigation [69]. Proposes a real-time stereo depth extraction method for an intelligent home assistant robot. [70] is able to achieve approximately 17fps on 640 × 480 images with 25 disparity range on a 2 GHz dual-core processor.

3.2. DSP Implementation

DSP has better signal processing capability because of better data processing architectures, lower cost and less power consumption than CPU. Although DSP can reach reasonable stereo matching performance, it has inherent disadvantages, such as data word alignment, bandwidth throughput issue, etc. Consequently, high quality algorithm is seldom realized by a DSP system and is only limited to window based algorithms. With powerful multimedia accelerators, high system clock frequency, optimized cache usage and interconnections between cores, multi-core processor is an effective way to increase stereo matching performance [71]. However, simply increasing the number of processing elements comes with the cost of higher power consumption. In addition, there is no linear relationship between the number of processor cores and the processing performance. As a result, the GPU architecture appeared.

3.3. GPU Implementation

GPU integrates hundreds of extremely powerful computation stream Processing Elements (PE) simultaneously, emphasizing coherent memory access, memory locality, and efficiency of data parallelism, with powerful floating-point arithmetic intensity and high memory bandwidth. For example, nVidia Tesla S1070 contains four GT200 processors, each has 240 PEs, and each processor provides about one Tera Flop (TF) of single-precision throughput over 100 GB/s memory bandwidth. As a comparison, 3.2 GHz Intel Core 2 Extreme can only operate at roughly 51.2 Gega Flops per second (GF/s). Some other GPU processing system, such as AMD FireGL, Qualcomm Snapdragon, and ST Ericsson's U8500 are also widely used in PC or embedded systems.

Software programming models, such as CUDA, OpenGL, and DirectX, are also developed by NVidia/AMD/Intel to assist General-Purpose Computation on GPU (GPGPU) programming for a broader community of application developers. Most stereo matching tasks perform the same computation on a number of pixels in parallel. So GPU stereo matching implementations are drawn much attention and obtained desirable speedup, which is benefited from GPU's hundreds of PEs and high-level software development platform. It is crucial to design application-specific GPU stream kernels and exploit the inherent parallelism of algorithms to adapt GPU's parallel computation core. Tradeoff should be sufficiently considered between accelerating speed and accuracy. With the rapid development of GPU architecture, there should be sufficient potential margin for GPU based stereo matching optimization.

3.4. FPGA/ASIC Implementation

Besides previously discussed software methods, dedicated FPGA/ASIC hardware approaches have more computation capability, and their costs are relatively lower. FPGA or ASIC is very suitable for pixel-wise operations, especially for the intensive complexity computational requirement of stereo matching methods. Differences between FPGA and ASIC implementations are that ASIC implementation needs IC design and implementation technology such as IC foundry United Microelectronics Corporation (UMC) and long development cycles, while FPGA implementation has reconfiguration capability with requiring more power and area consumption than dedicated ASIC chip. Thanks to increased hardware calculation capability, the performance of some FPGA or ASIC implementations are close to real-time processing for high quality stereo algorithms (GC, DP, or BP) with limited image resolution [72]. Gives a hardware processing comparison for FPGA and ASIC design, and presents powerful real-time vision engine in a single chip which integrates optical flow (with 1810 parallel PEs), stereo (with 1145 parallel PEs) and several local image feature extraction methods together FPGA or ASIC implementations have the following features.

- Parallel PE arrays: integrated into a signal chip. Each of the PE works parallel, focusing on specific algorithm execution to speed up the whole algorithm execution.
- Dedicated pipeline architecture: based on corresponding calculation progresses, dedicated pipeline architecture can improve chip clock frequency and execution efficiency, but with higher system design complexity. So there is a trade-off between execution performance and architecture complexity.

- Data reuse and memory allocation technique: heterogeneous processor's bottleneck is data exchanging for both off/on-chip memory. Data reuse strategy is tightly related with memory reuse and allocation method. It is critical to reduce system internal storage size, computation resource, and bandwidth requirement.

From the bandwidth's point of view, a high-end SoC with a fairly wide 128-bit bus can only support about 4GB/s bandwidth even with 50 percent bus utilization and 500MHz bus frequency. That is far below the maximum memory requirement of stereo matching. From processing performance's point of view, parallel data reuse method can increase performance and decrease memory bandwidth requirements, but it increases on-chip SRAM size and system cost. Therefore, the implementation of FPGA or ASIC calls for much attention on smart strategies for the selection between on-chip memory size and memory bandwidth.

4. Software and Hardware Processing Method Comparison

From CPU, DSP, GPU, to FPGA, ASIC, the processing performance increases sequentially, while cost and power consumption decreases correspondingly. Software methods have more flexibility and shorter development cycle, while hardware implementation needs longer design cycle with less design flexibility because of simultaneous consideration of algorithm optimization and hardware mapping issues. From the point view of practicality, hardware stereo processing system should be more acceptable for a real-time stereo vision system because of its lower cost and lower power consumption.

Stereo matching accuracy and speed evaluation are two critical points for stereo matching methods. Accuracy can be evaluated by the error rate, which is the average percentage of bad pixels of four benchmark data sets (Tsukuba, Venus, Teddy, and Cones). Speed is measured by system throughput, mainly including Millions of Disparity Estimations per Second (MDE/s), number of GF/s, and fps. To clearly indicate the differences between algorithms and processing platforms, software and hardware implementation comparisons for BP, DP and local stereo matching algorithms are shown in Table 4-6.

Although GC has higher quality, it is hard to reach the real-time requirement, and is seldom implemented by GPU, FPGA or ASIC accelerator. [104] proposes an early termination rule and prioritizing swap pair search order, and can reach 24.73s for Tsukuba image on Intel Core2Quad Q6600, 4G RAM [105]. Implements a reduced GC method where only some potential values in the disparity range are selected for each pixel, and can reach 83s for Tsukuba image on Intel 3.2 GHz P4 processor, 512 MB RAM.

Based on above comparisons, CPU and DSP are not suitable for real-time embedded applications. GPU, FPGA or ASIC have more advantages compared to CPUs, DSPs for their low power consumption and low cost embedded application systems. BP-based algorithms perform high image quality, but suffer from high computational complexity and memory storage requirement. They are more suitable to be accelerated by hardware. BP searches for an optimal solution of the entire image and requires multiple iterations; however DP uses a single pass to calculate the global optimal solution for each scan-line independently. As a result, DP-based approaches are faster and can generate disparity maps more quickly, but estimation results are prone to error with horizontal streaks in the generated disparity maps because of the difficulties in enforcing inter scan-line consistency by 1-D scan line process. Window based algorithms have higher processing speed because of lower data bandwidth and less computation complexity, but matching errors are higher. For the same algorithm, generally, the processing speed, quality and power consumption of FPGA or ASIC can outperform GPU accelerator generally, as proved by [80-91]. The disadvantages of FPGA or ASIC are longer developing time and less processing flexibility compared with GPU.

Table 4. Software and Hardware Processing Overview of Optimized BP Algorithms

Method (time)	Stereo Algorithm	Image size (Disparity Level)	Error Rate (%)												Technology	Per-formance	
			Tsukuba			Venus			Teddy			Cones					
			non-occ	all	occ	non-occ	all	occ	non-occ	all	occ	Non-occ	all	occ		fps(@MHz)	MDE/s
GPU [33](2004)	HBP	384 × 288(16)	1.9	3.8	10.1	1.2	2.2	15.6	23.1	30.9	33.8	20.6	27.6	29.0			
GPU [35](2006)	Converge-HBP	384 × 288(16)	1.5	3.4	7.9	0.8	1.9	9.0	8.72	13.2	17.2	4.61	11.6	12.4	nVidia GeForce 7900GTX	12.8	22.2
GPU [73](2006)	Multi-scale BP	384 × 288(16)		3.6											nVidia GeForce 6800GT	1.6	
GPU [16](2009)	Tile-based BP	450 × 375(32)													nVidia GeForce 8800GTX	1.68	
GPU [74](2009)	Sub-pixel HBP	584 × 388(5)		10.8			2.02								nVidia 8600M GT	3	
GPU [39](2010)	ConstSpace BP	800 ×600(300)		2.00			1.48			11.1			5.98		nVidia GeForce 8800GTX		
FPGA [75] (2006)	Phase BP	256 × 360(20)	19.6	37.6			10.5	31.5							Xilinx Virtex-4 2000E	30.3	55.2
FPGA [76] (2007)	Mem-opt BP	320 × 240(32)		1.9				0.8							Xilinx VirtexII pro-100	30	94014 LUTs
FPGA [77] (2009)	Truncated BP	1280 ×720(96)													Xilinx Virtex-5 330 VLX	2.5	23709 LUTs
ASIC [16] (2009)	Tile-based BP	640 × 480(32)													UMC 90nm	8.04	69.6 Kgates
ASIC [17] (2010)	Node-plane BP	320 × 240(32)													UMC 90nm		256.6 Kgates

Table 5. Software and Hardware Processing Overview of Optimized DP Algorithms

Method (time)	Stereo Algorithm	Image size (Disparity level)	Error Rate(%)												Technology	Perfo-rmance	
			Tsukuba			Venus			Teddy			Cones					
			non-occ	all	occ	non-occ	all	occ	non-occ	all	occ	Non-occ	all	occ		fps(@MHz)	MDE/s
CPU [78] (2004)	TwoStep DP	1024×1024(16)		2.9			15.6			6.4			25.3		AMD AthlonXP 2800+	6.14	100
GPU [79] (2004)	RealDP	384×288(100)		2.85			6.42								AMD AthlonXP 2800	18.9	209
GPU [51] (2005)	ORDP	320×240(16)		1.4			7.4			2.4			13.5				20.0
GPU [52] (2006)	Adaptive DP	320×240(16)	2.1	4.2	10.6	1.9	3.0	20.3	7.2	14.4	17.6	6.4	13.7	16.5	ATI Radeon XL1800	43	52.8
GPU [48] (2006)	Region-TreeDP	384×288(16)	1.39	1.64	6.85	0.22	0.57	1.93	7.4	11.9	16.8	6.31	11.9	11.8		67.9	
GPU [55] (2009)	TDP	384×288(16)	1.52	2.28	7.53	0.58	0.82	3.06	5.2	8.45	11.6	4.24	9.9	1.52	nVidia GT-9800	5.9	
GPU [57] (2010)	Multi-Resolution	1280×1024(256)													nVidia GTX 295 GPU	16	
GPU [80] (2010)	Symmetric DP	1024×768(128)													nVidia GeForce GTX 280	20	1760
GPU [1] (2011)	Rank DP	384×288(16)	19.6	27.2	34.0							13.5	22.6	22.0		30	1252
GPU [58] (2012)	Hybrid DP	384×288(16)	1.23	3.31	6.23	0.80	1.73	3.79	5.1	12.3	12.8	3.77	11.2	9.25	NVidia Quadro FX 4800	6.3	
FPGA [54](2007)	Trellis DP	320×240(128)		2.63						3.44					Xilinx Virtex II pro-100	30	
FPGA [80](2010)	Symmetric DP	1024×768(128)													Altera Stratix III	30	2600

Table 6. Software and Hardware Processing Overview of Optimized Window Based Algorithms

Method (time)	Stereo Algorithm	Image size (Disparity level)	Error Rate(%)												Technology	Per-formance	
			Tsukuba			Venus			Teddy	Cones							
			non-occ	all	occ	non-occ	all	occ	non-occ	all	occ	Non-occ	all	occ		fps(@MHz)	MDE/s
CPU [81](2008)	Effect-Aggr	463 × 370(75)		2.11			4.75			15.2			12.6		2.14GHz Intel Core	1.67	18.9
DSP [82](2007)	jigsaw	384 × 288(16)	20.4	20.6	47.9	15.3	16.6	29.5	25.1	32.4	34.1	22.9	31.1	30.6	TI C64x DSP	12.5	
GPU [23](2006)	Adaptive		1.4	1.9	6.9	0.7	1.2	6.1	7.9	13.3	18.6	4.0	9.8	8.3	AMD 2700+		
GPU [83](2007)	SAD		2.3	3.6	11.2	3.6	4.6	19.8	10.9	18.8	23.2	5.9	14.3	13.8	ATI Radeon X800	20.3GF/s	124.1
GPU [30](2007)	SAD		1.0	1.8	5.3	0.2	0.5	1.7	6.7	12.1	14.7	4.2	10.7	10.6			9.4
GPU [25](2007)	Adaptive	256 × 256(96)	7.63	13.0	14.3	9.04	19.4	9.20							Nvidia GeForce 7900	87	548
GPU [84](2007)	Laplacian	320 × 240(32)		4.22			2.98			14.4			13.7		ATI Radeon 9800	22	53.0
GPU [85](2007)	Gradient Guided	512 × 384(40)		2.48			3.91								ATI Radeon 9800XT	14.7	117
GPU [86](2009)	cross-based	384 × 288(16)	2.80	4.84	7.29	2.14	3.40	11.5	9.67	16.3	18.8	5.85	13.7	12.1	nVidia GeForce 7900GTX	17	30
GPU [87](2009)	Stream-Centric	450 × 375(60)	4.33	6.06	15.2	5.88	6.87	12.5	13.6	20.6	25.3	6.88	14.9	16.0	nVidia GeForce 8800 GT	100	
GPU [88](2010)	SAD		1.4	1.9	7.1	0.4	1.0	2.7	8.6	15.2	19.5	5.6	12.5	13.2	GTX 8800	350GF/s	144.4
GPU [89](2011)	Scalable	384 × 288(16)	1.64	2.13	6.43	0.57	0.90	2.90	9.70	14.8	19.3	7.00	12.5	13.6	GeForce8800 GTX	64	113.2
GPU [90](2011)	Profile Shape	384 × 288(31)	9.6	11.5		15.1	15.7		3.2	4.8		5.3	6.6		2.8GHz AMD	62	
GPU [91](2012)	SAD	1280 × 1024		241			161			311			381		nVidia GeForce GTX 280	49	
FPGA [21](2003)	SAD	640 × 480(64)													Xilinx 10000 Virtex II	31@10	
FPGA [92](2003)	SAD	640 × 480(80)													Xilinx XC2V6000	18.9@40	
FPGA [93](2005)	CBiased	512 × 512(96)		4.77			10.2								Nvidia Geforce 7900	24	605
FPGA [94](2007)	Phase based	1280 × 960(29)													Xilinx Virtex-II	52@65	1885
FPGA [95](2008)	SAD	320 × 240(64)		28											Nios II Cyclone II	325@100	800
FPGA [96](2010)	Census	640 × 480(64)	9.8	11.6	20.3	3.6	5.3	36.8	12.5	21.5	30.6	7.3	17.6	21.0	Xilinx Virtex-5	60@24.5	4522
FPGA [97](2010)	SAD	750 × 400(60)	5.81	7.14	22.6	2.61	3.33	25.3	9.79	15.5	25.7	5.08	11.5	15.0	AlteraStratixI (133MHz)	60	1080
FPGA [98](2011)	SAD	640 × 480(64)													Altera DE2-70	30@12.2	295
FPGA [99](2012)	SAD	640 × 480(64)	4.48	6.04	12.7	6.01	7.47	18.2	21.5	28.1	28.8	17.1	25.9	25.8	Xilinx Virtex-5LX110T	30@155	589
FPGA [100] (2012)	Combined	1280 × 1024(120)	9.26	10.4	28.2	11.0	12.1	28.9	21.4	29.1	41.3	17	25.3	33.4	Xilinx Virtex-5 LX110T	50	7864
FPGA [91](2012)	SAD	1280 × 1024		481			421			471			661		Xilinx Virtex-5 VLX 330	40@53	
ASIC [101](2004)	SSD-Census	256 × 192(25)													0.25um	50@75	
ASIC [102](2010)	Mini-census	352 × 288(64)		2.80			0.64			13.7			10.1		UMC 0.09um	42@95	272.5
ASIC[103](2012)	SAD	320 × 240(64)													0.18um	144@120	707

5. Future Research Direction

To improve system throughput with better disparity accuracy is still a challenging research topic although a number of near real-time systems which can achieve higher image resolution which have been implemented on GPU, FPGA or ASIC. For future software and hardware implementations, there are several points need to be emphasized:

- Super Resolution (SR) stereo vision processing: data bandwidth and memory storage requirements will be much tighter than current high resolution steam. New optimization methods should be considered during software and hardware optimization processes.
- Real-time and low power consumption requirements: the trend for hand-held stereo vision system with high image resolution, quality, and free-view features. Algorithm parallelization and specific PE design are effective technologies for it.
- Powerful GPU, FPGA and ASIC system optimization: will play more critical roles along with VLSI design technology development. Parallel calculation capability, specific PE, efficient memory allocation method, will bring stereo vision system a revolution in the near future.

To resolve issues including occlusion, image inconsistent, hardware resource limitations, etc., there are several optimization aspects to meet the upcoming stereo matching technology challenges.

5.1. Image Segmentation or Hierarchy Optimization

Segmentation or hierarchy approach has been widely employed in stereo matching to reduce algorithm complexity. It can be divided into three categories: over segmentation, color segmentation and coarse-to-fine layer department. They break the image apart into smaller ones, and then process the reduced images one by one. Color based segment assumes that the neighboring pixels with similar colors have similar depth values. Over segment gets trade-off between color segment and performance. Coarse-to-fine can reduce disparity calculation range by hierarchy search.

We also study the optimization method based on adaptive image segmentation [106]. We take advantages of chrominance component, intelligently comprehends object depth characteristics to pre-determine the inter-prediction block size, instead of selecting the best one after calculating all block sizes' cost function. Ignoring unrelated block size calculation saves computation resources and time. Smaller block size is pre-decided at an object boundary, and larger block size is for consecutive areas. Experiment results show that the method not only is effective for real-time stereo matching, but also performs well on both object boundary and consecutive areas, both block effect and prediction noises are optimized with accelerated prediction speed.

5.2. Occlusion and Consistency Handling

Stereo image discontinuous issues mainly include occlusion and color inconsistency. Occlusion is caused by viewing degree changes between images. Image inconsistency is always caused by various radiometric factors such as luminance changes, illuminate color or imaging device changes. Some optimization methods, such as left-right consistency criterion, interactively estimation, are already in use to identify and remove invalid matches or occlusion pixels. Normalized or unified matching algorithm [107] and Scan-line Optimization (SO) [108] are effective optimization techniques for image inconsistency issues. Depth map mismatches can be suppressed by penalizing large jumps in disparity between the scan-line points, or only dealing with propagated disparities along scan-line directions.

5.3. Matching Cost & Energy Optimization Improvement

A careful selection of cost functions or energy optimization methods is the foundation of local or global stereo matching algorithms [109]. compares a large scale possible combination of matching costs with window based method, SGM, and GC under various radiometric conditions. High order MRF model [110], Quadratic Pseudo-Boolean Optimization (QPBO) approaches [111], or high order strategy [112], Mutual Information (MI) [113], Winner-Take-All (WTA) [115] are all valid for efficient energy optimization. Cost function optimizations have always been used in a cooperative system to reduce computation redundancy.

5.4. Cooperative Optimization

High quality stereo matching results are always involved in combined processing methods. Integration of HBP, mean sift, color segmentation [116], combination of shiftable windows and global energy minimization framework [117], combination of WTA and DP [118], diffuses of matching costs and weights [119], integration of WTA and matching cost [120], combination of GC and SIFT [121], associated matching cost optimization and occlusion handling [114], integrated census transform and hamming distance calculation [91], as well as combined optical flow and feature extraction [122] are all examples of cooperation. Stereo matching precision enhancement, such as sub-pixel interpolation, can provide a dense disparity image, and is also effective for enhancing stereo matching image quality.

5.5. Efficient Memory Arrangement Method

Memory arrangement is critical for hardware performance enhancement because memory accessing is always a bottleneck of current hardware processing architecture. An optimized method is to divide a large memory accessing array into several sub-arrays, and allocate sub-arrays to different memory areas to exploit parallel access capability. The other method is to interchange memory content with various sub-array clusters based on algorithm data parallelism or pipeline architecture.

5.6. Advanced VLSI design Method

The advantages of VLSI chip have already brought high processing capability for stereo matching systems. CPU, GPU, FPGA and ASIC processing capabilities are getting much benefit from VLSI design technology development. Thanks to embedded DDR design technology, 3D VLSI design technology, *etc.*, VLSI chip's processing capability increases fast in recent years. More and more transistors will be integrated in one chip in future VLSI design, which is able to execute more parallel calculations simultaneously. It will bring more powerful calculation capability for both software and hardware realization.

6. Conclusion

In this paper, we summarize both software and hardware implementations for stereo matching algorithms. We analyze typical stereo matching algorithms based on different software and hardware implement techniques, including CPU, DSP, GPU, FPGA, and ASIC. We present the overall evaluation of processing performance, efficiency, and quality. We also point out optimization technologies to reduce computation complexity and increase calculation efficiency. Based on these comparisons, we conclude two points: First, there is optimization potential for both stereo matching algorithm optimization and software or hardware implementation in terms of speed, parallelism, data bandwidth, memory storage, etc. Second, GPU, FPGA and ASIC designs are future research trends in real-time embedded stereo vision application systems because of their high parallel processing capabilities and specific powerful calculation supporting components. GPU has more programming flexibility and powerful computation capability, while FPGA and ASIC have high performance, lower power consumption and cost. Stereo matching system could be enhanced fantastically along with software and hardware technology development.

Acknowledgements

This work was supported by Natural Science Foundation of Shandong Province ZR2013FQ006, Fund of independent innovation in Shandong Province 2013CXB3020, and State Key Laboratory of digital multimedia technology 2013-1-2569, and Shandong post-doctor innovation foundation grant 201002029. The authors would like to thank all research partners for their significant contributions in this work.

References

[1] S. H. Lee and S. Sharma, “Real-Time Disparity Estimation Algorithm for Stereo Camera Systems”, IEEE Transactions on Consumer Electronics, vol. 3, no. 57, **(2011),** pp.1018-1027.

[2] S. S. Hussmann and T. T. Liepert, “Three-dimensional TOF Robot Vision System”, IEEE Trans. Instrum. Meas., vol. 1, no. 58, **(2008),** pp. 141-146.

[3] D. Bruckner and R. Velik, “Behavior Learning in Dwelling Environments with Hidden Markov Models”, IEEE Trans. Ind. Electron., vol. 11, no. 57, **(2010),** pp. 3653-3660.

[4] Y. R. Horng, Y. C. Tseng and T. S. Chang, “VLSI Architecture for Real-Time HD1080p View Synthesis Engine, IEEE Transactions on Circuits and Systems for Video Technology, **(2011)** September 9-21 pp. 1329-1340.

[5] F. Shao, G. Y. Jiang, M. Yu, K. Chen and Y. S. Ho, “Asymmetric Coding of Multi-View Video Plus Depth Based 3-D Video for View Rendering, IEEE Transactions on Multimedia, vol. 1, no. 14, **(2012)**, pp. 157-167.

[6] M. K. Kang and Y. S. Ho, “Depth Video Coding Using Adaptive Geometry Based Intra Prediction for 3-D Video Systems”, IEEE Transactions on Multimedia, vol. 1, no. 14, **(2012),** pp. 121-128.

[7] “Joint Draft 6.0 on Multiview Video Coding”, ISO/IEC JTC1/SC29 and ITU-T SG16 Q.6 JVTZ209, **(2008)** January.

[8] S. Livatino, F. Banno and G. Muscato, “3-D Integration of Robot Vision and Laser Data with Semiautomatic Calibration in Augmented Reality Stereoscopic Visual Interface”, IEEE Transactions on Industrial Informatics, vol. 1, no. 8, **(2012),** pp. 69-77.

[9] D. Schleicher, L. M. Bergasa, M. Ocal, R. Barea and M. E. Liez, “Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion”, IEEE Transaction on Intelligent Transportation Systems, vol. 3, no. 10 **(2009),** pp. 440-452.

[10] S. Zinger, D. Ruijters and D. Luat, “View Interpolation for Medical Images on Auto-stereoscopic Displays”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 1, no. 22, **(2012),** pp. 128-137.

[11] A. Smolic, P. Kauff, S. Knorr, A. Hornung, M. Kunter, M. Muller and M. Lang, “Three-Dimensional Video Postproduction and Processing”, Proceedings of the IEEE, vol. 4, no. 19, **(2011),** pp. 607-626.

[12] T. R. Coffman and A. C. Bovik, “Efficient Stereoscopic Ranging via Stochastic Sampling of Match Quality”, IEEE Transactions on Image Processing, vol. 2, no. 19, **(2010),** pp. 451-461.

[13] M. Solgi and J. Weng, “Developmental Stereo: Emergence of Disparity Preference in Models of the Visual Cortex”, IEEE Transactions on Autonomous Mental Development, vol. 4, no. 1, **(2009),** pp. 238-243.

[14] M. Z. Brown, D. Burschka and G. D. Hager, “Advances in Computational Stereo”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, no. 25, **(2003),** pp. 993-1008.

[15] D. Scharstein and R. Szeliski, “Middlebury Stereo Evaluation: Version 2. [Online]”, Available: http://vision.middlebury.edu/stereo/eval.

[16] C. K. Liang, C. C. Cheng, Y. C. Lai, L. G. Chen and H. H. Chen, “Hardware Efficient Belief Propagation”, Proc. of IEEE Conf. Computer Vision and Pattern Recognition, **(2009)** June 20-25, pp. 80-87, Miami, FL, USA.

[17] Y. C. Tseng and T. S. Chang, “Architecture Design of Belief Propagation for Real-Time Disparity Estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 20, **(2010),** pp. 1555-1565.

[18] A. Haller, C. Pantilie, F. Oniga and S. Nedevschi, “Real-time Semi-global Dense Stereo Solution with Improved Sub-pixel Accuracy”, IEEE Intelligent Vehicles Symposium, **(2010)** June 21-24, pp. 369-376, San Diego, CA.

[19] Y. C. Tseng and T. S. Chang, "Architecture Design of Belief Propagation for Real-Time Disparity Estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 20, **(2010),** pp. 1555-1564.

[20] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 10, no. 28, **(2006),** pp. 1568-1583.

[21] S. Lee, J. Yi and J. Kim, "Real-time Stereo Vision on A Reconfigurable System", Proc. Fifth Int. Workshop on Systems, Architectures, Modeling, and Simulation, **(2005)** July 18-20, pp. 299-307, Samos, Greece.

[22] O. Veksler, "Fast Variable Window for Stereo Correspondence Using Integral Images", Proc. Of IEEE Conf. Computer Vision and Pattern Recognition, **(2003)** June 18-20, pp. 556-561, Madison, WI, USA.

[23] K. Yoon and I. Kweon, "Adaptive Support-weight Approach for Correspondence Search", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 4, no. 28, **(2006),** pp. 650-656.

[24] J. Lu, H. Cai, J. G. Lou and J. Li, "An Epipolar Geometry-based Fast Disparity Estimation Algorithm for Multi-view Image and Video Coding", IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 17, **(2007),** pp.737-750.

[25] L. Jiangbo, G. Lafruit and F. Catthoor, "Fast Variable Center-biased Windowing for High-speed Stereo on Programmable Graphics Hardware", in Proc. IEEE Int. Conf. Image Process., **(2007)** September 16-October 19, pp. 568-571, San Antonio, USA.

[26] J. Banks and M. Bennamoun, "Reliability Analysis of the Rank Transform for Stereo Matching", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 6, no. 31, **(2001),** pp. 870-880.

[27] M. Humenberger, T. Engelke and W. Kubinger, "A Census-based Stereo Vision Algorithm Using Modified Semi-global Matching and Plane-fitting to Improve Matching Quality", 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2010)** June 13-18, pp. 77-84, San Francisco, CA, USA.

[28] W. Zhu, X. Tian, F. Zhou and Y. W. Chen, "Fast Disparity Estimation Using Spatiotemporal Correlation of Disparity Field for Multiview Video Coding", IEEE Transactions on Consumer Electronics, vol. 2, no. 56, **(2010),** pp.957-964.

[29] X. M. Li, D. B. Zhao, S. W. Ma and W. Gao, "Fast Disparity and Motion Estimation Based on Correlations for Multi-view Video Coding", IEEE Trans. Consumer Electron, vol. 4, no. 54, **(2008),** pp. 2037-2044.

[30] Q. Yang, C. Engels and A. Akbarzadeh, "Near Real-time Stereo for Weakly Textured Scenes", Proceedings of the British Machine Vision Conference, **(2008)** September 12-14, Leeds, UK.

[31] W. D. Hu, K. Zhang, L. F. Sun, J. Y. Li, Y. J. Li and S. Q. Yang, "Virtual Support Window for Adaptive-weight Stereo Matching", 2011 IEEE Visual Communications and Image Processing (VCIP), **(2011)** November 6-9, pp. 1-4, Tainan, Taiwan.

[32] J. Sun, N. N. Zheng and H. Y. Shum, "Stereo Matching Using Belief Propagation", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 7, no. 25, **(2003),** pp. 787-800.

[33] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Belief Propagation for Early Vision", Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, vol. 1, no. 70, **(2004),** pp. 261-268.

[34] L. Zhang and S. M. Seitz, "Parameter Estimation for MRF Stereo", Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '05), **(2005)** June 20-25, pp. 288-295, San Diego, CA, USA.

[35] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao and D. Nister, "Real-time Global Stereo Matching Using Hierarchical Belief Propagation", In Proc. British Machine Vision Conference, **(2006)** September 4-7, pp. 989-998, Edinburgh, UK.

[36] T. Yu, R. S. Lin, B. Super and B. Tang, "Efficient Message Representations for Belief Propagation", in Proc. ICCV, **(2007)** October 14-21, pp. 1-8, Rio de Janeiro, Brazil.

[37] Y. C. Tseng, N. Chang and T. S. Chang, "Low Memory Cost Block-based Belief Propagation for Stereo Correspondence", 2007 IEEE International Conference on Multimedia and Expo., **(2007)** July 2-5, pp. 1415-1418, Beijing, China.

[38] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields", IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, no. 30, **(2008),** pp. 1068-1080.

[39] Q. Yang, L. Wang and N. Ahuja, "A Constant-space Belief Propagation Algorithm for Stereo Matching", In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2010)** June 13-18, pp. 1458-1465, San Francisco, CA, USA.

[40] S. Chen and Z. Wang, "Acceleration Strategies in Generalized Belief Propagation", IEEE Transactions on Industrial Informatics, vol. 1, no. 8, **(2012),** pp. 41-48.

[41] Y. Boykov, O. Veksler and R. Zabih, "Markov Random Fields with Efficient Approximations", Proc. IEEE Int. Conf. Pattern Analysis and Machine Intelligence, **(1998)** June 23-25, pp. 648-655, Santa Barbara, CA, USA.

[42] Y. Boykov, O. Veksler and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 23, **(2001),** pp. 1222-1239.

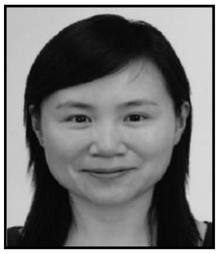
[43] L. Hong and G. Chen, "Segment-Based Stereo Matching Using Graph Cuts", Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, **(2004)** June 27- July 2, pp. 74-81, WASHINGTON, D.C, USA.
[44] O. Veksler, "Graph Cut Based Optimization for MRFs with Truncated Convex Priors", Proc. IEEE Conf. Computer Vision and Pattern Recognition, **(2007)** June 17-22, Minneapolis, MN, USA.
[45] P. Kohli and P. H. Torr, "Dynamic Graph Cuts for Efficient Inference in Markov Random Fields", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 29, **(2007),** pp. 2079-2088.
[46] M. Huan, K. Q. Wang, W. M. Zuo and Z. X. Li, "Template Based Stereo Matching Using Graph-cut", 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, **(2011)** October 21-23, pp. 303-306, Beijing, China.
[47] Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-scanline Search Using Dynamic Programming", IEEE Trans. Pattern Anal. Mach. Intell., vol. 2, no. 7, **(1985),** pp. 139-154.
[48] C. Lei, J. Selzer and Y. H. Yang, "Region-Tree Based Stereo Using Dynamic Programming Optimization", Proc. IEEE Conf. Computer Vision and Pattern Recognition, **(2006)** June 17-22, New York, NY, USA.
[49] J. C. Kim, K. M. Lee, B. T. Choi and S. U. Lee, "A Dense Stereo Matching Using Two-pass Dynamic Programming with Generalized Ground Control Points", Proc. of IEEE Int'l Conf. Computer Vision and Pattern Recognition, **(2005)** Jun. 20-25, pp. 1075-1082, San Diego, CA, USA.
[50] M. Gong and Y. H. Yang, "Fast Unambiguous Stereo Matching Using Reliability-based Dynamic Programming", IEEE Trans. Patt. Anal. Mach. Intell., vol. 6, no. 27, **(2005),** pp. 998-1003.
[51] M. Gong and Y. H. Yang, "Near Real-time Reliable Stereo Matching Using Programmable Graphics Hardware", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2005)** June 20-25, pp. 924-931, San Diego, CA, USA.
[52] L. Wang, M. Liao, M. Gong, R. Yang and D. Nister, "High Quality Real-time Stereo Using Adaptive Cost Aggregation and Dynamic Programming", Proc. of the Third International Symposium on 3D Data Processing, Visualization, and Transmission, **(2006)** June 14-16, pp. 798-805, Chapel Hill, NC, USA.
[53] O. Veksler, "Stereo Correspondence by Dynamic Programming on a Tree", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2005)** June 20-25, pp. 384-390, San Diego, CA, USA.
[54] S. Park and H. Jeong, "Real-time Stereo Vision FPGA Chip with Low Error Rate", Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering, **(2007),** pp. 751-756, Washington, DC, USA.
[55] C. H. Sin, C. M. Cheng, S. H. Lai and S. Y. Yang, "Geodesic Tree-based Dynamic Programming for Fast Stereo Reconstruction", 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), **(2009)** September 27-October 4, pp. 801-807, Kyoto, Japan.
[56] S. Moslah, A. Valles-Such, V. Guitteny, S. Couvet and S. Philipp-Foliguet, "Accelerated Multi-view Stereo Using Parallel Processing Capabilities of the GPUS", 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, **(2009)** May 4-6, pp. 1-4, Potsdam, German.
[57] R. Kalarot, J. Morris and G. Gimelfarb, "Performance Analysis of Multi-resolution Symmetric Dynamic Programming Stereo on GPU", 2010 25th International Conference of Image and Vision Computing New Zealand (IVCNZ), **(2010)** November 8-9, pp. 1-7, Queenstown, UK.
[58] Q. Q. Yang, L. H. Wang, D. X. Li and M. Zhang, "Hybrid Stereo Matching by Dynamic Programming with Enhanced Cost Entry for Real-time Depth Generation", 2012 International Conference on Audio, Language and Image Processing (ICALIP), **(2012)** July 16-18, pp. 557-563, Shanghai, China.
[59] D. G. Lowe, "Distinctive Image Features from Scale-invariant Key Points", Int. J. Comput. Vis., vol. 2, no. 60, **(2004),** pp. 91-110.
[60] C. Liu, J. Yuen and A. Torralba, "SIFT Flow: Dense Correspondence across Scenes and Its Applications", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, no. 33, **(2011),** pp. 978-994.
[61] J. M. Morel and G. Yu, "Asift: A New Framework for Fully Affine Invariant Image Comparison", SIAM J. Imag. Sci., vol. 2, no. 2, **(2009),** pp. 438-469.
[62] J. Zhang, H. S. Sang and X. B. Shen, "Overview of Approaches for Accelerating Scale Invariant Feature Detection Algorithm", 2011 International Conference on Electric Information and Control Engineering (ICEICE), **(2011)** April 15-17, pp. 585-589, Wuhan, China.
[63] M. Tomasi, M. Vanegas, F. Barranco, J. Dfaz and E. Ros, "High Performance Optical-flow Architecture Based on A Multiscale, Multi-orientation Phase-based Model", IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 20, **(2010)**, pp. 1797-807.
[64] Y. Qiao, W. Wang, N. Minematsu, J. Z. Liu, M. Takeda and X. O. Tang, "A Theory of Phase Singularities for Image Representation and its Applications to Object Tracking and Image Matching", IEEE Transactions on Image Processing, vol. 10, no. 18, **(2009)**, pp. 2153-2166.
[65] M. Solgi and J. Weng, "Developmental Stereo: Emergence of Disparity Preference in Models of the Visual Cortex", IEEE Transactions on Autonomous Mental Development, vol. 4, no. 1, **(2009),** pp. 238-252.

[66] M. Galar, J. Fernandez, G. Beliakov and H. Bustince, “Interval-Valued Fuzzy Sets Applied to Stereo Matching of Color Images”, IEEE Transactions on Image Processing, vol. 7, no. 20, **(2011)**, pp. 1949-1962.
[67] J. A. Martins, J. M. F. Rodrigues and J. M. H. Dubuf, “Disparity Energy Model Using A Trained Neuronal Population”, 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), **(2011)** December 14-17, pp. 287-292, Bilbao, Spain.
[68] A. Celebi, O. Urhan, I. Hamzao and S. Erturk, “Efficient Hardware Implementations of Low Bit Depth Motion Estimation Algorithms”, IEEE Signal Processing Letters, vol. 6, no. 16, **(2009),** pp. 513-516.
[69] D. S. Kim, S. S. Lee and B. H. Choi, “A Real-Time Stereo Depth Extraction Hardware for Intelligent Home Assistant Robot”, IEEE Transactions on Consumer Electronics, vol. 3, no. 56, **(2010),** pp. 1782-1788.
[70] M. Humenberger, C. Zinner, M. Weber, W. Kubinger and M. Vincze, “A Fast Stereo Matching Algorithm Suitable for Embedded Real-time Systems”, Comput. Vis. Image Understanding, vol. 11, no. 114, **(2010),** pp. 1180-1202.
[71] S. Gotchev, G. B. Akar, T. Capin, D. Strohmeier and A. Boev, “Three-Dimensional Media For Mobile Devices”, Proceedings of the IEEE, vol. 4, no. 99, **(2011),** pp. 708-742.
[72] M. Tomasi, M. Vanegas, F. Barranco, J. Divaz and E. Ros, “Massive Parallel-Hardware Architecture for Multiscale Stereo, Optical Flow and Image-Structure Computation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 2, no. 22, **(2012),** pp. 282-294.
[73] H. Brunton, C. Shu and G. Roth, “Belief Propagation on the GPU for Stereo Vision”, The 3rd Canadian Conference on Computer and Robot Vision, 2006, **(2006)** June 7-9, pp. 76-81, Quebec City, Canada.
[74] S. Grauer-Gray and C. Kambhamettu, “Hierarchical Belief Propagation To Reduce Search Space Using CUDA for Stereo and Motion Estimation”, 2009 Workshop on Applications of Computer Vision (WACV), **(2009)** December 7-8, pp. 1-8, Snowbird, UT, USA.
[75] S. Darabiha, J. MacLean and J. Rose, “Reconfigurable Hardware Implementation of A Phase Correlation Stereo Algorithm”, Machine Vision Appl., vol. 2, no. 17, **(2006),** pp. 116-132.
[76] S. Park, C. Chen and H. Jeong, “VLSI Architecture for MRF-based Stereo Matching”, 7th International Workshop, **(2007)** July 16-19, pp. 55–64, Samos, Greece.
[77] J. M. Perez, P. Sanchez and M. Martinez, “High Memory Throughput FPGA Architecture for High Definition Belief-Propagation Stereo Matching”, 2009 3rd International Conference on Signals, Circuits and Systems (SCS), **(2009)** November 6-8, pp. 1-6, Medenine, Tunis.
[78] S. Forstmann, J. Ohya, Y. Kanou, A. Schmitt and S. Thuering, “Real-time Stereo by Using Dynamic Programming”, in Proc. Computer Vision Pattern Recognition Conf., **(2004)** June 27-July 4, pp. 29-36, Washington, DC, USA.
[79] H. Hirschmuller, P. R. Innocent and J. Garibaldi, “Real-time Correlation Based Stereo Vision with Reduced Border Errors”, Int. J. Comput. Vision, vol. 1, no. 47, **(2004),** pp. 229-246.
[80] R. Kalarot and J. Morris, “Comparison of FPGA and GPU implementations of Real-time Stereo Vision”, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), **(2010)** June 13-18, pp. 9-15, San Francisco, CA, USA.
[81] T. H. Tsai, N. Chang and T. S. Chang, “Data Reuse Analysis of Local Stereo Matching”, in Proc. Int. Symp. Circuits Syst., **(2008)** May 18-21, pp. 812-815, Seattle, WA, USA.
[82] N. Chang, T. M. Lin, T. H. Tsai and Y. C. Tseng, “Real-time DSP Implementation on Local Stereo Matching”, IEEE Conf. Multimedia Expo, **(2007)** July 2-5, pp. 2090-2093, Beijing, China.
[83] M. L. Gong, R. Yang, L. Wang and M. W. Gong, “A Performance Study on Different Cost Aggregation Approaches Used in Real-time Stereo Matching”, Int. J. Comput. Vis., vol. 2, no. 75, **(2007),** pp. 283-296.
[84] L. Jiangbo, S. Rogmans, G. Lafruit and F. Catthoor, “Real-time Stereo Correspondence Using a Truncated Separable Laplacian Kernel Approximation on Graphics Hardware”, IEEE Int’l Conf. Multimedia Expo, **(2007)** July 2-5, pp. 1946-1949, Beijing, China.
[85] F. Tombari, S. Mattoccia, L. Di Stefano and E. Addimanda, “Near Real-time Stereo Based on Effective Cost Aggregation”, in Proc. Int. Conf. Comput. Vision Pattern Recognit., **(2008)** December 8-11, pp. 1-4, Tampa, FL, USA.
[86] J. Lu, K. Zhang, G. Lafruit and F. Catthoor, “Real-time Stereo matching: A Cross-based Local Approach”, in Proc. Int. Conf. Acou., Speech, Signal Process., **(2009)** April 19-24, pp. 733-736, Taipei, Taiwan.
[87] J. Lu, S. Rogmans, G. Lafruit and F. Catthoor, “Stream-centric Stereo Matching and View Synthesis: A High-speed Approach on GPUs”, IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 19, **(2009),** pp. 1598-1611.
[88] W. Yu, T. Chen, F. Franchetti and J. C. Hoe, “High Performance Stereo Vision Designed for Massively Data Parallel Platforms”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 20, **(2010),** pp. 1509-1519.
[89] K. Zhang, J. B. Lu, Q. Yang, G. Lafruit, R. Lauwereins and L.V. 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. 7, no. 21, **(2011),** pp.867-879.

[90] J. B. 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. 10, no. 21, **(2011),** pp. 1547-1554.
[91] P. Karl, T. Matteo, D. Alonso, K. Javier, R. Eduardo, V. Hulle and M. Marc, "A Comparison of FPGA and GPU for Real-Time Phase-Based Optical Flow, Stereo, and Local Image Features", IEEE Transactions on Computers, vol. 7, no. 61, **(2012),** pp. 999-1012.
[92] Y. Miyajima, "A Real-time Stereo Vision System with FPGA", 13th International Conference in Field programmable logic and applications, **(2003)** September 1-3, pp. 448-457, Lisbon, Portugal.
[93] G. Minglun and Y. Y. Hong, "Near Real-time Reliable Stereo Matching Using Programmable Graphics Hardware", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2005)** June 20-25, pp. 924-931, San Diego, CA, USA.
[94] J. Diaz, E. Ros, R. Carrillo and A. Prieto, "Real-time System for High Image Resolution Disparity Estimation", IEEE Trans. Image Processing, vol. 1, no. 16, **(2007),** pp. 280–284.
[95] J. A. Kalomiros and J. Lygouras, "Hardware Implementation of a Stereo Co-processor in A Medium-scale Field Programmable Gate Array", IET Comput. Digit. Tech., vol. 5, no. 2, **(2008),** pp. 336-346.
[96] S. Jin, J. Cho, X. D. Pham, K. M. Lee, S. K. Park, M. Kim and J. W. Jeon, "FPGA Design and Implementation of a Real-Time Stereo Vision System", IEEE Transactions on Circuits and Systems for Video Technology, vol. 1, no. 20, **(2010),** pp. 15-26.
[97] K. Ambrosch and W. Kubinger, "Accurate Hardware-based Stereo Vision", Computer Vision and Image Understanding, vol. 11, no. 114, **(2010),** pp. 1303-1316.
[98] N. H. Tan, N. H. Hamid, P. Sebastian and Y. V. Voon, "Resource Minimization in A Real-time Depth-map Processing System on FPGA", TENCON 2011-2011 IEEE Region 10 Conference, **(2011)** November 12-14, pp. 706-710, Bali, Indonesia.
[99] C. Ttofis and T. Theocharides, "Towards Accurate Hardware Stereo Correspondence: A Real-time FPGA Implementation of A Segmentation Based Adaptive Support Weight Algorithm", Design, Automation & Test in Europe Conference & Exhibition, **(2012)** March 12-16, pp. 703-708, Dresden, Germany.
[100] C. Ttofis, S. Hadjitheophanous, A. Georghiades and T. Theocharides, "Edge-Directed Hardware Architecture for Real-Time Disparity Map Computation", IEEE Transactions on Computers, vol. 4, no. 62, **(2012),** pp. 690-704.
[101] M. Kuhn, "Efficient ASIC Implementation of a Real-time Depth Mapping Stereo Vision System, Circuits and Systems", Proceedings of the 46th IEEE International Midwest Symposium, **(2003)** December 27-30, pp.1478–1481, Cairo, Egypt.
[102] N. Y. Chang, T. H. Tsai, B. H. Hsu, Y. C. Chen and T. S. Chang, "Algorithm and Architecture of Disparity Estimation With Mini-Census Adaptive Support Weight", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 20, **(2010),** pp. 792-806.
[103] S. K. Han, S. H. Woo, M. H. Jeong and B. J. You, "Improved-Quality Real-Time Stereo Vision Processor", 2009 22nd International Conference on VLSI Design, **(2009)** January 5-9, pp. 287–292, New Delhi, India.
[104] C. W. Chou, J. J. Tsai, H. M. Hang and H. C. Lin, "A Fast Graph Cut Algorithm for Disparity Estimation", 2010 Picture Coding Symposium (PCS), **(2010)** December 8-10, pp. 326-329, Nagoya, Japan.
[105] S. A. Fezza and S. Ouddane, "Fast Stereo Matching via Graph Cuts, 2011 7th International Workshop on Systems", Signal Processing and their Applications (WOSSPA), **(2011)** May 9-11, pp. 115-118, Tipaza, Algeria.
[106] L. Zhou, T. Sun and Y. Zhang, "Real-time Depth Map Prediction and Optimization Based on Adaptive Image Segmentation", International Journal of Advancements in Computing Technology, vol. 2, no. 5, **(2013),** pp. 621-631.
[107] Y. S. Heo, K. M. Lee and S. U. Lee, "Robust Stereo Matching Using Adaptive Normalized Cross Correlation", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 4, no. 32, **(2010),** pp. 807-822.
[108] X. Sun, X. Mei, S. H. Jiao, M. Zhou and H. Wang, "Stereo Matching with Reliable Disparity Propagation", 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), **(2011)** May 16-19, pp. 132-139, Hangzhou, China.
[109] D. Neilson and Y. H. Yang, "A Component-Wise Analysis of Constructible Match Cost Functions for Global Stereopsis", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 33, **(2011),** pp. 2147-2159.
[110] A. Gruber, E. Boros and R. Zabih, "A Graph Cut Algorithm for Higher-order Markov Random Fields", 2011 IEEE International Conference on Computer Vision (ICCV), **(2011)** November 6-13, pp. 1020-1027, Barcelona, Spain.
[111] V. Kolmogorov and C. Rother, "Minimizing Non-Submodular Functions with Graph Cuts-A Review", Technical Report MSRTR-2006-100, Microsoft Research, **(2006)**.

[112] J. Woodfordy, P. H. S. Torrz, I. D. Reidy and A. W. Fitzgibbon, "Global Stereo Reconstruction under Second Order Smoothness Priors", IEEE transactions on pattern analysis and machine intelligence, vol. 12, no. 31, **(2008),** pp. 2115-2128.
[113] H. Hirschmuller, "Stereo Processing by Semi-global Matching and Mutual Information", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 2, no. 30, **(2008),** pp. 328-341.
[114] D. Min and K. Sohn, "Cost Aggregation and Occlusion Handling with WLS in Stereo Matching", IEEE Trans. on Image Processing, vol. 8, no. 17, **(2008),** pp. 1431–1442.
[115] J. P. Abrahamsen, P. Hiliger and T. S. Lande, "Time Domain Winner-take-all Network of Integrate-and-fire Neurons", IEEE Int'l Symp. Circuits Syst., **(2004)** May 23-26, pp. 361-364, Vancouver, Canada.
[116] Q. Yang, L. Wang, R. Yang, H. Stewenius and D. Nister, "Stereo Matching with Color Weighted Correlation, Hierarchical Belief Propagation and Occlusion Handling", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 3, no. 31, **(2009),** pp. 492-504.
[117] S. B. Kang and R. Szeliski, "Extracting View-Dependent Depth Maps from a Collection of Images", Int'l J. Computer Vision, vol. 2, no. 58, **(2004),** pp. 139-163.
[118] X. F. Chang, Z. Zhou, L. Wang, Y. J. Shi and Q. P. Zhao, "Real-Time Accurate Stereo Matching Using Modified Two-Pass Aggregation and Winner-Take-All Guided Dynamic Programming", 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), **(2011)** May 16-19, pp. 73-79, Hangzhou, China.
[119] L. D. Maeztu, A. Villanueva and R. Cabeza, "Near Real-Time Stereo Matching Using Geodesic Diffusion", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 34, **(2012),** pp. 410-416.
[120] C. Rhemann, A. Hosni, M. Bleyer, C. Rother and M. Gelautz, "Fast Cost-volume Filtering for Visual Correspondence and Beyond", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 35, **(2013),** pp. 504-511.
[121] D. Zarpalas, E. Fotiadou, I. Biperis and P. Daras, "Anchoring Graph Cuts Towards Accurate Depth Estimation in Integral Images", Journal of Display Technology, vol. 7, no. 8, **(2012),** pp. 405-417.
[122] M. Tomasi, M. Vanegas, F. Barranco, J. Daz and E. Ros, "Massive Parallel-Hardware Architecture for Multiscale Stereo, Optical Flow and Image-Structure Computation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 2, no. 22, **(2012),** pp. 282-294.

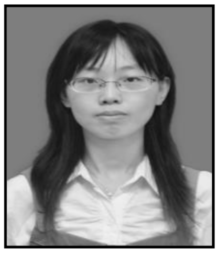
Authors



Li Zhou, received Ph.D degree on 2004 from Zhejiang University, China. On 2004, Dr. Zhou joined Freescale semiconductor R&D center as principle design architecture till 2009. As an architecture and core team member, Dr. Zhou participated in multiple National high-tech SoC and HDTV fundamental research projects, led multiple 65nm/90nm 10 million gate scale VLSI SoC chips, and joined many VLSI project and architecture researches. From 2009, Dr. Zhou is an assistant professor in Shandong University, China. Her current research interests include stereo vision system algorithm and hardware design, GPU architecture and VLSI design, high performance processor architecture and VLSI design, *etc.*

Tao Sun, received Ph.D degree on 2002 from Zhejiang University, China. Dr. Sun joined Suzhou China Core Co. Ltd as a researcher on CPU architecture and vice-general manager till 2007. From 2007 to 2009, Dr. Sun was a senior manager in Spansion China, led VLSI design projects. From 2009 till now, Dr. Sun is an associated professor in University of Jinan, China. His research interests include CPU/VLSI architecture, Solid storage architecture, *etc*.

Yuanzhi Zhang, received BE degree on 2011 from Shandong University, China. From 2011 till now, Zhang is pursuing his master's degree in Shandong University. His research interests include CPU/GPU architecture, human intelligence, image processing, *etc*.



Jia Wang, received BE degree on 2011 from Shandong University, China. From 2011 till now, Wang is pursuing his master's degree in Shandong University. Her research interests include image and video processing, VLSI design, GPU architecture, *etc*.