High Speed Low Complexity Guided Image Filtering Based Disparity Estimation

Charan Kumar Vala, Koushik Immadisetty, Amit Acharyya, *Member, IEEE,* Charles Leech, Vibishna Balagopal, Geoff V. Merrett, *Member, IEEE,* and Bashir M. Al-Hashimi, *Fellow, IEEE,*

***Abstract*—Stereo vision is a methodology to obtain depth in a scene based on the stereo image pair. In this paper we introduce a Discrete Wavelet Transform (DWT) based methodology for a state-of-the-art disparity estimation algorithm, that resulted in significant performance improvement in terms of speed and computational complexity. In the initial stage of the proposed algorithm, we apply DWT to the input images, reducing the number of samples to be processed in subsequent stages by 50%, thereby decreasing computational complexity and improving processing speed. Subsequently the architecture has been designed based on this proposed methodology and prototyped on a Xilinx Virtex-7 FPGA. The performance of the proposed methodology has been evaluated against four standard Middlebury Benchmark image pairs viz. *Tsukuba*, *Venus*, *Teddy* and *Cones*. The proposed methodology results in improvement of about 44.4% cycles per frame, 52% frames per second and 61.5% and 59.6% LUT and register utilization respectively, compared with state-of-the-art designs.**

***Index Terms*—Guided Image Filtering, Stereo-Matching, FPGA, Discrete Wavelet Transform, Low-Complexity, High-Speed.**

I. Introduction

STEREO matching, the task of matching images acquired by a pair of cameras (conventionally called reference and target images) and calculating the depth of objects in a scene, is being employed in sophisticated embedded vision applications including surveillance, autonomous vehicles and mobile robots [1]. A geometrical representation of the same is shown in Fig. 1. Since such systems are mostly used in mobile environments running from battery or harvested energy, the primary design-challenges are low power consumption and area-overhead reduction, whilst meeting the real-time processing requirement with acceptable precision. Hence the choice of matching algorithm and respective architecture implementation play an important role to tackle these challenges and meet these constraints for producing a viable embedded stereo matching system.

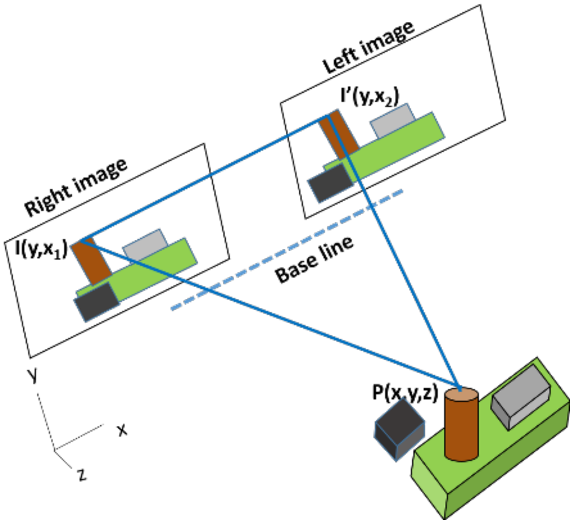


Fig. 1. Geometry of a stereo vision system where a pair of parallel cameras capture the scene from horizontally disparate viewpoints. Given that the input images are rectified, the correspondence of a pixel at coordinate (x,y) of the reference image can only be found at the same vertical coordinate y within a disparity range (D_{min} to D_{max}) in the target image.

Thorough analysis of the existing literature shows that, despite having low-error rate in the disparity computation, state-of-the-art DSPs can not support global stereo matching algorithms due to intensive computational needs [2], [3]. A viable alternative to this could be GPU-based implementations but at the expense of high cost and power-consumption for real-time designs [4]. On the other hand, due to reduced algorithmic complexity, local stereo matching algorithms were getting implemented over the FPGA and ASIC platform to meet real-time requirements with higher frames per second compared to its global matching counter-part [5]–[7]. However the concern with the local matching algorithms is its low precision. To mitigate this issue, investigations have been made to implement dedicated hardware architectures of more precise algorithms, such as Semi Global Matching (SGM) [8], [9] and Adaptive Support Weight (ADSW) [10], [11]. For the past few years, hardware implementations predicated on SGM and ADSW algorithms have become the preferred solution towards higher matching precision in embedded vision applications [5], [7], [12], [13]. In addition, modifications and improvements have been made to implementations to habituate

CKV, CL, GVM and BMAH are with the School of Electronics and Computer Science, University of Southampton, UK. E-mail: (C.K.V, cl19g10, gvm, bmah)@ecs.soton.ac.uk

Koushik Immadisetty with Qualcomm Innovation Center Inc.: E-mail: kimmadis@qti.qualcomm.com

Amit.A with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, India. E-mail: (amit_acharyya)@iith.ac.in

Vibishna Balagopalwith LumiraDx Technology, U.K E-mail: Vibishna.Balagopal@gmail.com

This research was supported in part by EPSRC grant EP/K034448/1 www.prime-project.org

Amit Acharyya would also like to acknowledge his Visvesvaraya Young Faculty Fellowship by the Ministry of electronics and Information Technology, Govt. of India.

Data supporting this research can be found at DOI http://doi.org/10.5258/SOTON/D0170

the algorithms for real-time processing, with improvement in execution times over existing designs [14] but at the cost of increased error rates when compared to state-of-the-art software implementations [15]. Furthermore, high memory and significant need of hardware resources are the bottleneck towards the scalability of these designs to higher resolution images. We believe that efficiently reducing the number of samples to be processed at different stages of the algorithm will reduce the consumption of hardware resources, improving the scalability of these designs. With this motivation, we introduce the Discrete Wavelet Transform (DWT) and the Inverse Discrete Wavelet Transform (IDWT) at the pre and post-processing stages of the traditional Guided Image Filter (GIF) based disparity estimation process flow, which results in reduced processing-data sizes in the intermediate stages and thus less complex hardware in the FPGA implementation and throughput improvement. It is to be noted that the DWT has been exploited in state-of-the-art disparity estimation for extracting the feature points of the image in [16] and for improving the quality of disparity map in [17], [18]. To the best of our knowledge it has never been attempted in GIF based disparity estimation on hardware complexity reduction.

The GIF algorithm has been shown to reduce the complexity of the cost aggregation step in local ADSW algorithms [19], [20]. Recently published literature detailed the fully pipelined and parallel GIF architecture and corresponding FPGA based implementations [21], [22]. However, instead of designing a stand-alone architecture, we believe holistic optimizations of algorithm and architecture would lead to improved performance. With this motivation, in this paper we:

- Introduce a Discrete Wavelet Transform (DWT) based methodology for the disparity estimation algorithm.
- Design the respective architecture and prototype it on a Xilinx Virtex-7 FPGA and validate it against the standard databases, comparing the performance to the outcomes of existing designs.
- Show that the proposed methodology results in improvement of about 44.4% Cycles per Frame (CPF), 52% Frames per Second (FPS) and 61.5% and 59.6% LUT and register utilization respectively compared with the state-of-the art designs [9], [13], [23].

The rest of the paper is organized as follows: Section II provides the necessary theoretical background, Section III introduces the proposed methodology and subsequent architecture, Section IV discusses the experimental results, compares performance of the proposed methodology with existing designs and finally Section V concludes the discussion. The acronyms used throughout the paper are explained in Table I.

TABLE I
FULL FORM OF ACRONYMS USED IN THE PAPER

Symbol	Explanation
BRAM	Block Random Access Memory
corr	Correlation
cov	Covariance
CPF	Cycles per Frame
CVC	Cost Volume Construction
CVF	Cost Volume Filtering
D_{max}	Maximum Disparity
D_{min}	Minimum Disparity
DWT	Discrete Wavelet Transform
FPS	Frames per second
GIF	Guided Image Filter
Hei	Hight of image
Wid	Width of image
IDWT	Inverse Discrete Wavelet Transform
LUT	Look Up Table
MDE/s	Million Disparity Estimation per second
SAD	Sum of Absolute Difference
WTA	Winner-take-all

II. Related Works

Stereo matching algorithms can broadly be classified into two categories: global and local [24]. Global algorithms are formulated as an energy minimization problem, which is solved with techniques such as Dynamic Programming, Graph Cuts and Belief Propagation. Such methods produce very accurate results at the expense of high computational complexity and memory needs. Semi-Global Matching (SGM) [8], [9], [15] methods renounce part of the accuracy by approximating a global 2D function using a sum of 1D optimizations from all directions through the image. SGM methods are therefore more affordable for dedicated hardware implementations, but still consume significant memory to store the interim cost of different aggregation paths. In contrast, local algorithms use block matching and winner-takes-all optimization to determine the disparity associated with a minimum cost function at each pixel [3]. Hence, they have lower computational complexity and memory requirements compared to global and SGM methods.

Among local algorithms, the most recent Adaptive Support Weight (ADSW) methods are currently the most accurate [2]. Despite their good results, ADSW algorithms cannot take advantage of the integral image or sliding window techniques, as the adaptive weights have to be recomputed at every pixel. This makes the cost aggregation's hardware complexity directly dependent on the support window size. To improve matching accuracy, a few attempts have been made by combining different stereo algorithms together or by implementing modified versions of SGM and ADSW algorithms. A modified version of the census transform in both the intensity and gradient images, in combination with the SAD correlation metric has been implemented in hardware [6]. A stereo algorithm based on the neural network and Disparity Space Image (DSI) data structure is introduced in [7] and implemented on an FPGA. Zhang *et. al.* combined both the mini-census transform and cross based cost aggregation for implementing real-time FPGA based stereo matching [25]. SGM-based stereo matching systems have been introduced in [8], [9], [15] and implemented on FPGAs and a hybrid FPGA/RISC architecture based platforms respectively. The VLSI design of an ADSW algorithm that adopted the mini-Census transform was implemented to improve the accuracy and robustness of the system to radiometric distortions [14]. Incorporating an ADSW algorithm and integration of pre and post- processing units, [11] proposed the implementation of a complete stereo vision system. Finally, a hardware oriented stereo matching

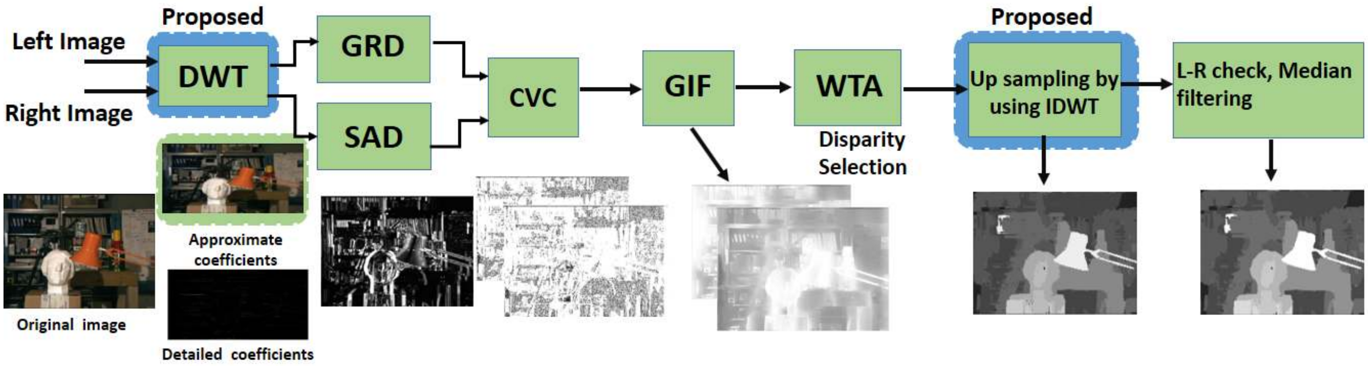


Fig. 2. Proposed DWT based methodology and the flow of Disparity estimation algorithm.

system based on the adaptive Census transform is presented in [10].

The aforementioned high-quality ADSW-predicated systems follow a homogeneous algorithm-to-hardware mapping methodology. The recently proposed Guided Image Filter (GIF) [19] has been employed in [20]–[22] to reduce the complexity of the cost aggregation step in ADSW methods, leading to a high-quality fast and simple algorithm consisting of the following steps: cost volume construction, cost volume filtering, disparity selection and disparity refinement. Cost Volume Construction (CVC) is a measure of calculating a cost between two pixels, one from the left image and one from the right image, to identify the disparity corresponding to the depth of a point in the scene [3]. The information obtained by matching single pixels is not sufficient for precise matching. So for accomplishing precise matching and to minimize the matching uncertainties, cost volume filtering is used in the next step. Following filter, the most probable disparity is selected by using a local winner-takes-all (WTA) strategy. Finally, a refinement step is used to reduce noise and improve the disparity map [3]. Recently, Ttofis *et al.* [21], [22] proposed a fully pipe-lined, parallel stereo matching FPGA-based hardware architecture based on the GIF, achieving real-time processing for high definition (HD) images.

III. Proposed Methodology and Architecture

A. Proposed Methodology

1) Overview: Fig. 2 shows the entire flow of the DWT based stereo vision algorithm, comprised of six stages. The proposed DWT is the first stage that receives the rectified stereo images captured from a stereo camera. The DWT transforms these images into approximate wavelet coefficients and feeds these coefficients into the second stage of CVC, which includes the Sum of Absolute Differences (SAD) and Gradient (GRD) stages. The GIF is used to filter each cost volume slice. In the fourth stage, the minimum among all cost volumes is selected by the WTA strategy for disparity selection. The first instance of the actual disparity (equivalent to the size of original image) is synthesized from the wavelet-domain using IDWT as part of the fifth stage, followed by post-processing in the sixth stage which includes left and right consistency check and median filtering.

2) Proposed DWT based methodology: In the context of stereo-vision algorithm, we propose to use a DWT as the first stage to reduce the number of samples required for subsequent computations [26]–[30]. Exploiting the property in stereoscopic images that neighboring pixels are highly correlated, without any loss of generality, the first resolution level of approximate wavelet coefficients can be considered for further computation and the detailed wavelet coefficients can be left apart, which results in simple down-sampling. This is evident in Fig. 2 where the original image, approximate wavelet coefficients and detailed wavelet coefficients are shown. When the approximate and detailed coefficients are compared, the approximate wavelet coefficients retain most of the features of the original image. Therefore, without sacrificing matching accuracy, the overall number of samples for subsequent computations can be reduced to half as compared to the original image. The Haar wavelet is chosen as the mother wavelet, because it can be implemented by using simple adder and shifter leading to a low-complexity hardware implementation. The approximate wavelet coefficients of the input images from the first resolution level, after applying a vertical $1D$ Haar mask $[0.5, 0.5]$, can be represented as:

$$I^i_{left}(j) = \Big(L^i(2j-1) + L^i(2j)\Big)/2 \tag{1}$$

$$I^i_{right}(j) = \Big(R^i(2j-1) + R^i(2j)\Big)/2 \tag{2}$$

where I^i_{left} and I^i_{right} are the approximate wavelet coefficients of the input stereo images respectively (represented as L and R), i denotes the color channel in RGB space and $j = 1 : Hei/2 \; \forall \; Wid$.

At this stage the number of pixels of the input images to be processed are halved, i.e from $Hei{*}Wid$ to $(Hei{*}Wid)/2$ and thereby significantly reducing the computational complexity of subsequent modules.

3) Processing of the Approximate Wavelet Coefficients using GIF Based stereo matching [20]: CVC involves the cost computation of each pixel between the stereo images over a range of disparities d by considering the truncated absolute difference of colors and gradients as follows:

$$C(p,d) = a \cdot min\big(T_c, M(p,d)\big) + (1-a) \cdot min\big(T_g, G(p,d)\big) \tag{3}$$

where,

$$M(p,d) = \sum_{i=1}^{3} \mid I^{i}_{left}(p) - I^{i}_{right}(p-d) \mid \tag{4}$$

and

$$G(p,d) = \mid \nabla_x I^{i}_{left}(p) - \nabla_x I^{i}_{right}(p-d) \mid, \tag{5}$$

where i denotes the color channel in RGB space, $\nabla_x I^{i}_{left}$ denotes the gradient in x direction computed at pixel p, a is used to balance the influence of the color and gradient terms, T_c, T_g are the truncation thresholds respectively.

Next, the GIF is used to filter the cost volume. The filtered cost of a pixel p at disparity-d is given by:

$$q(p,d) = \sum W_{i,j}(I)C(p,d) \tag{6}$$

where,

$$W_{i,j} = \frac{1}{\mid w \mid^2} \sum_{k:(i,j)\in\omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \varepsilon}\right) \tag{7}$$

In (6) and (7), i and j are pixel indices, $W_{i,j}$ is the weight corresponding to j in the window W_k with center i and radius r, $|w|$ is the number of pixels in W_k (i.e, $(2r+1)x(2r+1)$), μ_k and σ_k are the mean and the standard deviation of I (the first resolution level of wavelet approximate coefficients from the guidance image) in W_k. The weights $(W_{i,j})$ can be computed using linear operations [19] which can be decomposed into a series of mean filters with windows of radius r. The pseudo code of the GIF is given in Algorithm 1, where f_{mean} is a boxfilter (same as a mean filter) with a window of radius r. The abbreviations of correlation ($corr$), variance (var), and covariance (cov) hold their usual meaning.

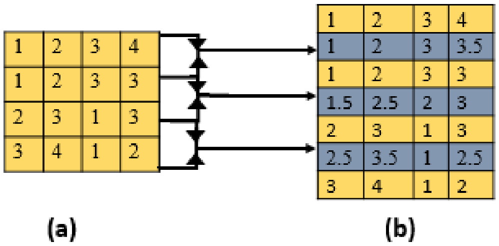
Disparity selection, involving the condensation of the cost volume back into a single image, is performed through the WTA strategy where $d \in [D_{min}, D_{max}]$:

$$d_p^{'} = argmin(q(p,d)) \tag{8}$$

4) Proposed IDWT based disparity estimation: The first disparity map,d_p, that is the size of the original image is synthesized from $d_p^{'}$ by applying the IDWT as follows:

$$d_p(2k) = \left(d_p^{'}(k) + d_p^{'}(k+1)\right)/2 \tag{9a}$$

$$d_p(2k-1) = d_p^{'}(k) \tag{9b}$$



1	2	3	4
1	2	3	3
2	3	1	3
3	4	1	2

1	2	3	4
1	2	3	3.5
1	2	3	3
1.5	2.5	2	3
2	3	1	3
2.5	3.5	1	2.5
3	4	1	2

Fig. 3. Computed disparity in the wavelet domain (left) and the up-sampled disparity after IDWT (right).

Algorithm 1 Guided Image Filter [20]

```
INPUT: guidance image I, guided image p
mean$_I$ = $f_{mean}$(I); mean$_P$ = $f_{mean}$(P); corr$_I$ = $f_{mean}$(I.*I); corr$_{IP}$ = $f_{mean}$(I.*P);
var$_I$ = corr$_I$ - mean$_I$.*mean$_I$; cov$_{IP}$ = corr$_{IP}$ - mean$_I$.*mean$_P$;
a = cov$_{IP}$/(var$_I$+$\varepsilon$); b = mean$_p$ - a.*mean$_I$;
mean$_a$ = $f_{mean}$(a); mean$_b$ = $f_{mean}$(b);
q = mean$_a$.*I+mean$_b$;
OUTPUT: output q (filtered image)
```

Algorithm 2 Proposed Disparity Estimation Algorithm

```
INPUT: pair of stereo images L(left), R(right)
for k = R,G,B do
    for i =1 to $Hei$/2 do
        for j = 1 to $Wid$ do
            $I_{left}$(i,j,k)=(L(2i-1,j,k)+L(2i,j,k))/2
            $I_{right}$ (i,j,k)=(R(2i-1,j,k)+R(2i,j,k))/2
        end for
    end for
end for
Find the left and right grayscale images
Cost Volume Construction:
for Disparity= $D_{min}$ to $D_{max}$ do
    for i =1 to $Hei$/2 do
        for j = 1 to $Wid$ do
            Find the SAD (4)
            Find the gradient (5)
            Find the cost (3)
        end for
    end for
end for
Cost Volume Filtering using GIF:
for CVC= $D_{min}$ to $D_{max}$ do
    Repeat GIF Algorithm 1
end for
Disparity Selection:
for CVF= $D_{min}$ to $D_{max}$ do
    $d_p{}' = argmin_d q(p,d)$
end for
Up-sampling of disparity:
for i =1 to $Hei$/2 do
    for j = 1 to $Wid$ do
        $d_p$(2i,j)=($d_p{}'$(i,j)+$d_p{}'$(i+1,j))/2
        $d_p$(2i-1,j)=$d_p{}'$(i,j)
    end for
end for
Post Processing:
Left-right consistency check and filling
Disparity refinement with a median filter
OUTPUT: Final Disparity
```

where $k = 1 : Hei/2 \; \forall \; Wid$. A demonstration of the up-sampling strategy for a 16-pixel template is shown in Fig. 2. To reduce noise and improve the quality of disparity map, a post-processing step is performed. This involves a left-right (L-R) consistency check. The L-R check requires the computation of both the left and right disparity maps. Pixels in the left disparity map are marked as inconsistent if the disparity value of its matching pixel in the right disparity map differs by more than one pixel. Inconsistent pixels are then filled by the disparity of the closest consistent pixel [20] and a median filter is used to smooth the filled regions and remove spikes. The pseudocode of the proposed DWT based disparity estimation methodology is given in Algorithm 2.

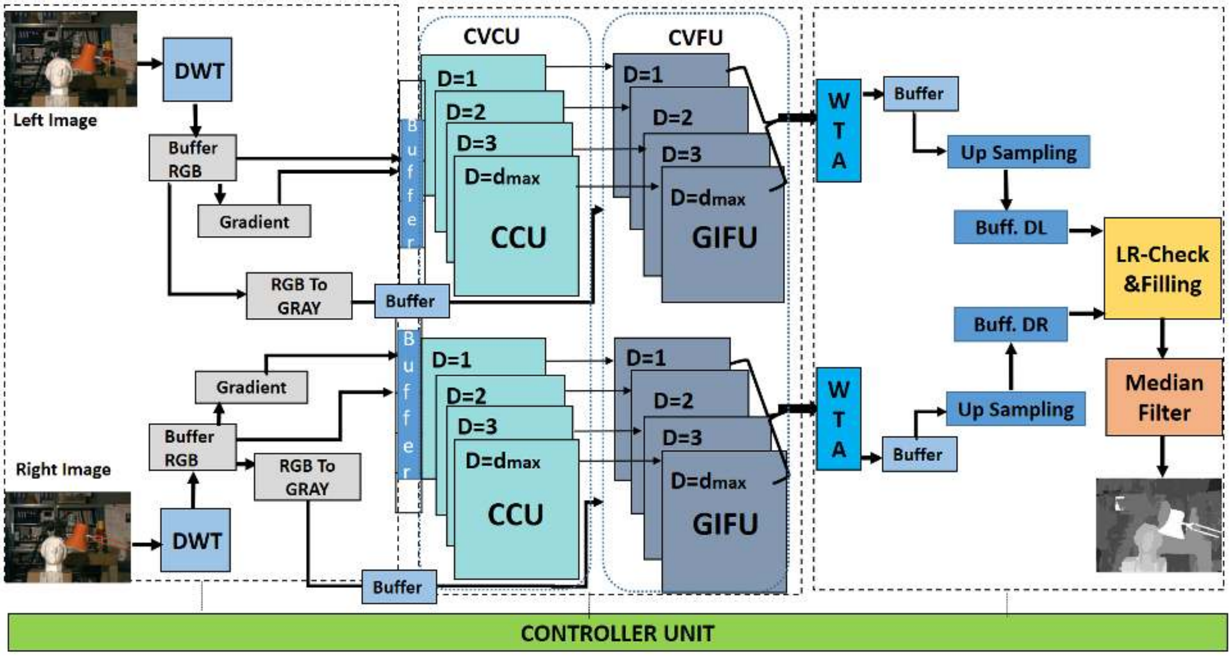


Fig. 4. Disparity estimation architecture based on the proposed DWT based methodology.

B. Architecture based on proposed methodology

Fig. 4 shows the disparity estimation architecture based on the proposed methodology. It can be observed that the architecture comprises of DWTs, RGB to gray scale image converters, gradient computation modules, Cost Computation Units (CCUs), Guided Image Filtering Units (GIFUs), WTAs, Upsampling modules, LR-check module, median filter and a controller. All the modules are fully pipelined to obtain high throughput. The rectified reference and target image's RGB data enters the processing pipeline consisting of the modules mentioned above. After multiple setup cycles from the pipeline latency, computed depth maps synchronized with the input pixel rate are forwarded successively through the scan-line to the output port.

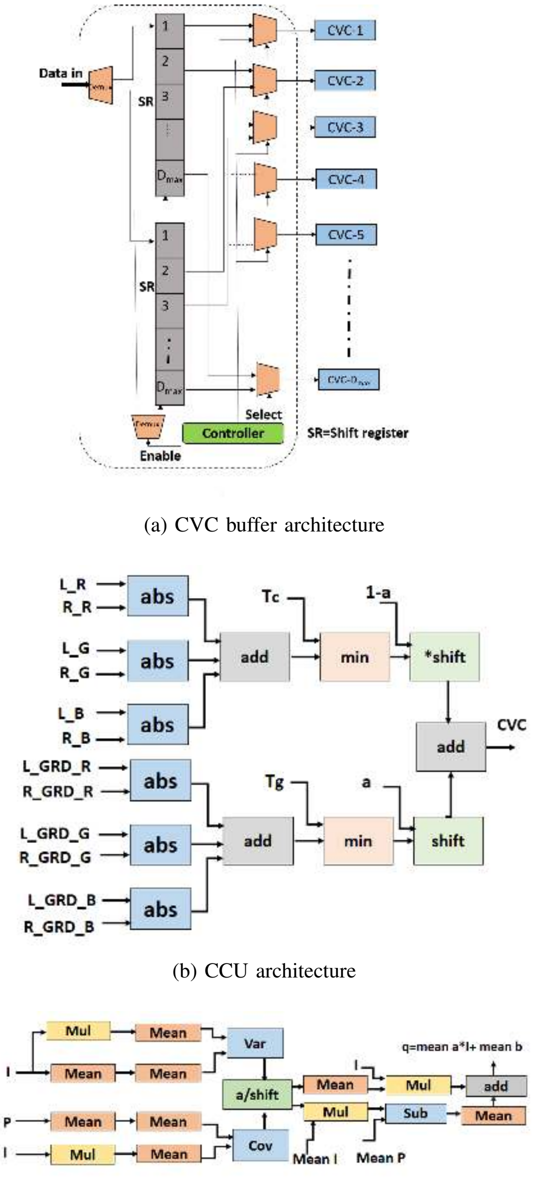
For the purpose of demonstrating the proof of concept, 6 DWT modules (one for each color channel) for left and right R,G,B color channels based on the Haar wavelet (Equations 1 and 2) have been designed, each consisting of one adder and one right shifter. The outputs of DWT module are stored in a BRAM. At this stage, the number of samples is reduced by half. The gradient module comprises of 6 subtracters for calculating the left and right RGB gradients in the X-direction. The gradient values of the left and right images are then stored in the buffer. Furthermore, these values are accessed in the next stage for CVC.

The inputs to the cost volume construction unit (CVCU) are sent through a buffer, Fig. 5a, where two shift registers (length D_{max}) send data to the corresponding CVC module through a $2x1$ mux depending on the select signal. The shift registers take the data from the input RGB BRAM alternately depending on the enable signal. All the remaining buffers in the design are realized using the FPGA's internal BRAMs.

The CVCU in Fig. 4 employs a cascade of CCUs to calculate the pixel-wise cost between the pixel p in the left image and the corresponding $(p-d)$ pixel in the right image. The architecture of CCU, based on Equations (3) - (5), is shown in Fig. 5b. It consists of absolute difference units, adders and comparators that calculate the truncated color and gradient costs, which are summed to compute the overall cost. Prior to the summation, the truncated color and gradient costs are multiplied by constant values to normalize the overall cost. To avoid multiplication and without any loss of generality, these constants are selected to be powers of 2, which can be realized with shifting operations alone.

The output of the CCU is fed into the cost volume filtering unit (CVFU) in a row-wise manner. The CVFU employs a cascade of GIFUs. The architecture of GIFU is based on the pseudocode of Algorithm 1 and is shown in Fig. 5c. Each GIFU comprises boxfilters, subtracters, an adder and a shifter. Among all the components in GIFU, the boxfilter is the computationally intensive component. Howevr, since the number of input samples to be processed has been reduced by the DWT module, the boxfilters benefit from the reduction in the number of computations per image.

The architecture of the boxfilter is shown in Fig. 6a. The boxfiltering operation for a 6x7 image template with window radius 1 is illustrated in Fig. 6b, and is explained as follows: If pixel-i of an image is updated then the mean of the box with pixel-i as end pixel is given by the boxfilter. The basic operation of the boxfilter is to maintain each column sum by adding the $(2r+1)$ pixels. The box sum is computed by adding the $(2r+1)$ adjacent column sums. From Fig. 6b, if pixel-i is updated, then the corresponding column sum is updated by adding and subtracting the new and old pixels respectively. In the same way, the box sum is computed by adding and subtracting the new and old column sums respectively. Storing

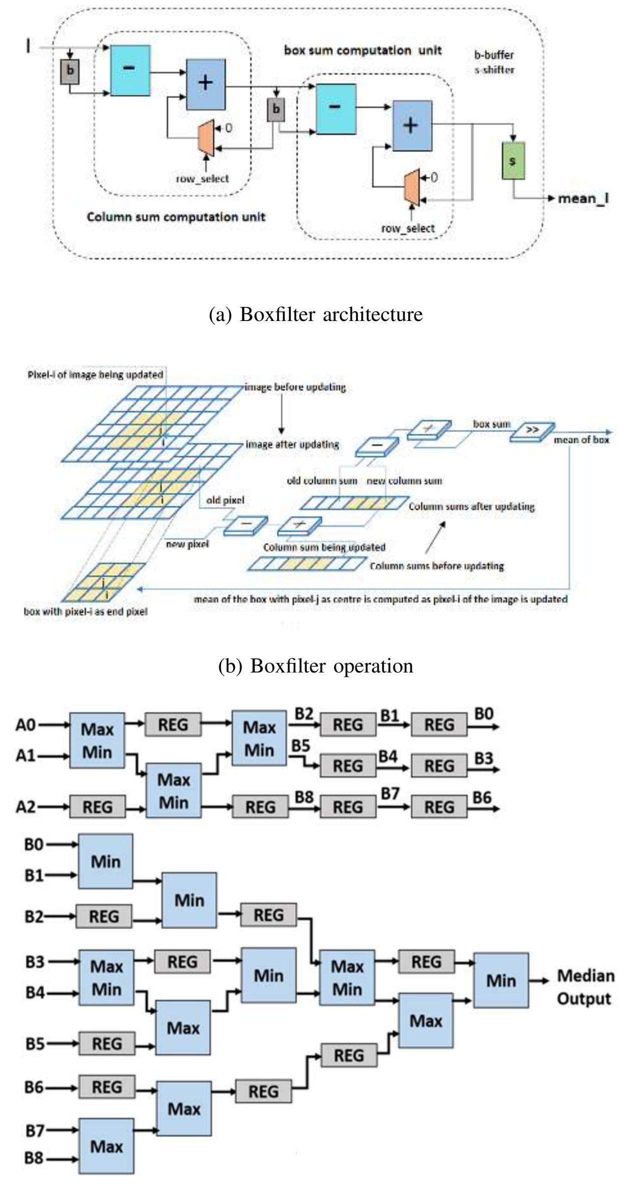


(a) CVC buffer architecture

(b) CCU architecture

(c) GIFU architecture

Fig. 5. The architectures of the CVC buffer, CCU and GIFU



(a) Boxfilter architecture

(b) Boxfilter operation

(c) Median filter architecture

Fig. 6. The architecture of the boxfilter, an illustration of its operation and the architecture of the median filter.

only $(2r+1)$ rows of pixels for boxfilter operations would be sufficient, resulting in reduction of memory consumption from entire image to just $(2r+1)$ rows of pixels. The boxfilter architecture (Fig. 6a) has two submodules; the column sum computation unit (where the column sum to be updated is computed) and the box sum computation unit (where the box sum to be updated is computed).

The operation of the GIFU for a $6x7$ image template and window radius 1 is shown in Fig. 7 and is explained according to the pseudocode of the GIF (algorithm 1) as follows: If pixel-i of an image is known then the mean of box with pixel-i as end pixel is computed by a box-filter (as described above). Similarly, from Fig. 7 as pixel-i of I (guidance image) and P (guided image) are updated, II and IP of pixel-i are computed by two multipliers. mean$_I$, mean$_P$, mean$_{II}$, mean$_{IP}$, var$_I$, cov$_{IP}$ of pixel-j are computed by four box filters, four multipliers and two subtracters. Consequently, the a and b values of pixel-j are computed by a shifter, multiplier and subtracter. In a similar way, since the a and b values of pixel-j are known, the mean of the box with pixel-j as the end pixel, i.e mean$_a$, mean$_b$ of pixel-k are computed by two box filters. Finally, q (i.e, filtered output) of pixel-k is computed by a multiplier and an adder. Thus the entire architecture of the CVFU is pipelined in parallel.

Two WTA modules, shown in Fig. 4, are used for the left and right disparity selection respectively. Each WTA module comprises of comparators organized in tree structures with $\log_2 D_{max}$ stages. The disparities obtained are stored in the buffers before being transferred to the up-sampling units which consist of an adder and shifter. The up sampled disparities are stored in the buffers DL and DR, as shown in Fig. 4, from where they are accessed by the post processing unit. The

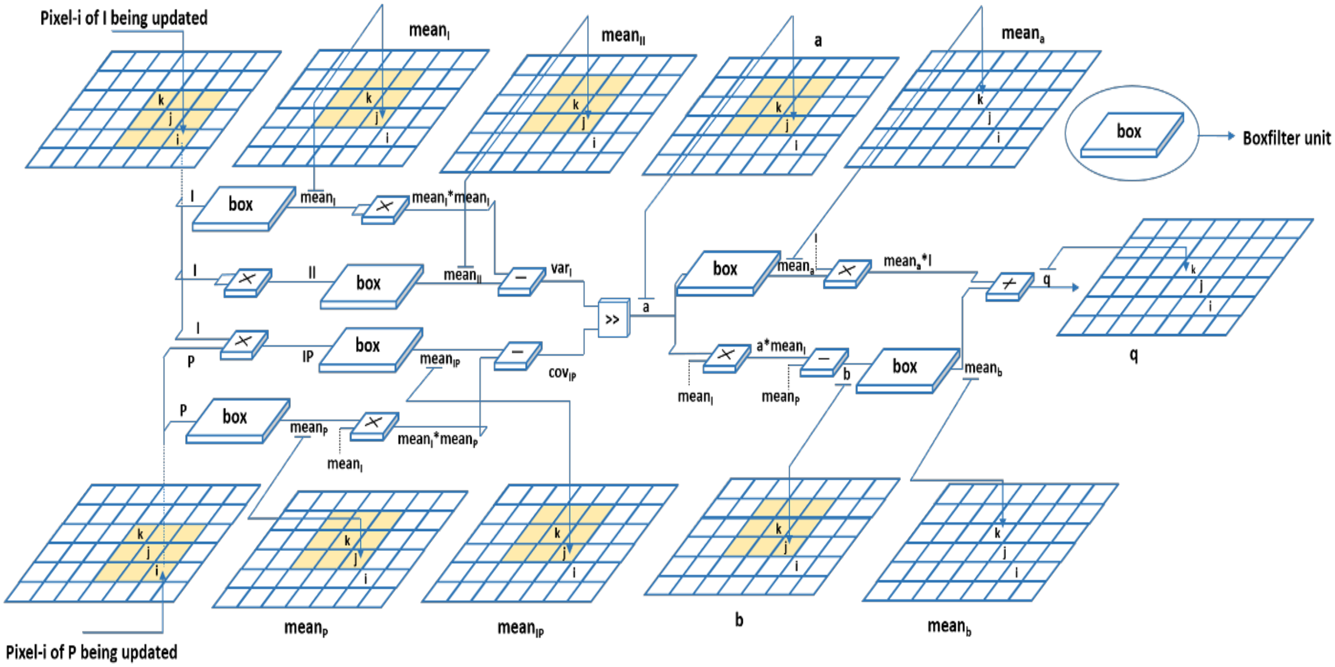


Fig. 7. The operation of the Guided Image Filtering Unit (GIFU).

Post Processing module implements the L-R check and filling (interpolation) through a set of comparators, multiplexer trees, and priority encoders that locate the nearest valid disparities in the left and right direction of the invalid pixels. To reduce noise and eliminate any remaining spikes in interpolated disparity map, an energy-efficient 3x3 median filter design has been adopted from [31], [32] as shown in Fig. 6c, where $A0, A1, A2$ are the input disparities. The reason for choosing a $3x3$ median filter is explained in Section IV.

IV. Results and Discussion

To validate the proposed methodology, the stereo matching core has been implemented on a Xilinx Virtex-7 FPGA, with system parameters set to the constant values: $\{r, \varepsilon, T_c, T_g\} = \{3, 0, 7, 2\}$ adopted from recent literature [20]–[22] and are used throughout our experiments. The metrics used to evaluate the proposed methodology are; error (nonocc, all, disc) (Table II), frames-per-second (FPS), millions of disparity estimations per second (MDE/s), cycles per frame (CPF) and hardware resource utilization (LUTs, slice registers, DSPs, BRAMs).

The performance of the system designed based on the proposed methodology has been evaluated against the standard Middlebury benchmark that is widely used in evaluating the quality of the stereo-matching algorithms. The four image pairs viz. *Tsukuba*, *Venus*, *Teddy* and *Cones* are processed by the proposed system and the results are shown in Fig. 8(a-l). A performance comparison of the proposed system with state-of-the-art designs in terms of error (i.e, the percentage of bad matching pixels when compared to the ground truth) is listed in Table II. From this table, it can be observed that the error of the proposed methodology is 5.71% in non-occluded regions, 9.89% across the whole disparity map, and 13.58% in discontinuous regions, resulting in an accuracy of 90.27% (calculated as 100 - average bad pixel percentage). To further validate the proposed methodology rectified real-time stereo images, captured using a stereo camera, are processed. The corresponding results are shown in Fig. 8(o,p). It can be seen that the disparity is sharp for clear edges.

The proposed methodology is verified by using different wavelets (db2, sym2, coif1, dmey) and without applying the DWT. The disparity maps obtained for the Tsukuba stereo pair are shown in Fig. 9a and the corresponding overall error graph is shown in Fig. 9b. The error for all the wavelets vary from 6.72% to 7.9%, which shows that any of the wavelets can be used for the proposed methodology as they are in acceptable range as per Table II. Since hardware complexity should also be considered, the Haar wavelet has been used in this paper as proof of concept as it has a low-complexity hardware implementation, using a simple adder and shifter.

The proposed algorithm is verified for different sizes of median filter (from 3 to 11 with step-size of 2) and the disparity maps are shown in Fig. 9d and the corresponding error of the obtained disparity maps are shown in Fig. 9c. Disparities obtained from the median filter with a smaller window size have a lower error compared to large sizes due to over smoothing. The complexity of the median filter increases as the window size of the median filter increases. Hence the median filter with a window size of 3x3 has been chosen for the implementation. The disparity map is obtained in two different cases, firstly by applying DWT in both dimension of the image (2D-DWT) and secondly by taking the second level approximate coefficients which are shown in Fig. 8(m,n) for the stereo image Tsukuba. In both cases, the computational complexity has been reduced, since the number of input samples is reduced by 75%, but at the cost of higher error rate

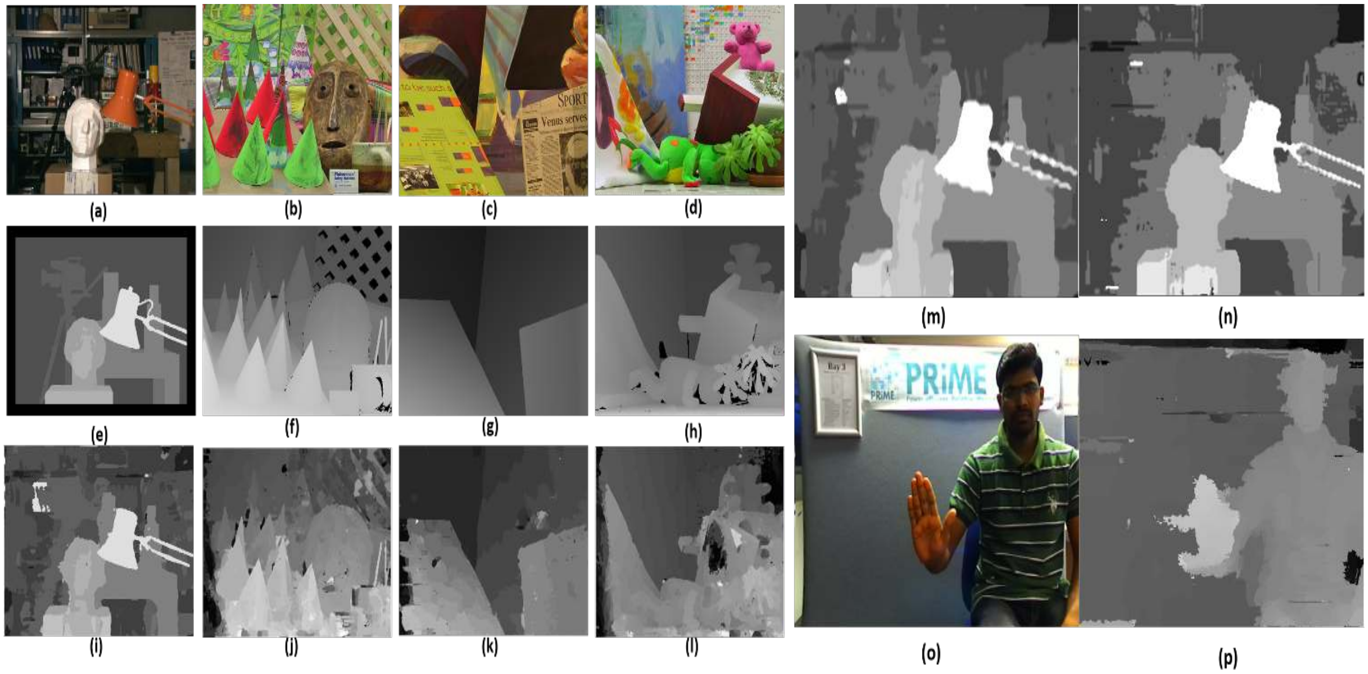


Fig. 8. Left images: (a) Tsukuba (b) Cones (c) Venus (d) Teddy; (e)-(h) Ground truth, (i)-(l) Outcome from the proposed methodology, Obtained disparity by applying (m) 2nd level DWT, (n) 2D DWT, (o) Real time left stereo image (p) Obtained disparity using proposed method.

TABLE II
COMPARISON OF ACCURACY USING THE MIDDLEBURY BENCHMARK. NONOCC: AVERAGE PERCENTAGE OF BAD PIXELS IN NON-OCCLUDED REGIONS; ALL: AVERAGE PERCENTAGE OF BAD PIXELS IN ALL REGIONS; DISC: AVERAGE PERCENTAGE OF BAD PIXELS IN DISCONTINUOUS REGIONS.

	Platform	Tsukuba			Venus			Teddy			Cones			Average percent
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	bad pixel
ADCensus [33]	GPU	1.07	1.48	5.73	0.09	0.25	1.15	4.10	6.22	10.9	2.42	7.25	6.95	3.97
Wenqiang [9]	FPGA	2.39	3.27	8.87	0.38	0.89	1.92	6.08	12.1	15.4	2.12	7.74	6.19	5.61
Jin et al. [23]	FPGA	1.66	2.17	7.64	0.40	0.60	1.95	6.79	12.4	17.1	3.34	8.97	9.62	6.05
Christos [21]	FPGA	2.38	3.01	9.38	0.40	0.7	3.62	7.23	12.7	17.2	2.87	8.59	8.27	6.36
Chang et al. [14]	DSP	-	2.80	-	-	0.64	-	-	13.7	-	-	10.1	-	6.81
SemiGlobal [34]	CPU	3.26	3.96	12.8	1.00	1.57	11.3	6.02	12.2	16.3	3.06	9.75	8.90	7.50
Banz et al. [15]	FPGA	4.1	-	-	2.7	-	-	11.4	-	-	8.4	-	-	noncc. = 6.7
MCADSR [35]	FPGA	3.62	4.15	14.0	0.48	0.87	2.79	7.54	14.7	19.4	3.51	11.1	9.64	7.65
Zhang et al. [25]	FPGA	3.84	4.34	14.2	1.20	1.68	5.62	7.17	12.6	17.4	5.41	11.0	13.9	8.20
Tree structured DP [36]	FPGA	1.43	2.51	6.60	2.37	2.97	13.1	8.11	13.6	15.5	8.12	13.8	16.4	8.71
Proposed	FPGA	4.5	6.72	11.34	2.3	3.88	9.8	8.61	14.2	18.04	7.28	14.67	15.07	9.73
S.perri [10]	FPGA	11.8	11.4	39.1	7.02	5.49	15.5	18.9	8.09	21	18	4.79	13.54	14.55
Jin et al. [5]	FPGA	9.79	11.6	20.3	3.59	5.27	36.8	12.5	21.5	30.6	7.34	17.6	21.0	17.2
Shan et al. [37]	FPGA	-	24.5	-	-	15.7	-	-	15.1	-	-	14.1	-	all = 17.3

of 26% for 2D-DWT and 22.6% for the second level DWT.

Fig. 10a shows a comparison of the proposed methodology with the state-of-the-art methodology of [20] in terms of computational complexity (i.e, number of pixels processed at every stage of algorithm). The x-axis shows the different stages of algorithm - DWT, CVC, GIF, WTA, IDWT and post processing as described in Section III-B, and the y-axis shows the number of pixels processed at every stage (where 1 is equivalent to size of an input image). The computational complexity of the proposed methodology is significantly reduced due to the reduction in the number of samples at the initial stage of the algorithm due to the application of the DWT. Fig. 10b gives the comparison of the latest designs [9], [22], [23], [35] with the proposed methodology in terms of the number of cycles needed for computing the disparity of a single frame ($CPF = Frequency/FPS$). The performance of proposed methodology does not degrade with the increase in image size.

Table III gives the comparison of the proposed methodology with related works in terms of quality (error), performance and hardware resource utilization. The resource utilization includes post-processing after FPGA-based optimizations. LUT utilization of proposed system is the least of all the approaches at the expense of 3.1 higher error on average compared to state-of-the-art designs [9], [21], [22], [35]. The error incurred is attributed to the application of the DWT and computing the approximate coefficients from the first resolution level of the DWT. Table IV provides the FPGA resource utilization for the system designed based on the proposed methodology in terms of LUTs, Registers, DSPs and BRAMs where core-$\{16, 32, 64\}$ is the maximum disparity range number. Module-

TABLE III
QUALITY, SPEED AND ERROR COMPARISON WITH THE RELATED WORKS.(N.A = NOT APPLICABLE, N.M = NOT MENTIONED)

Work	Image	D_{max}	Speed (fps)	MDE/s (10^6)	Error	platform	LUT's	Slice Registers
ADSW [38]	320*240	30	0.01	0.0263	6.53	CPU	N.A	N.A
Chang [14]	352*288	64	42.5	272.5	N.A	ASIC	N.A	N.A
Jin [5]	640*480	64	230	4522	16.5	FPGA	114214	N.M
Ambrosch [6]	750*400	60	60	1080	12.5	FPGA	139606	N.M
Banz [15]	640*480	128	103	4050	6.7	FPGA	68427	N.M
Ding [11]	640*480	60	51	940	11.9	FPGA	50585	35020
Hosni [20]	640*480	26	25	200	5.57	GPU	N.A	N.A
Zhang [25]	1024*768	64	60	3019	8.20	FPGA	53095	74109
MCADSR [35]	1024*768	128	129	13076	7.65	FPGA	60160	33291
Perri [10]	640*480	60	45	829	10.09	FPGA	N.M	80270
Jin [23]	1024*768	60	199.3	9362	6.05	FPGA	122900	N.M
Ttofis [13]	640*480	64	60	1179	9.79	FPGA	88791	117260
Christos [22]	1280*720	64	60	3538	6.80	FPGA	57492	71192
Wenqiang [9]	1024*768	96	67.8	10472	5.61	FPGA	125255	81092
Proposed	1280*720	64	103	6075	9.73	FPGA	34181	47368

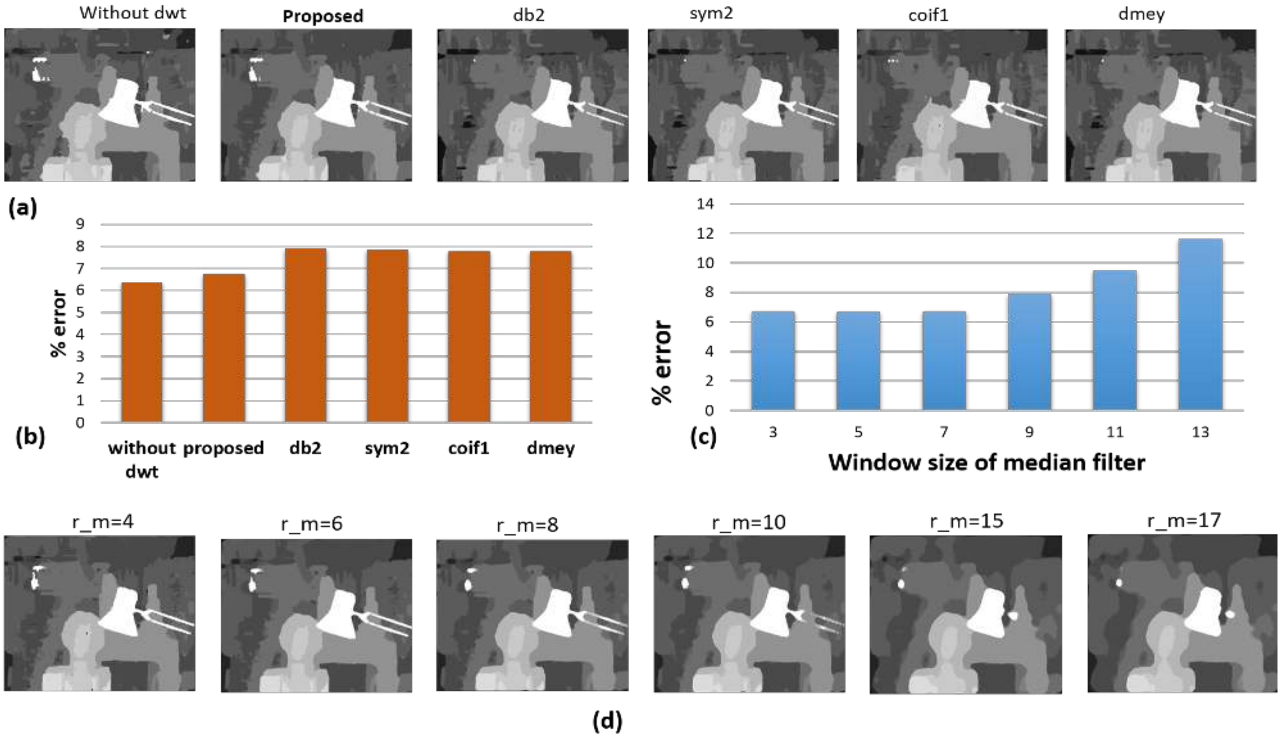
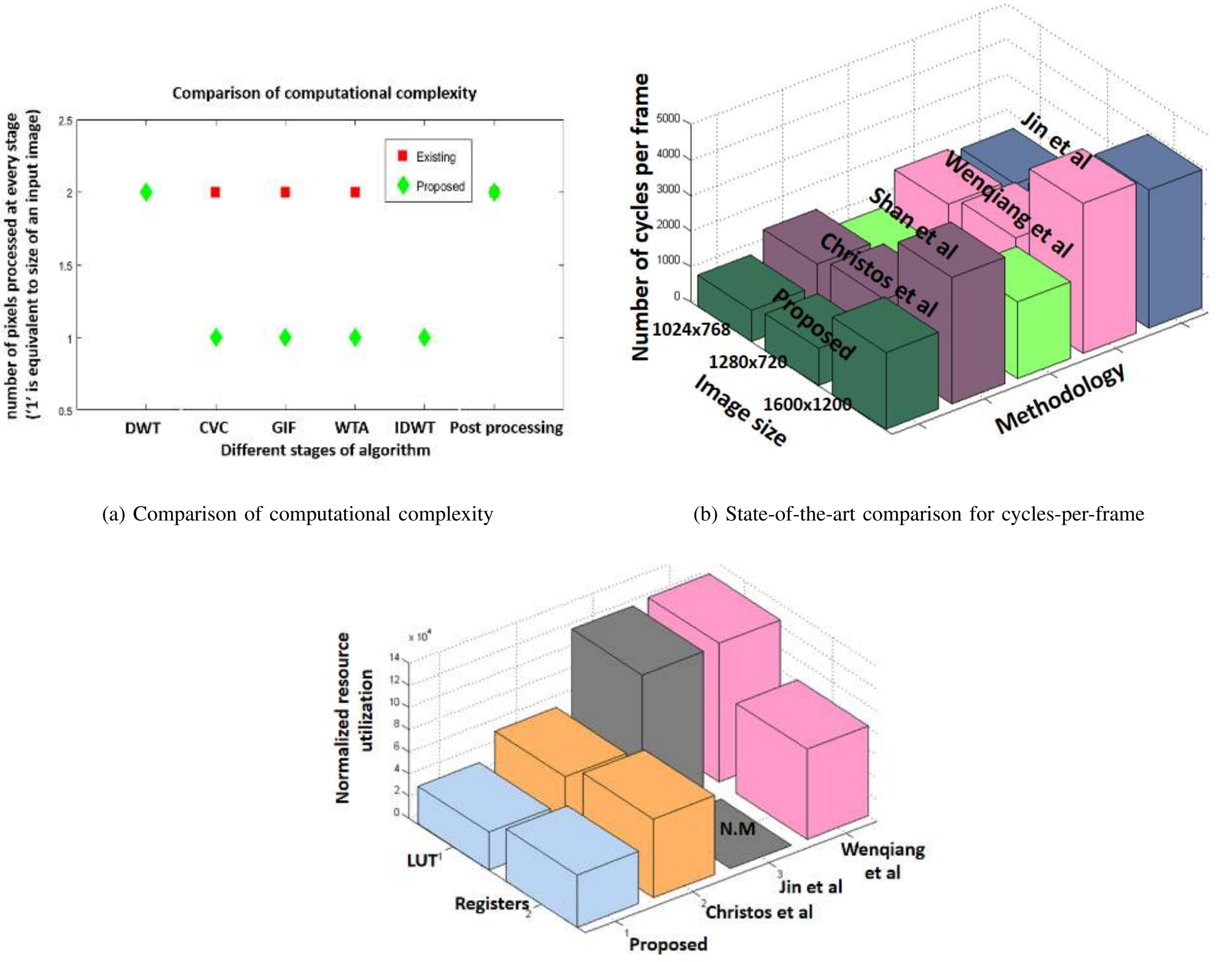


Fig. 9. (a) Obtained disparity for different wavelets where, 'db2' belongs to Daubechies family of wavelets, 'sym2' belongs to Symlets family of wavelets, 'coif1' belongs to Coiflets family of wavelets, 'dmey' belongs to Symlets family of wavelets, (b) Obtained disparity error for different wavelets, (c) Obtained error for Tsukuba image for different size of median filter (x-axis, y-axis represents window size of median filter and error when compared with ground-truth disparity respectively), (d) Error analysis by varying median filter widow size r_m over the range $\{3, 5, 7, 9, 11, 13\}$.

wise resource utilization for the proposed design is shown in Table V. Table VI gives a comparison of the proposed methodology with state-of-the-art designs in terms of CPF. This table shows that a 44.4% improvement is achieved by the proposed methodology as compared to the design of [23] for all the image resolutions. For the image size $1280x720$, the proposed methodology yields about 39.5% better performance in terms of CPF when compared with [22] because of the reduction in the number of samples.

A comparison of the system designed in terms of frequency, power and FPS is given in Table VII. A 52% improvement in FPS is observed when the proposed methodology is compared with one of the recently reported design [9]. In particular LUT utilization is reduced by 61.5%, 40.5%, 43.2%, 72.7%, 35.6% when compared to latest reported results in [9], [13], [22], [25], [35] respectively. Slice registers utilization is reduced by 59.6%, 33.46%, 41.6%, 36% when compared to [9], [13], [22], [25] respectively. A detailed comparison is given in Table VIII and Fig. 10c.



(a) Comparison of computational complexity

(b) State-of-the-art comparison for cycles-per-frame

(c) State-of-the-art comparison for resource utilization

Fig. 10. Jin *et. al.* [23], Christos *et. al.* [22], Wenqiang *et. al.* [9]. (a) Comparison of computational complexity between the proposed design and [20] (b) Comparison to [9], [22], [23], [35] in terms of CPF and (c) Comparison in terms of FPGA resource utilization LUTs, Registers (N.M=Not mentioned).

TABLE IV
FPGA RESOURCE UTILIZATION BY THE SYSTEM DESIGNED BASED ON THE PROPOSED METHODOLOGY.

Design	LUT	Slice register	DSP 48E	BRAM
Stereo Matching core-16	8709	12561	68	56
Stereo Matching core-32	18256	24157	129	120
Stereo Matching core-64	34181	47368	273	247

TABLE V
MODULE-WISE RESOURCE UTILIZATION

Module Name	LUT	Slice register	DSP 48E	BRAM
DWT	25	42	0	0
CVC	9328	5320	0	0
GIF	21053	39953	273	228
WTA	494	275	0	0
LR-Check Filling	515	720	0	0
Median filter	366	328	0	0

TABLE VI
NUMBER OF CYCLES PER FRAME COMPARISON (X10^3)

Methodology/Image Size	1024x768	1280x720	1600x1200
Jin 2014 [23]	1595	1869	3895
Shan 2014 [35]	790	926	2170
Wanq 2015 [9]	1730	2027	4224
Christos 2016 [22]	1464	1716	3576
Proposed	886	1038	2164

TABLE VII
FPS, FREQUENCY AND POWER COMPARISON (N.M= NOT MENTIONED)

	Image Size	Frequency (MHz)	Power (W)	FPS
Jin 2014 [23]	1024x768	318	10.6	199.3
Shan 2014 [35]	1024x768	102	N.M	129
Wang 2015 [9]	1024x768	180	N.M	67.8
Christos 2016 [22]	1280x720	103	2.8	60
proposed	1280x720	107	2.1	103

V. CONCLUSION

In this paper we introduced a DWT based methodology into a state-of-the-art disparity estimation algorithm, resulting in a significant performance improvement in terms of speed and computational complexity. The performance of a system designed from the proposed methodology has been evaluated against standard Middlebury benchmarks that are widely used in evaluating the quality of stereo matching algorithms.

TABLE VIII
HARDWARE RESOURCE UTILIZATION COMPARISON WITH STATE-OF-THE-ART DESIGNS (N.M= NOT MENTIONED)

Work	Image Size	LUT	Slice Register
Jin 2014 [23]	1024x768	122900	N.M
Shan 2014 [35]	1024x768	60160	33291
Wang 2015 [9]	1024x768	125255	81092
Christos 2016 [22]	1280x720	57492	71192
Proposed	1280x720	34181	47368

The four image pairs of *Tsukuba*, *Venus*, *Teddy* and *Cones* have been used to test the proposed system. It has been demonstrated that the system achieves an improvement of 44.4% cycles per frame, 52% frames per second and 61.5% and 59.6% LUT and registers utilization respectively on an FPGA compared with state-of-the-art designs. We believe that our system has significant impact for applications in autonomous vehicles and mobile robotics by meeting real-time processing requirements in a resource-constrained scenario.

REFERENCES

[1] B. Cyganek and J. P. Siebert, *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2011.

[2] 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*, vol. 11, no. 1, pp. 5–25, 2016.

[3] R. A. Hamzah and H. Ibrahim, “Literature survey on stereo vision disparity map algorithms,” *Journal of Sensors*, 2016.

[4] R. Kalarot, J. Morris, and G. Gimel’farb, “Performance analysis of multi-resolution symmetric dynamic programming stereo on gpu,” in *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*. IEEE, 2010, pp. 1–7.

[5] S. Jin, J. Cho, X. Dai 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. 20, no. 1, pp. 15–26, 2010.

[6] K. Ambrosch and W. Kubinger, “Accurate hardware-based stereo vision,” *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1303–1316, 2010.

[7] N. Baha and S. Larabi, “Accurate real-time neural disparity map estimation with fpga,” *Pattern Recognition*, vol. 45, no. 3, pp. 1195–1204, 2012.

[8] S. K. Gehrig, F. Eberli, and T. Meyer, “A real-time low-power stereo vision engine using semi-global matching,” in *International Conference on Computer Vision Systems*. Springer, 2009, pp. 134–143.

[9] W. Wang, J. Yan, N. Xu, Y. Wang, and F.-H. Hsu, “Real-time high-quality stereo vision system in fpga,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 10, pp. 1696–1708, 2015.

[10] S. Perri, P. Corsonello, and G. Cocorullo, “Adaptive census transform: A novel hardware-oriented stereovision algorithm,” *Computer Vision and Image Understanding*, vol. 117, no. 1, pp. 29–41, 2013.

[11] J. Ding, J. Liu, W. Zhou, H. Yu, Y. Wang, and X. Gong, “Real-time stereo vision system using adaptive weight cost aggregation approach,” *EURASIP Journal on Image and Video Processing*, vol. 2011, no. 1, p. 1, 2011.

[12] C. Colodro-Conde, F. J. Toledo-Moreo, R. Toledo-Moreo, J. J. Martínez-Álvarez, J. G. 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.

[13] C. Ttofis, C. Kyrkou, and T. Theocharides, “A hardware-efficient architecture for accurate real-time disparity map estimation,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 14, no. 2, p. 36, 2015.

[14] N. Y.-C. 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. 20, no. 6, pp. 792–805, 2010.

[15] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, “Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation,” in *Embedded Computer Systems (SAMOS), 2010 International Conference on*. IEEE, 2010, pp. 93–101.

[16] C. Prabhakar and K. Jyothi, “A wavelet-based multiresolution approach to stereo matching of face images,” in *Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on*. IEEE, 2012, pp. 316–320.

[17] T. Gao, “3-d video based disparity estimation and object segmentation,” *Global Applications of Pervasive and Ubiquitous Computing*, p. 194, 2012.

[18] S. Kumar, S. Kumar, N. Sukavanam, and B. Raman, “Human visual system and segment-based disparity estimation,” *AEU-International Journal of Electronics and Communications*, vol. 67, no. 5, pp. 372–381, 2013.

[19] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[20] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother, “Real-time local stereo matching using guided image filtering,” in *2011 IEEE International Conference on Multimedia and Expo*. IEEE, 2011, pp. 1–6.

[21] C. Ttofis and T. Theocharides, “High-quality real-time hardware stereo matching based on guided image filtering,” in *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 356.

[22] C. Ttofis, C. Kyrkou, and T. Theocharides, “A low-cost real-time embedded stereo vision system for accurate disparity estimation based on guided image filtering,” *IEEE Transactions on Computers*, vol. 65, no. 9, pp. 2678–2693, 2016.

[23] M. Jin and T. Maruyama, “Fast and accurate stereo vision system on fpga,” *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 7, no. 1, p. 3, 2014.

[24] 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.

[25] L. Zhang, K. Zhang, T. S. Chang, G. Lafruit, G. K. Kuzmanov, and D. Verkest, “Real-time high-definition stereo matching on fpga,” in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*. ACM, 2011, pp. 55–64.

[26] A. Acharyya, K. Maharatna, B. M. Al-Hashimi, and S. R. Gunn, “Memory reduction methodology for distributed-arithmetic-based dwt/idwt exploiting data symmetry,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 4, pp. 285–289, 2009.

[27] C. Zhang, C. Wang, and M. O. Ahmad, “A pipeline vlsi architecture for high-speed computation of the 1-d discrete wavelet transform,” *IEEE transactions on circuits and systems i: regular papers*, vol. 57, no. 10, pp. 2729–2740, 2010.

[28] M. Martina and G. Masera, “Low-complexity, efficient 9/7 wavelet filters vlsi implementation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 11, pp. 1289–1293, 2006.

[29] K. G. Oweiss, A. Mason, Y. Suhail, A. M. Kamboh, and K. E. Thomson, “A scalable wavelet transform vlsi architecture for real-time signal processing in high-density intra-cortical implants,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1266–1278, 2007.

[30] M. Martina and G. Masera, “Multiplierless, folded 9/7–5/3 wavelet vlsi architecture,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 9, pp. 770–774, 2007.

[31] A. Sanny and V. K. Prasanna, “Energy-efficient median filter on fpga,” in *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*. IEEE, 2013, pp. 1–8.

[32] D. Prokin and M. Prokin, “Low hardware complexity pipelined rank filter,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 6, pp. 446–450, 2010.

[33] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, “On building an accurate stereo matching system on graphics hardware,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 467–474.

[34] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[35] Y. Shan, Y. Hao, W. Wang, Y. Wang, X. Chen, H. Yang, and W. Luk, “Hardware acceleration for an accurate stereo vision system using mini-census adaptive support region,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 4s, p. 132, 2014.

[36] M. Jin and T. Maruyama, "A real-time stereo vision system using a tree-structured dynamic programming on fpga," in *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays*. ACM, 2012, pp. 21–24.

[37] Y. Shan, Z. Wang, W. Wang, Y. Hao, Y. Wang, K. Tsoi, W. Luk, and H. Yang, "Fpga based memory efficient high resolution stereo vision system for video tolling," in *Field-Programmable Technology (FPT), 2012 International Conference on*. IEEE, 2012, pp. 29–32.

[38] 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.

Charan Kumar Vala received his masters degree in microelectronics and very large scale integration (VLSI) from Indian Institute of Technology Hyderabad, INDIA. Currently he is a research scholar at School of Electronics and Computer Science, University of Southampton, U.K. His research interests include VLSI architectures, FPGA implementation, VLSI for Cyber physical systems, low-power design techniques, and analog/mixed-signal ASICs. He is runner-up in all India Cadence Circuit Design Contest.

Koushik Immadisetty received the B.Tech in Electrical Engineering from Indian Institute of Technology Hyderabad, Hyderabad, India, in 2017. His research interests include Efficient algorithms and architectures for image/video processing, stereo vision, VLSI architectures, low-power design techniques. He is currently working in Qualcomm Innovation Center Inc., India as a associate engineer.

Amit Acharyya (M11) received the Ph.D. degree from the School of Electronics and Computer Science, University of Southampton, U.K., in 2011. He is currently an Assistant Professor with IIT Hyderabad, Hyderabad, India. His research interests include signal processing algorithms, VLSI architectures, low power design techniques, computer arithmetic, numerical analysis, linear algebra, bioinformatics, and electronic aspects of pervasive computing.

Charles Leech is a senior research assistant for the PRiME project at the University of Southampton where he received his 1st class hons BEng Electronic Engineering degree. His work focuses on power and performance optimisation of computer vision applications on heterogeneous embedded systems. Additionally, he is involved in the development of software frameworks for runtime management of many-core systems. He has interests in approximate computing, stereo vision, and machine learning for embedded devices.

Vibishna Balagopal is a Senior Electronics Design Engineer at LumiraDx Technology, UK. She was awarded a first-class honors degree in Electronics and Communication Engineering from University of Calicut, India in 2008 and completed her Post Graduation Diploma in VLSI design in 2009 from National Institute of Electronics and IT, India. Currently, she is working in digital system design in embedded/ FPGA platforms for Point-Of-Care medical devices. Her research interests include run-time energy management in many/ multi-core Embedded systems. Previously, she was working for the design and development of beam steering controller for AESA Radar application. She was working as Senior Research Assistant at University of Southampton UK, Scientist at Electronics and Radar Development Establishment - DRDO, India and Project Engineer at Wipro Technologies, India.

Geoff V Merrett (GSM06-M09) received the B.Eng. degree (Hons.) in electronic engineering and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 2004 and 2009, respectively.

He is currently an Associate Professor in electronic systems with the University of Southampton. His current research interests include low-power and energy harvesting aspects of embedded & mobile systems. He has published over 100 articles in journals/conferences in the above areas.

Dr. Merrett was the General Chair of the Energy Neutral Sensing Systems Workshop from 2013 to 2015. He is a fellow of the The Higher Education Academy.

Bashir M. Al-Hashimi (M99-SM01-F09) is an ARM Professor of Computer Engineering, Dean of the Faculty of Physical Sciences and Engineering, and the Co-Director of the ARM-ECS Research Centre, University of Southampton, Southampton, U.K. He has published over 380 technical papers. His current research interests include methods, algorithms, and design automation tools for low-power design and test of embedded computing systems. He has authored or co-authored five books and has graduated 35 Ph.D. students.