Real-time Hybrid Stereo Vision System for HD resolution disparity map

Jiho Chang
changjh@etri.re.kr
Jae-chan Jeong
channij80@etri.re.kr
Dae-hwan Hwang
hdh@etri.re.kr

Intelligent Cognitive Technology
Research Department
Electronics and Telecommunications
Research Institute
Daejeon, Republic of Korea

Abstract

Stereo vision systems have been researched for decades and constitute the traditional method for extracting three-dimensional (3D) real-time depth information from images using sensors. However, passive stereo vision systems show a significant error in processing untextured regions, which are frequent in indoor environments. Also, modifications for processing in real time make error by using approximate algorithms. In this paper, we propose a hybrid stereo matching system that combines active and passive stereo vision. In order to implement it in a field-programmable gate array (FPGA) for real-time processing, we modify the existing stereo matching algorithm. Our system shows a significant improvement over current systems in processing untextured regions, and accurately calculates depth for a 1280 x 720-resolution image at 60fps in indoor environments.

1 Introduction

The use of three-dimensional (3D) depth information has become a research area of significant interest in computer vision in recent years, and has numerous and varied applications in fields such as human-computer interaction, robotics, surveillance, entertainment, etc. Since the release of Microsoft Kinect in particular, there has been rising interest in researching applications based on 3D depth sensors. However, applications that use 3D depth information have trouble detecting thin and long objects, such as a human finger, at a distance of more than three meters due to the limitations of 3D depth sensors and the corresponding algorithms. Stereo matching is a traditional method used to obtain 3D depth information and has been studied for decades. However, it is still difficult to apply stereo matching algorithms to practical devices due to real-time issues as well as the technique's inability to adequately handle untextured regions. Stereo matching can be classified into passive or active depending on whether or not light, such as a laser or a structured light, is projected onto the object. As there is no light or pattern to be generated in a passive approach, results may be inaccurate if the camera captures a large untextured region [3]. Furthermore, when the image is obtained in low illumination conditions, errors due to noise are possible. To reduce these errors and improve the accuracy of measured disparity, a global searching method has been

proposed. However, it requires a large number of computations and shows a blurring effect for thin objects [6].

A local stereo matching method using adaptive support-weight (ASW) cost aggregation was introduced in 2005 by Yoon and Kweon [13]. It involves fewer computations than a global system and produces better results than other local methods. However, ASW requires a large support window to handle untextured regions, and thus needs considerable memory and processing resources to implement a field-programmable gate array (FPGA) or a General-Purpose computation on Graphics Processing Units (GPGPU) for real-time processing [10]. For this reason, several researchers have used approximate algorithms to implement in real-time, but such algorithms have not performed as well as the original ASW method [6] [5].

Recent studies have proposed cross-shaped filtering and information permeability filtering in place of ASW for real-time processing. Information permeability filtering [4] in particular involves a much smaller number of calculations per pixel compared to ASW or approximate algorithms, and its complexity is independent of support window size. Jeon *et al*. [8] have implemented information permeability filtering in GPUs, and Stefano [7] and Aysu *et al*. [1] have implemented it in FPGAs.

In contrast, active stereo vision is a method for calculating the correspondence between the projected patterns of stereo images using illuminators. Active stereo systems reduce errors in untextured regions to a greater degree than passive stereo vision. Furthermore, because active stereo vision employs a light, it can reduce the effect of radiometric distortion generated by illumination variation. Active stereo is typically calculated by the method of space-time with images obtained by projecting different patterns to obtain a highly accurate disparity. However, because of problems with using multiple frames, the implementation of a real-time system with a high frame rate is not possible.

In this paper, we propose a hybrid stereo matching system to remedy the disadvantages of active and passive stereo vision. Our system includes laser diodes to project a pattern of random dots, and obtains pattern images to the left and the right in order to reduce error in untextured regions commonly encountered in real indoor environments. We obtain images through infrared light-emitting diodes (IR LEDs) for cost aggregation by utilizing the support weight. In this case, the cameras capture images without a pattern. We then generate the raw cost by locating the correspondence between the two pairs of images to the left and the right acquired in this way. We then perform cost aggregation by combining the raw cost with support weights by using non-pattern images.

This paper is structured as follows. In Section 2, we explain the stereo matching algorithm. We then detail the configuration of our proposed system in Section 3 and present its block diagram implemented using FPGA. Section 4 contains a performance comparison between our proposed system and current stereo matching algorithms. We also compare the performance of our system with that of Microsoft Kinect. We conclude and suggest avenues for future research in Section 5.

2 Stereo matching algorithm

Following Scharstein and Szeliski's taxonomy [11], we divide the stereo matching algorithm into four steps: matching cost computation, cost aggregation, disparity computation and disparity refinement. Our description of the algorithm takes into account issues in modifications for the hybrid system.

2.1 Matching cost computation (Rawcost)

Matching cost computation is an initial cost computation of the stereo matching algorithm. Methods for matching cost computation can be classified into parametric and non-parametric methods [11]. Parametric methods use pixel intensity absolute difference (AD), but are difficult to implement in the real-world environment because of their sensitivity to parameters of the cameras used as well as external conditions, such as radial difference, over-exposure, etc. On the other hand, non-parametric measures - such as rank, census transform or gradient-based measures - are more robust than parametric methods. They transform intensity data into feature data and thus exhibit high robustness to illumination variations and exposure differences [12].

In this paper, we calculate raw cost volume using the absolute difference (AD)-Census. The reason for combining the AD and Census Transform (CT) cost measures is that the AD-Census provides better matching accuracy than either the individual AD measure or the CT measure [14]. Equations 1 through 5 represent, respectively, the AD measure, the CT measure, cost calculation through hamming distance, cost combining with alpha-blending for the AD-Census, and the final raw cost that is the sum of the pattern cost (T_1) and the non-pattern cost (T_2).

$$C_d^{AD}(p) = \left|I(p) - \bar{I}(p-d)\right| \tag{1}$$

$$CT(x,y) = \begin{cases} 1 & if\ I(x_n,y_n) \geq I(x,y) \\ 0 & else \end{cases}, \ n = -r \sim r \tag{2}$$

$$C_d^{Census}(p) = HammingDistance(CT(p) - \overline{CT}(p-d)) \tag{3}$$

$$C_d^{T_n}(p) = \alpha_{T_n} \cdot C_d^{AD}(p) + (1-\alpha_{T_n}) \cdot C_d^{Census}(p) \tag{4}$$

$$C_d^{T}(p) = \alpha_p \cdot C_d^{T_1}(p) + (1-\alpha_p) \cdot C_d^{T_2}(p) \tag{5}$$

2.2 Cost aggregation

Since Yoon *et al*. proposed the adaptive support-weight approach for cost aggregation, many researchers have used similar cost aggregation methods [13]. The information permeability filtering proposed by Cevahir Cigla *et al*. [4] is one such approach that has simple parameters and provides constant operational time for calculating adaptive-weighted aggregation of cost values.

Cost aggregation using information permeability filtering is calculated in four fundamental directions (left, right, up and down), which are divided into the horizontal and vertical directions to obtain 2D support regions. Cevahir Cigla *et al*. compute the support weight of a pixel based on the strength of grouping by similarity. The following equation defines the update rule for calculating cost aggregation (left-to-right):

$$C_d^{CA_R}[x] = C_d(x) + \mu_R[x-1] \cdot C_d^{CA_R}[x-1], \quad \mu = e^{-\frac{\Delta I}{\sigma}} \tag{6}$$

However, because there is no proximity weight term, information permeability filtering can encounter problems with images containing large untextured regions. It is possible to attain a smaller bit size of aggregate cost than conventional methods by using a proximity weight term when implementing the hardware. Modified information permeability(MPF), including a proximity weight term, is defined as follows.

$$\mu = e^{-\frac{1}{\sigma_p}} \cdot e^{-\frac{\Delta I}{\sigma_s}} \tag{7}$$

$$C_d^{CA_R}[x] = C_d(x) + \mu_H[x-1] \cdot C_d^{CA_R}[x-1] \tag{8}$$

$$C_d^{CA_L}[x] = C_d(x) + \mu_H[x+1] \cdot C_d^{CA_R}[x+1] \tag{9}$$

$$C_d^{CA_H}[x] = C_d^{CA_L}[x] + C_d^{CA_R}[x] \tag{10}$$

$$C_d^{CA_T}[x,y] = C_d^{CA_H}(x,y) + \mu_V[x,y-1] \cdot C_d^{CA_T}[x,y-1] \tag{11}$$

$$C_d^{CA_B}[x,y] = C_d^{CA_H}(x,y) + \mu_V[x,y+1] \cdot C_d^{CA_T}[x,y+1] \tag{12}$$

2.3 Disparity computation

Disparity computation involves the calculation of the disparity that has been properly matched with the results of the calculation of raw cost. In the taxonomy of stereo matching, the classification between global systems and local systems is made on account of a difference in this step. Typically, a window with support weight uses the winner-take-all (WTA) method, which is used to select the disparity in minimum aggregated cost in the calculation of cost volume, as described in Section 2.2. WTA is simple and is represented by the following equation:

$$d(p) = \arg\min_{d \in S_d} (C_d^{CA}(p)) \tag{13}$$

Where S_d is the set of all possible disparities.

2.4 Disparity refinement

Disparity refinement reduces the error generated due to various problems during stereo matching. In this paper, we use left-right consistency (LRC) check and weighted median filtering (WMF) to eliminate error pixels from the results of disparity calculation. We also use sub-pixel estimation and weighted average filtering to obtain a more accurate disparity measure. A left-right consistency check is performed by comparing the left and right disparity maps, and we evaluate the performance of the refinement by a comparison with the ground truth using only the LRC. A quantitative comparison between our proposed system and Microsoft Kinect is difficult because the two have different fields of view (FOVs) and because the images in the latter are also post-processed. Thus, we only qualitatively compare our system with Microsoft Kinect.

3 Stereo vision system design

In this section, we describe our proposed system, which employs a time-multiplexed operation associated projection of the laser diode (LD)/LED. The system is composed of two parts, a stereo head and a stereo emulator. The stereo head includes a low-voltage differential signaling (LVDS) module for data transmission, an LD/LED projection module and two complementary metal-oxide semiconductor (CMOS) sensors. The stereo emulator is based on FPGA modules to obtain dense disparity maps from high-resolution stereo pairs. Our description of the algorithm takes into account issues in real-time implementation and modifications for implementing it using FPGA.

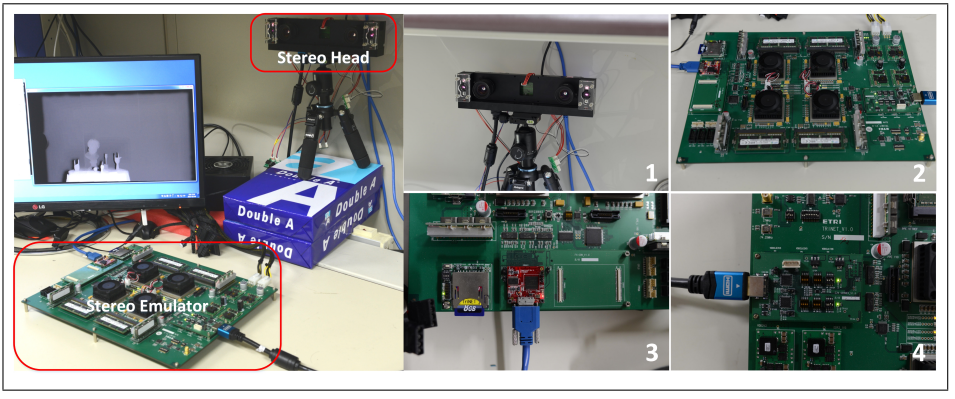


Figure 1: The entire system with stereo head and stereo emulator(1.Stereo head, 2.Stereo emulator, 3.USB3.0 Controller, 4.Deserialze module)

3.1 Stereo head

We use the EV76C661 sensor, which exhibits good quantum efficiency in the IR band. Input from the two CMOS sensors is received in the form of 10-bit monochrome images at a resolution of 1280 x 720 pixels (high-definition (HD) resolution) at a rate of 60fps. Image streams from the left and right cameras are transferred to the FPGA board through the LVDS module, which includes control signals (e.g., clock source, Vactive, Hactive and LD on/off). Video data from the stereo head is transferred to a computer through the USB 3.0 controller. The stereo head also receives control signals from the computer. The on/off cycle of the LD/LED, which is part of the control signal data, is transferred to LD and LED modules after it has been changed to signal the active form inside.

A pattern is made using a diffractive optic element (DOE), which can design the desired pattern using an 808-nm laser in the IR band. The projection uses a pseudorandom pattern designed by taking into account the brightness and density of the pattern. It also has LEDs of the same IR band, using which it acquires non-pattern images. We use the IR band because it is invisible to the human eye and allows us to control the light source.

3.2 Stereo emulator

The Xilinx Virtex-7 Family of FPGAs offers high bandwidth and a large memory. The XC7V2000T FPGA supplies two million logic cells, a 46Mb block random-access memory (RAM) and 2,160 digital signal processing (DSP) slices. These resources enable parallel processing architectures, such as image processing in high resolution. Our implementation is based on four FPGAs using Virtex-7 XC7V2000T.

Each FPGA can access the double data rate (DDR) memory via an interface provided by Xilinx. The stereo emulator contains a deserialized module, a USB 3.0 controller and four FPGAs. The deserialized module receives image data from the stereo head through the LVDS. The USB 3.0 controller transfers the results to the computer. Each of the four FPGAs runs a different algorithm depending on the task at hand, as described in Section 3.4. Figure 1 represents our entire system.

3.3 Timing diagram

The hybrid system proposed in this paper captures a pair of pattern images (left and right) and an alternating pair of non-pattern images to evaluate disparity. Both pairs of images are used to calculate the raw cost, whereas the non-pattern images are also used to generate the weight of the cost aggregation. The CMOS sensors are synchronized with the pattern projector in correct integration time, so that the images are obtained in operating time with the LD (pattern) in one frame and with the LED (non-pattern) in the next frame.

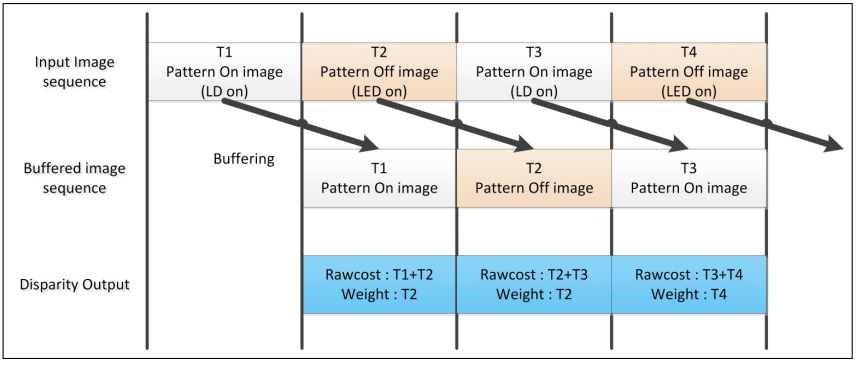


Figure 2: The timing diagram for hybrid stereo matching

As shown in Figure 2, the disparity in the system is calculated using previous frames that are stored and images being received at the time. The raw cost is calculated using T1 and T2 and the cost aggregate is determined using images from T2. In this way, the disparity output of our system has the same frame rate as the input image, and there is no frame dropping.

3.4 Implemeting stereo algorithm

Figure 3 is a block diagram of the stereo vision system. The stereo matching algorithm consists of four processing elements implemented in a single FPGA.

The pre-processing element (PrePE) is first composed through rectification and image filtering. Rectification is one of the most important parts of the stereo matching process because of the epipolar constraint. In order to perform cost aggregation, the PrePE must rectify an imput image under a one-pixel error in horizontal line. We implement rectification in this system based on the Caltech method. Using a square checkerboard, we extract the internal and external parameters from the left/right cameras. These rectification parameters are transferred to the host interface through the USB 3.0 controller. The PrePE then rectifies the pair of input images of each frame. Image filtering improves the results of cost aggregation and is applied, based on bilateral filtering, to the non-pattern images. The pattern selector determines the pattern image between the stored previous image and the current image by using a control signal (pattern projector "on") received from the stereo head.

The Main Processing Element Left (MPE Left) generates a left-referenced disparity that is implemented using the algorithm described in Section 2. In contrast, the Main Processing Element Right (MPE Right) generates a right-referenced disparity. Of course, the implementation can be carried out, using the method referenced by Georgia *et al*. [9], by shifting the cost volume of the left-referenced disparity. However, this does not satisfy the timing requirements of the FPGA.

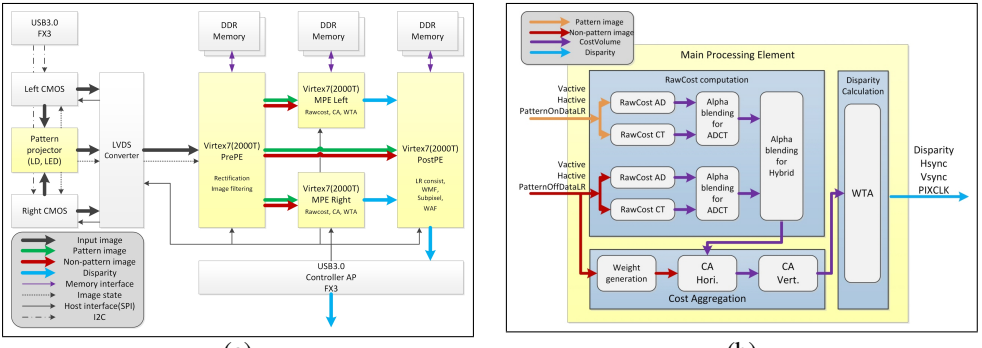


Figure 3: Block diagram of stereo vision system

Certain modifications are taken into account while implementing the system on the FPGA, and usually appear in the cost aggregation. One of the issues is whether the operations can be processed in a limited clock cycle. In order to solve this problem, we allow as much parallel processing and pipeline insertion as possible. Further, the form of the weight data is changed from exponent calculation to a look-up table (LUT). We configure to a power of two to use the data shifter instead of the divider.

Another issue is whether to use a filter for all four directions in cost aggregation. When implementing the FPGA image processing system, data from the CMOS sensors is entered in sequence. Therefore, as in a filter operation, buffering (e.g., register or memory) is required if we need previous data rather than data from the current pixel. In the information permeability filtering method, left-to-right processing is possible in accordance with the input image stream in this case. However, the cost aggregation of right-to-left processing, in the opposite direction from the image stream, cannot be predicted in advance. Thus, it is necessary to store one row of data. In the case of top-to-bottom processing, separate data buffering of each line is required for the same reason. However, in bottom-to-top processing, data from the entire image must be saved, and hence involves horizontal cost volume. If horizontal cost volume data is assumed to be 16-bit, bottom-to-top processing is needed, which requires at least 570MB (1280 x 720 x 2 x 256, when maximum disparity is 256).

For the above reasons, Stefano [7] and Aysu *et al.* [1] implement FPGA in only the horizontal direction. However, this causes streak noise and thus increases the error due to disparity. Our system is configured to run left-to-right, right-to-left and top-to-bottom while maintaining consistent performance and reducing memory consumption. In this case, we save two lines of aggregated cost and two lines of non-pattern images. Equation (11) represents the final cost volume.

Finally, instead of using μ for information permeability filtering, we reduce two bits in

Resource	Utilization(%)	Available
Slice LUTs	431938(35%)	1221600
Slice Registers	755406(31%)	2443200
Memory	679(53%)	1292
DSP	768(36%)	2160

Table 1: Resource utillization for MPE in Virtex7 2000T

the aggregated cost by adding a proximity term. Figure 3 shows a block diagram of the MPE for implementing the proposed hybrid system. Table 1 shows the resource utilization in Virtex7 2000T. The parameters of the MPE are maximum census size = 11 x 11, maximum disparity = 256, and image resolution = 1280 x 720.

The LRC eliminates the error due to disparity calculation results for occlusion regions. Sub-pixel estimation is implemented as a parabola fitting the costs and is calculated to divide 4 bits more than 256 steps that are separated by a maximum disparity. So, steps of final disparity are 4096(256x16). The modification according to the post-processing described above improves the sharpness of the disparity based on the original image and eliminates noise. However, it is difficult to quantitatively measure this performance improvement. Therefore, we propose qualitative results of post-processing for comparison of the shapes of the images. We present and evaluate quantitative measures only for LRC results.

4 Performance Evaluation

We evaluate performance in two ways: quantitative evaluation using the ground truth and qualitative evaluation using Microsoft Kinect.

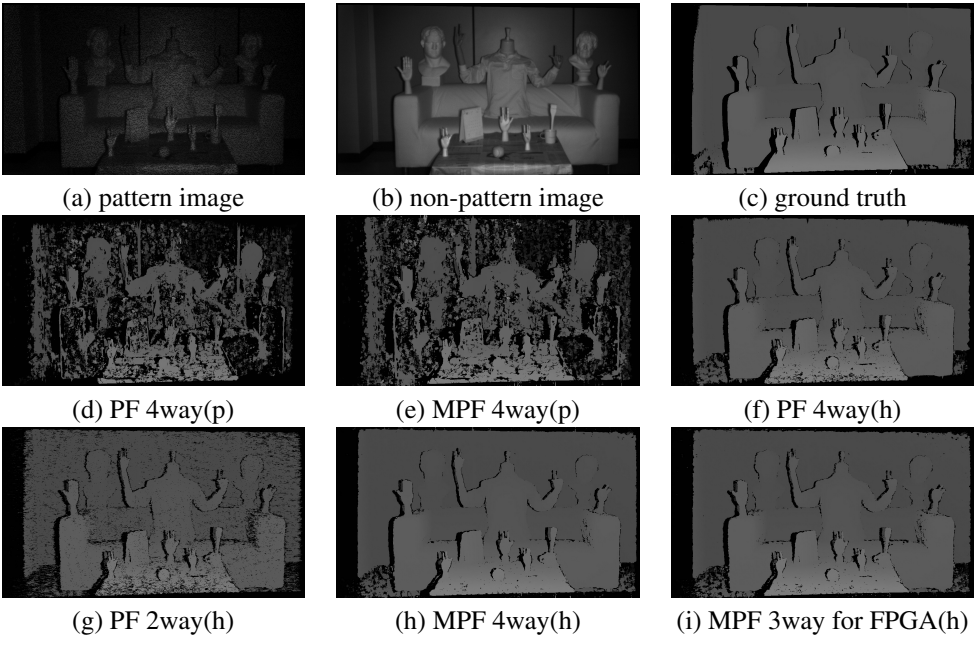
4.1 Quantitative Evaluation

The ground truth is generally created using a space-time stereo in order to evaluate the performance of stereo vision [7]. We create the ground truth using thousands of images captured by moving the pattern. Once we have captured two pairs of images - a pair of pattern images and a pair of non-pattern images - we fix the variable census window size (9 x 9), the value of alpha for ADCT (0.4) and for hybrid (0.8). The performance of the system is then analyzed by using different cost aggregation methods (information permeability, the proposed method, implementation to FPGA) and comparison between passive stereo matching and hybrid system using both of pattern images and non-pattern images.

Table 2 and Figure 4 show the results for each algorithm. We see that our proposed stereo system is better than the original passive stereo algorithm as well as variations of it to address untextured regions. Further, when applying the hybrid approach, the MPF has lesser streak noises than the conventional real-time two-way permeability filtering (PF). We also see that the three-way method for FPGA performs just as well as the four-way method, except for the fact that the sharpness of the object is different for the two methods.

Algorithm		error(%)
Not using pattern image (Passive stereo matching)	PF 4way	61.67%
	MPF 4way	54.54%
Using pattern image (Hybrid system)	PF 4way	5.14%
	PF 2way(Horizontal)	11.39%
	MPF 4way	3.69%
	MPF 3way	4.78%
	MPF 3way for FPGA	3.80%

Table 2: Error pixel rate on non-occlusion regions.



(a) pattern image (b) non-pattern image (c) ground truth

(d) PF 4way(p) (e) MPF 4way(p) (f) PF 4way(h)

(g) PF 2way(h) (h) MPF 4way(h) (i) MPF 3way for FPGA(h)

Figure 4: Disparity result of various algorithm

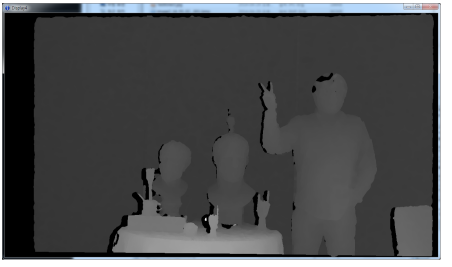
4.2 Qualitative Evaluation

To compare the proposed system with Microsoft Kinect, the field of view (FOV) of the sensors must match. The FOV of Microsoft Kinect is wider than that of our system. Thus, we place it approximately one meter closer to the object than the stereo head in order for both to have the same FOV. Figure shows the comparison. Even at short distances, the object image of Microsoft Kinect is generally blurry. Moreover, Kinect cannot detect long and thin objects, such as a pen or a human finger, at a distance of more than three or four meters. This is because Kinect has a lower resolution and uses structured light. The resulting image of our depth system is much sharper. For instance, it is difficult to distinguish the person's fingers in the Kinect image, whereas these are clearly distinguishable in the image generated by our proposed system.

5 Conclusion and Future work

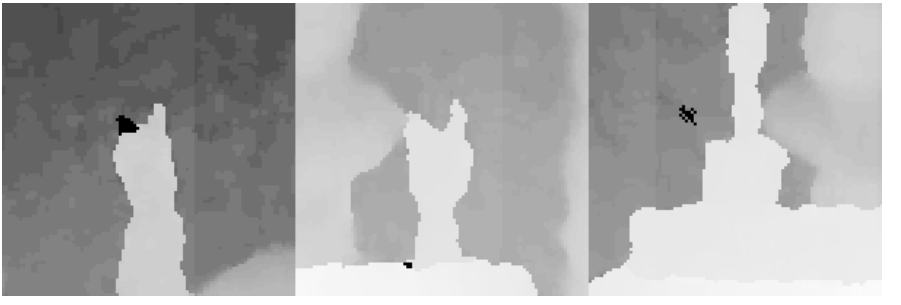
In this paper, we proposed a hybrid stereo matching system that combines active and passive stereo vision. Using the active pattern, our system successfully detected disparity in untextured regions. Through comparison with other algorithms using the ground truth and with Microsoft Kinect, we found that our proposed system exhibits a higher accuracy and better resolution in indoor environments.

The performance of our system suffers when the pattern or LED is weak. We intend to address this in future research. We also plan to improve the algorithm to make the system more robust under varied circumstances.



(a) The proposed hybrid system

(b) Kinect for XBOX360



(c) Detailed disparity results of Kinect for XBOX360

(c) Detailed disparity results of The proposed hybrid system

Figure 5: Qualitative comparison with Kinect

Acknowledgment. This work was supported by the ETRI R&D Program of KCC (Korea Communications Commission), Korea [11921-03001, Development of Beyond Smart TV Technology].

References

[1] A. Aysu, M. Sayinta, and C. Cigla. Low cost fpga design and implementation of a stereo matching system for 3d-tv applications. In *2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 204–209, Oct 2013.

[2] Jae chan Jeong, Hochul Shin, Jiho Chang, Eul-Gyun Lim, Seung Min Choi, Kuk-Jin Yoon, and Jae il Cho. High-quality stereo depth map generation using infrared pattern projection. *ETRI Journal*, 35(6):1011–1020, 2013.

[3] Chichyang Chen and Y.F. Zheng. Passive and active stereo vision for smooth surface detection of deformed plates. *IEEE Transactions on Industrial Electronics*, 42(3):300–306, Jun 1995.

[4] C. Cigla and A.A. Alatan. Efficient edge-preserving stereo matching. *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 696–699, 2011.

[5] Jingting Ding, Jilin Liu, Wenhui Zhou, Haibin Yu, Yanchang Wang, and Xiaojin Gong. Real-time stereo vision system using adaptive weight cost aggregation approach. *EURASIP Journal on Image and Video Processing*, 2011(1):1–19, 2011.

[6] A. Hosni, M. Bleyer, and M. Gelautz. Secrets of adaptive support weight techniques for local stereo matching. *Computer Vision and Image Understanding*, 117(6):620–632, 2013.

[7] S. Mattoccia. Stereo vision algorithms for fpgas. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 636–641, June 2013.

[8] Cuong Cao Pham and Jae Wook Jeon. Domain transformation-based efficient cost aggregation for local stereo matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(7):1119–1130, July 2013.

[9] G. Rematska, K. Papadimitriou, and A. Dollas. A low cost embedded real time 3d stereo matching system for surveillance applications. In *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*, pages 1–6, Nov 2013.

[10] Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, and NeilA. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *Computer Vision âĂŞ ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 510–523. 2010.

[11] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on*, pages 131–140, 2001.

[12] Xun Sun, Xing Mei, shaohui Jiao, Mingcai Zhou, and Haitao Wang. Stereo matching with reliable disparity propagation. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 132–139, May 2011.

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

[14] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision - ECCV '94*, volume 801 of *Lecture Notes in Computer Science*, pages 151–158. 1994.