Real-Time Object Segmentation of the Disparity Map Using Projection-Based Region Merging

Dongil Han
Vision and Image Processing Lab.
Sejong University 98 Kunja-dong, Kwagjin-gu, Seoul
Korea

1. Introduction

Robots have been mostly used in industrial environment, but modern developments of household robot-cleaner suggest the necessity of household robots as becoming in reality. Most industrial robots have been used for factory automation that perform simple and iterative tasks at high speed, whereas household robots need various interfaces with a man while moving in indoor environment like a household robot-cleaner does.

Robots activate in indoor environment using various sensors such as vision, laser, ultrasonic sensor, or voice sensor to detect indoor circumstance. Especially robot's routing plan and collision avoidance need three-dimensional information of robot's surrounding environment. This can be obtained by using a stereo vision camera which provides a general and huge amount of 3-D information. But this computation is too big to solve in real-time with the existing microprocessor when using a stereo vision camera for capturing 3-D image information.

High-level computer vision tasks, such as robot navigation and collision avoidance, require 3-D depth information of the surrounding environment at video rate. Current general-purpose microprocessors are too slow to perform stereo vision at video rate. For example, it takes several seconds to execute a medium-sized stereo vision algorithm for a single pair of images using one 1 GHz general-purpose microprocessor.

To overcome this limitation, designers in the last decade have built reprogrammable chips called FPGA(Field-Programmable Gate Arrays) hardware systems to accelerate the performance of the vision systems. These devices consist of programmable logic gates and routing which can be re-configured to implement practically any hardware function. Hardware implementations allow one to apply the parallelism that is common in image processing and vision algorithms, and to build systems to perform specific calculations quickly compared to software implementations.

A number of methods of finding depth information in video-rate have been reported. Among others, multi-baseline stereo theory is developed and the video-rate stereo machine has the capability of generating a dense depth map of 256x240 pixels at the frame rate of 30 frames/sec in [1-2]. An algorithm proposed from *parallel relaxation algorithm for disparity computation* [3] results reduction of error rate and enhancement of computational complexity

Source: Scene Reconstruction, Pose Estimation and Tracking, Book edited by: Rustam Stolkin,
ISBN 978-3-902613-06-6, pp.530, I-Tech, Vienna, Austria, June 2007

of problems. Also, an algorithm proposed from *depth discontinuities by pixel-to pixel stereo* [4] is concentrated on the calculation speed and rapidly changing disparity map. It is not possible to search for the exact depth of the discontinuities when there is no change in lightness of boundary. Also the high-accuracy stereo technique [5] mentioned the difficulty of drawing sharp line between intricate occlusion situations and some highly-slanted surfaces (cylinder etc.), complex surface shapes and textureless shapes. Nevertheless, for algorithm suggested in this chapter, we can use the post-processing as first half of process to get more neat disparity map produced by other many stereo matching algorithms, which can be used for the object segmentation.

To embody object segmentation, we used hardware-oriented technology which reduces tasks of the software, contrary to conventional software-oriented method. Also, it has great effectiveness that reduces software processing time by the help of real-time region data support, which containing various kinds of object information, that reduces total area of search process such as object or face recognition. Use of embedded software based on low-cost embedded processor, compare to use of high-tech processor, to conduct tasks of object recognition, object tracking, etc in real-time provides a suggestion of a household robot application.

This chapter is organized as follows: Section 2 describes a brief review of proposed algorithm. Section 3 explains refinement block while Section 4 explains segmentation. At the conclusion, the experimental results including results of depth computation and labeling are discussed in Section. 5

2. Algorithm Overview

In this chapter, we attempted to make clearer object segmentation using projection-based region merging of disparity map produced by applied trellis-based parallel stereo matching algorithm described in [6]. Throughout this experiment, we verified the performance. Necessity of post-processing algorithm application for many different characterized stereo matching has been ascertained through various experiment performed in this chapter.

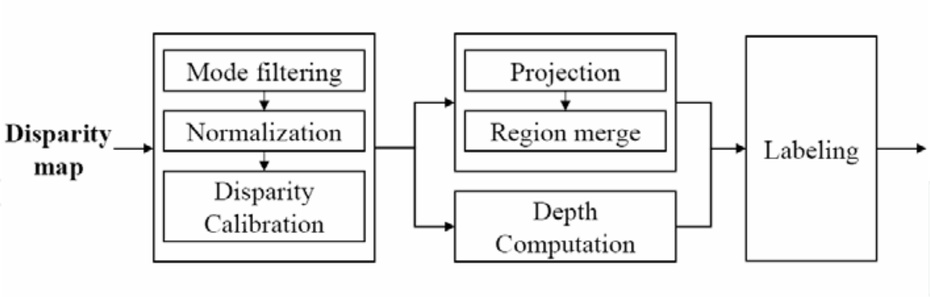


Figure 1. Block diagram of the post processing algorithm

The block diagram of the proposed post-processing algorithm is shown in figure 1. The post-processing algorithm is progressed in three big stages. The first stage is the refinement block, which carries normalization referenced from filtering and disparity max value, and elimination of noise using histogram consecutively. In second stage, the depth computation which helps to find out the distance between camera and original objects on disparity map and the image segmentation which takes responsibility of object partition are accomplished in a row. Finally in the last stage, information of object existed in original image is gathered and integrated with all information proposed in second stage.

The cause of noise in disparity map can be textureless object, background video, or occlusion etc. In stereo matching algorithm, possibility of textureless object and occluded area must be necessarily considered, but even through consideration has been applied, precise result may not be processed. Therefore, refinement stage like filtering must be included on the first half of post-processing to be able to segment the object with much more clear disparity map.

3. Refinement

In this stage, we try to obtain purified disparity map by the utilization of disparity calibration algorithm which used for mode filtering of disparity map out of trellis-based parallel stereo matching algorithm, with the normalization, and disparity calibration.

3.1 Mode filtering

The noise removal techniques in image and video include several kinds of linear and nonlinear filtering techniques. Through out the experiment, we adopted the mode filter technique for preserving boundary of image and effective removal of noise. The window size used for filtering has been fixed to 7x7, considering the complexity and performance of hardware when it is implemented. The numerical equation used for mode filtering is as follow:

$$C_i = \begin{cases} C_i + 1 & (D_{ij} = 0), 0 \le j < k \\ C_i & (D_{ij} \ne 0), 0 \le j < k \end{cases} \tag{1}$$

Here,

$$D_{ij} = x_i - x_j (0 \le i < k, 0 \le j < k) \tag{2}$$

And then, we can get

$$X_m = \begin{cases} x_i \;\; for \max_i(\forall C_i) & (\forall C_i \ne 1) \\ x_{center} & (\forall C_i = 1) \end{cases} \tag{3}$$

In equation (1) and (2), the value of *k* represents the window size. In this chapter, 7x7=49 is used. From equation (2), with given disparity map input x_i , and only changing the argument of pixel value *j* in the 7x7 window, we can calculate the difference between two pixel values. When D_{ij} value is 0 in equation (1), we increase the C_i value by one. If we can find the largest value of $C_{i,}$ then the mode value X_m can be decided. If all the values of x_i are different, we can not find the maximum value of C_i. In this case, we select and decide on the center value of window, x_{center}(window size 7x7 has been used in this chapter, thus x_{24} should be utilized).

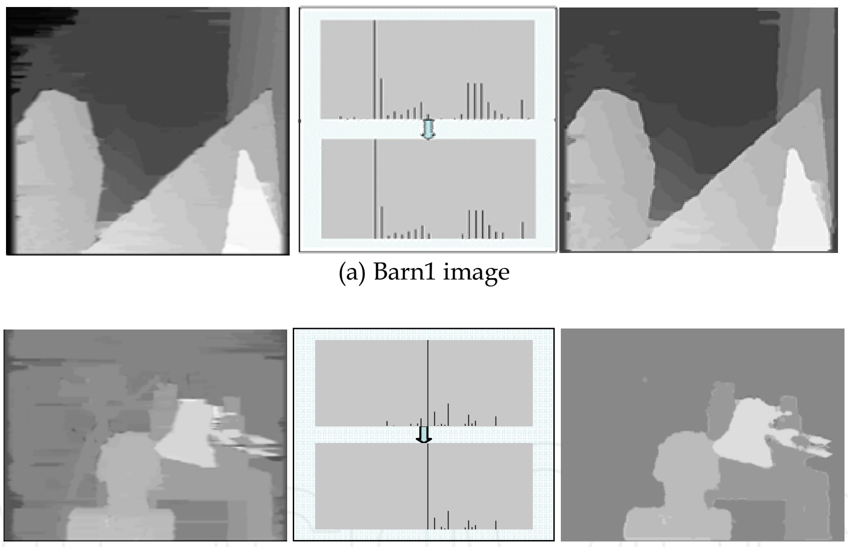
3.2 Normalization

After the mode filtering, noise removed disparity map can be obtained. Then by using the disparity max value used for getting the stereo matching image, the disparity values of mode filtered image are mapped out new normalized values with regular and discrete intervals.

The disparity max value can be decided in the stereo matching stage, which is the value to decide the maximum displacement of matching pixels which can be calculated from the left image to right image. In normalization stage, disparity map pixels, composed of 0~255 gradation values, is divided into 0~disparity max range (in barn1 image, disparity max value is 32). This process removes unnecessary disparity map. The value of 0~disparity max range is again multiplied to the pixel values calculated before, and finally restored to 0~255 gradation values.

3.3 Disparity Calibration

In disparity calibration stage, which is the final stage of refinement, the normalized disparity value is accumulated to form a histogram of each frame. During accumulation process, we ignore the disparity value under the given threshold value to remove the noise in dark area.



(a) Barn1 image

(b) Tsukuba image

Figure 2. The result of disparity calibration (*left: stereo matching result, middle: histogram comparison, right: calibrated disparity map*)

And in this histogram, the data under the predetermined frequency level can also be considered as noise. Thus, after the formation of the histogram, the accumulated pixel data are sorted out according to the frequency. The upper part of the histogram which consists of approximately 90% of total histogram area holds their pixel values. About the pixel frequency which does not reach the given specific threshold, the nearest value is selected among the accumulated pixel values which belong to the upper part of the sorted histogram. The center part of figure 2 (a) and (b) shows the histogram data before and after the disparity calibration. And the right part of figure 2 (a) and (b) shows the tsukuba and barn1 image after the disparity calibration stage.

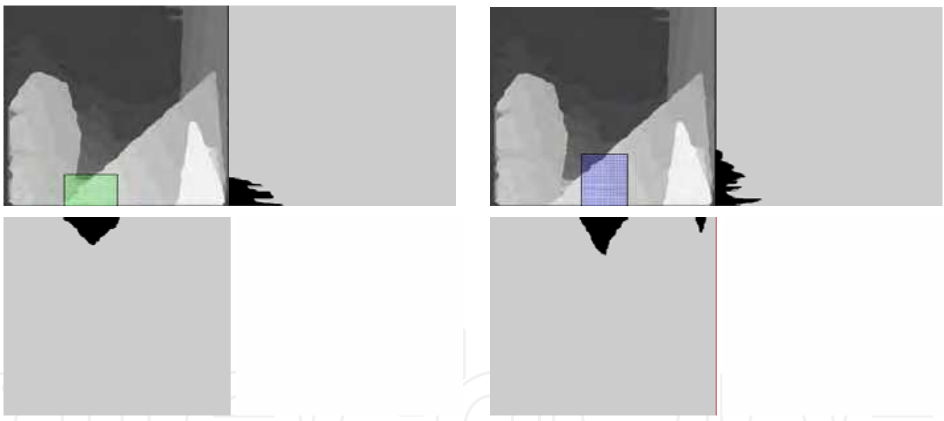
4. Segmentation

The objective of this block is to separate objects from the disparity map and to partition slanted objects to other objects. In this chapter, to achieve the objectives, we conducted the horizontal and vertical projection for each level of disparity map and sequential region merging with projection results.

4.1 Projection

The task to separate object from the distance information is completed by processing horizontal and vertical projection of each disparity map. The results of specific projections are shown in figure 3.

Using the horizontal and vertical projection for each disparity level, the region data for all level of disparity map could be obtained, and the horizontal position information of a region data is expressed by starting and ending point of vertical direction projection $P_x(n)=(X_s(n), X_e(n))$, while the vertical position information of a region data is expressed by starting and ending point of horizontal direction projection $P_y(n)=((n), Y_e(n))$. Also a region data is represented as $R(n)=(P_x(n), P_y(n))$.



(a) Second level (b) Third level

Figure 3. The projection examples about each disparity level

4.2 Region Merge

Whether to merge the region or not can be decided after all of the region information about each depth level is obtained. In the case of flat or slanted object, which produce wide range of distances from camera, the objects need to be recognized as one object. Therefore, regular rule is necessary to be applied on the merging algorithm.

In this chapter, the merging algorithm is such that the two region of depth level is overlapped and its difference of depth level is just one level, merging the regional information of two depth level. And this procedure is conducted until there are no remaining depth levels to merging. The above description is summarized as follows:

$$\begin{aligned} &P_x(n) = \{X_s(n),\, X_e(n)\} \\ &P_y(n) = \{Y_s(n),\, Y_e(n)\} \\ &R(n) = \{P_x(n), P_y(n)\} \quad n = r, \ldots, 3, 2, 1 \end{aligned} \tag{4}$$

$$\begin{aligned} &P_X(n) = (\min(X_s(n), X_s(n-1)), \max(X_e(n), X_e(n-1))) \\ &P_Y(n) = (\min(Y_s(n), Y_s(n-1)), \max(Y_e(n), Y_e(n-1))) \end{aligned} \tag{5}$$

$$R^{merge}(n) = \left\{ \begin{array}{ll} R'(n) = [P_X(n), P_Y(n)] & R(n) \in R(n-1) \\ R\ (n) = [P_x(n), P_y(n)] & R(n) \notin R(n-1) \end{array} \right\} \tag{6}$$

The *r* value in equation (4) represents the number of all separated region in each disparity depth level, and *n* in equation (4)~(6) is the level of disparity map. *$P_x(n)$, $P_y(n)$, $R(n)$* in equation (4) represents the obtained region data in projection block.

When the adjacent two regions are overlap each other, we regard two regions as one object, and merge two regional information by using the equation (5). The final region merging rule is described in equation (6).

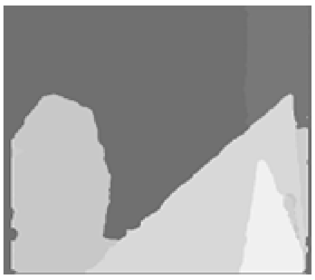


Figure 4. Disparity map after region merging (*barn1 image*)

Figure 4 shows disparity map after the region merging process. When considering the implementation of hardware, the result of this chapter shows the possibility of easy hardware implementation.

5 Experimental Results

5.1 Experimental environment

In this chapter, we proved the validity of proposed algorithm with C-language level implementation. And, after that, we implemented the proposed algorithms with VHDL level and we were able to get result of hardware simulation using Modelsim. Finally, the proposed post-processing algorithm is implemented in FPGA. We used 320x240 resolution and frame rates of 60 fps, 1/3" CMOS stereo camera, and the full logic is tested with Xilinx

Virtex-4 Series XC4VLX200. Figure 5 shows experimental environment. The stereo camera takes images to embedded system and the display monitor shows processed result in real-time. Control PC is linked to embedded system and to hub to conduct control task.

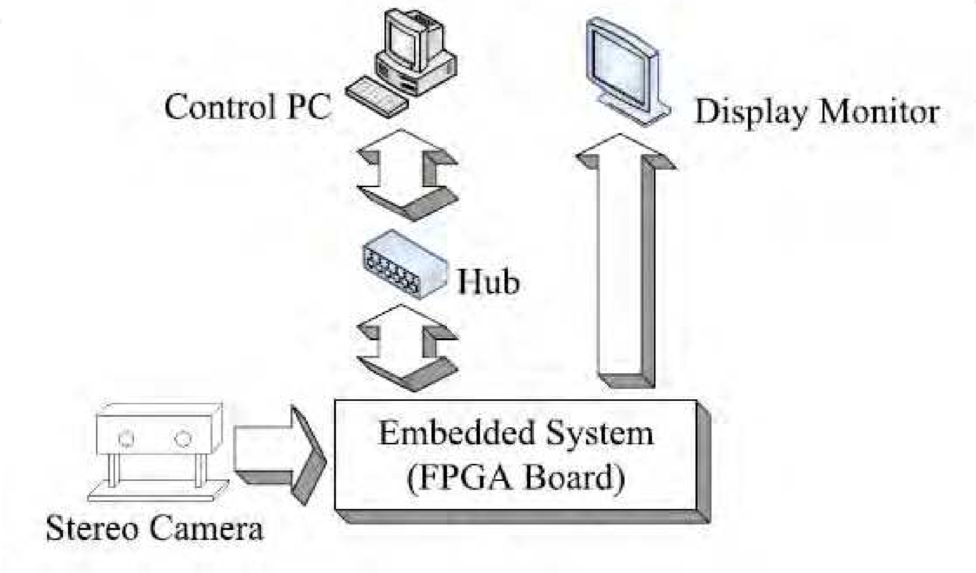


Figure 5. Experimental environment

5.2 stereo matching post processing FPGA logic simulation

Figure 6 shows the result of VHDL simulation to activate *stereo matching post processing* (SMPP) module. When Vactive sync is in high region, it takes 320x240-sized stereo image and shows it on the screen after post processing in real time. Also the control pc in Figure 5 can choose an object to be shown. Table 1 explains signals used in simulation established with FPGA.

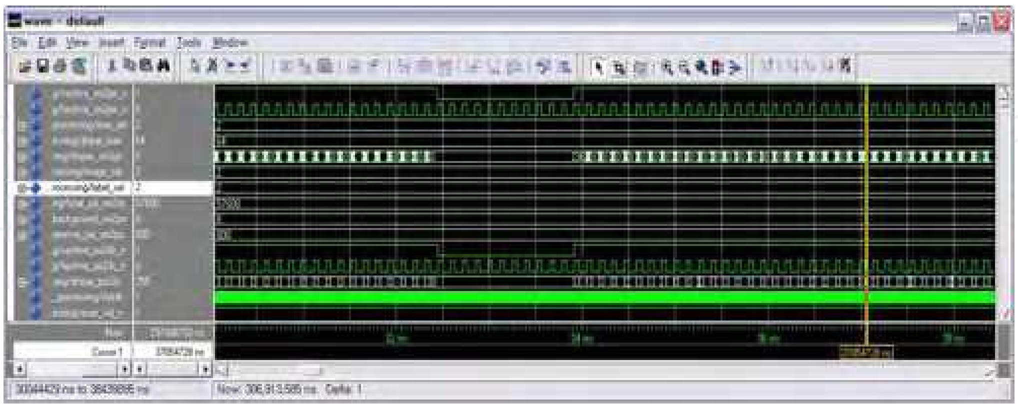


Figure 6. The result of VHDL simulation to activate SMPP module

Vactive_sm2po_n	Input vactive signal of SMPP
Hactive_sm2po_n	Input hactive signal of SMPP
Dispar_sm2po	Input disparity map signal of SMPP
Max_sel	Input register for selecting gray value about object
Dispar_max	Input register about Maximum disparity
Image_sel	Input register for selecting image
Label_sel	Input register for selecting label order
Total_pxl_se2re	Input register about total pixel number of threshold of histogram
Background_sm2po	Input register about background value
Remove_pxl_sm2po	Input register about noise threshold of histogram
Heighte_lb2dp_info	Output register about Height end point of segment object
Vactive_po2ds_n	Output vactive signal of SMPP
Hactive_po2ds_n	Output hactive signal of SMPP
Dispar_po2ds`	Output Disparity map signal of SMPP
CLK	Active clock of FPGA
RESET	Active reset of FPGA

Table. 1. Simulation signal

5.3 Result

This chapter examined the algorithms using various images within stereo matching database for first step and secured its validity. As shown in figure 4, we obtained perfect result with *barn1* image. We performed another experiment using *tsukuba* image and proved that the equal result can be gained. Also, in the result of applying post-processing algorithm in several other stereo images, we are able to obtain similar image as figure 4.

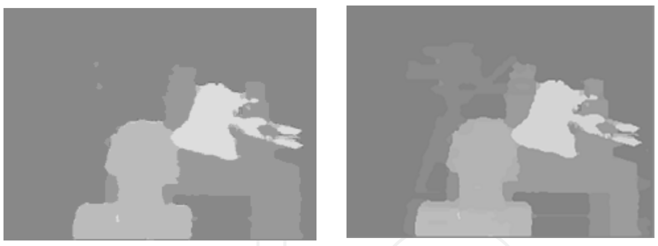


Figure 7. Disparity map after region merging (tsukuba image) (left: C simulation result, right: VHDL simulation result)

The proposed post-processing algorithm is also implemented in fixed-point C and VHDL code. The C and VHDL code test result about the *tsukuba* image is shown in figure 7 and we obtained same results. This result is passed onto labeling stage, with the depth information of camera extracted from depth calculation block. Synthesizing region information and depth information of segmented object is processed in labeling stage. Figure 8 shows the final labeling result of *tsukuba* and *barn1* images obtained from VHDL simulation. Figure 9 shows the BMP (Bad Map Percentage) and PSNR test results with *barn1*, *barn2* and *tsukuba* images.

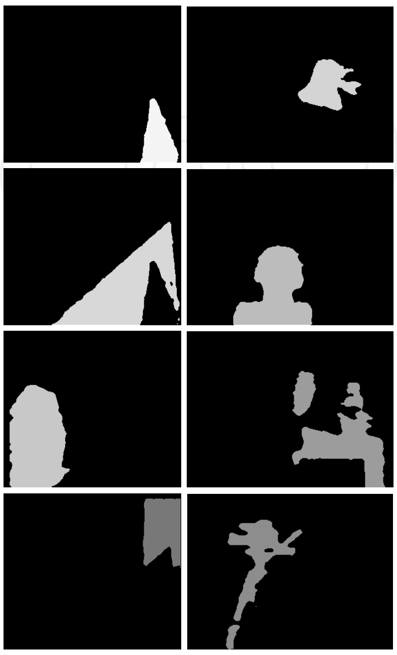
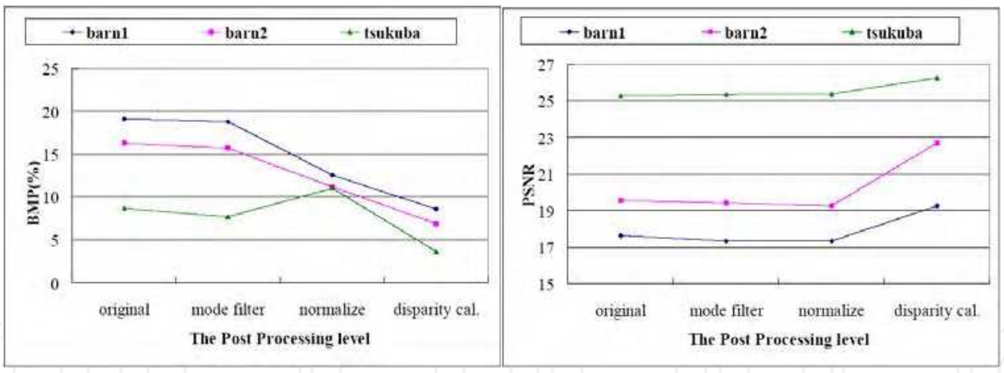


Figure 8. Labeling results (left: barn1 image, right: tsukuba image)



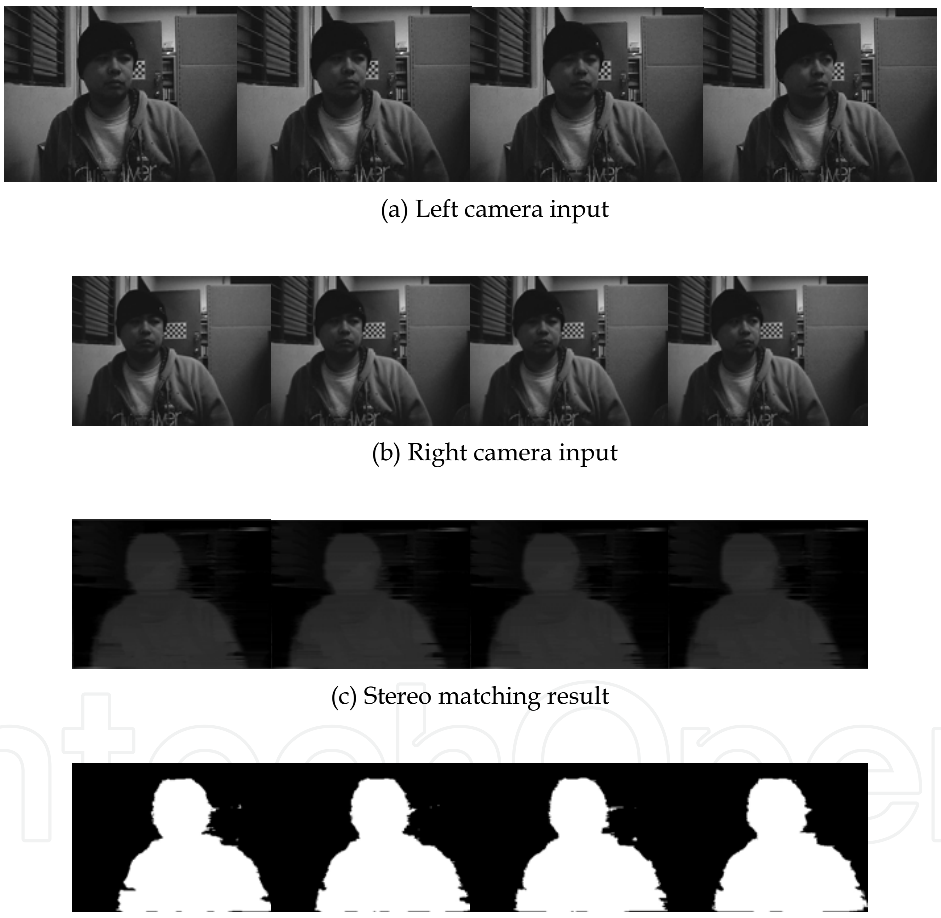
(a) BMP test results (b) PSNR test results

Figure 9. Image quality comparison with intermediate result images

We have designed unified FPGA board module for stereo camera interface, stereo matching, stereo matching post processing, host interface and display. And we also implemented embedded system software by constructing necessary device driver with MX21 350MHz microprocessor environment. Table 2 shows the logic gates of proposed SMPP module when retargeting FPGA. Figure 10 ~13 show real time captured images of stereo camera input and the results of SMPP modules using control pc.

	Virtex4 Available	**Unified module**	**SM module**	***SMPP module***
Number of Slice Flip Flops	178,176	38,658	11,583	17,369
Number of 4 input LUTs	178,176	71,442	25,124	40,957
Number of occupied Slices	89,088	55,531	19,917	29,507

Table 2. The logic gates for implementing the FPGA board



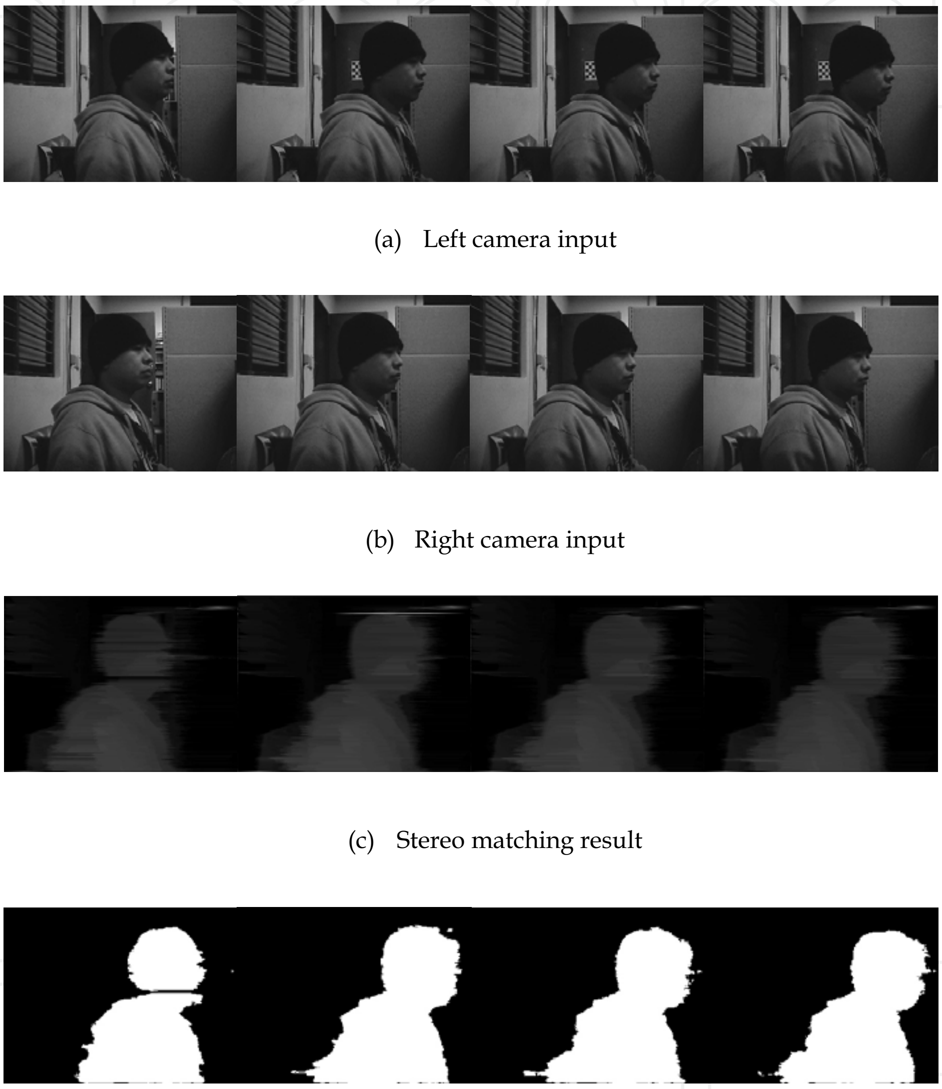
(a) Left camera input

(b) Right camera input

(c) Stereo matching result

(d) Nearest object segment result

Figure 10. Real-time test example 1



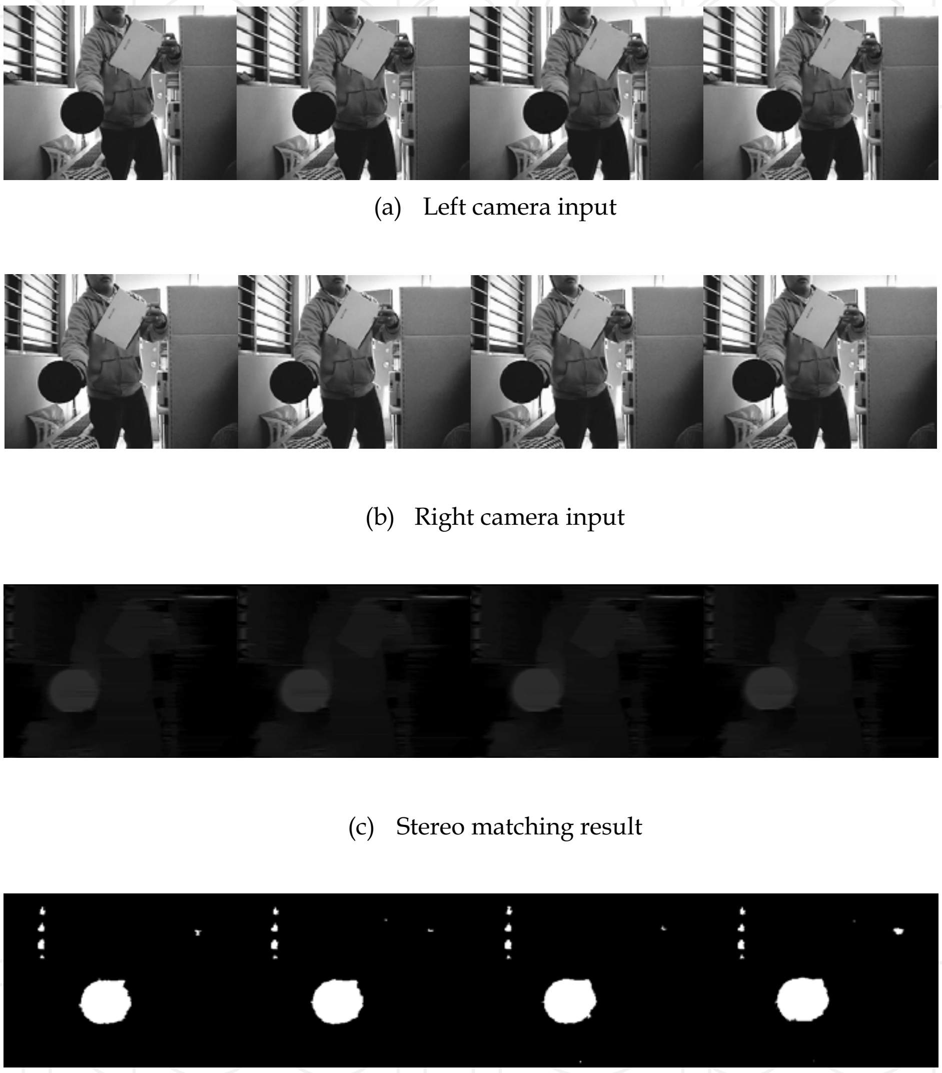
(a) Left camera input

(b) Right camera input

(c) Stereo matching result

(d) Nearest object segment result

Figure 11. Real-time test example 2



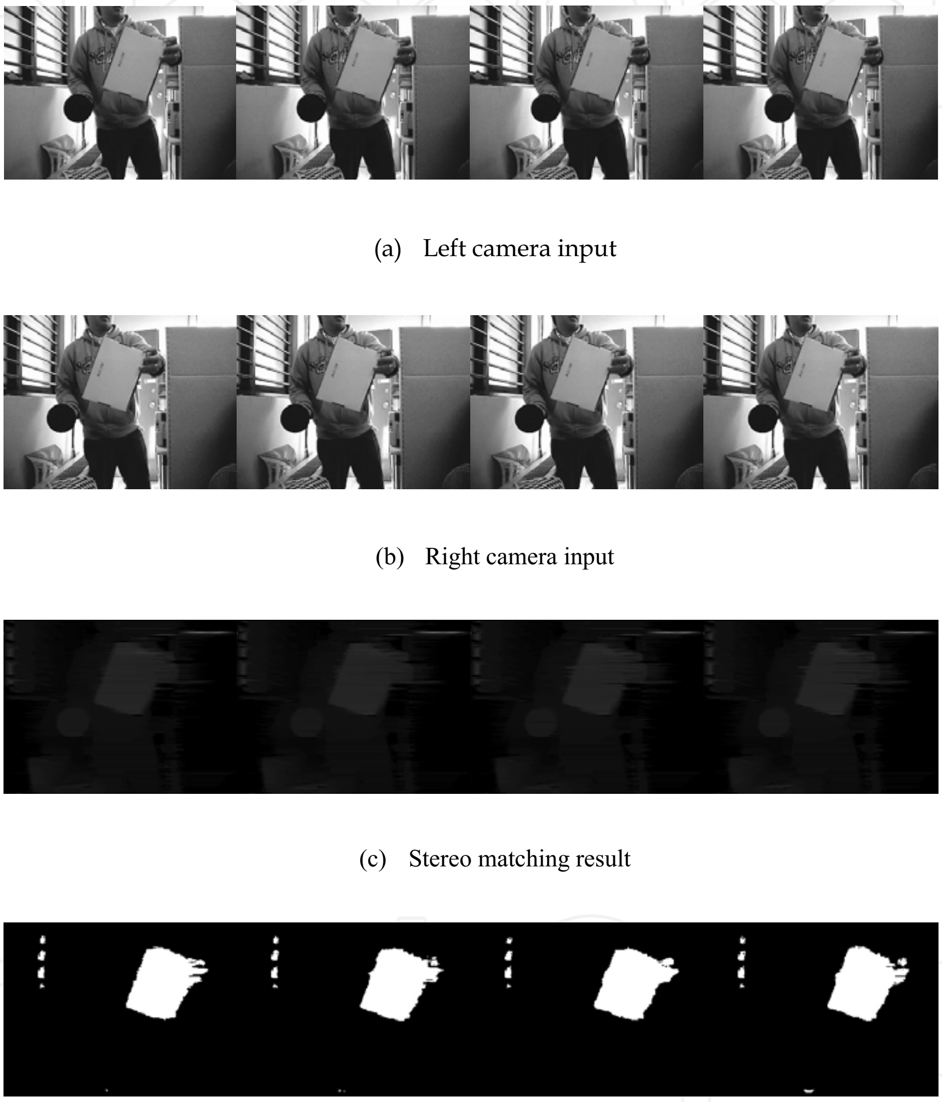
(a) Left camera input

(b) Right camera input

(c) Stereo matching result

(d) Nearest object segment result

Figure 12. Real-time test example 3



(a) Left camera input

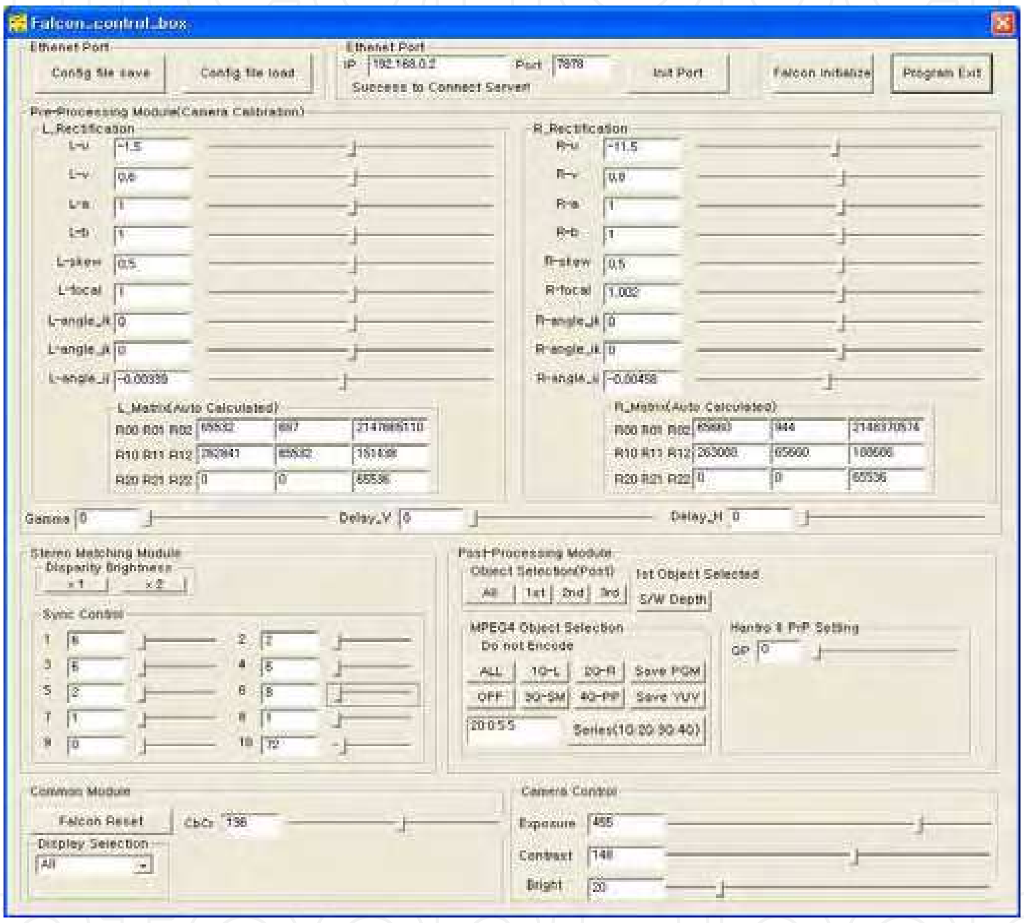
(b) Right camera input

(c) Stereo matching result

(d) Nearest object segment result

Figure 13. Real-time test example 4

Figure 14 shows control application program operated on control pc. This application program communicates to board and hub to calibrate camera and to modify registry of each modules. Also it can capture images on the screen which can be useful for debug jobs. Figure 15 shows image collecting stereo camera. Figure 16 shows implemented embedded system and unified FPGA board module.



IP 192.168.0.2 Port 7878

Success to Connect Server!

L-u	-1.5	R-u	-11.5
L-v	0.6	R-v	0.8
L-a	1	R-a	1
L-b	1	R-b	1
L-skew	0.5	R-skew	0.5
L-focal	1	R-focal	1.002
L-angle_jk	0	R-angle_jk	0
L-angle_ik	0	R-angle_ik	0
L-angle_ij	-0.00339	R-angle_ij	-0.00458

L_Matrix(Auto Calculated)

R00 R01 R02	65532	697	2147885110
R10 R11 R12	262941	65532	151438
R20 R21 R22	0	0	65536

R_Matrix(Auto Calculated)

R00 R01 R02	65680	944	2148370574
R10 R11 R12	263060	65660	100606
R20 R21 R22	0	0	65536

Gamma 0 Delay_V 0 Delay_H 0

Sync Control

1	6	2	2
3	6	4	6
5	2	6	8
7	1	8	1
9	0	10	72

1st Object Selected

Do not Encode

20 0 5 5

QP 0

CbCr 136

Display Selection All

Exposure 465

Contrast 148

Bright 20

Figure 14. The control applications operated on control pc

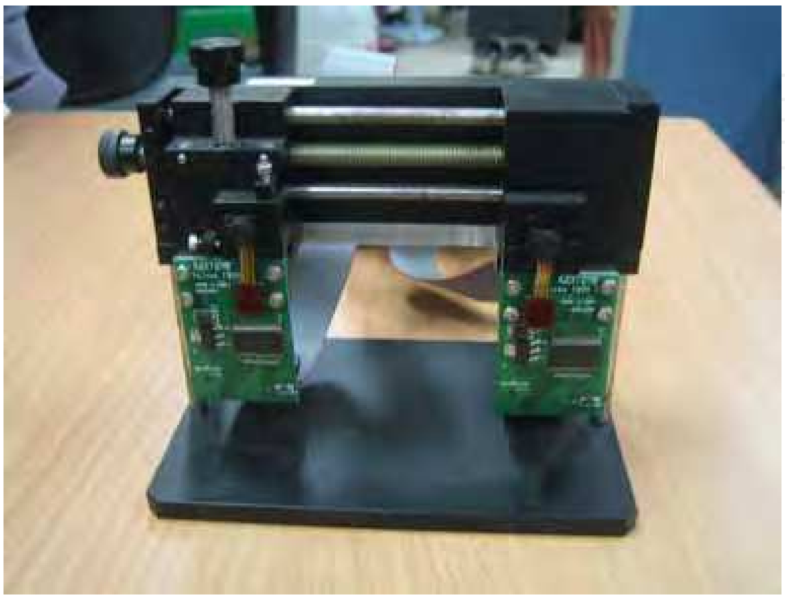


Figure 15. The stereo camera.

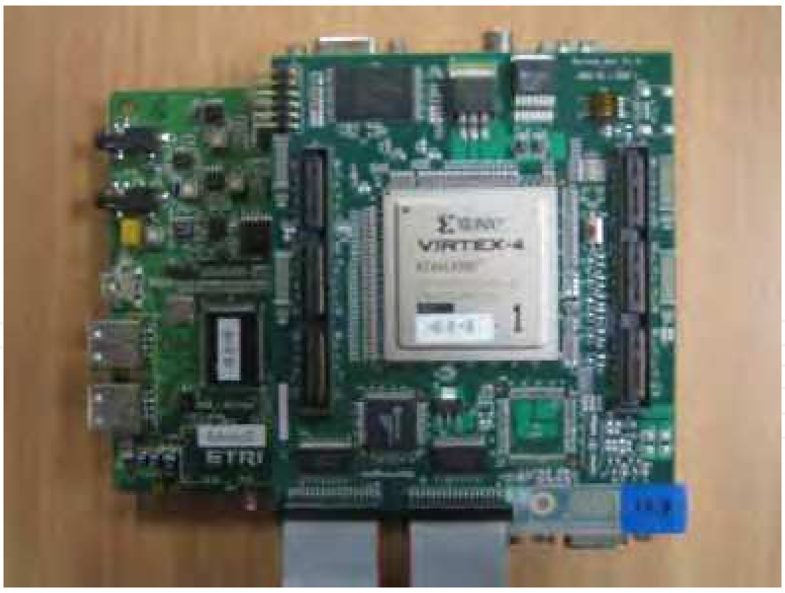


Figure 16. Embedded System and unified FPGA board module

5.4 Software application

A household robot has to perform actions like obstacle avoidance or human recognition activity. One of systems used widely can recognize human by extracting possible human-like areas among those with motions within the screen. However, the system can have performance drops when human doesn't move or the robot moves.

The algorithm suggested in this chapter extracts human shapes on depth map using stereo matching to get relative distances between camera and other objects in real-time, as it also can separate each area in real-time, which keeps performance regardless of human's or robot's motions mentioned above.

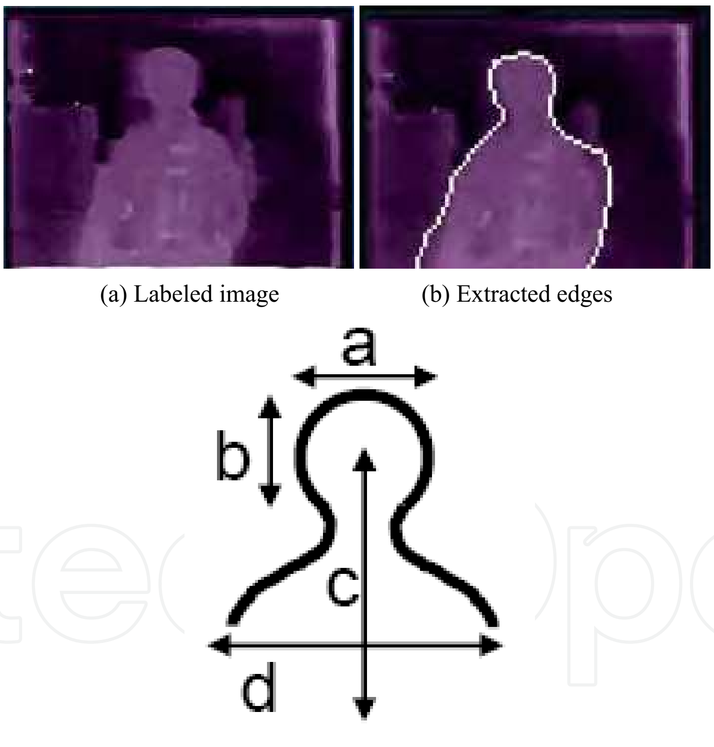
A. Application to human recognition

The followings are description of the human recognition method using results of our study.

Step. 1. Extract edge of screen in 80x60 size from the labeled image (Fig 17.(a),320x240).

Step. 2. Recognize Ω/A pattern (Fig. 17. (c)) among extracted edges.

Step. 3. Determine possibility of human exist considering face size (a,b), height of face (c), width of shoulders, distances, or etc with edges of Ω/A pattern.



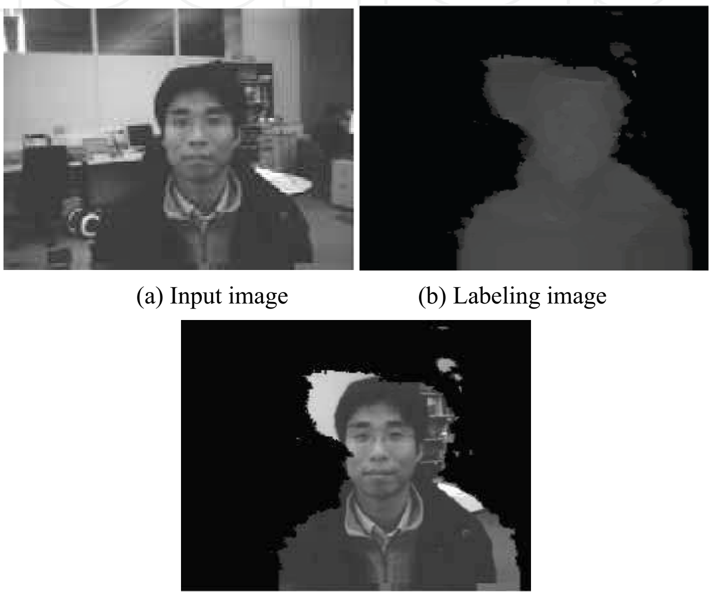
(a) Labeled image (b) Extracted edges

(c) Ω/A type pattern

Figure 17. Example of human recognition with software application

B. Application to face recognition

Figure 18 shows an application of our study to face recognition issue. Figure 18 (a) is an input image, and (b) is an area of object segmentation produced by the algorithm suggested in this chapter. Figure 18 (c) is an overlapped image that has an advantage of faster processing speed by focusing target area to possible human position using segmentation information , compare to total search for face recognition.



(a) Input image (b) Labeling image

(c) Overlap image

Figure 18. Application to face recognition

6. Conclusion

If we can get more accurate result than the conventional stereo vision system, performance of object recognition and collision avoidance will be improved in robot vision applications. So, we used the process of stereo matching algorithm with post processing in this chapter.

The problems such as lack of texture and existence of occlusion area must be carefully considered in matching algorithm and accurate dividing objects must be processed. Also post processing module is necessary as removal of remaining noise. Thus, this chapter developed stereo matching post process algorithm that is able to provide distance between robot and the object regarding trellis-based parallel stereo matching result and to provide the object's area data in real time and examined it by real time FPGA test.

The developed stereo matching post process algorithm is considering possibility of hardware implementation and implemented it using C-algorithm for first step. Then we examined it with various images registered in stereo matching database to secure validity. Also we have developed VHDL and on-boarded it to unified FPGA board module to examine various real time tests using stereo camera on various indoor environments for second step. As the result of many experiments, we were able to confirm quality improvement of stereo matching images.

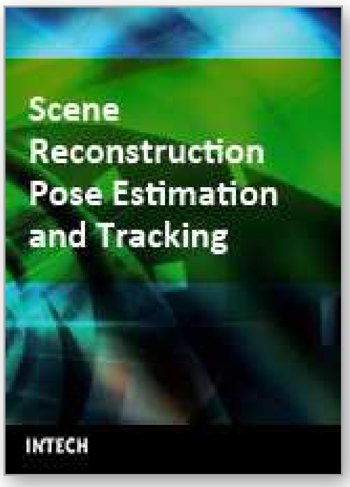
To embody object segmentation, we used hardware-oriented technology which reduces tasks of the software. Also, it has great effectiveness that reduces software processing time by the help of real-time region data support, which containing size and distance information of various kinds of objects, that reduces total area of search process for face or object recognition. Use of embedded software based on low-cost embedded processor to conduct tasks of object recognition, object tracking, etc in real-time provides a suggestion of a household robot application.

7. Acknowledgments

This work is supported by ETRI. The hardware verification tools are support by the NEXTEYE Co., Ltd and the IC Design Education Centre.

8. References

1. Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano and Masaya Tanaka: A Stereo Machine for Video-rate Dense Depth Mapping and Its New applications. Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 18–20 June (1996) 196–220
2. Dongil Han, and Dae–Hwan Hwang: A Novel Stereo Matching Method for Wide Disparity Range Detection. Proceeding of LNCS 3656 (Image analysis and Recognition). Sep. –Oct. (2005) 643–650
3. Jung–Gu Kim, Hong Jeong: Parallel relaxation algorithm for disparity computation. IEEE Electronics Letters, Vol. 33, Issue 16. 31 July (1997) 1367–1368
4. Birchfield S. Tomasi C: Depth discontinuities by pixel-to-pixel stereo. Proceeding of Computer Vision, 1998. Sixth International Conference on 4–7. Jan. (1998) 1073–1080
5. Scharstein D. Szeliski R: High–accuracy stereo depth maps using structured light. Proceeding of Computer Vision and Pattern Recognition, 2003. IEEE Computer Society Conference, Vol. 1. 18–20 June (2003) 195–202, Vol. 1. Digital Object Identifier 10.1109/CVPR.2003. 1211354.
6. Yuns Oh, Hong Jeong: Trellis–based Parallel Stereo Matching. Proceeding of Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. 2000 IEEE International Conference, Vol. 6. June (2000) 2143–2146



Scene Reconstruction Pose Estimation and Tracking
Edited by Rustam Stolkin

ISBN 978-3-902613-06-6
Hard cover, 530 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

This book reports recent advances in the use of pattern recognition techniques for computer and robot vision. The sciences of pattern recognition and computational vision have been inextricably intertwined since their early days, some four decades ago with the emergence of fast digital computing. All computer vision techniques could be regarded as a form of pattern recognition, in the broadest sense of the term. Conversely, if one looks through the contents of a typical international pattern recognition conference proceedings, it appears that the large majority (perhaps 70-80%) of all pattern recognition papers are concerned with the analysis of images. In particular, these sciences overlap in areas of low level vision such as segmentation, edge detection and other kinds of feature extraction and region identification, which are the focus of this book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821