Implementation of Auto-rectification and Depth Estimation of Stereo Video in a Real-time Smart Camera System

Xinting Gao, Richard Kleihorst, and Ben Schueler
NXP Semiconductors, Corporate I&T / Research
High Tech Campus 32, 5656AE Eindhoven, The Netherlands
http://www.nxp.com
{xinting.gao,richard.kleihorst,ben.schueler}@nxp.com

Abstract

In this paper, we present a real-time, low power, and wireless embedded stereo vision system. The system consists of WiCa board (Wireless Camera board) and the methods developed using a smart camera processor, IC3D. IC3D is an SIMD video-analysis processor that has 320 processor elements. The proposed auto-rectification method is suitable for a parallel stereo system like WiCa. It is based on the matching of a planar background. After rectification, the two images of the background are coincident with each other, i.e., both the vertical and horizontal disparity of the background plane is removed. Then a dense matching method is implemented to achieve the depth map of the foreground object. Left-to-right checking and reliability checking is applied to reduce the error of the estimated depth. The system runs at 30fps and handles disparity up to 37 pixels in CIF (320x240 pixels) mode.

1. Introduction

Stereo vision is an important technique in a wireless smart camera system. By passive sensing with relatively low cost sensors, it provides 3D information from two or more images of the same scene. Such 3D information can be used as a reliable distance clue for computer vision systems such as video surveillance and robot vision. In WiCa system, it can also be used as a basic segmentation tool to implement background subtraction.

A wireless smart camera system like the WiCa provides a platform that makes many new applications practical. WiCa is a low-cost, low power and small-sized embedded system. Such systems can for instance be used in consumer electronics applications, mobile smart vision systems and ambient intelligence systems. The core component, the IC3D, is especially designed for low-level video processing [4]. In this paper, we focus on the stereo vision function developed using the IC3D.

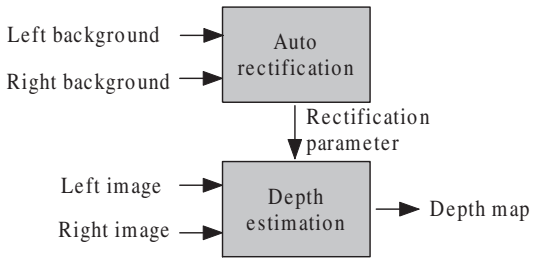


Figure 1. Schematic of the auto-rectification stereo vision system in WiCa1.1.

Many methods have been developed for stereo systems, but mainly using the PC [8, 3]. Rectification is an important process for many stereo vision systems that aligns the conjugate epipolar line collinear and parallel to the image scan lines. In most of the existing embedded stereo systems, the rectification of the two images is obtained manually or with the help of a computer [9, 5]. This is not convenient for a portable system as the parameters of the rectification may change over time, especially when the mechanical support of the sensors is not solidly built. In the WiCa1.1 stereo system, we developed an automatic rectification method. During the auto-rectification, a planar background is used. By matching the stereo images of the background plane, the rectification parameters are obtained. After the rectification, the two epipolar lines will be collinear. The epipolar geometry reduces the correspondence searching from two dimensions into one dimension, and simplifies subsequent algorithms.

As an SIMD processor, IC3D is advantageous for a dense matching algorithm [10]. Therefore, a dense disparity estimation method has been implemented on WiCa1.1. The correspondence searching is based on a local similarity measurement. Left-to-right checking and reliability checking is applied to reduce the errors. The system handles up

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

to 37 pixels disparity at 30fps in CIF mode. The schematic of the implemented stereo vision system is shown in Fig. 1.

The paper is organized as follows. In Section 2, the configuration of the smart camera system is introduced. In Section 3, the auto-rectification method is presented. Section 4 describes the depth estimation method implemented in the real-time system of WiCa. The implementation process and the results of the system are also presented in Section 3 and Section 4, respectively. We give conclusions in Section 5.

2. WiCa: a smart camera system

WiCa is a wireless smart camera system that is suitable for real-time video processing. Due to the advances of integration, WiCa provides a low cost, low power and programmable platform. The advantages of WiCa make it promising in mobile or ambient computing. A detailed introduction about the WiCa system can be found in [4]. In the following, we give a description of the components as far as it is relevant to the stereo vision implementation.

2.1. WiCa1.1 hardware

The WiCa1.1 smart camera system contains the following components: two VGA color image sensors, an SIMD processor - IC3D, an 8051 microcontroller, a communication module and RAM module. The RAM in WiCa1.1 is SRAM which is accessed as a *Dual Port RAM* (DPRAM) using a CPLD. The RAM functions as the communication buffer between the two processors (IC3D and 8051) and enables them to work in their own clock domains. It also serves as a loop-back frame buffer for the IC3D. Fig. 2 shows the WiCa1.1 board (the 8051 is on the other side of the board). The IC3D and the 8051 are coupled using the DPRAM that enables them to have a shared workspace. The main function connection of WiCa1.1 is shown in Fig. 3. The two VGA color image sensors provide either CIF (320x240) or VGA (640x480) images to the stereo system.

The IC3D is one of the SIMD processors of Philips' and NXP's Xetal series. Xetal is designed specifically for low-level video processing. Fig. 4 shows the basic internal blocks of the IC3D. The main component of the chip is the Linear Processor Array (LPA) with 320 RISC processors. Each of the processors has simultaneous read and write access within one clock cycle to the corresponding memory positions or its left and right neighbors in the parallel line memory. Both the memory address and the instruction of the processors is shared in SIMD sense, i.e. the one setting of the memory address or one instruction is applied to the whole 320 pixels simultaneously. The line-memory block can store 64 lines of 320 pixels. Each pixel is up to 10 bits. The Global Control Processor (GCP) controls the IC3D and does some global DSP operations. The peak pixel performance of the IC3D is around 50GOPS at 80MHz. Despite its high performance, IC3D is an inherently low power processor. Table 1 summarizes the specifications of the IC3D.

Figure 2. Architecture of the WiCa1.1.

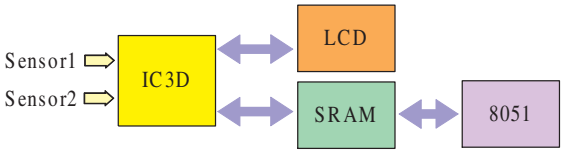


Figure 3. Main function connection of WiCa1.1.

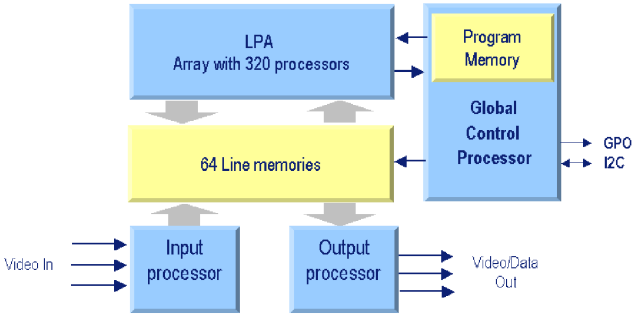


Figure 4. Architecture of the IC3D.

DPRAM is another important component in the stereo vision application. One of the stereo image is written to DPRAM. It is transformed and read back during the rectification and depth estimation. DPRAM works as the asynchronous connection between IC3D and 8051. The size of the memory is 64KB for each bank and WiCa1.1 has 8 banks in total. One image of 256x256 pixels can be directly stored into one DPRAM bank.

The host controller 8051 is suitable for high-level pro-

Features	IC3D
Array Width	320
Line Memories	64
Data Path	10-bit
Video Input	3 or 4
Video Output	3
Performance	50GOPS
Power Consumption	15mW/GOP
Power Supply	1.2 ... 1.8Volt

Table 1. Specifications of IC3D.

Figure 5. Setup of the system.

cessing and communicates to the outside world. The ZigBee module is the transceiver part of the wireless camera. The 8051 and ZigBee provides the peer-to-peer connections in a smart camera network.

2.2. WiCa1.1 software

Programs for the WiCa1.1 are written in a language called XTC. It is similar to C, but with implicit parallel data-types and without pointers [11]. For next generation, WiCa1.2, the programming language C will be used and all XTC programs can be converted directly.

2.3. Setup of the hardware

Fig. 5 shows the setup of the system. The LCD is for debugging display. The WiCa is connected to the PC through a USB port. The program is compiled by the specific compiler [6]. A PC program called "WiCaEnv" is used to configure the system and upload the program from the PC to the WiCa board.

3. Auto-rectification of the stereo system

For the dense depth estimation methods based on epipolar geometry, most of the existing papers assume that the two images/videos are already rectified. In practice, the system is rectified by either manually adjusting the hardware or using some existing tools with the assistance of the PC [9, 5]. As an embedded system, this is not convenient, especially when it is applied in a mobile situation. On the other hand, an accurate rectification is important since it has great effect on the quality of the depth estimation process. In this section, an auto-rectification method is presented.

3.1. The auto-rectification method

For a convergent stereo system, both the internal parameters and the external parameters are needed to correct and rectify the stereo images as illustrated in Fig. 6. Due to the rigid connection of the stereo cameras to the WiCa board, our stereo system is parallel (or near parallel). Therefore, only the translation parameters need to be achieved in the rectification. This is for the following reasons. Firstly, prior calibration of the cameras through the free software showed that the parameters are close to the non-distortion case [2] for this hardware configuration. Secondly, the assumption of the parallel stereo system works well for the current applications.

During the rectification, some planar textured background is provided to the parallel stereo cameras. This background plane is used to obtain zero disparity values for the stereo cameras. Thus, after rectification, any foreground object will have nonzero disparity values as we expect. To make the background plane give zero disparity values we need to align the two background images obtained from the stereo cameras. Therefore, we translate the left image and compare it with the right one. The aligned two images should have no (ideal case) or little difference (due to light sensitivity differences in the two image sensors). To measure the difference between the two images, we compute the *Sum of Absolute Difference* (SAD) within the central parts of the whole images.

$$C(m,n) = \sum_{(i,j)} |I_l(i-m, j-n) - I_r(i,j)|. \qquad (1)$$

Here, m represents the vertical shift of the left image and n the horizontal shift. $C(m,n)$ is the correlation measurement. I_l and I_r is the luminance value of the left and right image, respectively. Here the SAD is a global measurement based on the whole images (central parts).

The translation parameters are corresponding to the minimum SAD obtained during the searching process.

$$(v,h) = arg_{(m,n)} min\{C(m,n) | -M \leq m \leq M, -N \leq n \leq N\} \qquad (2)$$

3.2. Implementation of the auto-rectification

Due to the real-time performance of the system, it is better to distribute the computation over multiple frames during the auto-rectification. The right image is taken as the reference image, while the left image is shifted. To distribute the calculation, a state variable is used in the program [2]. Each state change corresponds to one pixel shift

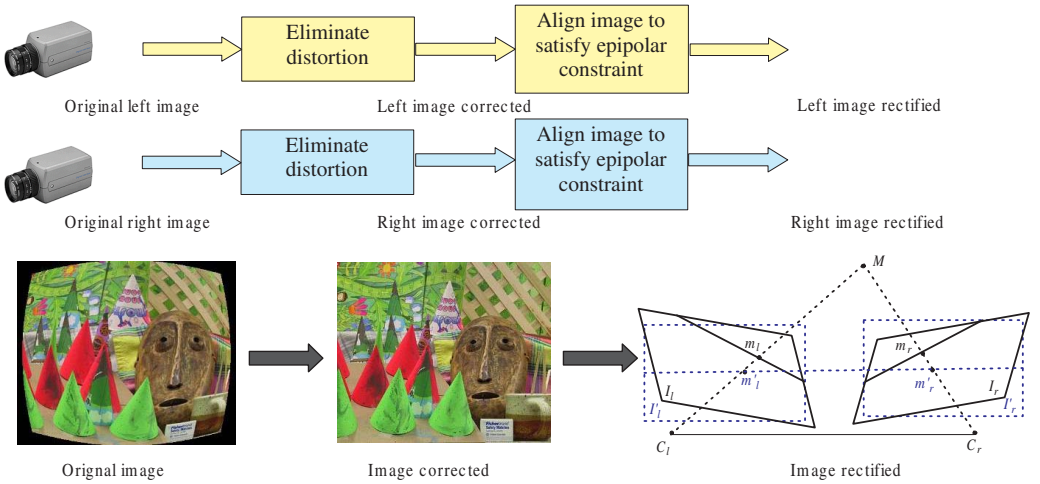


Figure 6. Illustration of correction and rectification of stereo images.

of the left image either horizontally or vertically. The state variable is updated for every frame until the rectification is finished.

There are two methods to obtain the geometry transformation: forward transformation and backward transformation [7]. Forward transformation obtains the destination images by scanning the source image. The pixel location of the destination image is calculated as follows.

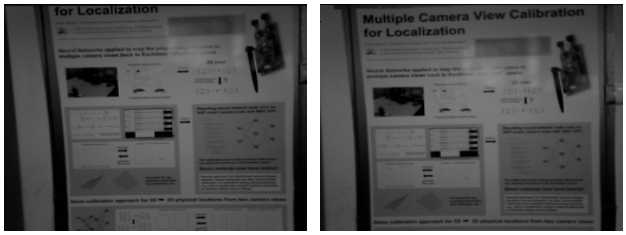
$$\begin{bmatrix} i_d \\ j_d \end{bmatrix} = \begin{bmatrix} i_s & - & m \\ j_s & - & n \end{bmatrix}. \quad (3)$$

Backward transformation scans the destination image, and calculates the pixel location of the source image which generates the destination image as in Eq. (4).

$$\begin{bmatrix} i_s \\ j_s \end{bmatrix} = \begin{bmatrix} i_d & + & m \\ j_d & + & n \end{bmatrix}. \quad (4)$$

It is easy to understand the concept by using the forward transform. However, the backward transformation has the advantage of finishing the operation in a single pass on the IC3D [7]. Therefore, the backward transformation is applied in the implementation of both rectification and depth estimation.

To do the rectification, we need to shift one of the images and compare it with the other one. Thus, one of the images (or part of it) must be saved in memory. One way is to save a part of the image in line-memories. As we have 64 line-memories, this should be enough for the vertical difference between the two images from the sensors in WiCa. The other way is to save a part of or the whole image into one DPRAM bank. We choose to save one of the image in DPRAM, where the transformation can be achieved by the backward operation in a single pass. The line-memories can



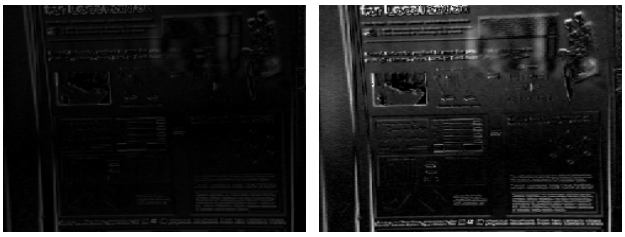
(a) (b)

Figure 7. The images before rectification: (a) the left image, (b) the right image.

be saved to save the intermediate variables. The WiCa system supports 480 video lines per frame. In CIF mode, the video data is input in the odd video lines. As we can either read or write from/to DPRAM in one video line, the whole central part of the CIF image can be saved in DPRAM, i.e., 256x240 pixels of the left image are written into DPRAM. Data is written into DPRAM in odd video lines, while it is read and calculated in even video lines.

3.3. Results of the auto-rectification

Fig. 7 shows the original images before rectification. The left sensor of the system is the slave sensor, while the right one is the master sensor. Apart from the spatial configuration, the slave sensor has a vertical shift compared to the master one due to slightly different reset times. The vertical and horizontal shift between them is obvious. Only the central parts of the images are used to compute the translation parameters. As the right image is taken as the reference image, and the left image is written into the DPRAM and shifted during the searching, the right image remains the same after rectification, while the left one is shifted. Fig. 8 (a) shows the absolute difference between the two



(a) (b)

Figure 8. The absolute difference of the two rectified images: (a) absolute difference, (b) the magnified absolute difference with the scaling factor 3.

rectified images, while Fig. 8 (b) shows the magnified absolute difference by a factor of 3, demonstrating that the two images are aligned very well. At some places they are slightly misaligned. This is due to the following two main reasons. First, only translation is considered in the transform. Second, the background is not a perfect plane, which makes the global rectification method fail in some local parts of the images. The time for the rectification process depends on the horizontal and vertical searching ranges that are set in the rectification process. For example, if we set $-15 \leq n \leq 15, -30 \leq m \leq 30$, then we have 31x61 = 1891 states, which will take less than 64 seconds since the WiCa has 30fps and each state takes one frame. However, it does not exploit the full computational power of the IC3D during the rectification. The time can be shortened by optimizing the implementation.

4. Depth estimation of the stereo videos

There are two basic types of depth estimation method. One type is based on feature matching. The other type is based on correlation measurement [1]. The feature matching method produces a sparse depth map, while the correlation based method can produce a dense depth map. For our application, segmentation of the sparse map does not give enough 3D information of the foreground object, thus dense matching is preferred. As the IC3D SIMD processor is suitable for dense matching methods due to its parallel processing characteristics, the dense depth estimation method was implemented on the WiCa system.

4.1. The dense depth estimation method

For the dense depth estimation methods, Scharstein and Szeliski give a detailed overview in their seminal paper [8]. The dense matching algorithms generally consist of four steps: matching cost computation, cost aggregation, disparity computation and disparity refinement. Due to the simplicity and relatively good performance of the SAD [2], we choose SAD as the similarity measurement, which covers the first two of the above steps. The size of the SAD window is 7x7. For each shift within the disparity searching range, the SAD, $C(i, j, d)$, is computed. Then, the *Winner Takes All* (WTA) method is applied in the disparity computation, i.e., the disparity corresponding to the minimum SAD during the searching is taken as the correct one.

$$D(i, j) = arg_d min\{C(i, j, d)|0 \leq d \leq 37\}. \quad (5)$$

There are many factors that can result in wrong matching, i.e., the wrong disparity value is achieved. Two refinement processing steps are adopted in the system. First, the reverse searching is conducted, which is particularly effective for wrong matching caused by occlusion around the border area of an object. This is called left-to-right checking. For example, for the original searching, the left image is taken as the reference. The right image is shifted and the SAD is computed for each shift step. Finally, the shift value corresponding to the minimum SAD (D_{LR}) is taken as the disparity for the pixel. Then, in the reverse searching, the right image is taken as the reference, and the left image is shifted and the process is repeated. The disparity is recorded as D_{RL}. If the difference between the original searching result and the reverse searching result is too big, it is marked as a wrong matching.

$$D_{LR} = \begin{cases} D_{LR} & if\ |D_{LR} - D_{RL}| \leq t1; \\ 0 & if\ |D_{LR} - D_{RL}| > t1. \end{cases} \quad (6)$$

Here, we simply put the wrong matching parts to background by setting it to zero. Second, the reliability is checked for each matching pixel. During the matching process, the correlation corresponding to the second best matching, C_{2nd}, is recorded as well as the best matching result, C_{best}, for each pixel. If the two values are too close to each other, it is taken that the pixel is within a homogeneous region, making the matching result unreliable.

$$D = \begin{cases} D & if\ |C_{2nd} - C_{best}| \geq t2; \\ 0 & if\ |C_{2nd} - C_{best}| < t2. \end{cases} \quad (7)$$

After these steps, the remaining disparity map is taken as the final result.

The relative depth is the inverse of the disparity. Consequently, it is convenient to show the disparity value directly on the screen. The brighter is the disparity, the closer is the object to the cameras.

4.2. Implementation of the depth estimation method

During the depth estimation, one of the images is saved into the DPRAM. Backward transformation is adopted to achieve the computation in one pass. To guarantee that the comparing data for the left and right images is both from the current frame, the upper image must be written into the DPRAM. By writing the upper image into the DPRAM and taking the lower image as the reference image to calculate and display the disparity map , the data we need from the

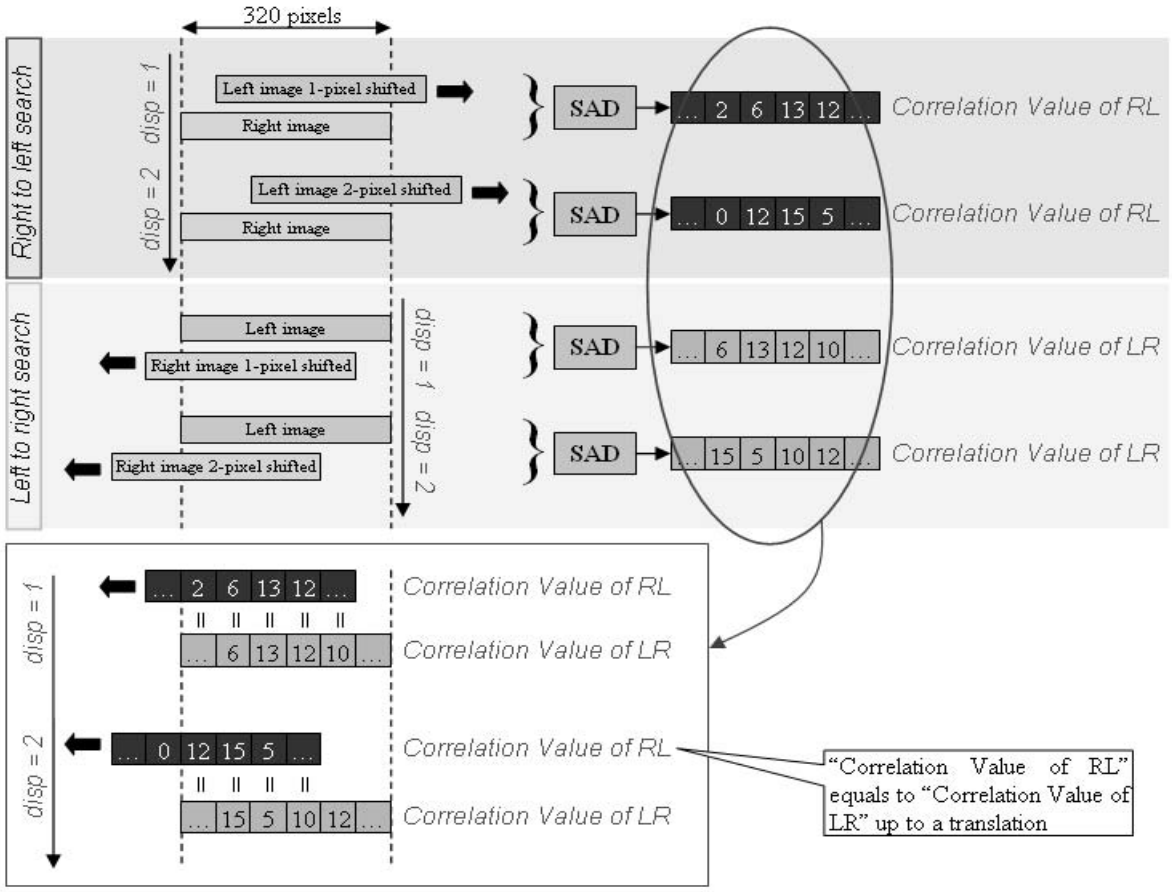


Figure 9. Demonstration of the relationship between the "left to right (LR) searching" and "right to left (RL) searching: RL searching can be obtained from the LR searching according to a shifting.

upper image is already saved in the DPRAM for the current frame. Otherwise, the data we use from the DPRAM to compute the disparity map is from the previous frame. The latter case cause one frame delay between the two stereo images and can not get right results when the foreground object moves vertically.

The reverse searching process will have the same computational cost as the original searching if we compute the SAD for it again. Looking into the computation process of the original searching and the reverse searching, we find that they are corresponding to each other up to a translation as shown in Fig. 9. Therefore, the SAD in the reverse searching can be obtained by shifting the results of the original search properly [2].

At present, we just set the wrong matching value to zero disparity. For the occlusion region, zero disparity is a suitable value as the background has zero disparity after rectification. For a homogeneous region of the foreground object, it is not correct to set the disparity to zero as the disparity corresponding to this region should be the same as of other parts of the object. This issue will be considered in the future work.

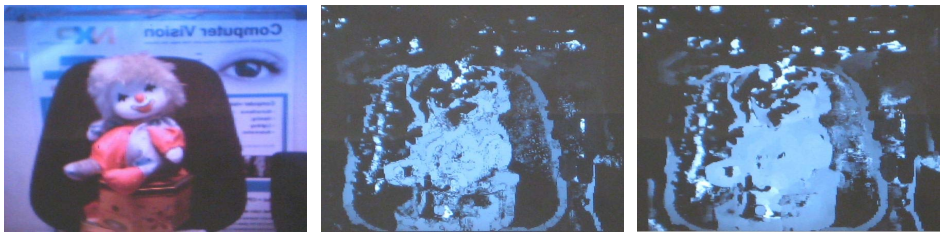
The disparity range in our system is up to 37 pixels. To show the disparity map on the LCD in a clear way, the disparity is adjusted by a gain factor, 6, as follows.

$$D' = D * gain \tag{8}$$

Therefore, the intensity range for the disparity map is [0 222].

4.3. Results of the depth estimation

The system runs at 30 fps and achieves the disparity range up to 37 levels. The threshold $t1$ and $t2$ is set as 2 and 8 respectively. Fig. 10 shows the result of the depth estimation. Fig. 10 (b) is the result with the reliability checking, while Fig. 10 (c) is the result without the reliability checking. The foreground object is clearly detected by the disparity estimation, but there remain some holes in the homogeneous region. Comparing the two results shown in Fig. 10 (b) and (c), it is found that the reliability checking produces more holes within the object while reducing the



(a) (b) (c)

Figure 10. Results of the depth estimation: (a) the original image from the right camera, (b) the depth map with reliability checking, and (c) the depth map without reliability checking.

Resources	used/available
program memory	983/2048
Line-memory	47/64
instructions/video line	4436
IC3D power consumption	≈ 350mWatts

Table 2. The resource usage of the WiCa1.1 system for the depth estimation application running at 30fps with 37 disparity levels in CIF mode.

wrong matching in the background. Only the border of the chair is detected because it is homogeneous.

Table I shows the resource usage for the depth estimation application runs at 30fps with 37 disparity levels in CIF mode. In CIF mode, data is read at every odd lines and processed. The full computational power at the odd video lines is used to achieve the 37 disparity levels. However, IC3D has a computational power of 480 video lines per frame. The computational power for the even lines is still available and can be used by other programs. In Table I, the instructions/pixel is measured at the odd line. For the even lines, the only operation is to write the upper image into the DPRAM. The IC3D power consumption is calculated according to the above analysis.

5. Conclusion

In this paper, we present a stereo vision method in a smart camera system - WiCa1.1. The WiCa system is designed to provide an efficient embedded platform for real-time video data processing with low power consumption. In this stereo system, we implement an auto-rectification method and a dense depth estimation method. The implemented auto-rectification method makes the system more practical. For the depth estimation, the system uses the dense disparity estimation method. By taking advantages of the SIMD processor, the system runs at 30fps and achieves satisfactory results for disparities up to 37 pixels.

6. Acknowledgement

The authors would like to thank all the team members for Xetal project, especially Peter Meijer, Joost Hart, Alexander Danilin, Herman Budde, and Zoran Zivkovic for their assistance and discussions during the work.

References

[1] M. R. Dhond and J. Aggarwal. Structure from stereo - a review. *IEEE Trans. on Systems, Man, and Cybernetics*, 19:1489–1510, Nov./Dec. 1989. 5

[2] D. Grelaud. Development and implementation of perceptual algorithm for mobile robots, 2007. Technical Report, available at <http://xvp.ddns.nl-htc01.nxp.com/Xetal/pubs.html>. 3, 5, 6

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 1

[4] R. Kleihorst, A. Danilin, and J. Schueler. Wireless smart camera with high performance vision system. *Telematik*, 3:20–25, 2006. 1, 2

[5] G. Kraft and R. Kleihorst. Computing stereo-vision in video real-tie with low-cost simd-hardware. In *ACIVS*, pages 697–704, 2000. 1, 3

[6] S. Mouy. Compiler design of the xetal simd processor, 2004. Technical Report, available at <http://xvp.ddns.nl-htc01.nxp.com/Xetal/pubs.html>. 3

[7] R. Rootseler. Inter-frame operations on the wica platform, 2006. Technical Report, available at <http://xvp.ddns.nl-htc01.nxp.com/Xetal/pubs.html>. 4

[8] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(6):7–42, April-June 2002. 1, 5

[9] J. van der Horst, R. van Leeuwen, H. Broers, R. Kleihorst, and P. Jonker. A real-time stereo smartcam, using fpga, simd and vliw. In *Proc. 2nd Workshop on Applications of Computer Vision*, pages 1–8, 2006. 1, 3

[10] W. van der Mark and D. Gavrila. Real-time dense stereo for intelligent vehicles. *IEEE Trans. on Intelligent Transportation Systems*, 7(1):38–50, Mar. 2006. 1

[11] B. Zwaans. Programming architecture ic3d, 2005. Technical Report, available at <http://xvp.ddns.nl-htc01.nxp.com/Xetal/pubs.html>. 3