Proceedings of the Fifth IASTED International Conference
VISUALIZATION, IMAGING, AND IMAGE PROCESSING
September 7-9, 2005, Benidorm, Spain

ENHANCEMENT OF FAST ACQUIRED DISPARITY MAPS USING A 1-D CELLULAR AUTOMATON FILTER

Leonidas Kotoulas[1], Antonios Gasteratos[2], Georgios Ch. Sirakoulis[1], Christos Georgoulas[1] and Ioannis Andreadis[1]
[1]Democritus University of Thrace, Department of Electrical and Computer Engineering, GR-67 100 Xanthi, Greece
[2]Democritus University of Thrace, Department of Production and Management Engineering, GR-67 100 Xanthi, Greece
lkotoula@ee.duth.gr, agaster@pme.duth.gr, gsirak@ee.duth.gr, cgeorg@ee.duth.gr, iandread@ee.duth.gr
tel: +302541079359, fax:+302541079343

Abstract

Among others, stereo vision analysis deals with the extraction of depth in a scene, using the disparity between the images acquired by a stereo camera setup. The disparity calculation for the whole of an image is mostly a computation demanding procedure, commonly being performed by dedicated hardware. In this paper a hardware architecture for real time extraction of disparity maps is proposed, capable of processing images of 1MPixels in less than 25ms. The produced disparity maps are indented to be used for real-time navigation of a mobile platform, as well as in other time critical applications which demand 3D information.

Key Words

Disparity maps, FPGA, Cellular Automata.

1. Introduction

Stereo matching is one of the most complex tasks in machine vision. It refers to associating points of one image on a stereo rig to the other. Assuming an accurately calibrated stereo setup, corresponding points reside on the same epipolar line and thus the search is reduced theoretically within a scan line. The horizontal distance of these points is the disparity. A disparity map consists of all the possible disparity values in an image [1]. Such a map is basically a representation of the depth of the perceived scene. Therefore, the disparity maps have been utilized to efficiently address problems such as 3D reconstruction, positioning, mobile robot navigation, obstacle avoidance, vehicle surveillance, teleconferencing etc [2-5]. Their fast and accurate computation is, therefore, still an active and incessant research topic. This problem has been addressed using area-based [6, 7] and feature based algorithms [8, 9]. Area based techniques have been favoured for fast, dense disparity map extraction. Feature based techniques, although presenting improved accuracy around the extracted features, usually produce sparse disparity maps and require image interpolation. Feature extraction and interpolation cause a computational cost which renders these techniques unsuitable for real-time imaging. Thus, the real-time applications of nowadays tend to adopt local area methods, usually correlation-based ones, to obtain dense depth maps in real time. Nonetheless, it is only the last quinquennium that fast implementations of stereo vision algorithms on common PCs were reported. To this end the contribution of the progress in CPU processing speeds and the utilization of special extensions of the CPU instruction set were essential. However, while some of these algorithms can outperform the 100 fps in software [10-12], they almost do exclusive use of the CPU, which leaves very few cycles to perform the rest of the tasks. The adoption, therefore, of specialized hardware for the computation of the disparity maps is essential, in order to leave the CPU to perform the rest of the laborious processing of the application in which the disparity map is utilized.

In this work, we propose a fast area based technique, oriented for highly parallel hardware implementation in a pipelining fashion. The resulting hardware structure can be implemented in a single FPGA device. Our method consists of three steps. First, the two images are filtered using a 1-D weighted mean filter. The initial disparity map is then extracted. This preliminary disparity map exhibits a very high level of noise, which is removed using a 1-D cellular automaton (CA) filter. The main advantage of the above sequential steps is that only one pass of the images is required. Therefore, the algorithm exhibits excellent potentiality for hardware implementation. The pipelining architecture for real-time disparity maps computation is also described in Section 3.

2. Proposed Method

2.1 Disparity Calculation

The proposed algorithm is a pixel based one that was chosen in order to maximize speed and hardware efficiency. For each pixel on the left image, a search is performed within a window on the right image. This window is 1xD pixels, where D is the maximum disparity value. For each pixel on the right image, the absolute difference of the intensity values is calculated as a distance measure. The relative position of the pixel presenting the smallest distance is stored as the disparity value. This is one of the simplest reported techniques and

it presents the major advantage of very fast implementation.

2.2 Pre-processing

In order to reduce the effects of noise, due to viewpoint change, a one dimensional weighted mean filter is first applied on the images. The size of the mask of the filter was experimentally chosen to be three pixels. The simplicity of this linear filter is twofold: it meets with the requirements for image pre-processing as well as with the high speed ones imposed by the application.

2.3 Disparity Map Filtering

The disparity maps produced exhibit a very high level of noise. Due to the nature of this noise, typical linear or ordered filtering does not perform its satisfactory removal. A CA approach was preferred. CAs are models of physical systems, where space and time are discrete and interactions are local [13]. A CA is characterized by five properties: the number of spatial dimensions (in our case 1-d) [14]; the width of each side of the array (w) where w_j is the width of the jth side of the array, where j = 1, 2, 3, …, n; the width of the neighborhood of the cell (d) where d_j is the width of the neighborhood along the jth side of the array; the states of the CA cells (usually binary) and the CA rule, which is an arbitrary function F (in our case Boolean), where F is a function of the state of this cell at time step (t) and the states of the cells in its neighborhood at time step (t) [15]. Based on CAs the above following four rules are taken into account for the disparity map filtering:

-For each pixel of the disparity image, search if no pixels of its 3-neighbourhood exhibit the same disparity with the current pixel. In this case, set its disparity value to unchanged. It should be mentioned that in case of 3-neighborhood, the neighborhood of the (i) cell consists of the (i) cell itself and of the two adjacent cells (i-1) and (i+1) on either side.

-For each pixel of the disparity map, search if both pixels of its 3-neighbourhood present the same disparity value, different than the current pixel. In this case, set its disparity value to the dominant value of the neighbourhood.

-For each pixel of the disparity image, search if less than two pixels of its 3-neighbourhood exhibit the same disparity with the current pixel. Additionally, check if less than two pixels of its 5-neighbourhood exhibit the same disparity with the current pixel. In this case, set its disparity value to unchanged. It should be also mentioned that in case of 5-neighborhood, the neighborhood of the (i) cell consists of the (i) cell itself and of the four adjacent cells (i-2), (i-1), (i+1) and (i+2) on either side.

-For each pixel of the disparity map, search if more than two pixels of its 5-neighbourhood present the same disparity value, different than the current pixel. In this case, set its disparity value to the dominant value of the neighbourhood.

In order to increase the noise removal capabilities, a second identical filter may be used, for the vertical direction of the image. However, this requires additional circuitry, and about 4KB of RAM, for the system to be fully pipelined.

2.2 Experimental Results

The left and right images of the Tsukuba pair are shown in Figure 1. The blurred image, depicted in Figure 2, is the result of the linear filter on the left image.

Fig.1 Stereo Pair

Fig.2 Filtered Left Image

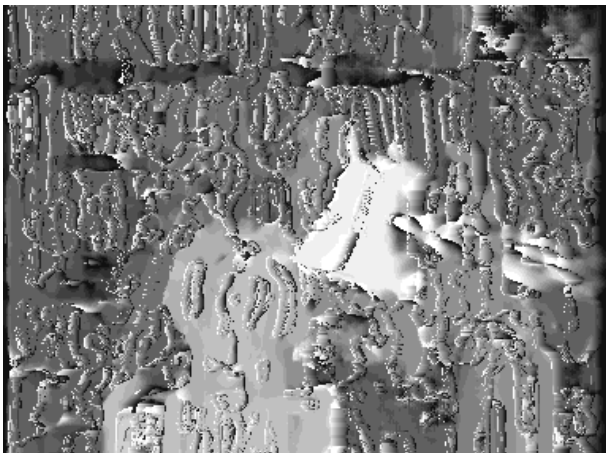


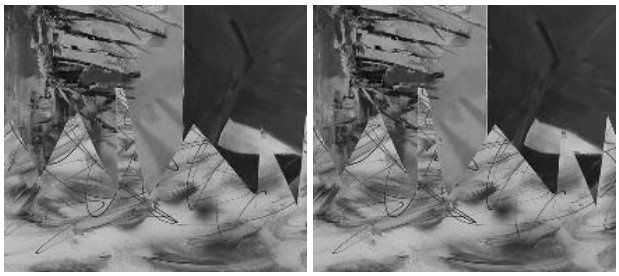
Fig. 3 Initial Disparity Map

The initial computation of the disparity map is presented in Figure 3. The noise level at this stage is obvious. Apart the fact that this map is little comprehensible, several objects such as the easel and the

video-camera in the back are not distinguished at all. However, with the use of the CA filter the depth perception is ameliorated significantly, as shown the final result, presented in Figure 4.

Fig.4 Final Result

The resulting disparity maps exhibit increased noise compared to the ones reported on literature. However, in applications where processing time is critical, whereas detail preservation is not, the proposed method presents a speed advantage of at least one order in magnitude. Additional experimental results can be seen in Figures 5 and 6.



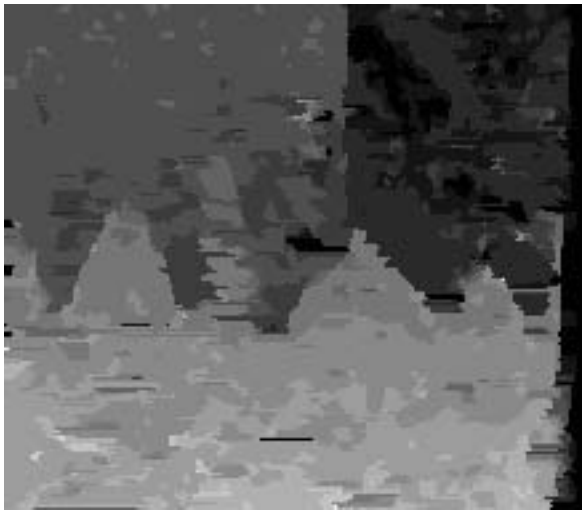
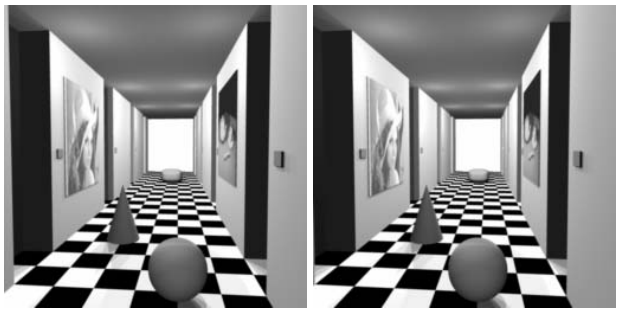


Fig.5 The "Sawtooth" pair and resulting disparity map



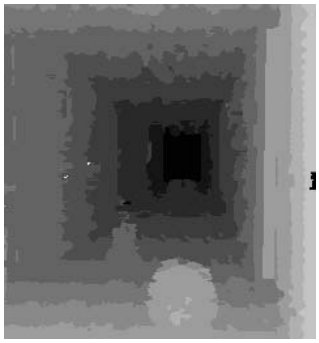


Fig.5 The "Corridor" pair and resulting disparity map

3. Hardware Description

3.1 Mean Filtering

The first step of the proposed procedure is a simple weighted mean filter implementation. This reduces the effects of random noise on the stereo matching. In order to reduce latency and chip area, a one dimensional weighted filter was preferred. The schematic of the filter is shown in Figure 6. The two images are processed simultaneously, so two identical filters have been implemented. The filter can be described by the following equation:

$$F(x,y) = \tfrac{1}{4}\left(f(x-1,y) + f(x+1,y)\right) + \tfrac{1}{2} f(x,y)$$

where f is the original image, and F the filtered one.

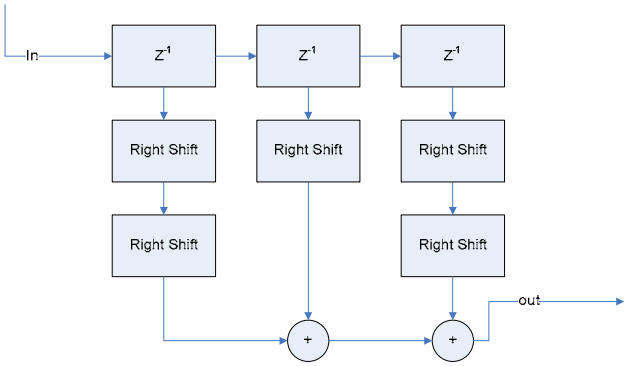


Fig.6 Mean Filter Circuitry

3.2 Disparity Calculation

The most time consuming process of most disparity map calculation methods is the matching phase.

In this work, we address this problem by using the highly parallel structure shown in Figure 7. In consists of a one-dimensional array of W 8-bit adders, where W is the width of the images, the same number of comparators, 4W comparators and a 10-bit counter. Current FPGA devices meet the requirements for such an architecture. The right image is fed in a parallel manner into the adders, while the left is fed serially. On each column of the array, the absolute difference of the pixels of the two images is calculated, and compared to the current minimum. After each scan line has been processed, the disparities of the pixels are computed and are sent to the output of the circuit.

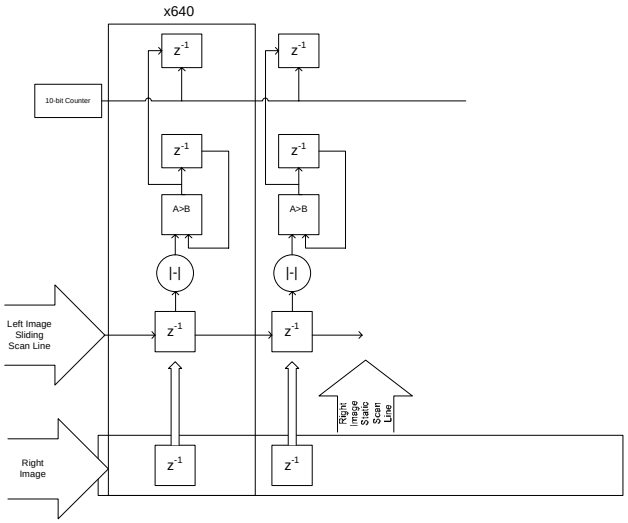


Fig.7 Disparity Calculation Unit

3.3 Post-Processing

A simple, one dimensional CA filter fully pipelined, taking into account the two aforementioned neighbourhoods and the exceptionally simple four cases rule previously described, was preferred because of the appealing results and its straightforward implementation. The schematic of the filter is shown in Figures 8(a) and (b), respectively.

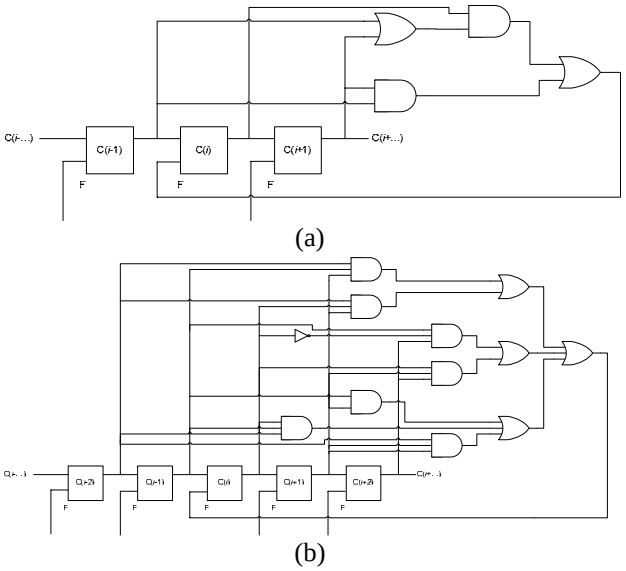


Fig.8 CA Filter in case (a) of 3 neighbours and (b) of 5 neighbours

If, additionally, a vertical filter is needed to improve the noise reduction, the unit shown in Figure 9 must be included. It should be noted, that both units operate at a fully pipelined manner, and do not affect the total frame rate of the system.

After the first three lines have been stored in the serial memories, 1x3 blocks are fed into the CA filter, while the next line is read. The control logic units are responsible for routing the image input to the respective memory. This effectively allows pipelined processing of vertical blocks. A similar architecture can be implemented for 5 pixel blocks, requiring 2 additional memory blocks.

3.4 Speed Issues

The above circuitry is fully pipelined, requiring one pass of the image for the weighted mean filtering, one pass for the disparity calculation and one pass for the horizontal CA filtering and an optional one for the vertical filtering. Thus, the first image requires approximately 4NM clock cycles, and each consecutive image NM clock cycles, for images of size NxM pixels. The relation between the size of the images and the images processed per second is shown in Figure 10.

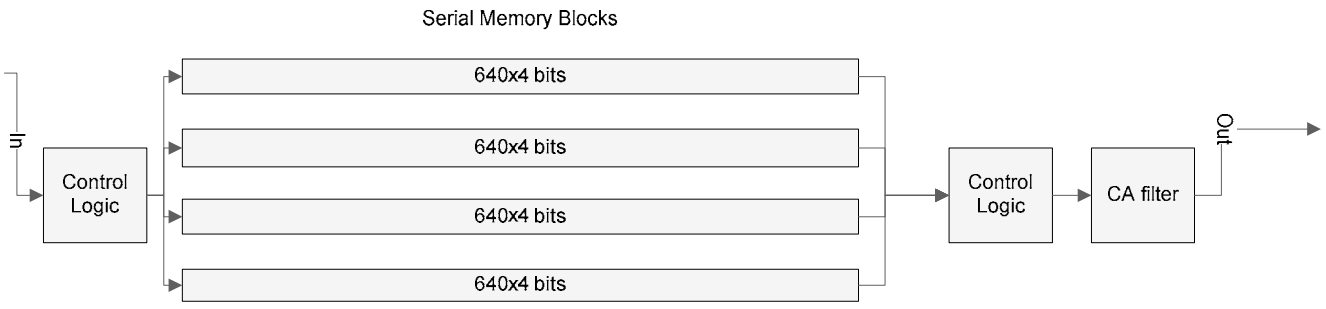


Fig.9 Vertical CA Filter Circuitry

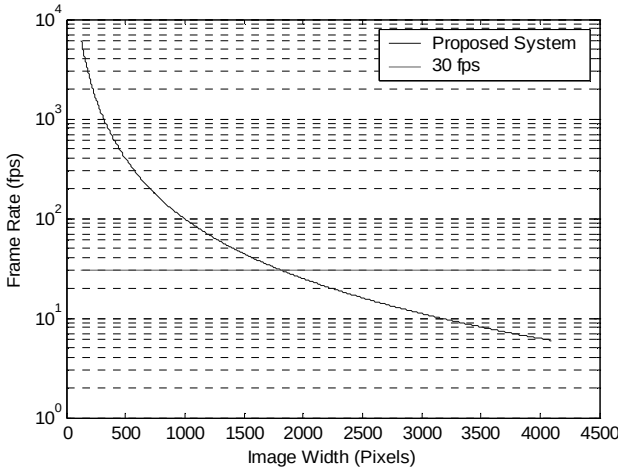


Fig.10 System Frame Rate

3.4 Chip Area Issues

The described architecture consists of a large number of logic elements; namely, approximately 640 8-bit adders and 20k memory bits, as well as a small amount of other control elements. Large FPGA devices, such as the ones of ALTERA Stratix-II family, are capable to treat such a volume of logic. However, since a small amount of data is sent from each unit to the next, the architecture can readily be expanded to two or more smaller and cheaper devices.

4. Conclusions

A novel disparity map calculation technique was presented in this paper. The proposed technique combines desirable properties that such a technique should exhibit for robot navigation. That is, it provides dense disparity maps in real-time, taking advantage of a fully pipelined architecture. The method is hardware oriented and it is intended to be used for autonomous mobile robot navigation.

5. Acknowledgements

The work is supported by the E.U. funded project "RESCUER", IST-2003-511492.

References

[1] S.T. Barnard & M.A.Fischler, *Stereo vision*, (Encyclopedia of Artificial Intelligence, pp. 1,083–1,090, New York: John Wiley, 1987).

[2] J. Corso, D. Burschka, & G. Hager, Direct plane tracking in stereo images for mobile navigation, *Proc of the 2003 IEEE Intl Conf on Robotics and Automation*, Taipei, Taiwan, 2003, 875-880

[3] D. Burschka & G. Hager, Scene classification from dense disparity maps in indoor environments, *Proc. 16th Intl Conf Pattern Recognition, vol. 3*, Quebec City, Canada, 2002, 708- 712.

[4] Z. Sun, G. Bebis, & R. Miller, On-road vehicle detection using optical sensors: a review, *IEEE International Conference on Intelligent Transportation Systems*, Washington, DC, 2004.

[5] N. Atzpadin, P. Kauff, & O. Schreer, Stereo analysis by hybrid recursive matching for real-time immersive video conferencing, *IEEE Trans. on Circuits and Systems for Video Technology*, *14*(3), 2004, 321-334.

[6] L. Di Stefano, M. Marchionni & S. Mattoccia, A fast area-based stereo matching algorithm, *Image and Vision Computing*, *22*(12), 2004, 983-1005.

[7] R. Maas, B.M. ter Haar Romeny, & M.A. Viergever, Area-based computation of stereo disparity with model-based window size selection, *Computer Vision and Pattern Recognition (CVPR)*, Fort Collins, CO, 1999, 106-112

[8] G.O. Wei, & G. Hirzinger, Intensity and feature based stereo matching by disparity parametrization, *International Conference on Computer Vision, ICCV98*, Bombay, India, 1035-1040.

[9] T. Frohlinghaus, & J.M. Buhmann, Regularizing phase-based stereo, *13th International Conference on Pattern Recognition, ICPR96, Vol. A*, Vienna, Austria, 1996, 451–455.

[10] H. Hirschmuler, Improvements in real-time correlation-based stereo vision, *Proc of IEEE Workshop on Stereo and Multi-Baseline Vision*, Kauai, Hawaii, 2001, 141-148.

[11] H. Hirschmuller, P. Innocent, & J. Garibaldi, Real-time correlation-based stereo vision with reduced border errors, *International Journal of Computer Vision*, *47(1-3*), 2002, 229-246.

[12] K. Mühlmann, D. Maier, J. Hesser, & R. Männer, Calculating dense disparity maps from color stereo images, an efficient implementation, *International Journal of Computer Vision*, *47(1-3*), 2002, 79-88.

[13] J. von Neumann, *Theory of self-reproducing automata* (University of Illinois, Urbana, IL, 1966.)

[14] S. Wolfram, *Theory and applications of cellular automata* (World Scientific, Singapore, 1986).

[15] G.Ch. Sirakoulis, I. Karafyllidis, & A. Thanailakis, A CAD system for the constructions and VLSI implementation of cellular automata algorithms using VHDL, *Microprocessors and Microsystems*, *27*, 2003, 381-396.