A FAST POST-PROCESSING TECHNIQUE FOR REAL-TIME STEREO CORRESPONDENCE

Georgios - Tsampikos Michailidis, Leonidas Kotoulas and Ioannis Andreadis
Democritus University of Thrace, Department of Electrical and Computer Engineering, GR-67 100 Xanthi, Greece

Keywords: Stereo vision, real time, disparity maps.

Abstract: In computer vision, the extraction of dense and accurate disparity maps is a computationally expensive and challenging problem, and high quality results typically require from several seconds to several minutes to be obtained. In this paper, we present a new post-processing technique, which detects the incorrect reconstructed pixels after the initial matching process and replaces them with correct disparity values. Experimental results with Middlebury data sets show that our approach can process images of up to 3MPixels in less than 3.3 msec, producing at the same time semi-dense (up to 99%) and accurate (up to 94%) disparity maps. We also propose a way to adaptively change, in real time, the density and the accuracy of the extracted disparity maps. In addition, the matching and post-processing procedures are calculated without using any multiplication, which makes the algorithm very fast, while its reduced complexity simplifies its implementation. Finally, we present the hardware implementation of the proposed algorithm.

1 INTRODUCTION

Stereo vision has been traditionally one of the most extensively investigated topics in computer vision. In general, stereo algorithms can be divided into two major categories, global and local methods (Brown et al., 2003). Global methods are more accurate and can produce dense disparity maps but they are computationally more expensive and usually they are unsuitable for real-time applications. Local methods attempt to determine the corresponding points using area or window-based algorithms, they yield less accurate disparity maps but they are better qualified for real-time stereo matching due to the reduced computational complexity.

In this paper, we present an area-based technique that is capable to generate fairly accurate disparity maps of pictures up to 3MPixels in real-time. The whole architecture can be accommodated in a single FPGA device, operating in a highly parallel and fully pipelined manner. Our algorithm comprises three steps: pre-processing, disparity computation using AD algorithm and post-processing using a new filtering technique. A fundamental characteristic of the proposed algorithm is that the user can use an optional external parameterization, in order to modify, in real time, the density and the accuracy of the output results. This advantageous feature is important for many real-time applications, since it is possible to increase the density of the extracted disparity map in order to obtain a more detailed view of the scene structure, or to increase its accuracy in order to obtain a more accurate localization. It is also worth noticing that the matching and post-processing procedures can be calculated without using any multiplications. This is another advantage, since we reduce the complexity of the algorithm by exploiting only the relationships between neighboring pixels.

2 RELATED WORK

Using Dynamic Programming, (Gong and Yang, 2003) introduce a weak consistency constraint, which expresses the visibility constraint in the image space by re-formulating and extending the consistency check. For evaluating the reliability of a given match, a reliability measure is introduced. It is based on the cost difference between the globally best disparity assignment that includes the match and the globally best assignment that does not include the match (Gong and Yang, 2005). As a result, instead of relying on the smoothness

Michailidis G., Kotoulas L. and Andreadis I. (2008).
A FAST POST-PROCESSING TECHNIQUE FOR REAL-TIME STEREO CORRESPONDENCE.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 490-495
DOI: 10.5220/0001073404900495
Copyright © SciTePress

constraint to remove mismatches, the approximate reliability measure to detect mismatches is used, in order to selectively assign disparities to pixels when the corresponding reliabilities exceed a given threshold. A generalized ground control points (GGCPs) scheme is used in (Kim et al., 2005), where multiple disparity candidates are assigned to all pixels by local matching using the oriented spatial filters.

A different method is presented in (Boykov et al., 2001). Using graph cuts, dense features are defined and extracted during the correspondence process. The boundary condition is enforced to the whole boundary of a dense feature, producing accurate results in areas where features are detected and no matches in featureless regions. A similar algorithm is presented in (Veksler, 2002), where dense features are defined as sets of connected pixels such that the intensity edges on the boundary of these sets are stronger than their matching errors. After computing all dense features, pixels that belong to a dense feature will be assigned with the same disparity value.

3 PROPOSED ALGORITHM

3.1 Pre-Processing and Disparity Estimation

Since in many practical cases the initial intensity values are unreliable, a Laplacian prefilter is applied first in the initial frames for intensity normalization. Then, a weighted mean filter is used to reduce the noise on the initial disparity estimation. The filter can be described by the following equation:

$$F(x,y)=\frac{1}{4}(f(x-1,y)+f(x+1,y))+\frac{1}{2}f(x,y) \quad (1)$$

where f is the original image, and F the filtered one. Of course, a two-dimensional filter produces better results, but also increases the computational cost.

Then, assuming that the source images are rectified, the matching cost for a scanline is calculated using the Absolute Differences (AD) of intensities, which is given by the following equation:

$$d(x,y)=\min_D(\left|I_L(x,y)-I_R(x+D,y)\right|) \quad (2)$$

where D is the disparity value that belongs to the interval $[0, d_{max}]$ and I_L, I_R are the intensity values in the left and right image, respectively.

3.2 Post-processing

While an AD algorithm is fast and simple, it does not exhibit high accuracy and introduces several mismatches in the initial disparity maps. Thus, an efficient post-processing filtering is required. Typical linear or ordered filtering techniques have performed inadequately, as they tend to oversmooth objects and distort their edges. A new non-linear filtering technique is proposed instead.

Assuming that the scene is piecewise constant, a mode filtering is applied first in the initial disparity map. It is based on the ranking of the pixels in a small neighborhood according to their disparity values. Then, the mode value in this ordered list can be used as the depth value for the central pixel. Of course, the computational effort required rises quickly with the number of disparity values to be sorted. For this reason, a 3x3 neighborhood is chosen, although an increase in the number of neighbor pixels contributes to better results.

Next, an one-dimensional filtering technique is employed, in order to incorporate in a computationally efficient manner all the available disparity information between scanlines. Two horizontal and two vertical simple filters are used to modify single pixels with different values in a small neighborhood, while two adaptive filters are used in larger areas. Since the incorrect reconstructions are randomly distributed on the initial disparity maps, a soft modification procedure is adopted, where incorrect disparities are gradually replaced, making at the same time the reliable areas more reliable.

In order to separate the incorrect disparities from the correct ones, the following heuristics are used:

1. Any reliable area in the disparity map must have more than 3 pixels of the same disparity value in range. Any area smaller than this will be an unreliable one and its disparity values will be set to undefined.
2. Any undefined area between a near and a far object belongs to the near object. This heuristic may be justified by the observation that these undefined areas are mainly caused by occlusions, where far objects are occluded by near objects.

Although it is difficult to determine accurate depth values at object boundaries, experimental results show that these heuristics work well in practice and produce satisfactory results. Next, we will examine the post-processing filters separately and then we will present the block diagram of the proposed algorithm. The rules for the two horizontal simple filters are as follows:

• Check for each pixel in the disparity map if the right and left pixels exhibit the same disparity, different from that pixel. In this case, set its disparity value equal to the other two pixels.
• Check for each pixel in the disparity map if the right and left pixels do not exhibit the same disparity value, different from the central pixel. In this case, set its disparity value to undefined.

Figure 1 illustrates the horizontal filtering rules. For vertical filtering, two identical filters are used with exactly the same rules but in vertical direction.

For adaptive filtering, two separate filters with complementary functions are used. The first one is applied to areas where adjacent pixels have similar incorrect disparity values and its size is adaptively changed depending on the number of incorrect pixels. The second filter is applied to undefined areas, in order to propagate reliable disparities. In default operation, its size is adaptively modified depending on the number of undefined pixels in range. The rule for this filter is as follows:

• If the detected undefined area is among objects with the same disparity value, then replace it with the disparity value of this object. Otherwise replace it with the disparity value of the nearest object.

An example of each filter is shown in Figures 2 and 3. In the rest of this work we will mention these filters as Adaptive Undefined (AU) filter and Adaptive Propagation (AP) filter, respectively.

The advantage of the proposed post-processing technique is that the user, through some external parameter setting, is able to increase the accuracy of the resulted disparity map with respect of its density and vice versa. Its use is optional, while the default filtering operation has been described above. To do this, we modify the effectiveness of the AP filtering by introducing three separate variables. The first variable determines the maximum undefined search area (*win_siz*) and the second one the maximum undefined area between two different objects (*und_rep)* that can be replaced by reliable disparities. However, experimental results with real scenes show that the replacement of large undefined areas between two different objects produce more wrong "corrections", caused either by the second

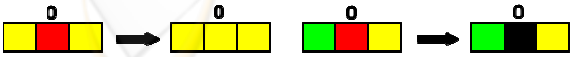


Figure 1: Rules for horizontal filtering.

Figure 2: Example of AU filtering.

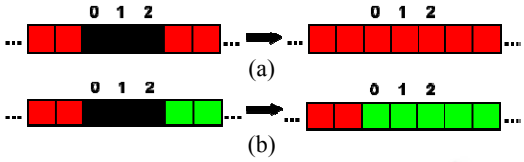


Figure 3: Examples of AP filtering: (a) When the undefined area is among objects with the same disparity value and (b) when the undefined area is among objects with different disparity values.

heuristic we used or by the assumption that an undefined area between two different objects has only one disparity value that depends only from the nearest object. To eliminate these errors, we introduced a third variable (*max_und*) to determine the maximum undefined area between two different objects that can be replaced by only one disparity value. If the undefined area exceeds this threshold, the filter determines the mid point of the undefined area and propagates the disparity values of each object in each part.

To summarize, Figure 4 shows a diagram of the proposed post-processing algorithm. Horizontal and vertical filters are used interchangeably in order to use better the local depth information. In addition, AP filters are used between horizontal and vertical filters to propagate the correct information in undefined areas. In order to improve the visibility of the diagram, we have separated it into two blocks. Notice that the AU filter is used only in the second block, when reliable features have become stronger and the detection of unreliable areas is easier.

4 HARDWARE DESCRIPTION

4.1 Pre-processing

The implementation of a simple weighted mean filter is straightforward and can be seen in Figure 5. The filters process one pixel per clock cycle, so they

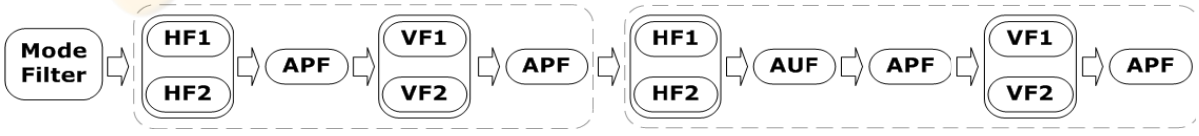


Figure 4: Diagram of the proposed post-processing algorithm.

do not impose any speed degradation on the system.

4.2 Disparity Estimation

Due to the great computational complexity of the disparity map estimation, a highly parallel structure has been implemented, as shown in Figure 6. 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 next unit of the system.

4.3 Post-processing

The mode filter is the input block of the post-processing unit. In order to calculate the mode value in a 3x3 neighborhood, the unit shown in Figure 7 must be included. After the first three lines of disparity values have been stored in the serial memories, 3x3 blocks are fed into the mode filter, while the next line is read. The control logic units are used to route the input image to the respective memory block, allowing the pipelined processing of each 3x3 block.

In the first stage of mode filter, which is depicted in Figure 8, each 'Neighb_Comp' sub-block compares one disparity value with the other eight of the 3x3 neighborhood, and if it stands more than 4 times, then the output is assigned as logic one. The priority encoder generates an output based on the highest 'Neighb_Comp' sub-block that emits a logic one and, finally, the mux selects the mode disparity for the central pixel.

The horizontal and vertical filtering blocks present the simplest hardware architecture of our system. For horizontal filters, only three $\log_2 D$-bit comparators are used to compare the neighboring pixels and provide the proper results. Vertical filters use a similar architecture, while the unit of Figure 7 must be included once again.

In AU filtering block, after the detection of a reliable area, a counter calculates the unreliable pixels in range. If the filtering rules are accomplished, the unreliable pixels are modified to undefined and the others remain unmodified, driving the output of the filter in every clock cycle. AP filter, which is demonstrated in Figure 9, is fed with undefined pixels after the detection of a reliable area. The 'Und_Counter' sub-block counts them and

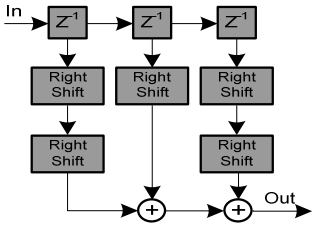


Figure 5: Block diagram of weighted mean filter.

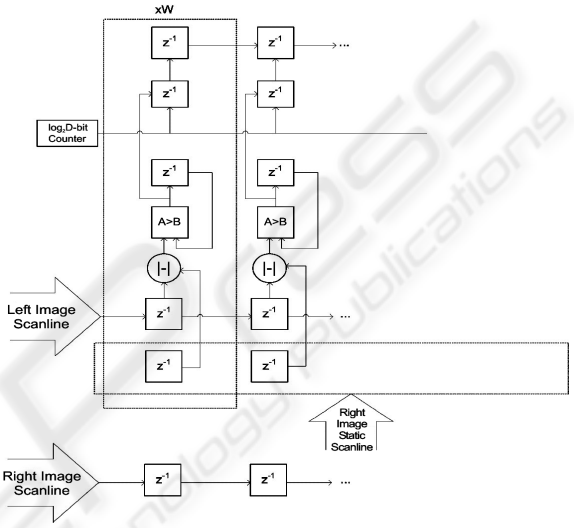


Figure 6: Disparity estimation unit.

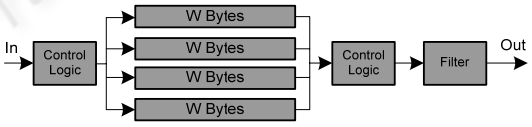


Figure 7: Memory block for two-dimensional filtering.

the result is sent to 'Replace' sub-block. The 'Sel_Disp' sub-block selects the proper disparity value and sends it to 'Replace' sub-block, in order the undefined pixels to be replaced with that value.

4.4 Circuit Characteristics

All units described above operate in a fully pipelined manner. Output latencies are not of importance, since they are in the order of a few microseconds. After an initial latency period, output is given once per clock cycle. The total output latency of the system depends on the width of the input images and the values of the filtering parameters, that is $11W$ + 4'Win_Siz' + 38 clock cycles. This architecture was implemented on an FPGA device of the Cyclone II

family of Altera Devices and the maximum operating clock frequency was found to be 150 MHz. The proposed hardware architecture requires $3W+21$ 8-bit registers, $2W+18D$ $\log_2 D$-bit registers, $W+60$ comparators, W subtraction elements, 1 4-bit and 1 8-bit counter, 6 8-bit adders, 20 8-bit shifters, 4 $\log_2 D$-bit MUXes and a small number of logic gates.

5 EXPERIMENTAL RESULTS

In this section, we present results for some image pairs with different disparity ranges, using the test procedure reported by Scharstein and Szeliski (Scharstein and Szeliski, 2002), available at www.middlebury.edu/stereo. The initial and the resulting disparity maps for the default filtering operation are shown in Figure 10, where black pixels represent the undefined pixels and not zero disparity.

It can be seen that before the filtering process, the initial disparity maps present high number of incorrect reconstructions and object boundaries are not clearly distinguishable. After post-processing, they are significantly cleaner and the cluttered background has been significantly improved. For example, the camera on the tripod in Figure 12(d) is clearly distinguishable, while in Figure 12(c) it is part of the background. As with all area-based methods, our algorithm performs better on textured areas, whilst in textureless and occluded regions the replacement of incorrect disparities is satisfactory.

The proposed algorithm is very fast and can be implemented in real-time stereo systems like autonomous mobile robot applications. In Table 1, a comparison of our algorithm with other semi-dense approaches is presented. We also mention that density for Teddy data set is 36.48%, where 71.3% of them are found correctly in 3.1 msec. In terms of performance, we tested our algorithm on a notebook Intel Pentium M 1.5 GHz, while the execution times for other algorithms are as given by their authors. Quantitative results in Table 1 show that the proposed algorithm presents higher map density than most of the compared algorithms, but also higher error rate. However, other related approaches use some of the state-of-the-art algorithms and are lacking in robustness, while in our approach we use only some simple computations. Furthermore, in many real-time applications, it is more important to identify adequately and fast the space occupied by each object in the scene, rather than to have an accurate but slow reconstruction of it. Therefore, an increase in error rate can be balanced by the significant increase in computational speed, which is essential for time critical applications. Moreover, results on Teddy data set indicate that it can retain its robustness even for large-size images with difficult scenes and larger disparity ranges.

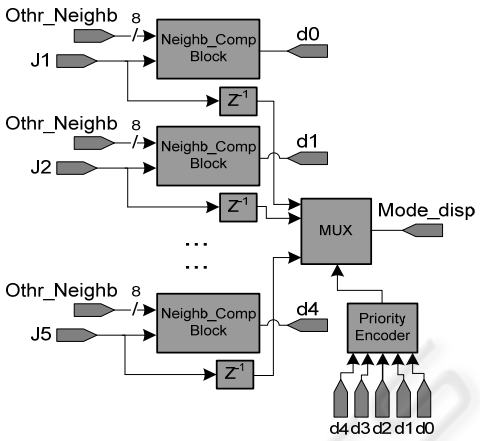


Figure 8: Mode filtering block.

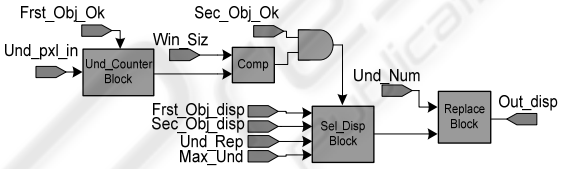
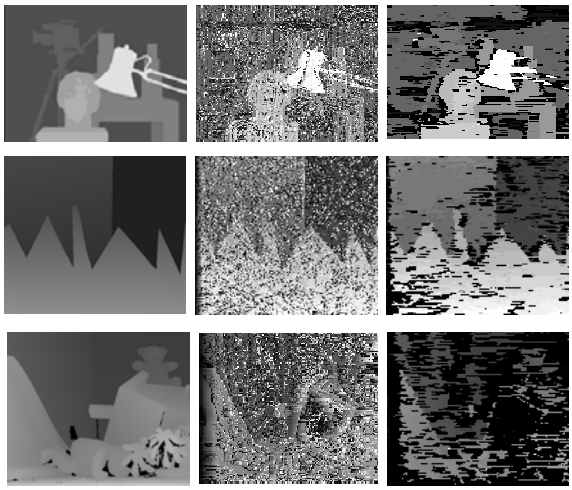


Figure 9: AP filtering block.

Figures 11 and 12 show the plots of density and error rate as a function of variables *win_siz* and *und_rep*. The results indicate that images with larger undefined areas and larger disparity range present smaller density and higher error rate than the smaller ones. We should also notice that our approach is not dependent on the disparity range but only on the size of the images.

6 CONCLUSIONS

In this paper, we have presented a new post-processing algorithm and its hardware implementation. A non-linear filtering procedure and a way to adaptively change in real time the density and the accuracy of the extracted disparity maps, provide a unique feature against other related methods, taking advantage of a fully pipelined architecture. The extracted disparity maps are semi-dense but the localization of objects is quite good, suitable for many real-time applications, where high performance and satisfying accuracy are essential.



(a) (b) (c)

Figure 10: Results on Tsukuba (top row), Sawtooth (middle row) and Teddy (bottom row) data sets. (a) Ground truth, (b) initial disparity map, (c) final disparity map.

Table 1: Comparative results. Density (*D*) is the percentage of matches generated, error rate (*e*) is the percentage of bad pixels far from the true disparity by more than 1.

	Tsukuba			Sawtooth		
	D(%)	e(%)	Time(s)	D(%)	e(%)	Time(s)
Proposed Algorithm	81.7	9.8	0.0021	87.9	33.7	0.0025
Gong and Yang (2003)	71	1.03	0.047	72	0.23	0.093
Gong and Yang (2005)	76	0.32	0.062	89	0.07	0.141
Veksler (2002)	66	0.38	1	76	1.62	6
Kim et al. (2005)	95.2	0.24	4.4	98.9	0.07	11.8
Veksler (2003)	75	0.36	6	87	0.54	13
Szel.&Scharst. (2002)	73	4	-	-	-	-
Sara (2002)	45	1.4	-	52	1.6	-

Experimental results with real-world images have demonstrated that the proposed algorithm is comparable to other methods, indicating at the same time a clear advantage regarding computation time.

REFERENCES

Boykov, Y., Veksler, O., Zabih, R., (2001). Fast Approximate Energy Minimization via Graph Cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), pp. 1222-1239.

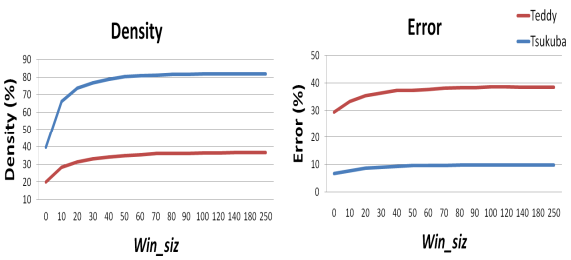


Figure 11: Density and error rate as a function of *win_siz*.

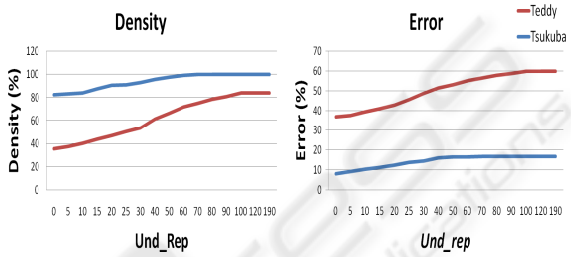


Figure 12: Denstiy and error rate as a function of *und_rep.*

Brown, M.Z., Burschka, D., Hager, G.D., (2003). Advances in Computational Stereo. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8), pp. 993–1008.

Gong, M. and Yang, Y.-H., (2003). Fast Stereo Matching Using Reliability-Based Dynamic Programming and Consistency Constraints. In *Proc. IEEE Conf. on Computer Vision*, pp. 610-617.

Gong, M. and Yang, Y.-H., (2005). Fast unambiguous stereo matching using reliability-based dynamic programming. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6), pp. 998–1003.

Kim, J., Lee, K.M., Choi, B.T., Lee, S.U., (2005). A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 1075-1082.

Sara, R., (2002). Finding the Largest Unambiguous Component of Stereo Matching. In *Proc. European Conf. Computer Vision*, vol. 2, pp. 900-914.

Scharstein, D. and Szeliski, R., (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. In *International Journal of Computer Vision*, 47(1), pp. 7-42.

Szeliski, R. and Scharstein, D., (2002). Symmetric Sub-pixel Stereo Matching. In *European Conference on Computer Vision*, vol. 2, pp. 525-540.

Veksler, O., (2002). Dense Features for Semi-Dense Stereo Correspondence. In *International Journal of Computer Vision*, 47(1-3), pp. 247-260.

Veksler, O., (2003). Extracting Dense Features for Visual Correspondence with Graph Cuts. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 689–694.