Improving local stereo algorithms using binary shifted windows, fusion and smoothness constraint

Mircea Paul Muresan, Mihai Negru, Sergiu Nedevschi
Technical University of Cluj-Napoca, Computer Science Department
Mircea.Muresan, Mihai.Negru, Sergiu.Nedevschi@cs.utcluj.ro

Abstract — **Stereo cameras are a viable solution for reconstructing 3D scenes and are well suited for advanced driver assistance systems, autonomous driving and robotics applications. Modern stereo reconstruction algorithms offer good results, but require very much memory and their real time capabilities are limited on modern day processors. On the other hand, local window aggregation algorithms have a small memory footprint, they are very fast and can be ported to embedded devices, although they provide a lower number of 3D reconstructed points and are more error prone in the case of occluded and slanted surfaces. In this paper we propose a novel, local block matching method which has increased quality and is suitable for real time processing with hardware acceleration (satisfying running time). Our first contribution consists in the introduction of two new binary descriptors used for block matching. The second contribution lies in the shifting method implemented for the matching windows, in order to capture surfaces which are slanted, together with the fusion of the results obtained for fronto-parallel surfaces. Here we propose and compare two fusion methods: a naive and a gradient based approach. The final contribution consists in a smoothness constraint applied to neighboring pixels. The results have been tested on images from the Middlebury benchmark and also on real traffic scene.**

Keywords – local stereo correspondance; dense stereo reconstruction; block matching; slanted and fronto-parallel surfaces; disparity at sub-pixel level; binary descriptors; smoothness costraint

I. Introduction

Many advanced driver assistance systems or robotics applications depend on the perception of the environment and on the surrounding objects. Using stereoscopic depth for obstacle detection and road surface estimation remains a popular choice, mainly because of the reduced costs of the imagining devices and their acceptable performance compared with laser scanner technology. Retrieving accurate distance information from a pair of cameras for every pixel has been an intensive research area. The solutions available are split into three main categories: local methods, semi-global methods and global methods.

Some of the best results can be obtained with global methods however they are not suitable for real time applications. Global methods work by imposing some constraints in the disparity selection phase. These constraints will be modeled into an energy function on the whole image that must be minimized. Semi-global methods were introduced to obtain good performance and reconstruction accuracy in real time. Like in global methods, this is also achieved by the minimization of an energy function. One of the issues with semi-global methods is that they require more memory, and in order to achieve real time performance they need an improved hardware solution based on a graphics processing unit (GPU) with the tradeoff of increased power consumption. Local algorithms use a finite support region around each interest point. The methods are based on a matching metric and usually apply some matching aggregation for smoothing. The minimum disparity for each pixel is searched. Some common matching metrics used by these algorithms are SAD, SSD, RT, Census, and ZNCC [1]. Local methods are a great solution for systems that would require low memory and low power consumption. Because our focus is real time performance this paper deals with improving local algorithms for stereo reconstruction.

In all block matching techniques pixels within a rectangular patch from the left and right images are compared. However, these algorithms assume that the disparities within a certain block are constant and consequently the objects in the scene are perpendicular to the cameras optical axis. This assumption does not hold in the case of ground robots or vehicles, since the road surface and objects on the road are not fronto-parallel. We propose a solution to this problem by shifting parts from the matching windows, based on binary masks. Furthermore, a smoothness constraint is added to favor smooth transitions between neighboring pixels.

The rest of the paper is structured as follows: the next section presents the relevant related works, in the field of stereo vision, that have tried to improve the problem of stereo matching, especially in local correspondence algorithms. In the third section we present our contributions, highlighting the stages of our new stereo pipeline. In the fourth section experimental results are illustrated. The presented algorithm is

978-1-4673-8200-7/15/$31.00 ©2015 IEEE

compared with classical local stereo algorithms, following an evaluation pattern similar to the one on the Middlebury data-set [2], [3]. The last section highlights the conclusions of the paper and presents future improvements for our algorithm.

II. Related work

A real time solution that deals with local stereo correspondence is presented in [4]. This solution, called the DeepSea processor, is implemented in FPGA and ASIC and uses a local correspondence correlation combined with Census transform in order to obtain increased accuracy and high data rate.

In 2008 [5], Hirschmüller introduced semi-global matching (SGM) with mutual information. This semi-global solution is one of the most renowned stereo correspondence algorithms in literature. The algorithm treats the problem of slanted surfaces by the penalty P1 and depth discontinuities by a bigger penalty P2. The structure of the algorithm is well suited to be processed by highly parallel hardware architectures like FPGAs or GPUs.

Center symmetric LBP is another local block matching technique which provides a more compact representation, by comparing only the center symmetric pairs of pixels. In addition, an intensity threshold (*T*) is introduced. The authors in [6] have proven that the best results for this threshold are obtained in the [0, 0.02] interval. The equation for the CS-LBP is highlighted below:

$$CS-LBP(X,\mathrm{Y})=\sum_{i=0}^{\frac{N}{2}-1}\mathrm{sign}(\mathrm{n}_i-n_{i+\frac{N}{2}}-T)\cdot 2^i \tag{1}$$

A variety of articles are dedicated towards the correct reconstruction of slanted surfaces. An interesting approach is highlighted in the work of Ranft et. al [7]. In this paper the effect of slanted surfaces is corrected by fusing the matching information from multiple scaling and shearing of the images. The results of the algorithm have been implemented on GPU in order to achieve real time performance. Other approaches that try to solve the problem of slanted surfaces are presented in several articles of Nils Einecke [8], [9] and [10]. The work presented in [10] has a similar approach as Ranft [7]. In this method the original disparity map computed by a local stereo method is iteratively improved through a process of depth interpolation and image warping based on the interpolated depth. The presented structure, which is based on image warping, gives a mechanism for testing the validity of the interpolated depths, allowing for incorrect depth estimations to be discarded.

In [8] the authors solve the problem of slanted surfaces by rotating the aggregation window. In case the best matching block is not fronto-parallel a penalty is introduced to the matching cost block. The best matching cost is computed by rotating the aggregation window on several disparities (±1, ±2). If the best cost comes from a disparity different from the standard rectangular patch that cost is being penalized. Different penalties are proposed, depending on whether the best match is obtained from one or two disparities away. The idea of penalizing costs from different disparities comes from SGM.

In [9] the problem of slanted planes is solved by an image warping taking into consideration the camera parameters.

Other methods try to deal with depth discontinuities by means of adaptive support weight [11] which use weighted aggregation based on spatial similarity and appearance of the pixels within a patch.

III. Proposed Method

As mentioned in section I, one of the main issues why the dense disparity maps are deteriorated when using standard window matching techniques is that of non-frontal surfaces. The block matching on these surfaces fails because the distance to the camera is not constant within a matching window.

In our approach we crop parts of the matching window, shift them left and right with (±1, ±2) disparity positions in order to find its best matching position. The best matching solutions are fused into a correlation cost along with penalties for each shifting. We finally perfect the resulted disparity map by imposing a smoothness constraint among its neighboring pixels. Even though we will be able to compute disparities correctly for slanted surfaces, some fronto-parallel surfaces will be reconstructed incorrectly. For this reason we are also computing the stereo correspondence with the standard block matching method in parallel and fusing the two algorithms to obtain a better disparity map. In the following three subsections, of this section, we will detail the main contributions of this paper.

We will test the performance of our algorithm against the algorithms presented in [4], [6].

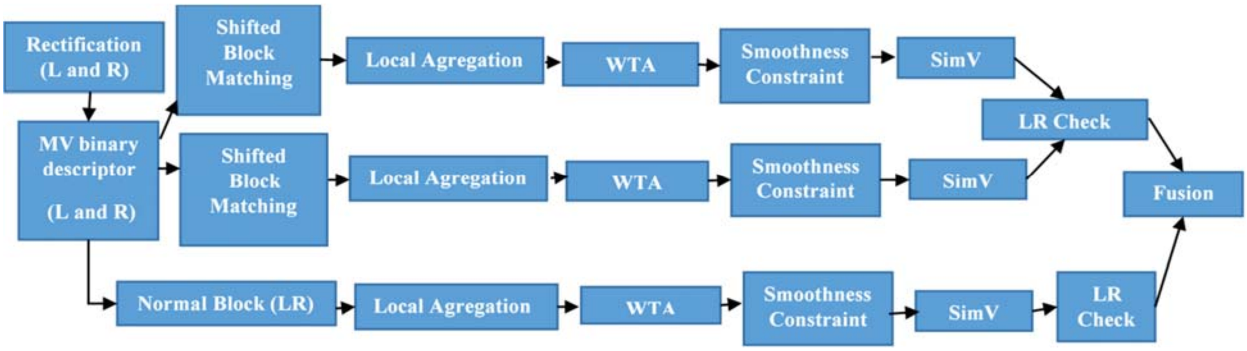


Figure 1. Pipeline of the matching algorithm

$$C(u,v)=\begin{cases}00,\ I(u,v)-t\geq I(u+s\cdot i,v+s\cdot j)\\01,\ I(u,v)-t<I(u+s\cdot i,v+s\cdot j)\,\&\,I(u,v)+t\geq I(u+s\cdot i,v+s\cdot j)\\11,\ I(u,v)+t<I(u+s\cdot i,v+s\cdot j)\end{cases}\left|\begin{array}{l}i,j\in\{-1,0,1\}\\s\in\{2,4\}\end{array}\right. \tag{2}$$

In our implementation we will use a descriptor which has similar structure to the one presented in [7]. Our descriptor also makes comparisons with pixels near the center, not just the ones at a distance of 4px. In our approach we also devised a method of shifting blocks from a patch, such that the final cost does not come from only standard block matching. The shifting of the matching windows is not uniform meaning we could shift two thirds of the window in one direction and the other third could be shifted in the total opposite direction. Furthermore we shift the blocks from our aggregation window using a set of disproportional binary masks.

The resulting cost is penalized depending on whether it comes from two disparities or just from one disparity away. The shifting process will be presented further in the paper. Another key difference between our algorithm and the algorithm presented in [8] is that we are also incorporating a penalty in our cost volume, similar to SGM, for each shifting.

The general architecture of the stereo correspondence algorithm pipeline is presented in Figure 1.

After the rectification process we apply our descriptor to the rectified images. We then apply our shifting process to each image. In the same time we are also calculating the cost volume by using the standard patch matching. The classical block matching is applied in order to repair the disparities which got an erroneous value in the shifted window block matching approach.

A local cost aggregation with a 7x7 window is needed since single cost matches can be erroneous. Winner takes all methodology is applied and the resulted disparity map is corrected by a smoothness constraint and an interpolation at sub pixel level. We will elaborate on each of the original contributions of this paper further in this section.

A. Block Matching Descriptors

The matching score computation is an important part of our algorithm. The correct choice of the matching score will affect the selection of the best disparity and so, the pixel and sub-pixel data will also be influenced. One of the most challenging tasks is to limit erroneous matching scores caused by lack of texture, reflective surfaces and repetitive patterns. For the matching metric we propose two binary descriptors. The binary descriptors have been chosen because of their properties of being invariant to additive and multiplicative offsets in intensity [4]. Although global algorithms deal with this issue, we have to extract more powerful features which differentiate certain regions.

Single pixel matching can give many errors and therefore instead comparing just one pixel a neighborhood around the pixel is chosen. A binary descriptor that reveals the comparison of each pixel in the matching block with the center pixel of the matching window is formed. For a 5x5 window our descriptor would contain 24 bits. Such small matching windows are often not sufficient to capture enough information regarding the compared pixel position on the epipolar line, its neighborhood and its variation from neighboring pixels. In order to solve this issue larger matching windows can be used. However increasing the matching windows size means more comparisons and more memory for storing each comparison. The running time increase is another consequence of the aforementioned statements. This issue is solved by choosing only specific pixels from the matching block and keeping the small memory structures similar to the ones used for 5x5 matching window. Of course some information from the matching is lost when we choose only specific pixels for comparison. In order keep the small footprint and also store information from the matching window, we use not only the comparisons with the center pixels but also the comparison with the mean of the window storing for each comparison 2 bits. Two bits are an optimal solution for each pixel comparison. In our pattern we have chosen 16 pixels for comparison, for each storing 2 bits: one being the comparison of each pixel with the center and the other is the comparison with the mean of the matching window. The second comparison we have done in order to capture some information regarding matching window.

Our first binary descriptor (mean variation – MV) is performed on a 9 x 9 window in order to provide immunity to noise. The image of the binary descriptor window is shown in Figure 2.

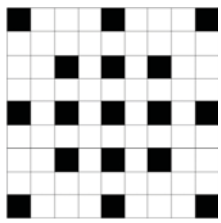


Figure 2. Binary pattern used for stereo block matching, with black we mark the selected pixel

For each evaluated pixel, 2 values are stored. The first value is the comparison of the respective pixel's intensity with the center pixel, like in the census transform, the second bit is formed by comparing the same pixel intensity with the mean of the intensity over the entire 9 x 9 window. The result is a 32 bit bit-string that can be easily compared with another bit string using the synergy between the XOR instructions with the SSE POPCNT instruction. The black squares represent the selected pixels, while the white ones are not considered for the pattern creation.

An alternative binary descriptor proposed (modified census transform – MCT) uses the same 9 x 9 window structure presented in Figure 2. However, in this second pattern a reflection invariance, t, has been introduced. We are verifying if each compared pixel is within a certain interval of intensity and for each comparison we store 2 bits depending on whether the pixel is inside the specified interval or outside. We formed this pattern taking into consideration an idea from [7]. Equation (2) reveals how the second pattern is formed.

The best solution has been found for a value of t = 1. To form this pattern, s will be at first equal to the value 2 and then 4; and i, j will be at first equal to 1 and then -1.

B. Stereo Block Matching and fusion

After computing the descriptor on the input images, we must perform the cost matching step. For this we will use a set of precomputed binary masks and extract from adjacent positions the exact part of the window which we want to match against a portion of the reference window. Considering our 32 bit bit-string obtained after the matching process, we split it into five distinct regions. This splitting is depicted in figure 3. The blue squares correspond to regions where we have a bit of 1 and the white regions to bits of 0.

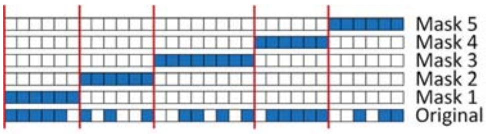


Figure 3. Bit string masks and original bit string value

When the final matching cost is formed, from „parts" of adjacent strings, we will also include a penalty for each component. For example: if our final matching block is formed from two regions coming from disparity +1 we will include a penalty P1 to each of the regions to the final cost of the matching window. If we would have a matching block coming from disparity +2 and one from disparity -1, we would include for the first a penalty P2 and for the second a penalty P1. The choice of selecting the best section is similar to a window shift for each region of the matching window. All the resulted costs are fused together to form the final patch cost. The method is performed for d disparities for the left and right images. The idea of introducing a penalty when calculating the cost of a matching came from [5] and [8]. To obtain the best results P2 penalty should be greater than P1. In our case it is considered to be the square of the P1 penalty, another viable selection could be the double P1. Comparison of pixels from a block, from left and right image is performed using hamming distance. Since the matching values are limited for each pixel a 7 x 7 aggregation of the cost volume is performed. By performing an aggregation over a small window we achieve better smoothness and a larger spread of the matching cost. The algorithm is called shifted windows – SW and is depicted in Figure 4.

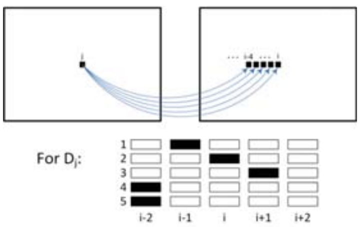
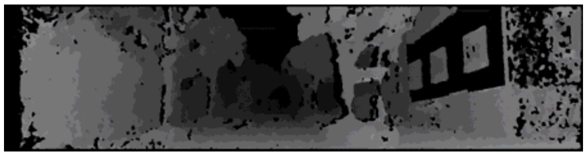


Figure 4. The process of window shifting for a block D_j in the left image and its corresponding blocks in the right image

a. Original scene

b. Disparity map with naive fusion



c. Disparity map with gradient fusion

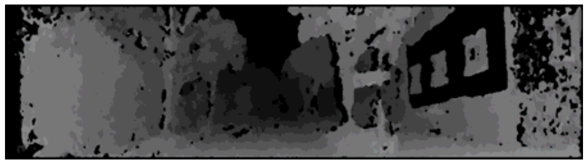


Figure 5.Comparison between the naive fusion type and the gradient fusion type.

Window shifting solves the problem of slated surfaces, however fronto-parallel surfaces may receive a wrong disparity value. Therefore, the authors propose a further improvement of the stereo algorithm by fusing the classical local stereo algorithm, using the same descriptor, to the previously resulted disparity map.

An alternative way of fusion has been studied. The second fusion method implemented was using the gradient image to identify when the matching block should be shifted i.e. finding slanted surfaces, and not shifting the windows for each disparity. A set of 36 orientations have been selected and broken in two classes, for slanted and for frontal surfaces. Before performing block matching we first decide whether the block has to be slanted or not. Even though the resulted disparity map has less reconstructed points than the naive version of fusion presented above, the number of points reconstructed incorrectly seems to be much larger when compared with the ground truth. In Figure 5 we can see the results of the naive and gradient way of fusing on a stereo image pair from the Kitty data-set [12].

More concrete results will be presented in the experimental results section.

C. Smoothness constraint

In this section we suggest a possible improvement for local stereo reconstruction algorithms. The idea for this improvement was first presented for semi-global matching in [13]. After generating a disparity map, we want to correct transitions between neighboring pixels so that they are as smooth as possible. We apply this smoothness constraint only where the difference between neighboring pixels are smaller than a threshold T. This would have as an effect the smoothing of the surfaces that are belonging to a specific item. In our experiments we have used the value of T = 2.

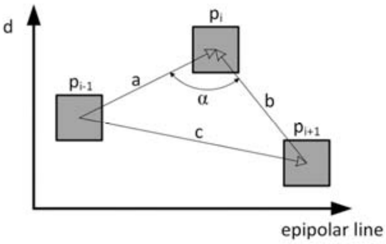


Figure 6. Smoothness angle computation

Considering three consecutive pixels p_{i-1}, p_i and p_{i+1} along a row of the generated disparity image, see Figure 6, with disparities d_{i-1}, d_i, d_{i+1} which satisfy the following condition:

$$d_{i+1} - d_i < T \quad (3)$$

the three points define a triangle in the 3D space, with disparities being the third coordinate. The angle α can be computed using equation (4).

$$\alpha = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad (4)$$

The three values a, b and c are being defined as:

$$a = \left\| (p_{i-1}, d_{i-1}), (p_i, d_i) \right\|_2$$
$$b = \left\| (p_i, d_i), (p_{i+1}, d_{i+1}) \right\|_2 \quad (5)$$
$$c = \left\| (p_{i-1}, d_{i-1}), (p_{i+1}, d_{i+1}) \right\|_2$$

The operations denoted by $\|x\|_2$ is the Euclidian distance. The maximum angle can be of π (which corresponds to a very smooth transition).

We will compute the smoothness function f, as:

$$f = \frac{\alpha}{\pi} \quad (6)$$

We search for that angle α, which is formed with disparities situated at one pixel difference from the values of the neighbors' disparities and maximizes the given function (6). The goal of the function is to favor smooth pixel transitions.

Sub-pixel interpolation refines the disparity image at sub-pixel level so that accurate distance information can be provided. Usually, parabola interpolation is used, however as highlighted in [1] this method lacks the necessary accuracy for automotive systems, due to the pixel locking effect. The sub-pixel function used is symmetric V and its expression is highlighted in equation (7).

$$Disp_{final} = Disp_{integer} + \begin{cases} 0.5 - 0.25 \cdot \left(\dfrac{(M_3 - M_1)^2}{(M_2 - M_1)^2} + \dfrac{M_3 - M_1}{M_2 - M_1} \right) & , if\ M_2 > M_3 \\ -\left(0.5 - 0.25 \cdot \left(\dfrac{(M_2 - M_1)^2}{(M_3 - M_1)^2} + \dfrac{M_2 - M_1}{M_3 - M_1} \right) \right) & , if\ M_2 \le M_3 \end{cases} \quad (7)$$

where M_1, M_2, M_3 are the correlation values belonging to the current winning disparity and its neighbors. By implementing the sub-pixel interpolation function we are able to estimate more precisely the depth to the objects in the scene.

IV. EXPERIMENTAL RESLTS

In this section we are going to evaluate the presented algorithm having as reference other local block matching stereo methods. The system on which we tested our method contains an Intel i5 processor at a 3.3 GHZ frequency. The algorithms presented have been implemented in C++ using Open MP for parallelization, no hardware acceleration methods have currently been used with in the current implementation, apart from one SSE instruction, POPCNT.

To be noted that the current result adopts no post-filtering, pre-filtering and disparity refinement methods, making the method more error prone. However the algorithm is compared against other local block matching stereo in which no such refinement methods are implemented.

Our input data comes from the Middlebury data-set [2], [3] considering a 2px error. When using this data-set the method is evaluated with respect to two criteria: running time and result quality, using a metric similar with the one from Middlebury.

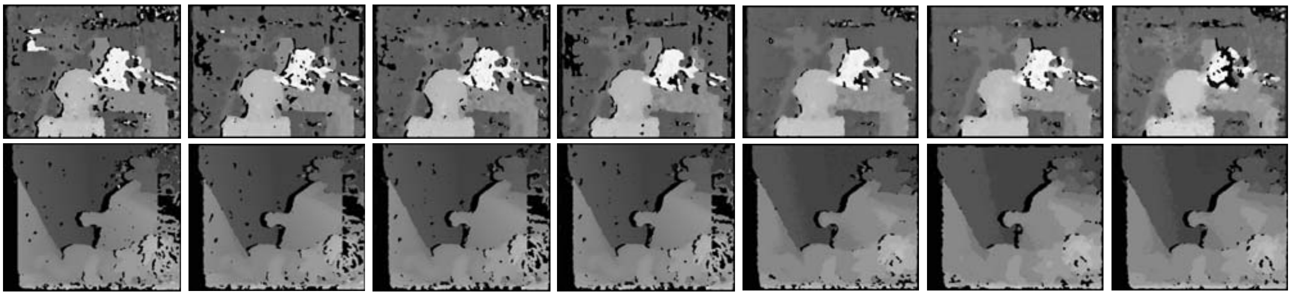
Sample images from the Middlebury data-set are presented in the figure bellow. We have displayed only the Tsukuba and Teddy images for each algorithm in order to conserve space. Left column presents the original images and the right column presents the ground truth for the Tsukuba and Teddy images respectively.

Figure 7. Tsukuba and Teddy reference images and their corresponding ground truth from the Middlebury data-set

The compared Block Matching methods are: Census [4], CS-Census [6], Modified Census Transform (MCT), Mean Variation (MV), Shifted Windows Mean Variation (SWMV), Shifted Windows Modified Census Transform (SWMCT) and Gradient fusion applied for Mean Variation. Figure 8 presents the obtained disparity images for all the above mentioned algorithms on the stereo pairs from the Middlebury data-set presented in Figure 7. All the comparisons were done on the Middlebury benchmark and the results of the evaluation on this benchmark are provided in table I. The lower scores overall are due to the lack of post refinement. However in this paper we are only interested in the block matching techniques and we want to see how our method deals with surfaces that are not fronto-parallel.

obtained for other local matching metrics like ZNCC, RANK etc. are presented in [14].

In Figure 9 we can see a comparison of the results provided



Census CS-Census MCT MV SWMV SWMCT Gradient Fusion

Figure 8. Obtained Disparity Maps with the compared algorithms for the Tsukuba and Teddy images from the Middlebury data-set

TABLE I. EVALUATION ON MIDDLEBURY DATA-SET

Method	Results on several images				
	Tsukuba	Venus	Teddy	Cones	Average bad pixels
Census	14.0	79.2	21.7	19.4	38.2
CS-CENSUS	25.2	6.59	21.1	19.2	26.1
MCT	11.2	6.59	21.1	19.2	23.0
MV	12.9	6.53	21.0	19.3	23.3
SWMV	8.47	3.99	14.7	12.1	19.3
SWMCT	11.1	5.38	14.7	12.2	20.2
Gradient Fusion	28.8	31.8	25.2	22.4	35.9

We have presented only the results from the "all" column, because otherwise the table would have been too large. The significance of the all method is: All (including half-occluded) regions (white) and border regions (black). This part highlights the errors in a region of interest, the black borders of the image are not taken into account.

We can see that our method using mean variation (MV) block matching, clearly surpasses other methods with respect to the quality of the results. We can also observe that the gradient fusion offers very poor results compared with classical block matching even though at a first glance the results seem better.

MCT and MV refer to the modified binary descriptor and mean variation binary descriptors which are the proposed patterns applied without the window shifting method. We can see that by applying the window shifting we obtain an increase of 4% in case of the MV (SWMV) and 2.8% in case of the MCT (SWMCT). We also see that the proposed patterns outperform the CS-Census and Census descriptors. To be noted that in case of the MCT pattern 5 threshold values have been tested (t = 0, 1, 2, 3, 4). We have observed that after the value of 1 the disparity maps begin to degrade. We have also tested the above mentioned threshold values for the case of the shifted windows. In this case, we also conclude that the best threshold value for the t parameter is 1. Alternative results

by our SWMV algorithm against the TYZX reconstruction. The images were obtained with a stereo system in driving scenarios.

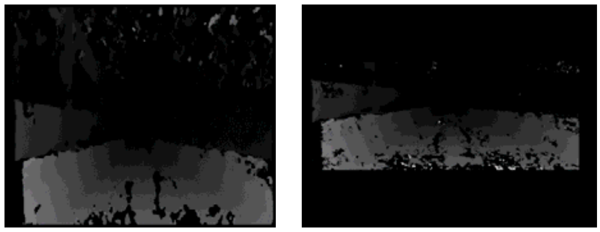


Figure 9. Our Result compared to the Tyzx version

In the Table II we can see the running times of the algorithms implemented on the system that was specified earlier. All the specified methods have been implemented and run on the test computer for a number of 20 times and the average time value has been extracted. The maximum disparity considered is 64 and the matching block size is of 9 x 9. The size of the tested images is 512 x 383 pixels.

TABLE II. RUNNING TIME OF THE BLOCK MATCHING ALGORITHMS

Method	Results on several images
	Average time in milliseconds (ms)
Census	55.6
CS-CENSUS	35.5
MCT	50.5
MV	58.8
Our Method (SWMV)	288,3
Our Method (SWMCT)	258.6
Gradient Fusion	297.3

TABLE III. CHOOSING THE CORRECT PENALTY FOR OUR ALGORITHM

Penalty	Results of several penalties applied to the SWMV algorithm				
	Tsukuba	Venus	Teddy	Cones	Average bad pixels
0.65	9.42	4.23	14.8	12.7	19.9
0.68	9.27	4.15	14.7	12.6	19.8
0.75	8.89	3.99	14.7	12.6	19.5
0.95	8.47	3.99	14.8	12.1	19.3

Although the CS-Census method is the fastest one, due to its low number of comparisons, our method with SWMV produces the best results from the reconstruction quality point of view. It is clear that further optimizations have to be applied in order to lower the processing time, but this should not pose a problem due to the highly parallel structure of our reconstruction algorithm.

Several experiments have been done in order to select the best penalties for the algorithm. In table III we present several penalties and results for the Shifted Windows with Mean Variation (SWMV) binary descriptor. We have tested all penalties starting from 0 until 1 having a step of 0.01. We have found that the results are similar for the MCT descriptor.

The average error will start to grow after 0.95, therefore the value chosen for our penalty is 0.95. To be noted that the authors in [8] have also found that their best results were obtained for a penalty of 0.95.

All the implemented local stereo block matching methods use a single 3D matrix for computing the cost volume, unlike the semi-global or global approaches that require much more memory. For this reason these type of algorithms are more appropriate for embeded systems.

V. Conclusions and Further Work

In this paper we have presented a variant for a Block Matching stereo algorithm which uses limited resources for computing good quality disparity maps. The main focus was to capture both types of surfaces (fronto-parallel and slanted) and offer a smooth transition among neighboring disparities. For this purpose we have proposed two types of binary descriptors, which in essence have the same structure and we have implemented a state of the art method of shifting each row of the descriptors, such that the effects caused by slanted surfaces are mitigated or removed. The algorithm was fused with the classical Block Matching algorithm, using the same descriptor, in order to capture frontal surfaces that could otherwise obtain a bad disparity. The obtained disparity map is smoothed using our smoothness constraint and a symmetric V interpolation is applied, in order to obtain a sub-pixel accuracy of the disparity map. The main focus was on the stereo matching algorithm and no effort was, yet, given to the post processing of the disparity map. This is why experiments and comparisons with other BM algorithms that do not benefit from post refinement were carried out on the Middlebury benchmark.

More experiments have been done in order to prove the methods advantage over classical BM. Several penalties were tested and, as a byproduct, the penalty which is best suited for both binary descriptors was found.

In future work, we will first focus on improving the running time of our stereo reconstruction method by using paralelization and hardware acceleration mechanisms (GPU implementation or SSE intrinsics) and then analyze, design and implement the refinements and post processing steps of the obtained disparity maps, which are equally important for improving the accuracy of stereo reconstruction algorithm.

VI. Acknowledgement

This work has been supported by ARTEMIS JU in the frame of the European FP7 research project Reconfigurable ROS-based Resilient Reasoning Robotic Cooperating Systems (R5COP), grant agreement no. 621447 / 2014.

References

[1] I. Haller and S. Nedevschi, "Design of Interpolation Functions for Subpixel-Accuracy Stereo-Vision Systems," *Image Processing, IEEE Transactions on,* vol. 21, pp. 889-898, 2012.

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

[3] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, pp. I-195-I-202 vol.1.

[4] J. I. Woodfill, G. Gordon, D. Jurasek, T. Brown, and R. Buck, "The Tyzx DeepSea G2 Vision System, ATaskable, Embedded Stereo Camera," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, 2006, pp. 126-126.

[5] I. Ernst and H. Hirschmüller, "Mutual Information Based Semi-Global Stereo Matching on the GPU," in *Advances in Visual Computing*. vol. 5358, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli*, et al.*, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 228-239.

[6] R. Spangenberg, T. Langner, and R. Rojas, "Weighted Semi-Global Matching and Center-Symmetric Census Transform for Robust Driver Assistance," in *Computer Analysis of Images and Patterns*. vol. 8048, R. Wilson, E. Hancock, A. Bors, and W. Smith, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 34-41.

[7] B. Ranft and T. Strauss, "Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 1941-1947.

[8] N. Einecke and J. Eggert, "Block-matching stereo with relaxed fronto-parallel assumption," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, 2014, pp. 700-705.

[9] N. Einecke and J. Eggert, "Stereo image warping for improved depth estimation of road surfaces," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 189-194.

[10] A. Murarka and N. Einecke, "A Meta-Technique for Increasing Density of Local Stereo Methods through Iterative Interpolation and Warping," in *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, 2014, pp. 254-261.

[11] Y. Kuk-Jin and K. In So, "Adaptive support-weight approach for correspondence search," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, pp. 650-656, 2006.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Rob. Res.,* vol. 32, pp. 1231-1237, 2013.

[13] S. Hermann, R. Klette, and E. Destefanis, "Inclusion of a Second-Order Prior into Semi-Global Matching," in *Advances in Image and Video Technology*. vol. 5414, T. Wada, F. Huang, and S. Lin, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 633-644.

[14] G. Saygili, L. Van Der Maaten, and E. A. Hendriks, "Stereo Similarity Metric Fusion Using Stereo Confidence," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, 2014, pp. 2161-2166.