Patch warping and local constraints for improved block matching stereo correspondence

Mircea Paul Muresan, Sergiu Nedevschi, Radu Danescu
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
{mircea.muresan, sergiu.nedevschi, radu.danescu}@cs.utcluj.ro

***Abstract*—Depth estimation of the surrounding environment using a stereoscopic camera setup is an important and fundamental research topic in computer vision. Due to its running time and quality performance in real situations the semi global matching algorithm is often used. The biggest disadvantage of the semi global approach is its large memory footprint. On the other hand, block matching stereo is leaner when it comes to memory consumption and therefore it is commonly used in applications where we do not have many resources, in order to obtain coarse depth information of the environment. The poor quality performance of such algorithms make them impractical for many real life applications. In this paper we focus on improving the quality of the classical block matching (BM) stereo method by proposing a novel approach which tackles the problem of stereo matching for slanted and fronto-parallel surfaces by using different types of binary masks on the matching window. Another improvement consists in the usage of different types of local constraints in the generation of the winning disparity for a specific position, such that possible outliers are eliminated from the start. The validation of our results has been done on the KITTI stereo benchmark dataset.**

Keywords—stereo correspondence, block matching, 3D reconstruction, local stereo.

I. INTRODUCTION

Depth estimation from binocular imagery is a core issue to many applications like intelligent vehicles [2], biometry [1], metrology [3], etc. A stereo system usually consists of two side by side cameras observing the same scene. After rectifying the images acquired from the stereo sensor, the 3D information can be obtained by finding the offset between the column coordinates in the left and in the right image, for the same observed feature of the environment. This offset is called disparity (d) and it is inversely proportional to the distance (Z) to the object as presented in equation 1. The letters B and f denote the baseline of the stereo sensor and the equivalent focal length of the canonical system obtained from rectification.

$$d = \frac{Bf}{Z} \tag{1}$$

According to the taxonomy proposed by Szelinski[4], a typical stereo matching algorithm consists of the following steps: matching cost computation, cost aggregation, optimization and refinement. Depending on how we approach these stereo correspondence steps, there exist two main directions for finding disparity maps, i.e. local and global block matching. The global block matching approach tries to solve the stereo problem by minimizing a global energy function. This method offers very good end results at the cost of large computational complexity and high running time. A subclass of the global methods is represented by the semi global algorithms [5]. The energy function for this subclass of algorithms is simpler than the one used for global algorithms, and therefore this method can offer good results in decent running time; this is the reason why the semi global method has gained popularity in over the years. Local block matching is the second class of stereo correspondence algorithms where the pixels from one image are compared with the ones from the second image, and there is no global energy minimization constraint. Single pixel matching is more likely to provide erroneous results, and therefore the matching is usually done using rectangular windows called blocks. The drawback of this matching scheme relies in the fact that it considers the same disparity across the matching window, and consequently assumes that the image consists of frontally viewed planes which are perpendicular to the optical axis of the camera.

This assumption does often not hold for applications which have to observe other surfaces, such as the road, side objects and building facades. For such objects, the quality of disparity computation will be lower.

Even though local block matching algorithms produce results of lower accuracy, these algorithms tend to be much faster than the global or semi-global ones. Another advantage of block matching compared to other classes of algorithms is the low memory requirement. In order to preserve the advantages of high speed and low memory footprint, while also improving the quality of the disparity estimation results, a lot of research has been done towards tackling the problems which are likely to cause the stereo correspondence errors.

The rest of the paper is structured as follows: in the following section we present related work for stereo block matching, in section three we present the paper's main contributions, and in the fourth section the experimental results are shown. The last section highlights the conclusions of the paper and presents further directions for our work.

978-1-5090-3899-2/16/$31.00 ©2016

II. RELATED WORK

The idea behind block matching consists in finding the correspondences between patches (usually called blocks) in the left and right image by searching on the epipolar lines. There exist a lot of functions used as matching criteria[6], out of which the most frequently used are: sum of absolute differences (SAD), rank transform (RT), census transform (CT), normalized cross correlation (NCC). An embedded real time solution called the DeepSea, which was created by the TYZX Company and uses local block matching stereo is presented in [7]. This solution is implemented in FPGA and ASIC and runs with the classical BM having the census transform as matching criterion.

More modern BM approaches that are trying to address the problems of slanted surfaces are presented in [8, 9, 10, and 11]. In [8] each individual row of a matching block is shifted and a penalty is incorporated in the final score in case the best matching cost does not come from the classical rectangular block. In [9] multiple operations of sheering and scaling of the original image pairs are performed in order to capture non-frontal planes. The resulted disparities are fused together to form the final depth map. In [10] the authors perform a linear image warping according to the expected disparity for the ground plane. Finally in [11] the non-frontal problem is approached by computing the cost from different disparity values. The authors penalize the score whenever the best cost does not come from the fronto-parallel block; however the penalty is not encapsulated in the final matching value. Another interesting BM method is presented in [12]. In this method multiple blocks of different shapes and sizes are aggregated in order to eliminate the effect caused by non-frontal surfaces. This aggregation method also solves the fattening effect which is usually caused by large block sizes.

III. PROPOSED SOLUTION

In this paper we will tackle the improvement of BM stereo in the presence of slanted surfaces. The block matching on these surfaces fails because the distance to the camera is not constant within the matching window. By using a multi block matching approach many errors are filtered out. However, due to the fact that when we are computing the matching descriptors we are not considering how the surfaces are tilted, we may be introducing auxiliary matching errors. In this section we will demonstrate that by creating oriented matching descriptors for the blocks in the right image, we are able to improve the matching quality. We will also illustrate that if we make local constraints on individual values in the winner takes all stage of the stereo correspondence pipeline we will be able to filter out many erroneous results.

Due to their low computational complexity and invariance to additive and multiplicative offsets in intensity we will be using binary descriptors in our paper.

A. Matching descriptors

The descriptor cost computation plays an important role in finding correct correspondences. Binary descriptors like census and its many variations [8], are offering good results mainly because they are invariant to additive and multiplicative offsets in intensity and the time needed to compute them is relatively small. For this reason in our approach we have chosen a dense census descriptor. In the proposed method different shapes are chosen for the left and right patch descriptors. For the left census image the shape of the matching window remains the same as in the classical fronto-parallel approach (a 7x7 descriptor as described in equation 2). For the right descriptor patch we add to the descriptor cost four more pixels, from the upper right corner and four more pixels from the lower left corner. The expression which leads to the formation of the right image descriptor is expressed in equation 4. These pixels are selected in order to have the necessary information when warping an image patch. An intuitive depiction of the image patches is illustrated in figure 1.

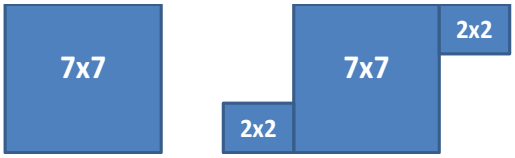

Fig. 1. Descriptor patch shapes for the left and right images

$$T(u,v) = \bigoplus_{i=-n}^{n} \bigoplus_{j=-n}^{n} \varepsilon(I(u,v), I(u+i,v+j)) \tag{2}$$

We denote by the symbol "$\oplus$" the bit-wise concatenation of bit strings. The auxiliary function ε is defined in equation 3.

$$\varepsilon(x,y)=\begin{cases}0, if\ x\le y\\ 1, if\ x>y\end{cases} \tag{3}$$

$$\begin{aligned} T(u,v) = &\bigoplus_{i=-n}^{n} \bigoplus_{j=-n}^{n} \varepsilon\left(I(u,v),I(u+i,v+j)\right) \mid \\ &\bigoplus_{i=-n}^{-n+2} \bigoplus_{j=n}^{n+2} \varepsilon\left(I(u,v),I(u+i,v+j)\right) \mid \\ &\bigoplus_{i=n-2}^{n} \bigoplus_{j=-n-2}^{-n} \varepsilon\left(I(u,v),I(u+i,v+j)\right) \end{aligned} \tag{4}$$

By the "|" operator we denote the bit-wise concatenation of the descriptor blocks. The bit-strings from the 2x2 blocks are appended to the end of the 7x7 descriptor.

B. Hamming Distance Computation

In order to compute the matching cost between the left and right descriptor images the hamming distance is used. In the classical approach, for each pixel in the left image, a pixel with similar appearance is searched in the right image over a number of positions (candidate disparities). The winning disparity represents the position where we have obtained the minimum Hamming distance.

In our solution we have used a set of binary masks with whom we are choosing specific pixels from the right descriptor image. The creation of the new descriptor is composed by a series of logical ands, binary shifts, and concatenation operations. The binary masks were chosen such that they can cover several warping angles. These warping angles were chosen arbitrarily. We are using the binary masks in order to tilt the descriptor patch window so that we can better capture the orientation of a surface.

After the new descriptor has been formed, the Hamming distance computation is used. The advantage of using binary masks consists in the fact that we know exactly what the position of each bit is and the extraction operations are done very fast. In figure 2 we illustrate an example of cropping the bits for the fronto-parallel case and for one non-frontal scenario. In order to maintain a decent running time we have chosen just three warping levels.

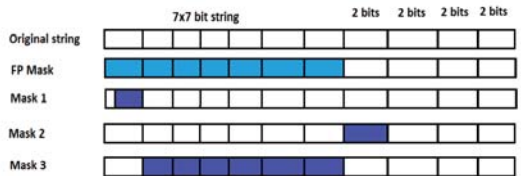

Fig. 2. Bit extraction using binary masks example. The colored positons represent values of one and the white regions represent values of 0.

In case of the frontal surfaces the final value is obtained as described in equation 5.

$$FinRez_{FP} = OriginalString \ \& FPMask \tag{5}$$

In case of the non-frontal scenario we compute a new bit string descriptor.

$$\begin{aligned} FinRez_{NFP} &= ((Mask1 \ \& OriginalString) \ll 2) \mid \\ &((Mask2 \ \& OriginalString) \ll 42) \mid \\ &(Mask3 \ \& OriginalString) \end{aligned} \tag{6}$$

By the "&" operator we denote the logical and operation, by the "<<" symbol we denote the shift left operation and by the "|" operator we denote the appending operation. All the used binary masks are hard coded so we know exactly the position of each pixel.

The result is the minimum Hamming distance obtained for all the frontal and non-frontal surfaces. The final score for the correlation of left image *i*th pixel with right image pixel at position (disparity) *j* using mask *k* is given in equation 7. By the "∑" operator we denote an iteration procedure, not a summation operation. In order to favor frontal surfaces, we add an extra penalty in case the best score comes from a slanted patch. In our solution we have obtained the best results for a penalty of 5.

$$\begin{aligned} FinScore_{i,j} &= Min(Hamming(ImageLeft_i, FinRezFrontalRight_j), \\ &\sum_{k=1}^{3} Hamming\left(ImageLeft_i, FinRezsSlantRight_{j,k}\right) + Penalty) \end{aligned} \tag{7}$$

After computing the Hamming distance a multi block aggregation procedure is used for improving the quality of the depth map and for reducing the possible outliers. The block sizes used are: 1x137, 137x1, 17x17, 7x7. The aggregation scheme is similar to the one presented in [11], the only difference consisting in the block sizes used as presented in equation 8. These block sizes were chosen based on experimental results. The sizes of the matching blocks were varied and the blocks for which we had the best matching scores were selected. We have used the KITTY data set for adapting the block sizes. We have also tried to introduce auxiliary matching blocks, however the results were not satisfactory.

$$AggregatedValue = Max(val_{1x137}, val_{137x1}) * val_{17x17} * val_{7x7} \tag{8}$$

By $val_{blockSize}$ we denote the value obtained by summing the individual values within a block for one of the specified sizes. By max we are referring to a function that takes the maximum number between two inputs.

C. Winner takes all

As it can be seen from equation 9, in the winner takes all step a disparity value is chosen from the computed cost volume *C* for a certain pixel position *p*.

$$D(p) = argmin_d \ (C(p,d)) \tag{9}$$

The generated disparity values may be erroneous for multiple reasons: problems caused by reflectance, lack of texture, repetitive patterns etc. For this reason in this paper we propose several types of local constraints which are used in the process of generating the winning disparity. In order for a disparity value to be valid all the constraints have to be passed.

First of all, instead of finding just one minimal value and its corresponding index, we are searching for the first three minimal values.

The first constraint we have put is that the ratio between the first minimum and the third minimum is lower than a threshold. This constraint would ensure that the minimal value is at a very steep position. This condition is doubled by a second check where we verify how close the second minimal value is to the smallest value. We label these two conditions as the confidence flag constraint (10), and two confidence constraint thresholds have been determined experimentally for both of the cases described above. The confidence check also gives us a measure of how correct a certain pixel got reconstructed. For example, a small ratio value would mean that the selected disparity is very good while a large ratio value would imply that the selected value may not be correct.

$$ConfidenceFlag = \begin{cases} 1, \ \dfrac{min3}{min1} \le Confidence1 \ AND \ |min1 - min2| \le Confidence2 \\ 0, otherwise \end{cases} \tag{10}$$

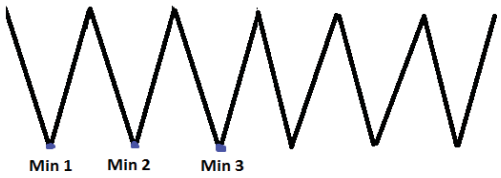

Fig. 3. Graphical representation of the periodicity in the cost volume.

The second constraint refers to the minimal value's periodicity (11). In case the minimal value is periodical, it means the reconstructed surface may be un-textured or it may have repetitive patterns. In this case we cannot say very much about that surfaces disparity.

$$PeriodicityFlag = \begin{cases} 1, min1 = min2 \;\; AND \; min1 = min3 \\ 0, otherwise \end{cases} \quad (11)$$

A graphical depiction of this phenomenon can be observed in figure 3. Each spike represents a value where the disparity reaches a minimum point. In case there are many such points the surface corresponding to that pixel has very little information.

The last constraint refers to the presence of an upside down "hill" type structure in the cost volume near the winning disparity. In case the minimal values obtained are different, and they pass the confidence constraint, we have to determine how "sharp" is the position where the minimum is located. Figure 4 intuitively demonstrates this concept and in equation 12 the analytical expression is revealed.

$$HillFlag = \begin{cases} 1, |\min 3 - \min 1| > Threshold \; AND \; |\min 1 - \min 2| > Threshold \\ 0, Otherwise \end{cases} \quad (12)$$

The values used for the two confidence thresholds are 15 and 20000, and the value for the hill threshold is 50000. All values were identified experimentally. A disparity value is generated if the confidence flag is 1, the periodicity is 0 and the hill flag is set to 1, otherwise we do not generate a disparity value for that corresponding pixel. The next step consists in a subpixel interpolation procedure, such that the final disparity value is more precise. For this step we have used symmetric V, which was presented in [13]. The subpixel interpolation equation is presented in 13.

$$Disp_{final} = Disp_{integer} + \begin{cases} 0.5 - 0.25 \cdot \left(\dfrac{(M_3 - M_1)^2}{(M_2 - M_1)^2} + \dfrac{M_3 - M_1}{M_2 - M_1} \right), if \; M_2 > M_3 \\ -\left(0.5 - 0.25 \cdot \left(\dfrac{(M_2 - M_1)^2}{(M_3 - M_1)^2} + \dfrac{M_2 - M_1}{M_3 - M_1} \right) \right), if \; M_2 \le M_3 \end{cases} \quad (13)$$

The parameters M_1, M_2, M_3 are the correlation values belonging to the current winning disparity and its neighbors. After the subpixel interpolation step two additional stages are carried out in order to refine the disparity map.

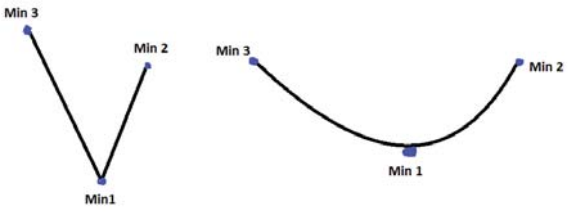

Fig. 4. Illustration of the sharpness of the position where the minimum value is located

These steps are a background fill in stage, which is performed in case of occlusions [14] and LR check .The analytical expressions for these steps are described in equation 14.

The final disparity map is further refined using a speckle removing technique and a median filtering blur using 1x9 and 9x1 kernels. The speckle size that is considered to be filled is 200 pixels.

$$\begin{cases} correct, if \left| d - D^R(pd) \right| < 1 \; for \; d = D^L(p) \\ missmatch, if \left| d - D^R(pd) \right| < 1 \; for \; any \; other \; d \\ occlusion, otherwise \end{cases} \quad (14)$$

No background interpolation technique has been implemented.

IV. Experimental results

In this section we present an evaluation of the proposed algorithm in terms of quality. We will compare our results to the results of classical stereo block matching algorithms and to other existing algorithms as well. The evaluation has mainly been done on the KITTI stereo dataset, however we have also tested our solution on Middlebury [15] stereo images.

Fig. 5. Bottom image is the left intensity image, in the second image the output of our algorithm is presented, in the first image the result of the census algorithm without our optimization is illustrated.

The system on which we implemented our method contains an Intel I5-2500 CPU with 3 Ghz frequency.

No hardware acceleration methods have been used. Open MP has been used to parallelize some parts of the code. The error threshold used in our testing is 2px. The evaluation metric which is of interest to us is Out-All, meaning we evaluate our image on all pixels for which we have ground truth information, even if they are occluded.

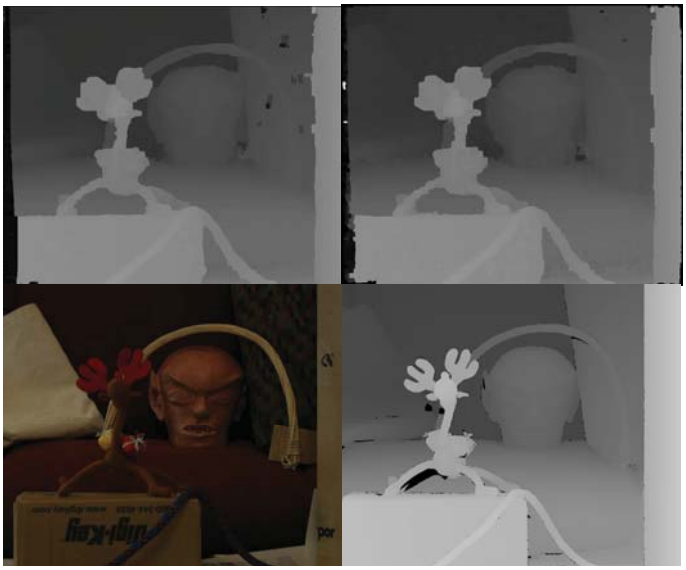

Fig. 6. Bottom image is the left intensity image and its ground truth, in the top left image we have the image obtained with the census descriptor, in the top right image the image obtained with our approach is revealed

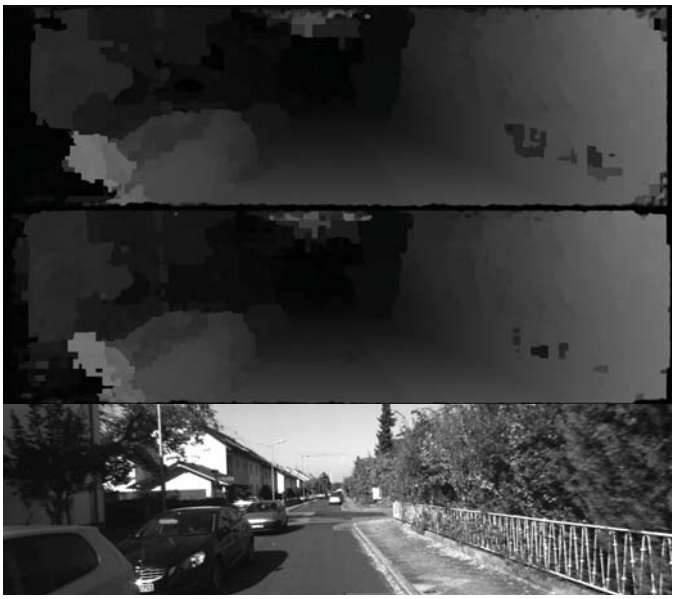

Fig. 7. The bottom image is the left intensity image, second image depicts the disparity map using our algorithm, the first image represents disparity image using the sparse census approach

In figure 5 we illustrate the result of our algorithm in comparison to a 7x9 census descriptor. The same aggregation and refinement scheme has been used for both algorithms. As we can see our solution is able to reconstruct better the fence on the right and other slanted surfaces present in the image.

In figure 6 a set of stereo images from the Middlebury database are presented. The classical 7x9 census transform was used on the top left image and our method was applied for the top right image. The same parameters were used for both images. As it can be seen the multi block method approach using census does not reconstruct, as good as the method presented in this paper, the colored pillow which is tilted towards the head.

Figure 7 illustrates another outdoor automotive scenario where our solution has better results compared to the classical approaches. As it can be seen the right slanted wall presents some artifacts when using the sparse census approach. When generating the disparity image with our algorithm the artifacts are reduced.

In figure 8 we present the Tsukuba image with its ground truth and in figure 9 we illustrate the result of our algorithm on this image in comparison to the census block matching algorithms.

It should be noted that for the Middlebury images we had to change the aggregation block size. For the Tsukuba image pairs we have used 61x1, 1x61, 13x13 and 3x3 blocks, while for the image with the reindeer we used 150x1, 1x150, 21x21, 7x7 block sizes. The penalty for our algorithm when we were using the Middlebury dataset was set to 4. In future work we will have to make an analysis of how the image size and content affect the aggregation blocks dimensions.

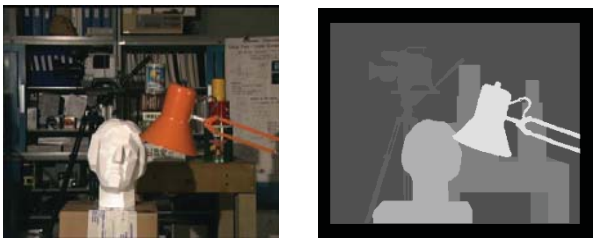

Fig. 8. Middlebury tsukuba image with ground truth

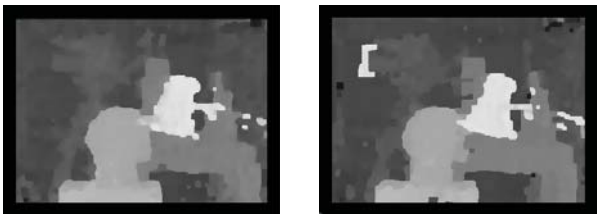

Fig. 9. The left image is obtained using our method while the right image is using a classical census descriptor.

In table 1 we present the comparison of our algorithm and other block matching algorithms which are based on binary descriptors. The metric used is out-all. The density illustrates the percentage of the image that has been filled. No background interpolation has been used in with any of the

tested methods from table I. This is why for all implemented solutions the density is less than 100%.

TABLE I. EVALUATION ON KITTY DATA-SET

Method	Results on Kitty images	
	Density	Out-All error
Census	99.67%	11.865%
WCS-CENSUS	99.98%	11.24%
MCT	99.68%	11.21%
OurMethod	**99.98%**	**10.27%**
Sparse Census	99.51%	13.29%

The KITTY dataset contains outdoor traffic scenarios. Many researchers have tested their algorithms on this dataset, so the benchmark contains local, semi global and global submissions. A rough estimate of our algorithms position on the kitty benchmark can be seen in table 2.

TABLE II. COMPARISON WITH EXISTING METHODS FROM THE KITTY STEREO DATA-SET

Position	Method	Performance on Kitty dataset	
		Density	Out-All error
39	RBM	100%	10.05%
40	SGM	85.80%	10.16%
41	DLP	100%	10.19%
42	**OurMethod**	**99.98%**	**10.27%**
43	MSGM-LDE	100%	10.68%
44	Ensemble	100%	10.89%

The running time for the proposed algorithm on the KITTY dataset is 0.4 s. The current implementation can be accelerated in order to obtain a better running time; in the current paper however we have focused on improving the quality of the reconstruction. The only extra memory used in this approach is the one needed for storing the array of binary masks.

V. CONCLUSIONS AND FURTHER WORK

In this paper we have presented a new scheme for block matching stereo which uses limited resources for computing good quality depth maps. During the process of block matching the depth within a matching window is considered constant. This assumption does not often hold for various types of tilted surfaces. For this reason in the proposed solution the best descriptor for a surface is chosen by using a set of binary masks. The binary masks are used in order to tilt the matching window so that the best surface descriptor is created. A penalty is encapsulated in the matching value in case the best score comes from a slanted window. The second contribution of this paper consists in the creation of local constraints which are imposed on the cost volume when generating a winning disparity for a certain pixel position. These constraints help eliminate the outlying disparities from the start. We have conducted experiments using different kinds of binary descriptors and the proposed method proved superior in terms of quality. In our approach we have used a multi block aggregation approach in order to filter outliers and eliminate the fattening effect caused by large rectangular blocks. The sizes of the various blocks were taken experimentally using the kitty dataset ground truth as adjustment reference. We have observed that when using other images, the block sizes have to be adjusted. For our future work we plan to investigate how different types of images affect the sizes of the aggregation blocks used.

In future work, we will tackle the creation of a more flexible method of capturing slanted surfaces, which will work with any binary descriptor. We will also focus on improving the running time of our algorithm.

VI. ACKNOWLEDGEMENT

This work has been supported by UEFISCDI (Romanian National Research Agency) in the national research project Multi-scale multi-modal perception of dynamic 3D environments based on the fusion of dense stereo, dense optical flow and visual odometry information (MultiSens), project no. PNII- ID-PCE-2011-3-1086.

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.

This work was supported by the MULTIFACE grant (Multifocal System for Real Time Tracking of Dynamic Facial and Body Features) of the Romanian National Authority for Scientific Research, CNDI–UEFISCDI, Project code: PN-II-RU-TE-2014-4-1746.

REFERENCES

[1]. Huang, J., V. Blanz, and B. Heisele, Face Recognition with Support Vector Machines and 3D Head Models. Center for Biological and Computer Learning, M.I.T, Cambridge, MA, USA and Computer Graphics Research Group, University of Freiburg, Freiburg, Germany

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

[3]. Lyndon N. Smith ; Melvyn L. Smith; Stereo vision technology for object measurement. Proc. SPIE 5011, Machine Vision Applications in Industrial Inspection XI, 307 (May 19, 2003);

[4]. Scharstein, D., Szeliski, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", International Journal of Computer Vision, vol. 47 no.1-3, pp. 7-42, April-June 2002.

[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]. 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.

[7]. M. P. Muresan, M. Negru and S. Nedevschi, "Improving local stereo algorithms using binary shifted windows, fusion and smoothness constraint," Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on, Cluj-Napoca, 2015, pp. 179-185.

[8]. B. Ranft and T. Strauß, "Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching," 17th International

IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, 2014, pp. 1941-1947.
[9]. N. Einecke and J. Eggert, "Stereo image warping for improved depth estimation of road surfaces," Intelligent Vehicles Symposium (IV), 2013 IEEE, Gold Coast, QLD, 2013, pp. 189-194.
[10]. N. Einecke and J. Eggert, "Block-matching stereo with relaxed fronto-parallel assumption," *2014 IEEE* Intelligent Vehicles Symposium Proceedings, Dearborn, MI, 2014, pp. 700-705.
[11]. N. Einecke and J. Eggert, "A multi-block-matching approach for stereo," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, 2015, pp. 585-592
[12]. 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.
[13]. Žbontar, Jure, and Yann LeCun. "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." arXiv preprint arXiv:1510.05970 (2015).
[14]. 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.
[15]. 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-202vol