2010 IEEE Intelligent Vehicles Symposium
University of California, San Diego, CA, USA
June 21-24, 2010

Temporal Consistent Fast Stereo Matching for Advanced Driver Assistance Systems (ADAS)

Mohamed El Ansari, Stéphane Mousset, Abdelaziz Bensrhair, George Bebis

Abstract— **In this paper, we present a new fast method for matching stereo images acquired by a stereo sensor embedded in a moving vehicle. The method consists in exploiting the matching results obtained in one stereo pair (frame) for computing the disparity map of the following stereo pair. This can be achieved by finding a temporal relationship, which we named association, between consecutive frames. The disparity range of the current frame is deduced from the disparity map of the preceding frame and the association between the two frames. Dynamic programming technique is considered for matching the image features. The proposed approach is tested on virtual and real stereo image sequences and the results are satisfactory. The method is fast and able to provide about 20 millions disparity maps per second on a HP Pavilion dv6700 2.1GHZ.**

I. Introduction

An intelligent vehicle (IV) can perform road obstacle detection by knowing its environment. Stereo vision [1] is a well-known method used to obtain accurate and detailed 3D representation of the environment around an IV. The key problem in stereo vision consists in finding correspondence between pixels of stereo images taken from different viewpoints [2]. Exhaustive surveys on the methods tackling the correspondence problem are available in [3], [4]. A taxonomy of dense stereo correspondence algorithms together with a testbed for quantitative evaluation of stereo algorithms is provided by Scharstein and Szeliski [5]. The taxonomy shows that graph cuts-based methods [6] outperform the other methods, but they are time consuming which make them not suitable for real-time applications, e.g. ADAS.

Although there is strong support that the incorporation of temporal information can achieve better results [7], [8], [9], only a small amount of research has been devoted to the reconstruction of dynamic scenes from stereo image sequences. We believe that by considering the temporal consistency between consecutive frames the stereo matching results could be improved [10]. Based on this principle, this paper presents a new real-time stereo matching approach dedicated to ADAS. The current work constitutes an improvement of that presented in [10]. The features to be matched from stereo images are declivities (edge points) [11]. The declivity operator is chosen because it is precise, fast, and self-adaptive. The main idea of the method we are presenting is the use of the temporal relationship between current frame and its preceding one to determine the disparity range (possible disparities) of the former one. Dynamic programming technique [12] is considered for matching edge points of the stereo sequences. The new method is tested on both virtual and real stereo image sequences and gives promising results.

The remainder of the paper is organized as follows. Section II overviews some of the stereo methods handling stereo sequences and using temporal consistency. Section III presents the method used to extract primitives. The new stereo method is detailed in section IV. Experimental results are shown in section V. Section VI concludes the paper.

II. Related work

In the recent years, several techniques have been proposed to obtain more accurate disparity maps from stereo sequences by utilizing temporal consistency [7], [9], [13], [8]. Most of these methods use either optical flow or spatiotemporal window for matching stereo sequences. In their approach, Tao et al. [13] proposed a dynamic depth recovery in which a scene representation, that consists of piecewise planar surface patches, is estimated within an incremental formulation. Such a representation is derived based on color segmentation of input images. Each segment is modeled as a 3D plane. The motion of this plane is described using a constant velocity mode. The spatial match measure and the scene flow constraint [14] are investigated in the matching process. The accuracy of the results and the processing speed are limited by the image segmentation algorithm used. Vedula et al. [14] present a linear algorithm to compute 3D scene flow based on 2D optical flow and estimate 3D structures from the scene flow. In [15], the temporal consistency was enforced by minimizing the difference between the disparity maps of adjacent frames. This approach is designed for offline processing only, i.e. it takes pre-captured stereo sequences as input and calculates the disparity maps for all frames at the same time. In [9], an algorithm has been developed to compute both disparity maps and disparity flow maps in an integrated process. The disparity map generated for the current frame is used to predict the disparity map for the next frame.

M. El Ansari is with LabSIV, Department of Mathematics and Computer Science, Faculty of Science, University of Ibn Zohr, 80000 Agadir, Morocco. He is also an associate member at the LITIS Lab., INSA-Rouen, France. Currently, M. El Ansari is a Fulbright Visiting Scholar at Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA. m.elansari@univ-ibnzohr.ac.ma

S. Mousset is with the LITIS Lab., INSA Rouen, B.P. 08, F-76131 Mont-Saint-Aignan Cedex, France stephane.Mousset@insa-rouen.fr

A. Bensrhair is with the LITIS Lab., INSA Rouen, B.P. 08, F-76131 Mont-Saint-Aignan Cedex, France Abdelaziz.Bensrhair@insa-rouen.fr

G. Bebis is with the CVL Lab., Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA. G. Bebis is a visiting professor at Department of Computer Science, King Saud University, Saudi Arabia. bebis@cse.unr.edu

978-1-4244-7868-2/10/$26.00 ©2010 IEEE

The disparity map found provides the spatial correspondence information which is used to cross-validate the disparity flow maps estimated for different views. Programmable graphics hardware have been used for accelerating the processing speed.

Zhang et al. [8], propose to extend the existing traditional methods by using both spatial and temporal variations. The spatial window used to compute SSD cost function is extended to a spatiotemporal window for computing sum of SSD (SSSD). Their method could improve the results when we deal with static scenes and with structured light. It fails to do so with dynamic scenes. Davis et al. [7] have developed a similar framework as the one in [8]. However, their work is focused on analyzing and presenting results for geometrically static scenes imaged under varying illumination. Given an input sequence taken by a freely moving camera, Zhang et al. [16] propose a novel approach to construct a view-dependent depth map for each frame. Their method takes a one sequence as input and provides the depth for the different frames, i.e. offline processing. It is not applicable in an IV.

Our approach is different from the aforementioned ones. It uses neither optical flow nor spatiotemporal window. As temporal integration, we propose to use what we call *association* (see section IV-B) between consecutive frames. Once the association between the current frame and its preceding one is found, a pre-estimated disparity map of the current frame can be inferred. By analyzing the pre-estimated disparity we can determine the disparity range authorized for each scanline (potential matching candidates). The disparity range is used by the dynamic programming algorithm for selecting the valid nodes.

III. Image segmentation

The first step in stereo vision consists in extracting significant features from the stereo images to be matched. In this work, we are interested in edge points as features to consider in the matching process. In order to be suited for computer vision applications, e.g. IV applications, the edge detector we choose should satisfy the following constraints : fastness, precision, and self-adaptivity. Therefore, we consider the so-called declivity [11] as edge detector because it meets the above mentioned constraints. In an image line, a declivity is defined as cluster of contiguous pixels, limited by two end-points which correspond to two consecutive local extrema of grey level intensity, i.e. one maximum and one minimum. As shown in Fig. 1, Dec_i and Dec_{i+1} are two adjacent declivities. The declivity Dec_i is limited by two end-points l_i and r_i. The grey-level intensities at the end-points are respectively $I(l_i)$ and $I(r_i)$. The same for the declivity Dec_{i+1}, their end-points are l_{i+1} and r_{i+1}, respectively. Each declivity is characterized by its amplitude, e.g. $a_i = I(r_i) - I(l_i)$ is the amplitude of Dec_i and $a_{i+1} = I(r_{i+1}) - I(l_{i+1})$ is the amplitude of Dec_{i+1}. Relevant declivities are extracted by thresholding these amplitudes. To be self-adaptive, the threshold value is defined by

$$a_t = 5.6\sigma \tag{1}$$

where σ is the standard deviation of the white Gaussian noise component in each image line, which is computed using the cumulative histogram of the absolute value of the gradient [11].

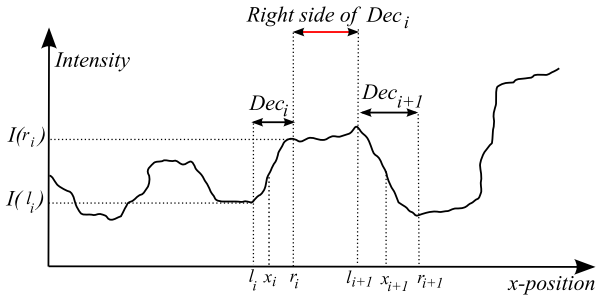

Fig. 1. Image line : characteristic parameters of a declivity.

The position of a declivity is computed using the mean position of its points weighted by the gradients squared. As an example, the position x_i of Dec_i is calculated as follows (See 1).

$$x_i = \frac{\sum_{x=l_i}^{r_i-1}[I(x+1)-I(x)]^2(x+0.5)}{\sum_{x=l_i}^{r_i-1}[I(x+1)-I(x)]^2} \tag{2}$$

For each declivity Dec_i, the following characteristics should be known to be used in the matching process:

- The x-coordinate x_i of Dec_i in the image line as defined in equation 2. x_i define the position of the edge point detected by the declivity operator. We note that in the subsequent of the paper *edge point* or *declivity* has the same meaning.
- The left and right end-points of Dec_i : l_i and r_i.
- The set of intensities of pixels situated between the right end-point r_i of Dec_i and the left end-point l_{i+1} of Dec_{i+1}, i.e. the declivity on the right side of Dec_i (see Fig 1). We call this set of pixels the *right side* of Dec_i.

More details about the declivity operator and how to determine the parameter σ are available in [11].

IV. Stereo matching algorithm

This section details the steps followed by the proposed method for matching stereo images provided by the stereo sensor mounted aboard an IV. We note that the stereoscopic sensor used in our experiments provides rectified images, i.e., the corresponding pixels have the same *y-coordinate*.

A. Matching constraints

In order to discard false matches, we consider some local constraints. The first one is geometric resulting from the sensor geometry, which assumes that a pair of declivities d_i^l and d_j^r appearing in the left and right scanlines, respectively, represent possible match only if the constraint $x_i^l > x_j^r$ is satisfied [17]. x_i^l and x_j^r are the x-coordinates of d_i^l and d_j^r, respectively. The second constraint is the slope constraint, which means that only pairs of declivities with the same slope sign are considered as possible matches.

B. The association

The aim of this subsection is to describe the method used to find association between declivities of consecutive frames. Let I_{k-1} and I_k be two consecutive images of the same sequence, e.g. left sequence. Let C_{k-1} be a curve in the image I_{k-1} and C_k be its corresponding one in the image I_K. Consider two declivities P_{k-1} and Q_{k-1} belonging to the curves C_{k-1} and their corresponding ones P_k and Q_k belonging to the curve C_k (see Fig. 2). We define the associate point of the point P_{k-1} as the point belonging to the curve C_k which has the same *y-coordinate* as P_{k-1}. Note that the association is not correspondence neither motion. Two associate points are two points belonging to two corresponding curves of two consecutive images of the same sequence and having the same y-coordinate. From Fig. 2, we remark that the point Q_k meets these constraints. Consequently, Q_k constitutes the associate point of the point P_{k-1}. In practice, we assume that the movement of the objects from one frame to the other is small. So, if x_1 and x_2 represent the x-coordinates of P_{k-1} and Q_k, respectively, x_2 should belongs to the interval $[x_1 - \Delta x, x_1 + \Delta x]$, where Δx is a threshold to be selected. This constraint allows the reduction of the number of associate candidates. The gradient magnitude is used to choose the best associate one. As a similarity criterion, the absolute difference between the gradient magnitudes of the declivities is used. As we see in Fig. 2, the point P_k represents the match of the point P_{k-1}. However, the point Q_k constitutes the associate of the point P_{k-1}. We remark that the points P_k and Q_k are different because of the movement of the point P_k in the image I_k.

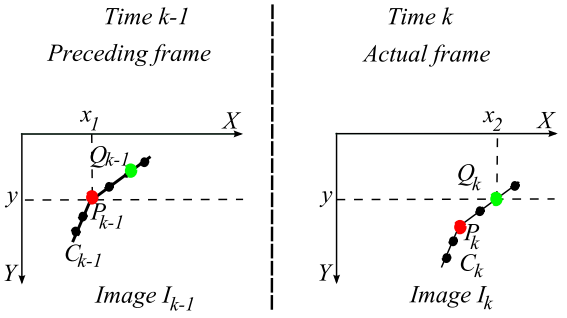

Fig. 2. I_{k-1} and I_k represent consecutive images of the same sequence, e.g. left sequence. The point Q_k in the image I_k constitutes the associate point of the point P_{k-1} in the image I_{k-1}. The points P_k and P_{k-1} are in red color. The points Q_k and Q_{k-1} are in green color. We mean declivity with point.

C. The pre-estimated disparity map

We define the so-called *pre-estimated disparity map* of a current frame as the disparity map deduced from its preceding frame and the temporal link between the two frames. In the rest of this subsection we show how to compute the pre-estimated disparity map of the current frame. Let I^L_{k-1} and I^R_{k-1} be the left and right stereo images of the frame f_{k-1} acquired at time $k-1$ and d_{k-1} is the corresponding disparity map. I^L_k and I^R_k are the left and right stereo images of the frame f_k acquired at time k. For each declivity in the image I^L_k (resp. I^R_k) we look for its associate one in the image I^L_{k-1} (resp. I^R_{k-1}), if any, by following the technique detailed in section IV-B. Knowing the association between the declivities of the frames f_k and f_{k-1} the pre-estimated disparity map of f_k is computed as follows:

```
for  i=1  to N do
Dec = Declivity(i);
if AssociateOf(Dec)  exists
     aDec = AssociateOf(Dec);
     if MatchOf(aDec)  exists
        maDec = MatchOf(aDec);
        if AssociateOf(maDec) exists
            amaDec = associate(maDec);
            disparity(Dec) = amaDec-Dec;
        endif
     endif
endif
endfor
```

The algorithm is executed independently for each image scanline. N denotes the number of the declivities present in the scanline for which the algorithm is performed. The association can be searched from frame f_k to frame f_{k-1}, and vice versa.

D. Disparity range

The accurate choice of the maximum disparity threshold value for almost any known stereo processing method is crucial to the quality of the output disparity map and the computation time [5], [18]. The following of this section describes how to compute the possible disparities (disparity range) of a current frame (f_k) on the basis of its pre-estimated disparity map (pd_k). Let H be a function of the image variable pd_k such that $H(pd_k) = vpd_k$. The image vpd_k is called the *v-disparity* image [19]. H accumulates the points with the same disparity that occur on a given image line. Details on how to construct the v-disparity image are available on [19]. The processing of the v-disparity image provides geometric content of road scenes. It was demonstrated in [19] that the obstacles and the road appeared as vertical and oblique lines, respectively. Let us assume that we have a road scene containing four objects. The corresponding v-disparity image should be as shown in Fig. 3. We remark that the v-disparity image contains four vertical lines representing four obstacles and one oblique line representing the road map. For computing the disparity range, we divide the v-disparity image into two parts: the top part containing the objects and bottom part containing the road map. The two parts are separated by the line $y = L_0$. We propose to find the disparity range independently for each part.

Let us start by the top part of the v-disparity image. A disparity value is associated to each object in the scene. We can deduce from the top part that the disparities of the detected objects belong to the interval $[d_1, d_2]$, where d_1 is the disparity of the farthest object and d_2 is the disparity of the closest object. In order to take into account the

uncertainty inherent to the computation, the disparity range can be chosen as $[d_1 - d, d_2 + d]$, where d is a threshold to select. d controls the number of possible candidates in the matching process. The authorized disparities at the scanlines $\{y = y_i\}_{1,..,L_0}$ should belong to the interval $[d_1 - d, d_2 + d]$, which is represented by the area situated between the lines $(D1)$ and $(D2)$ (the lines in blue color in Fig. 3).

In the bottom part, the road map is represented by an oblique line. We have only one possible disparity value for each scanline. For the scanline y_i the only possible disparity is $a * y_i + b$, where a and b are the oblique line equation parameters. In order to take into account the uncertainty inherent to the computation, the possible disparities at the scanline $\{y = y_i\}_{L_0+1,..,M}$, where M is the image height, should be between $a * y_i + b - d$ and $a * y_i + b + d$. In Fig. 3, the possible disparities is the area situated between the lines $(D3)$ and $(D4)$ (in green color). We remark that the disparity range in the top part is the same for all the image lines. However, it varies from scanline to scanline in the bottom part.

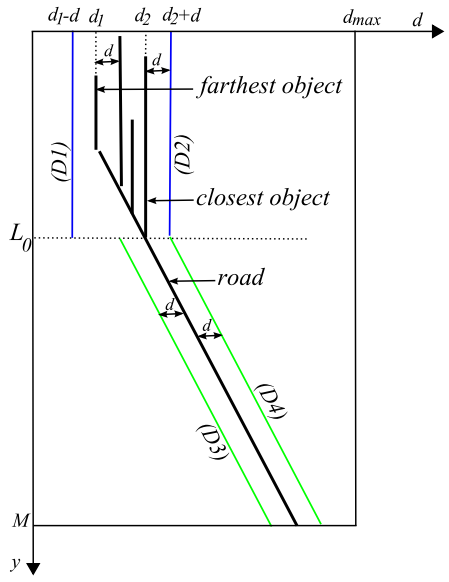

Fig. 3. v-disparity of the pre-estimated disparity map. The vertical axis refers to the image lines and the horizontal axis represents the disparities. M is the image height. d_{max} is the maximum disparity value. The possible disparities are the area between the lines $(D1)$ and $(D2)$ for the top part and the area between the lines $(D3)$ and $(D4)$ for the bottom part.

E. Cost function

As a similarity criterion between corresponding declivities, we use a cost function based on the variance of the intensities at the pixels situated on the right sides of the matched declivities. Let d_i^l and d_j^r be two declivities belonging to two corresponding epipolar lines on the left and right images, respectively. We denote by $S_l = \{f_m^l\}_{m=1,..,M^l}$ and $S_r = \{f_n^r\}_{n=1,..,M^r}$ their corresponding right sides, respectively. M^l and M^r are the numbers of pixels in S_l and S_r, respectively. We assume that corresponding declivities on the stereo images should have the same intensities at their right sides. Let $S = \{f_1^l, .., f_{M^l}^l, f_1^r, .., f_{M^r}^r\} = \{f_i\}_{i=1,..,M^l+M^r}$ be the union of S_l and S_r. Corresponding declivities should have similar right sides, i.e. the intensities of S_l and S_r should be similar or very close to each other. We use the variance of the intensities of S as a similarity criterion between d_i^l and d_j^r. Corresponding declivities should give a small variance value. The cost function is defined as follows.

$$C(d_i^l, d_j^r) = \frac{1}{M^l + M^r} \sum_{i=1}^{M^l+M^r} \left(f_i - \bar{f}\right)^2 \qquad (3)$$

where $\bar{f}$, is the mean of the intensities of S, defined as

$$\bar{f} = \frac{1}{M^l + M^r} \sum_{i=1}^{M^l+M^r} f_i \qquad (4)$$

F. Dynamic programming

Let $\{d_i^l\}_{i=1,..,N^l}$ and $\{d_j^r\}_{j=1,..,N^r}$ be two sets of declivities ordered according to their coordinates in an arbitrary l right and l left epipolar scanlines. N^l and N^r are the numbers of the declivities on the left and right scanlines, respectively. The problem of obtaining correspondences between declivities on right and left epipolar scanlines can be solved as a path finding problem on 2D plane [12]. Fig. 4 illustrates this 2D search plane. The vertical lines show the positions of declivities on the left scanline and the horizontal ones show those on the right scanline. We refer to the intersections of those lines as nodes. Nodes in this plane correspond to the stages in dynamic programming where a decision should be made to select an optimal path to that node. They represent the candidate matches. Optimal matches are obtained by the selection of the path which corresponds to minimum value of the global cost. The optimal path must goes from the upper left corner **S** to the lower right corner **G** monotonically due to the condition on ordering. Because of the non reversal ordering constraint, starting from **S**, a path can be extended towards only one of the three directions: east, south, or southeast.

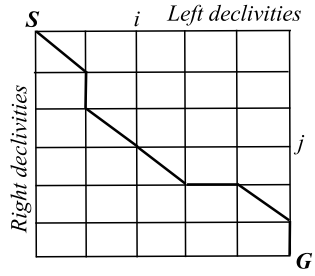

Fig. 4. 2D search plane. The horizontal axis corresponds to the left scanline and the vertical one corresponds to the right scanline. Vertical and horizontal lines are the declivity positions and path selection is performed at their intersections.

Based on the subsections IV-A and IV-D the possible matches (valid nodes in the search plane) on the left and right scanlines are searched as explained in subsections IV-A and IV-D. The cost function (Eq. 3) is used to fill in the valid

nodes in the search plane. After looking for the optimal path in the 2D search plane, the pairs of corresponding declivities on the corresponding scanlines are determined. We note that the matching process is achieved independently for each scanline.

V. Experimental results

In order to evaluate the performance of the proposed approach, it has been applied to virtual and real stereo sequences. In the subsequent of this section, we propose to call the new method temporal consistent matching (TCM) method. We call space matching (SM) method, the TCM method deprived of the disparity range computation step. The only difference between the two methods is that the TCM method computes the disparity range for each scanline, whereas the SM method uses a fixed maximum disparity value for the whole image. The comparison between TCM and SM methods illustrates the advantages of the computation of the disparity range before performing the matching process.

A. Virtual stereo image sequences

We have tested our method on the MARS/PRESCAN virtual stereo images available in [20]. The size of the images is 512×512. Fig. 5 illustrates the left stereo images of the frames #293 and #294 of the same sequence. We have applied the new method (TCM) to the original virtual sequences. The extracted edge points are depicted in Fig. 6. The disparity maps computed by the TCM method are shown in Fig. 7. We have used false colors for representing the disparity maps. The SM method, with maximum disparity value $d_{max} = 200$, has been used to initialize the TCM method.

Fig. 5. Virtual stereo sequences (left images of the frames #293 and #294).

Fig. 6. Edge points of the images shown in Fig. 5.

Table I summarizes the matching results obtained. It shows the number of matched edge points (NME), the percentage of correct matches (PCM), the number of correct matches (NCM), and the number of false matches (NFM) for the frames #293 and #294. In order to assess the performance of the TCM method, the SM method has been applied to the virtual sequence (Fig. 5) and the disparity maps obtained are depicted in Fig. 8. Table II summarizes the results provided by the SM method. We remark from the tables I and II that TCM succeeds in providing more correct matches and less mismatches when compared to SM. The proposed approach has been applied to multiple frames of both virtual and real sequences and the results are satisfactory. We did not show the whole results because of the limitation of the space allowed to the publication of the paper.

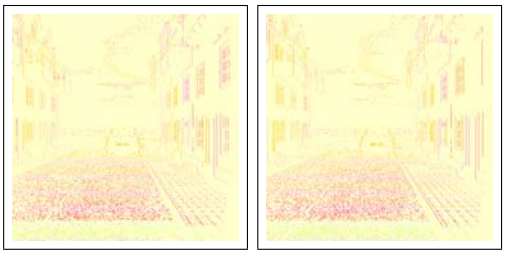

Fig. 7. Disparity maps computed for the frames shown in Fig. 5 by the TCM method.

Frame	NME	PCM	NCM	NFM
293	15685	80.87	13312	2373
294	15934	88.03	14027	1907

TABLE I

Summary of the results obtained by the TCM method when applied to the stereo images shown in Fig. 5.

B. Real images sequences

The proposed method has been tested on the real sequence depicted in Fig. 9. The image size is 384×288. The stereo sequence was acquired by stereo vision sensor embedded in a moving car. The velocity of the car is 90km per hour. The stereo vision sensor provides 10 frames per second. The

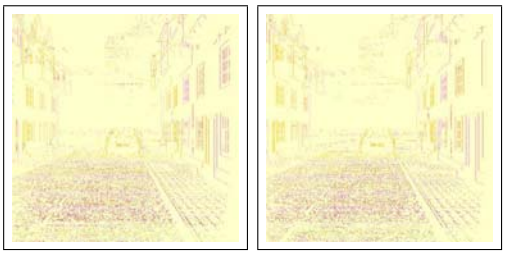

Fig. 8. Disparity maps computed by the SM method ($d_{max} = 200$) for the frames shown in Fig. 5.

Frame	NME	PCM	NCM	NFM
293	16499	72.24	11919	4580
294	16772	73.78	12375	4397

TABLE II

Summary of the results obtained with the SM method.

Fig. 9. Real stereo sequences (left images of the frames #4185 and #4186).

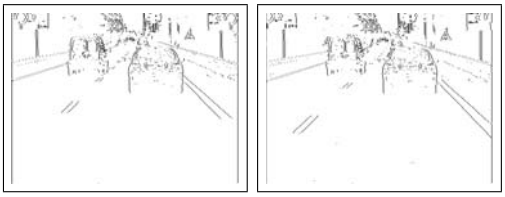

Fig. 10. Edge points of the real stereo sequence (frames #4185 and #4186).

extracted edge points are shown in Fig. 10. The disparity maps computed by the TCM and SM methods are illustrated in Figs. 11 and 12, respectively. It is clear that the disparity maps computed by the TCM method are smoother than those computed with the SM method. The SM disparity maps are more noised. With real sequence, however, there is no ground truth available to judge the results. in order to do so, let us consider the disparity maps computed by the two methods at the sub-images covering the left car (LC) and the right car (RC) appearing in the frame #4185. Fig. 13 depicts the two sub-images.

Let us start by analyzing the computed disparities in the area containing RC. Fig. 14 shows sub-disparity maps taken from the disparity maps of Fig. 12 and 11. They are enlarged before insertion in the present paper. The left and right maps depict the disparity maps estimated with the SM and the TCM methods, respectively. According to the disparity smoothness constraint, the edge points belonging to the same contour should have very close or similar disparity values. If we focus on the contour points of RC, we remark clearly that those on the left image are more noised, which means that the left image contains more false matches. However,

Fig. 11. Disparity maps computed by the TCM method.

Fig. 12. Disparity maps computed by the SM method.

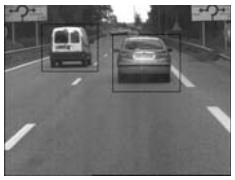

Fig. 13. Sub-images covering the left and right cars.

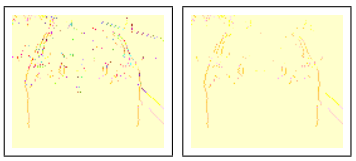

Fig. 14. Disparity maps at RC computed by (left) the SM method and (right) the TCM method.

in the right image the disparity values at the car contour points are homogeneous. Consequently, the disparity map on the right image presents a very small number of false matches when compared to the left disparity map. The same observations can be made on the top right and the top left areas of the images. We can see that the disparity values on the left image are more noised. The left disparity map of the car contains more false matches which are represented by different colors. The correct matches in the car contour points in the left image should have the same color as the car contour points in the right image. All the points with different color are considered as false matches. In the area situated between the vertical contours of the car, we see that mismatches were made in the left image, which is not the case in the right image.

After analyzing the results obtained, we can deduce that the edge points of RC should have a disparity value equal to 9 pixels. We consider the edge points with this value as correct matches. The number of correct matches with the SM and TCM methods are 74 and 89, respectively. The TCM has more correct matches, which is equal to 20% of the correct matches with SM method.

The same comparison can be made for LC appearing in the stereo images. Fig. 15 shows the sub-disparity maps for LC sub-images. The left and right maps represent the disparities estimated with the SM and TCM methods, respectively. The correct matches in the car contours should have the same color as in the vertical car contour on the right image. There are few false matches in the right image. In the left image, there are a lot of false matches lying on the vertical contours of LC. The remarks are valid when we look at the right side and the middle part of LC. There are more false matches in

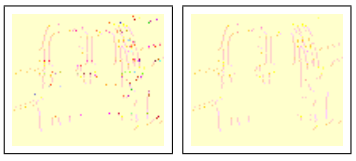

Fig. 15. Disparity maps at LC computed by (left) the SM method (right) the TCM method.

the map computed with the SM method. The improvements are clear when we analyze the results obtained for the other frames. The TCM method gives promising results.

After analyzing the disparity maps, we can deduce that the correct disparities at the edge points of LC should have a disparity value equal to 7 pixels. The number of edge points having this value are 206 with the SM method and 234 with the TCM method. We remark that the TCM method correctly matches 13% more edge points than the SM method. Table III summarizes the results obtained for both LC and RC by using TCM and SM methods.

Sub-image	RC	LC
SM	74	206
TCM	89	234
PMI	20%	13%

TABLE III

NUMBER OF CORRECT MATCHES WITH THE TWO METHODS AND THE PERCENTAGE OF MATCH IMPROVEMENT (PMI) BY TCM.

C. Running time

The hardware used for the experiments is a HP Pavilion dv6700 2.1GHZ running under Windows Vista. Table IV illustrates the time consumed by different methods per frame. The time needed in the TCM matching process is less than the SM method for all the sequences. However when we take into account the time consumed by the disparity range computation step, the TCM method needs more time than the SM for matching. This is due to the technique used to find association between consecutive frames. Despite the time used by the disparity range computation step, the TCM is still very fast and able to process about 20 millions frames per second.

Sequence	Declivity	SM	TCM	
			Disparity range	Matching
Virtual	72.21	135.72	65.01	97.10
Real	27.14	20.14	11.85	15.85

TABLE IV

RUNNING TIME CONSUMED WITH DIFFERENT ALGORITHMS IN NANOSECOND (NSEC)

VI. CONCLUSION

In this paper, we have presented a real-time stereo matching method devoted to ADAS. The temporal information integrated in the matching process allows to determine the disparity range for each scanline and consequently increases the number of the correct matches. The proposed method processes about 20 millions frames per second on a HP Pavilion dv6700 2.1GHZ running under Windows Vista. Given that the proposed approach is applied independently to each image line, the running time can be further reduced by using GPU card. The new method has been tested on virtual and real stereo image sequences and the results are satisfactory.

VII. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. Special thanks to François Gavillon for his appreciated comments.

REFERENCES

[1] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous Systems*, vol. 32, pp. 1–16, 2000.

[2] S. Barnard and M. Fisher, "Computational stereo," *ACM Comput. Surveys*, vol. 14, pp. 553–572, 1982.

[3] U. R. Dhond and J. K. Aggarwal, "Sructure from stereo - a review," *IEEE Trans. on Syst. Man and Cybern.*, vol. 19, pp. 1489–1510, 1989.

[4] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, 2003.

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

[6] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, p. 12221239, 2001.

[7] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: a unifying framework for depth from triangulation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 1–7, February 2005.

[8] L. Zhang, B. Curless, and S. M. Seitz, "Spacetime stereo: shape recovery for dynamic scenes," in *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Madison, WI, USA, 2003, pp. 367–374.

[9] M. Gong, "Enforcing temporal consistency in real-time stereo estimation," in *Proc. European Conference on Computer Vision*, Graz, Austria, May 7-13 2006, pp. III–564–577.

[10] M. El-Ansari, S. Mousset, and A. Bensrhair, "A new stereo matching approach for real-time road obstacle detection for situations with deteriorated visibility," in *Proc. IEEE Intelligent Vehicle Symposium*, Eindhoven University of Technology, Eindhoven, The Netherlands, June 4-6 2008.

[11] P. Miché and R. Debrie, "Fast and self-adaptive image segmentation using extended declivity," *Ann. Telecommun.*, vol. 50, no. 3-4, pp. 401–410, 1995.

[12] Y. Otha and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 139–154, 1989.

[13] H. Tao, H. S. Sawhney, and R. Kumar, "Dynamic depth recovery from multiple synchronized video streams," in *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawai, USA, 2001.

[14] S. Vedula, S. Baker, P. Rander, R. Collins, and T. kanade, "Three-dimensioal scene flow," in *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, September 1991, pp. II–722–729.

[15] C. Leung, B. Appleton, B. C. Lovell, and C. Sun, "An energy minimisation approach to stereo-temporal dense reconstruction," in *Proc. of IEEE Int. Conf. on Pattern Recognition*, Cambridge, UK, 2004, pp. 72–75.

[16] G. Zhang, J. Jia, T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 974–988, 2009.

[17] M. Hariti, Y. Ruichek, and A. Koukam, "A voting stereo matching method for real-time obstacle detection," in *Proc. of the 2003 IEEE Int. Conf. on Robotics & Automation*, Taipei, Taiwan, September 14-19 2003, pp. 1700–1704.

[18] B. Cyganek and J. Borgosz, "An improved variogram analysis of the maximum expected disparity in stereo images," in *J. Bigun and T. Gustavsson (Eds.), SCIA 2003, LNCS 2749*, 2003, pp. 640–645.

[19] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereo vision on non flat road geometry through v-disparity representation," in *Proc. IEEE Intelligent Vehicle Symposium*, Versailles, France, June 2002.

[20] "Stereo data for algorithms evaluation," http://stereodatasets.wvandermak.com/.