Temporal consistent real-time stereo for intelligent vehicles

Mohamed El Ansari, Stéphane Mousset, Abdelaziz Bensrhair

Temporal Consistent Real-Time Stereo for Intelligent Vehicles

Mohamed El Ansari [b,a,*] Stéphane Mousset [a]

Abdelaziz Bensrhair [a]

[a] *LITIS EA 4108, INSA de Rouen, Avenue de l'Université, BP 8*

76801 Saint-Etienne-du-Rouvray Cedex, France

[b] *LabSIV, Department of Mathematics and Computer Science, Faculty of Science,*

Ibn Zohr University, BP 8106, 80000 Agadir, Morocco

Abstract

This paper presents a real-time stereo image sequences matching approach dedicated to intelligent vehicles applications. The main idea of the paper consists in integrating temporal information into the matching scheme. The estimation of the disparity map of an actual frame exploits the disparity map estimated for its preceding frame. An association between the two frames is searched, i.e. temporal integration. The disparity range is inferred for the actual frame based on both the association and the disparity map of the preceding frame. Dynamic programming technique is considered for matching the image features. As a similarity measure, a new cost function is defined. 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.

Key words: Stereo vision, Stereo matching, Stereo image sequences, Obstacles

detection, Intelligent vehicles, Real-time, Temporal consistency.

1 Introduction

An appealing application of intelligent transportation systems (ITS) is the automatization of transport of people and goods in inner city environments. Reliable, robust and real-time obstacle detection methodologies [9,29] are needed to enable the safe operation of these types of IV among other traffic participants such as cars and pedestrians. The intelligent vehicle (IV) can achieve the obstacle detection by knowing its environment. Stereo vision has the advantage that it is able to obtain an accurate and detailed 3D representation of the environment around a vehicle, by passive sensing and at a relatively low sensor cost.

The key problem in stereo vision consists in finding correspondence between pixels of stereo images taken from different viewpoints [6]. Exhaustive surveys on the methods tackling the correspondence problem are available in [18,12,13]. An updated taxonomy of dense stereo correspondence algorithms together with a testbed for quantitative evaluation of stereo algorithms is provided by Scharstein and Szeliski [28]. It is demonstrated from [28] that graph cuts methods [11,22,32,21] produce good results. However, they are time consuming which make them not suitable for real-time applications,e.g.

* Corresponding author : Mohamed El Ansari
Email addresses: m.elansari@univ-ibnzohr.ac.ma,melansari@yahoo.com (Mohamed El Ansari), stephane.mousset@insa-rouen.fr (Stéphane Mousset), abdelaziz.bensrhair@insa-rouen.fr (Abdelaziz Bensrhair).

IV applications.

The reader should keep in mind that the stereo vision approach we are presenting in the actual work is devoted to IV applications. The IV stereo vision system is configured as follows. The IV is equipped with a stereo sensor which provides pair of stereo images at each time instant, i.e. stereo sequences. Both the IV and objects of the scene, e.g. cars and pedestrians, are moving. Therefore, the stereo approach we propose should deal with dynamic scenes.

Although there is strong support that the incorporation of temporal information can achieve better results [17,30,34,19], only a small amount of research has been devoted to the reconstruction of dynamic scenes from stereo image sequences. All the stereo approaches in the survey papers mentioned above match each frame independently. We believe that by considering the temporal consistency between successive frames the stereo matching results could be improved better [5]. Based on this principle, this paper presents a new real-time stereo matching approach dedicated to IV applications. The method provides a sparse disparity map and the so-called declivity operator [26] is used for extracting edge points from the stereo images. The declivity operator is precise, fast, and self-adaptive which justify its choice in our framework. The main idea of the proposed approach consists in exploiting the disparity map estimated at one frame for the computation of the disparity map of the next frame. A pre-estimated disparity map is computed for the last frame and used to deduce its possible disparities (disparity range) for each image line. A new cost function is defined to measure the similarity between pairs of edge points. Dynamic programming technique [27,25,8] 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 good 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.

2 Related work

In the recent years, several techniques have been proposed to obtain more accurate disparity maps from stereo sequences by utilizing temporal consistency [17,19,30,34]. Most of these methods use either optical flow or spatiotemporal window for matching stereo sequences. In their approach, Tao et al. [30] 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 [31,35] are investigated in the matching process. The accuracy of the results and the processing speed are limited by the image segmentation algorithm used. Zhang et al. [35] compute 3D scene flow and structure in an integrated manner, in which a 3D motion model is fit to each local image region and an adaptive global smoothness regularization is applied to the whole image. They later improve their results by fitting parametric motion to each local image region obtained by color segmentation, so that discontinuities are preserved [36]. Carceroni and Kutulakos [14] present a method to recover piecewise continuous geometry and parametric

reflectance under non-rigid motion with known lighting positions. Vedula et al. [31] present a linear algorithm to compute 3D scene flow based on 2D optical flow and estimate 3D structures from the scene flow. In [24], 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 [19], 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. 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. [34], 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. However, It fails to do with dynamic scenes. Davis et al. [17] have developed a similar framework as the one in [34]. 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. [33] 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 can't be applicable in 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* between successive frames. The association is defined later. Once the association is found between the actual frame and its preceding one, a pre-estimated disparity map of the actual frame can be inferred. The pre-estimated disparity map allows to determine the disparity range authorized for each image line. The disparity range is provided to the dynamic programming algorithm which performs the matching between edge points of the same image line. The authors have developed different stereo matching [3,4] methods. However, it is difficult to adapt these methods to IV applications because of the features and segmentation they use in the matching process.

3 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 satisfies the following constraints : fastness, precision, and self-adaptivity. Therefore, we consider the so-called declivity [26] 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 [26].

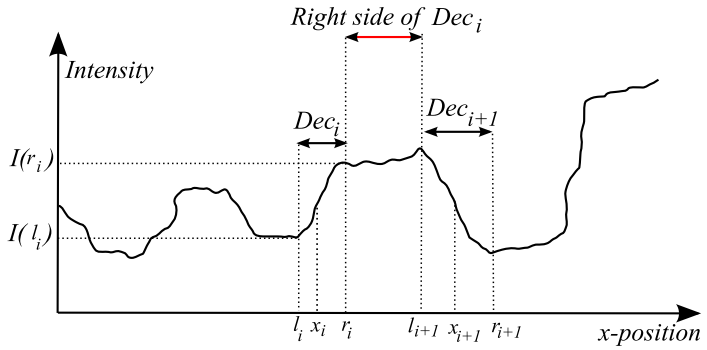

Fig. 1. Image line : characteristic parameters of a declivity.

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 as the *right side* of Dec_i.

More details about the declivity operator and how to determine the parameter σ are available in[26].

4 Stereo matching algorithm

In this section, we present the proposed method for matching pairs of stereo images provided by stereo sensor mounted aboard a car. We start by mentioning the constraints the pairs of corresponding declivities should meet. As disparity constraint, we propose a technique which defines the possible disparities for each scanline independently. A new cost function is defined to measure the similarity between candidates pairs of declivities. The last subsection describes the dynamic programming algorithm used for the matching process. We note that the stereoscopic sensor used in our experiments provides rectified images, i.e., the corresponding pixels have the same *y-coordinate*.

4.1 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 [20]. 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.

4.2 Disparity range

The accurate choice of the maximum disparity threshold value for almost any known stereo processing method [28] is crucial to the quality of the output disparity map and the computation time [16]. In this subsection, we propose a new approach which is able to find the minimum and maximum disparity value for each image scanline based on the image content, i.e. objects appeared in the stereo images. The main idea consists in exploiting the disparity map computed for the preceding frame to compute the disparity range of the actual frame. To achieve such a task we need to find a relationship between successive frames. We refer to that relationship as *association* between declivities of successive frames. For each declivity in the actual frame, we search its associate one in the preceding frame, if any. The next step computes a *pre-estimated* disparity map for the actual frame. The disparity range is derived from the pre-estimated disparity map. The rest of the subsection details the three main steps followed to compute the disparity range.

4.2.1 The association

The aim of this subsection is to describe the method used to find association between declivities of successive frames. Let I_{k-1} and I_k be two successive im-

ages 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 successive 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.

4.2.2 The pre-estimated disparity map

We define the so-called *pre-estimated disparity map* of a pair of stereo images as the disparity map deduced from the disparity map of its preceding pair of stereo images. The goal of this subsection is to demonstrate how to compute the pre-estimated disparity map at an actual frame from its preceding one.

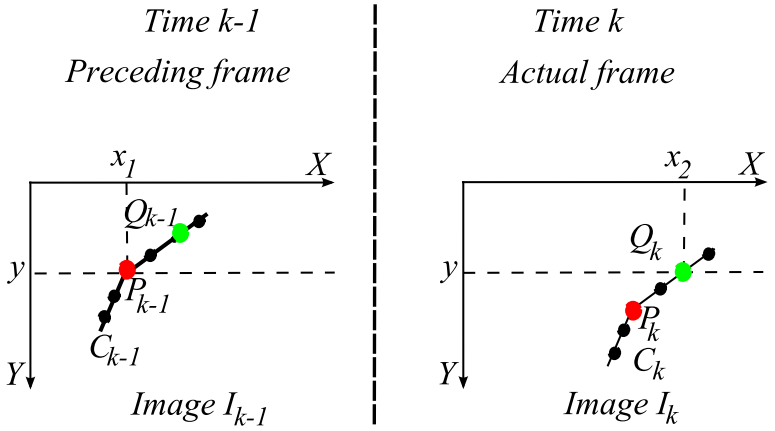

Fig. 2. I_{k-1} and I_k represent successive images of the same sequence, e.g. left sequence. The point Q_k in the image I_k constitutes the associate point of the point P_{k-1} in the image I_{k-1}. The points P_k and P_{k-1} are in red color. The points Q_k and Q_{k-1} are in green color. We mean declivity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

Let I_{k-1}^L and I_{k-1}^R 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_k^L and I_k^R are the left and right stereo images of the frame f_k acquired at time k. The declivities are extracted by the method presented in [26]. For each declivity in the image I_k^L we look for its associate one in the image I_{k-1}^L, if any, by following the approach detailed in section 4.2.1. The same process is performed for the declivities of the images I_{k-1}^R and I_k^R. The subject now consists in computing the pre-estimated disparity map of the frame f_k based on the knowledge of the association between their declivities and those of the preceding frame f_{k-1}, and the disparity map d_{k-1}. The method we propose for such a task is 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. Fig. 3 illustrates the different steps of the algorithm. We have two frames f_{k-1} and f_k. P, Q, R, and S are declivities belonging to the l-scanlines of the four images . The first step consists in finding the associate of P, which we name Q (Fig. 3). In the second step, we get the match S of Q based on disparity value computed for the frame f_{k-1}. The third step looks for the associate R of S. The last step deduces that R is the match of P. Consequently, we can compute the disparity at the point P in the image I_k^L or at the point Q in the image I_k^R. The same technique will be done for all image scanlines ($l = 1, .., image\ height$).

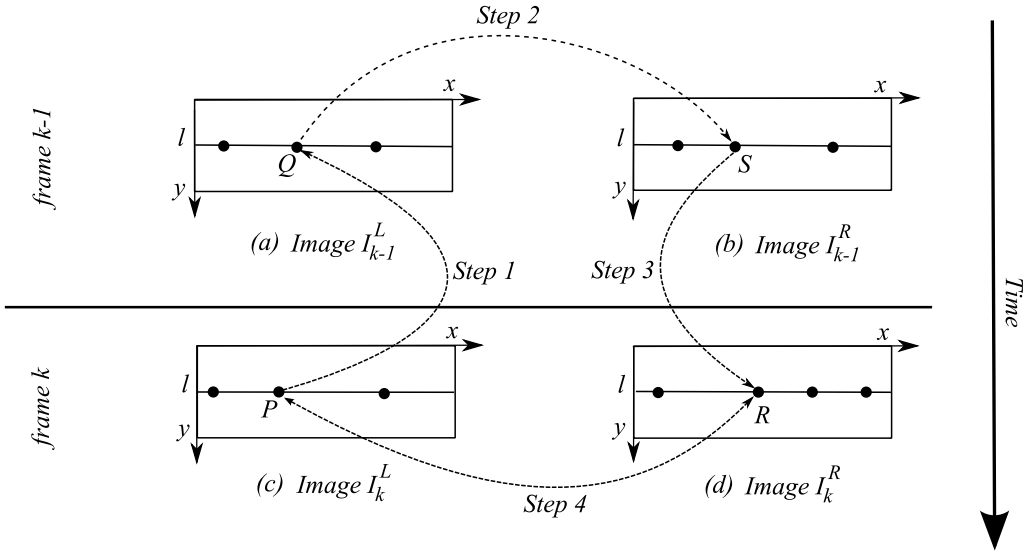

Fig. 3. Stereo images of two successive frames. (a) Left image of the frame f_{k-1}, (b) right image of the frame f_{k-1}, (c) left image of the frame f_k, (d) right image of the frame f_k. Q represents the associate of P. S represents the match of Q. R constitutes the associate of S. R is the match of P.

4.2.3 Disparity range

We suppose that the pre-estimated disparity map pd_k of the frame k has been computed as described in section 4.2.2. The subsequent of this subsection details how to compute the disparity range of the frame f_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 [23]. 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 [23]. The processing of the v-disparity image provides geometric content of road scenes. It was demonstrated in [23] that the obstacles and the road appeared as vertical and oblique lines, respectively. Assume that we have a road scene containing four objects. The corresponding v-disparity image should be as shown in Fig. 4. 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 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 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. 4).

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. 4, 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.

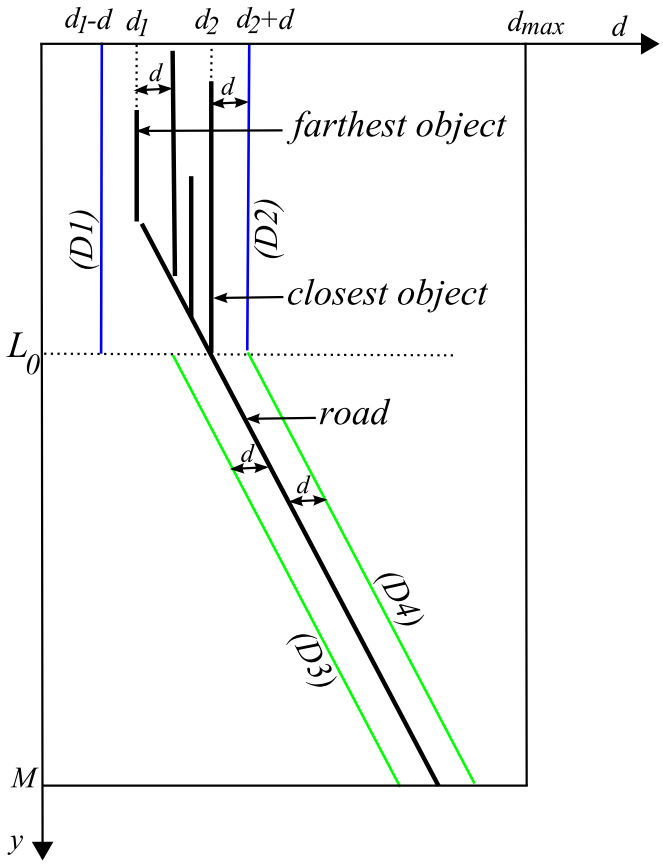

Fig. 4. v-disparity of the pre-estimated disparity map. The vertical axis refers to the image lines and the horizontal axis represents the disparities. M is the image height. d_{max} is the maximum disparity value. The possible disparities are the area between the lines $(D1)$ and $(D2)$ for the top part and the area between the lines $(D3)$ and $(D4)$ for the bottom part.

4.3 Cost function

As a similarity criterion between corresponding declivities, we propose a new cost function which we define 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 propose to 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. We define the cost function as follows.

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

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

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

4.3.1 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 sacanlines, 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 [27]. Fig. 5 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. 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.

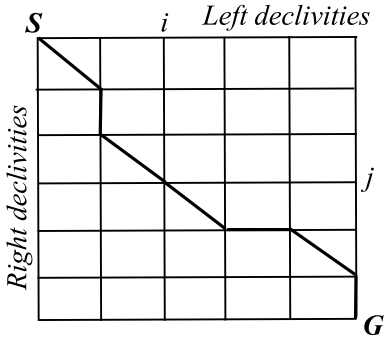

Fig. 5. 2D search plane. The horizontal axis corresponds to the left scanline and the vertical one corresponds to the right scanline. vertical and horizontal lines are the declivity positions and path selection is done at their intersections.

Based on the subsections 4.1 and 4.2 the possible matched pairs of declivities on the left and right scanlines are searched. The pairs which do not meet the above constraints will be discarded and their nodes on the search plane will be noted as invalid nodes. 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. The matching process is achieved independently for each scanline.

5 Experimental results

In order to evaluate the performance of the proposed approach, it has been applied to virtual and real stereo sequences. We propose to call the new method

as temporal consistent matching (TCM) method in the sequel of this section. We call space matching (SM) method, the TCM method deprived of the disparity range computation step. The SM method uses the dynamic programming with a predetermined disparity maximum value for all the image lines. To assess the performance of the TCM method, particularly the disparity range computation step, the SM method has been applied to data used in our experimentation.

5.1 Virtual stereo image sequences

We have tested our method on the MARS/PRESCAN virtual stereo images available in [2]. The archives contain the original left and right stereo sequences with and without added distortions and noise, and the ground truth. The size of the images is 512×512. Before using them they are converted into grey level as our approach deals with grey level images. At first, we have applied the new method (TCM) to the original virtual sequences. Fig. 6 illustrates the left stereo images of the frames #293 to #295 of the same sequence.

Fig. 6. Virtual stereo sequences (left images of the frames #293 to #295).

The extracted edge points are depicted in Fig. 7. The disparity maps computed by the TCM method are shown in Fig. 8. We have used false colors for representing the disparity maps. For the initialization of the proposed approach,

Fig. 7. edge points of the images shown in Fig. 6.

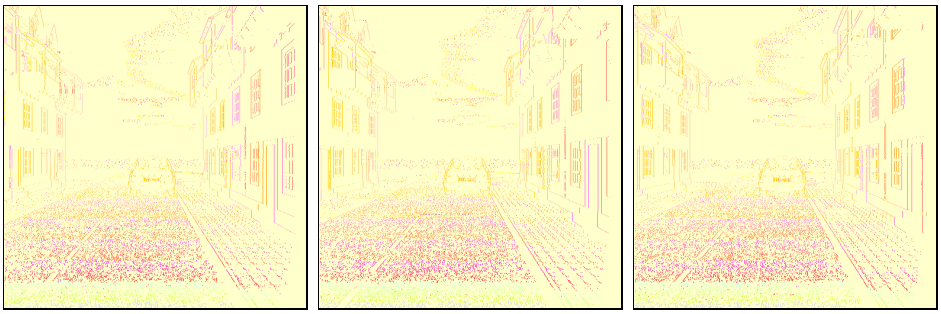

Fig. 8. Disparity maps computed for the frames shown in Fig. 6 by the TCM method.

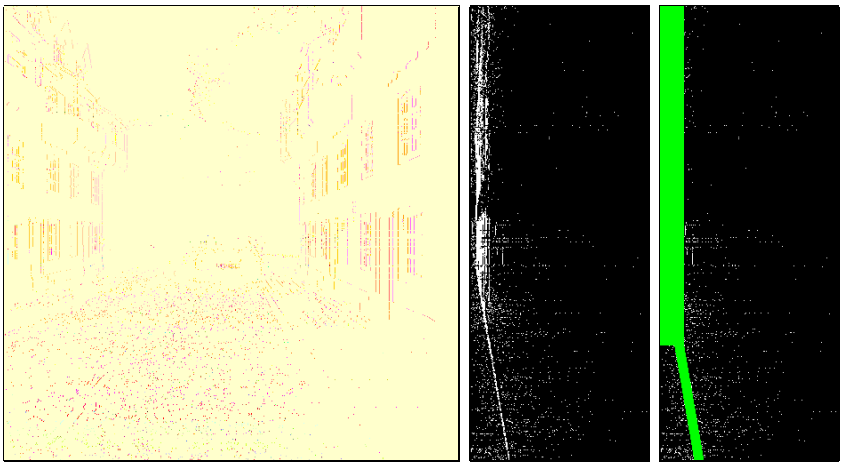

Fig. 9. Computation of the disparity range for the frame #293. (left) The computed pre-estimated disparity. (middle) V-disparity of the pre-estimated disparity in (left). (right) The disparity range computed by the proposed method. It is shown in green color.

the SM method is used for the first frame and then the TCM method is used for the following frames. The maximum disparity value is set to $d_{max} = 200$. The results related to the disparity range computation are shown in Fig. 9. In

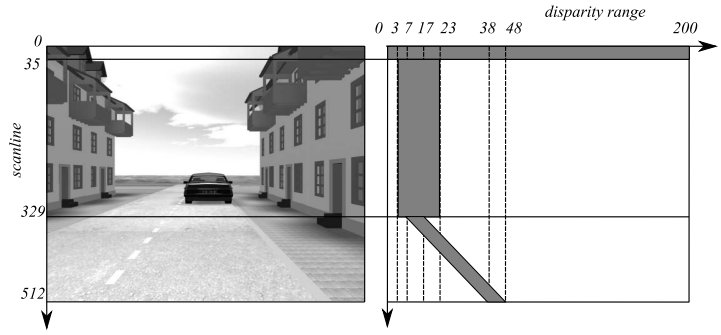

Fig. 10. Disparity range computed for the frame #293. (left) left stereo image of the frame #293. (right) the disparity range found (the area in black color).

the left of the figure, we have the pre-estimated disparity map for the frame #293. It is deduced from the disparity computed for the frame #292 and the association between the frames #292 and #293. The image in the middle of Fig. 9 represents the v-disparity of the pre-estimated disparity map. The disparity range is illustrated in green color on the right of Fig. 9. For the computation of the disparity range for an actual frame, we need to extract straight lines from the v-disparity image of its pre-estimated disparity map. The Open Computer Vision Library (OpenCV) [1] was used for such a task. Numerically speaking, The disparity range computed for the frame #293 is as follows.

$$\begin{aligned} 0 \leq d(l) \leq d_{max} \qquad & if \qquad 1 \leq l \leq 35 \\ 3 \leq d(l) \leq 23 \qquad & if \qquad 36 \leq l \leq 329 \\ 0.17l - 43.57 \leq d(l) \leq 0.17l - 38.57 \qquad & if \qquad 330 \leq l \leq 512 \end{aligned} \tag{5}$$

where l denotes the scanline index. The same disparity range is depicted in Fig. 10.

Between the scanlines 1 and 35 no lines are detected. Therefore, d_{max} was

Frame	NME	PCM	NCM	NFM
293	15685	80.87	13312	2373
294	15934	88.03	14027	1907
295	15740	85.81	13507	2233

Table 1

Summary of the results obtained by the TCM method when applied to the stereo images shown in Fig. 6.

kept. From scanline 36 to scanline 329 some lines was detected. The lines corresponding to farthest and closest objects are lying on the disparities 8 and 18, respectively. We have used 5 as the tolerance value for our algorithm. The disparity range becomes $[3, 23]$. The disparities and the interval are mentioned in pixel. From the line 330 to the line 512 an oblique line was detected, which has the equation $d = 0.17l - 43.57$, where d is the disparity and l is the scanline index. The possible disparities for each l-scanline are the interval $[0.17l - 43.57 - 5, 0.17l - 43.57 + 5]$. Table 1 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 to #295.

The SM method has been applied to the sequence shown in Fig. 6 in order to assess the performance of the TCM method. The estimated disparity maps obtained by the SM method are depicted in Fig. 11. Table 2 summarizes the results provided by the SM method. Comparison results are illustrated in Table 3 from which we remark clearly the improvements due to the TCM approach. The TCM method has matched correctly more pairs of edge points and mismatches less pairs of edge points than the SM method for the different

Fig. 11. Disparity maps computed for the frames shown in Fig. 6 by the SM method ($d_{max} = 200$).

Frame	NME	PCM	NCM	NFM
293	16499	72.24	11919	4580
294	16772	73.78	12375	4397
295	16503	72.77	12010	4493

Table 2

Summary of the results obtained with the SM method.

frames of the sequence. As an example, let's take the frame #293. The number of pairs of edge points matched correctly by the methods SM and TCM are 11919 and 13312, respectively. The TCM approach matches 1393 pairs more correctly, which correspond to 12% of the correct matches of the SM method. The number of mismatches by the methods SM and TCM are 4580 and 2373, respectively. More mismatches has been made by the SM methd. The same remarks are true for all the frames of the virtual stereo sequences, which prove the success of the TCM method.

The TCM and SM methods have been applied also to the virtual sequences with added distortions and noise. Table 4 summarizes the results. The performances of the TCM against the SM method are very obvious. It gives more

Frame	NME		PCM		NCM		NFM	
	SM	TCM	SM	TCM	SM	TCM	SM	TCM
293	16499	15685	72.24	84.87	11919	13312	4580	2373
294	16772	15934	73.78	88.03	12375	14027	4397	1907
295	16503	15740	72.77	85.81	12010	13507	4493	2233

Table 3

Matching results when the SM and TCM methods applied to the virtual stereo sequences.

Frame	NME		PCM		NCM		NFM	
	SM	TCM	SM	TCM	SM	TCM	SM	TCM
293	14362	13432	67.54	80.98	9699	10877	4662	2555
294	14163	13372	66.15	77.93	9369	10421	4794	2951
295	13974	13120	67.01	78.64	9364	10317	4610	2803

Table 4

Matching results when the SM and TCM methods applied to the virtual stereo sequences with added distortions and noise.

correct matches and less false matches which demonstrates its effectiveness.

5.2 Real images sequences

The proposed method has been tested on the real sequence #1 depicted in Fig. 12. The image size is 384×288. The stereo sequence was acquired by stereo vision sensor embedded in a car. The velocity of the car is 90km per hour.

Fig. 12. Real stereo sequences #1 (left images of the frames #4185 to #4187).

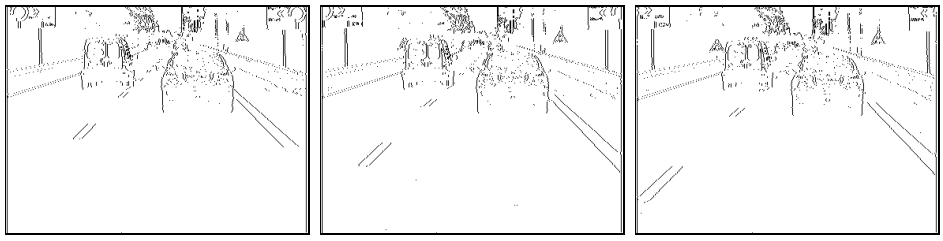

Fig. 13. Edge points of the real stereo sequence #1 (frames #4185 to #4187).

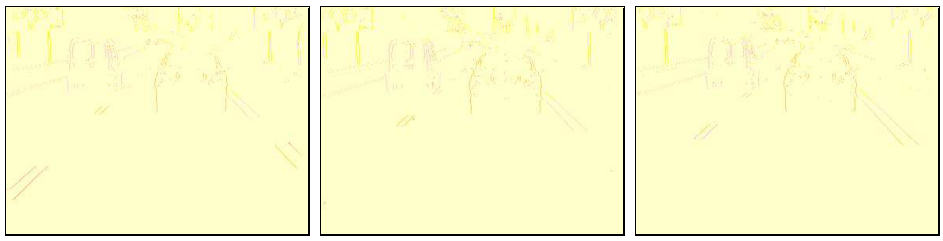

Fig. 14. Disparity maps computed by the TCM method.

The stereo vision sensor provides 10 frames per second. The extracted edge points are shown in Fig. 13. The disparity maps computed by the TCM and SM methods are illustrated in Figs. 14 and 15, respectively. It is clear that the disparity maps computed by the TCM method are more smooth than those computed with the SM method. The SM disparity maps are more noised. With real sequence, there is no ground truth available like for virtual ones to judge the results. To achieve such a task, let 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. 16 depicts the two sub-images.

Let start by analyzing the computed disparities at the area containing RC. Fig. 17 shows sub-disparity maps taken from the disparity maps of Fig. 15

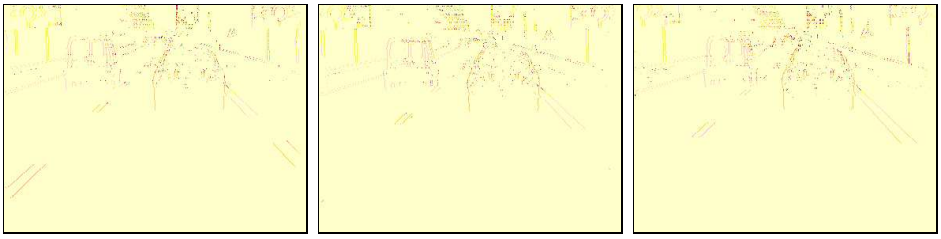

Fig. 15. Disparity maps computed by the SM method.

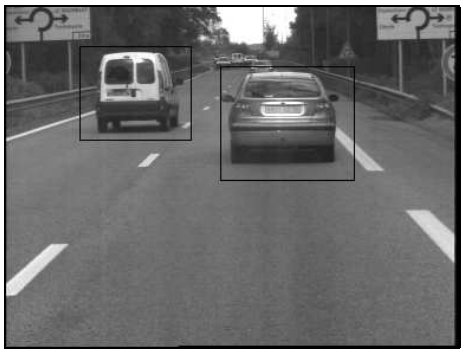

Fig. 16. Sub-images covering the left and right cars.

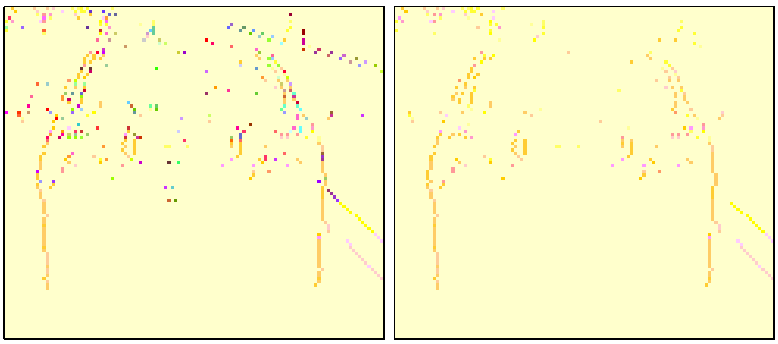

Fig. 17. Disparity maps at RC computed by (right) the SM method and (left) the TCM method.

and 14. They are enlarged before insertion in the manuscript. The left and right maps depict the disparity maps estimated with the SM and the TCM methods, respectively. Inspired of the smoothness constraint of the disparity, 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. That means that the left image contains more false matches. However, in the right image the disparity values

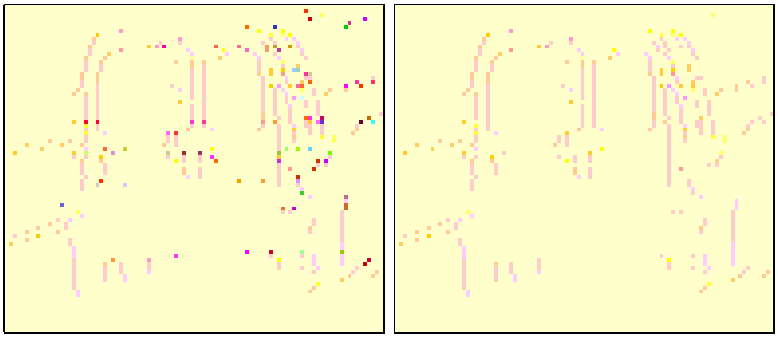

Fig. 18. Disparity maps at LC computed by (right) the SM method (left) the TCM method.

at the car contour points are homogeneous. Consequently, the disparity map on the right image presents very small number of false matches for comparison to the left disparity map. The same 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 was made in the left image, which is not the case in the right image.

After analyzing the results obtained, we 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 done for LC appeared on the stereo images. Fig. 18 shows the sub-disparity maps for LC sub-images. The left and right

maps represent the disparities estimated with 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 is little 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 see on the right side and the middle part of LC. There is more false matches in 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 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 matches correctly 13% more edge points than the SM method.

The proposed stereo matching approach has been applied also to the real stereo sequences #2 depicted in Fig. 19. Instead of showing the results corresponding to the frames of the Fig. 19 with small size, we illustrate only the results obtained for the frame #1983 with large size. This makes it easy to comment the disparity map obtained. The comments made for the frame #1983 will be true for the other frames. The declivity of the frame #1983 is depicted in Fig. 20. The computed disparity maps by both the TCM and SM methods are shown in Figs. 21 and 22, respectively. We remark clearly the difference between the two disparity maps. The one obtained by the TCM is more smooth than the other obtained by the SM method.

To have more comparison results we concentrate our comments on the areas in the images where the cars are situated. The first area (A1) is the sub-

Fig. 19. Left real stereo sequence #2 (frames #1980, #1983, and #1989).

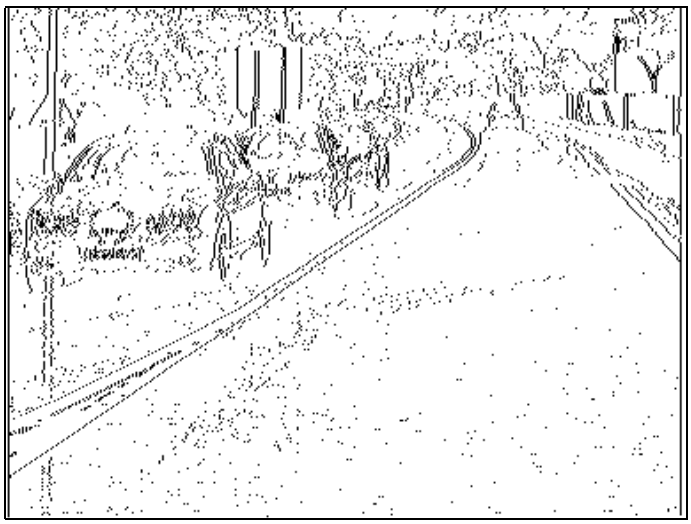

Fig. 20. Declivity of the left image of the frame #1983 of the real sequence #2.

Fig. 21. Disparity map found by the TCM method for the frame #1983.

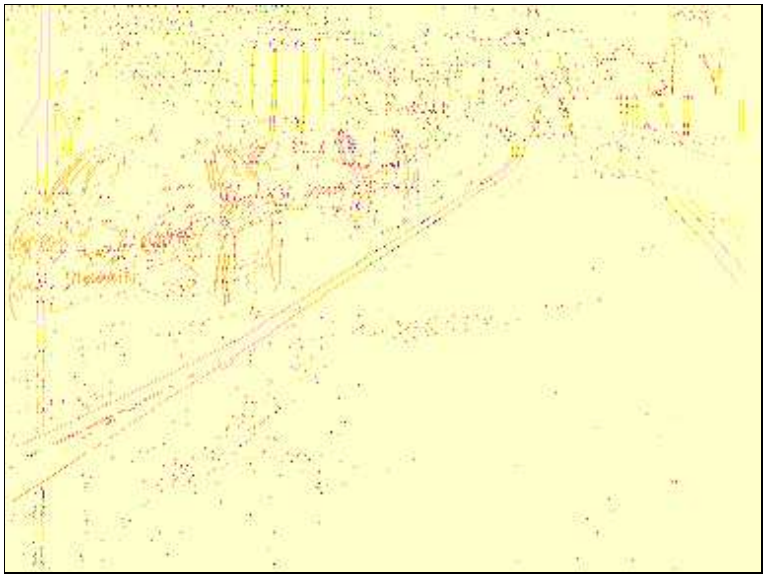

Fig. 22. Disparity map found by the SM method for the frame #1983.

Fig. 23. The two sub-images used for commenting the results obtained by the methods SM and TCM.

image containing the three cars and the second area (A2) is the sub-image containing the small car (see Fig. 23). The disparity maps computed for A1 (resp. A2) by the methods SM and TCM are depicted in Fig. 24 (resp. Fig 25). In both A1 and A2, the disparity maps estimated by the TCM method are more smooth than those of estimatd by the SM method, i.e. the TCM method

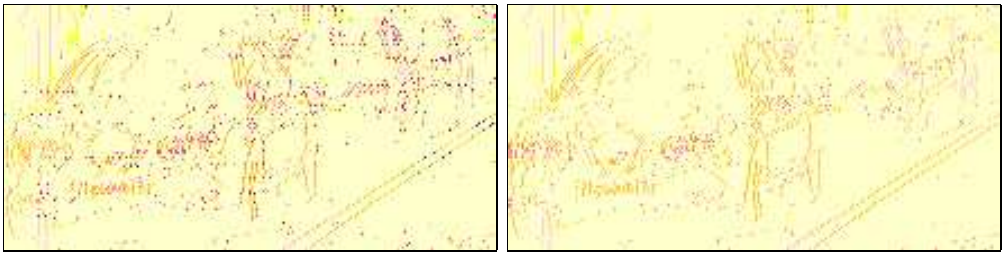

Fig. 24. Disparity map at the edge points of A1 computed by (left) the SM method and (right) the TCM method.

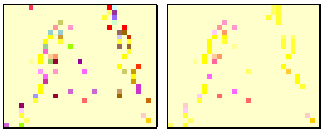

Fig. 25. Disparity map at the edge points of the A2 computed by (left) the SM method and (right) the TCM method.

Sequence	Real Sequence #1		Real Sequence #2	
Sub-image	RC	LC	A1	A2
SM	74	206	1062	16
TCM	89	234	1273	28
PMM	20%	13%	20%	75%

Table 5

Number of correct matches with the two methods and the percentage of the more matched (PMM) by TCM.

gives less mismatches. We are persuaded that the disparities in A1 should be approximately between 7 and 10. The edge points in A1 having their disparities in the interval [7,10] are considered as correct matches. The number of correct matched obtained with SM and TCM are 1062 and 1273, respectively. The same for the A2 the correct matches should have a disparity value equal to 4. The number of correct matched obtained with SM and TCM are 16 and 28,

respectively. We remark that the TCM method gives more correct matches than the SM method. The TCM method matches correctly 20% more edge points at A1. Although, the small car in A2 is very far, the TCM matches 75% more edge points, which is very interesting in the road applications for which the proposed approach is devoted. The same performance obtained for all the frames of the sequence. This shows clearly the performance of the proposed method. Table 5 summarizes the results obtained on different selected areas in the two real sequences when the TCM and SM method have been applied. It provides the number of correct matches with the two methods and the percentage of the more matched pairs correctly by the TCM.

5.3 Running time

The hardware used for the experiments is a HP Pavilion dv6700 2.1GHZ running under Windows Vista. Table 6 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 int 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 successive frames. Although 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
Virtual with noise	69.43	102.53	52.12	77.07
Real #1	27.14	20.14	11.85	15.85
Real #2	27.71	26.85	14.57	20.84

Table 6

Running time consumed with different algorithms in nanosecond (nsec)

6 Conclusion

A new real-time stereo matching method has been proposed to match the stereo image sequences of dynamic scenes. The method is dedicated to IV applications. Believing its advantages, the temporal information has been integrated in the matching process. The proposed method is very fast and can process about 20 millions frames per second on a HP Pavilion dv6700 2.1GHZ running under Windows Vista. The running time can be reduced more by using GPU card for our implementations as we have used dynamic programming technique which can be performed independently for each image line. The new method has been tested on virtual and real stereo image sequences and the results are satisfactory.

The method, we use for finding association between edge points of successive frames, is based on the gradient information. In the future work, we plan to improve this step of the proposed method. The future association technique should provides more pairs of associate points which gives a more dense pre-

estimated disparity maps. The eventual association technique should need less running time. On the other hand, the dynamic programming technique used in our stereo approach ignores the inter-scanline consistency [27,15,10,7]. This is another point to investigate in the future.

References

[1] OpenCV. http://www.sourceforge.net/projects/opencvlibrary/.

[2] Stereo data for algorithms evaluation. http://stereodatasets.wvandermak.com/.

[3] M. El Ansari, L. Masmoudi, and A. Bensrhair. A new regions matching for color stereo images. *Pattern Recognition Letters*, 28(13):1679–1687, 2007.

[4] M. El Ansari, L. Masmoudi, and L. Radoune. A new region matching method for stereoscopic images. *Pattern Recognition Letters*, 21:283–294, 2000.

[5] 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.

[6] S. Barnard and M. Fisher. Computational stereo. *ACM Comput. Surveys*, 14:553–572, 1982.

[7] P. N. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237260, 1996.

[8] A. Bensrhair, P. Miché, and R. Debrie. Fast and automatic stereo vision matching algorithm based on dynamic programming method. *Pattern Recognition Letters*, pages 457–466, 1996.

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

[10] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(4):401406, 1998.

[11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):12221239, 2001.

[12] L. G. Brown. A survey of image registration techniques. *Computing Surveys*, 24(4):325–376, 1992.

[13] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[14] R. L. Carceroni and K. N. Kutulakos. Scene capture by surfel sampling: from multiview streams to non-rigid 3d motion, shape and reflectance. In *Proc. Int. Conf. on Computer Vision*, July 2001.

[15] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *CVIU*, 63(3):542567, 1996.

[16] 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*, pages 640–645, 2003.

[17] 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*, 27(2):1–7, February 2005.

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

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

[20] 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*, pages 1700–1704, Taipei, Taiwan, September 14-19 2003.

[21] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 74–81, Washington, DC, USA, 2004.

[22] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *In Int. Conf. on Computer Vision*, volume II, page 508515, 2001.

[23] 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.

[24] 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*, pages 72–75, Cambridge, UK, 2004.

[25] Z. N. Li. Stereo correspondence based on line matching in hough space using dynamic programming. *IEEE Trans. on Systems, Man, and Cybernetics*, 24(1):144–152, 1994.

[26] P. Miché and R. Debrie. Fast and self-adaptive image segmentation using extended declivity. *Ann. Telecommun.*, 50(3-4):401–410, 1995.

[27] Y. Otha and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1989.

[28] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo sorrespondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[29] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006.

[30] 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.

[31] 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*, pages II–722–729, September 1991.

[32] O. Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999.

[33] G. Zhang, J. Jia, T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(6):974–988, 2009.

[34] 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*, pages 367–374, Madison, WI, USA, 2003.

[35] Y. Zhang and C. Kambhamettu. Integrated 3d scene flow and structure recovery from multiview image sequences. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages II–674–681, South Carolina, June 2000.

[36] Y. Zhang, C. Kambhamettu, and R. Kambhamettu. On 3d scene flow and structure estimation. In *In IEEE Conf. on Computer Vision and Pattern Recognition*, pages 778–785, 2001.