Dense spatio-temporal stereo matching for intelligent driving systems

Zakaria Kerkaou1 | Mohamed El Ansari1 | Lhoussaine Masmoudi2 | Redouan Lahmyed1

1 LabSIV, Department of Computer Science, Faculty of Science, Ibn Zohr University, Morocco
2 Department of Physics, Faculty of sciences, Mohammed V University of Rabat, Morocco

Correspondence
Zakaria Kerkaou, LabSIV, Department of Computer Science, Faculty of Science, Ibn Zohr University, BP 8106, 80000 Agadir, Morocco.
Email: kerkaou.zakaria@gmail.com

Funding information
Minister of Higher Education and Scientific Research of Morocco (MESRSFC)

Abstract
This paper addresses the problem of matching stereo images acquired by a stereo system mounted aboard an intelligent vehicle. The main idea behind the new method consists in involving temporal matching between a current stereo pair and its preceding one to achieve the spatial matching of the former stereo by involving the matching results obtained at the last frame. The proposed method is achieved in three main steps. First, an edge based disparity map is derived from the disparity map of the preceding frame, we call it the assisting disparity map (ADM). Second, for each scan-line, a set of local ranges and global ranges are deduced from the ADM to keep only potential matching candidates. Third, the matching is done on the basis of dynamic programming algorithm by involving in the resulting local and global ranges we get from the last step. The proposed approach has been tested on both real and synthetic stereo sequences and the results demonstrate its effectiveness.

1 INTRODUCTION

Stereo vision is a measurement method of finding correspondence between two or more input images in order to obtain a detailed 3D representation of a scene. It has shown an interesting increase in research efforts due to numerous applications such as 3D navigation, obstacle detection and 3D reconstruction. One of the main applications of the stereo vision methods is advanced driver assistance systems (ADAS)[1–3], which aims at improving driving comfort and safety of people and goods in the road. Some works propose to solve the problem of matching stereo images using local algorithms based on block matching or pixel correlation due to their simplicity [4–6]. Other works suggest to use global methods such as graph-cut (GC) [7, 8], belief propagation [9–11] or dynamic programming (DP) [2, 12, 13]. These approaches have become very popular due to their accurate and fast results.

This work is tackling the problem of stereo matching methods in ADAS. The stereo sensor providing stereo images is fixed on an intelligent vehicle. The proposed stereo matching approach should deal with dynamic scenes since the intelligent vehicle, and the other vehicles and pedestrians around it are moving. We believe that integrating the temporal information in a stereo matching approach can improve the results by considering the consistency between temporally adjacent frames [2, 13–15]. Therefore, we present in this paper a dense stereo matching method based on spatial and temporal information. The main novelties of the proposed method are: The computation of the assisting disparity map from which global and local ranges are deduced. The usage of a novel cost function that takes advantage of different image components such as gray intensity, gradient magnitude, gradient orientation and the census transform, and the utilization of a two way dynamic programming algorithm. To initialise the system, the first disparity map is generated without using the temporal information (i.e. by applying dynamic programming directly on the images). In the next frames, we start by detecting edge curves in the stereo pair, the spatial matching between the preceding stereo pair is deduced from the preceding disparity map, and the temporal matching between temporally adjacent image is obtained using optical flow [16, 17]. From this correspondence we can generate an edge based disparity map called assisting disparity map (ADM) from which global and local ranges of disparity are integrated. These ranges are used as control points to drive the dynamic programming in the search process for matching the current stereo pair in the aim of obtaining a dense disparity map.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. IET Image Processing published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology
The paper is organised as follow: Section 2 pins point the state of the art stereo matching algorithms using the temporal information. Section 3 details the proposed matching algorithm. Section 4 presents the experimental results and comparisons. Finally, Section 5 concludes the paper.

2 RELATED WORKS

Stereo matching methods can be categorised into two main categories: Local and global methods. Local methods match every pixel in the reference image independently, typically using pixel-based matching cost such as block matching or pixel correlation [4–6, 18] almost all these methods use the strategy of winner take all in which the disparity of a pixel is chosen based on the lowest aggregate cost. The matching cost is aggregated using an aggregation function that basically assumes that the disparity in homogeneous areas have little difference and share the same disparity. Aggregation functions can be classified into window based method [19], filter based method [20, 21], or segment-tree-based method [22, 23]. On the other hand, global methods try to minimise an energy function either on a single scan-line or over the entire image. Algorithms such as graph-cut (GC) [7, 8], dynamic programming (DP) [2, 12, 13], semi global matching (SGM) [24] and belief propagation (BP) [9–11] have become very popular due to their accurate results. [25] presents a taxonomy and evaluation of these global methods. It was demonstrated that GC and BP approaches give the best results at the expense of computational time, while DP and SGM give good results but are way faster due to their one-dimensional optimisation solution. As mentioned in [25], the disparity range is a major key to the quality of the disparity map for all stereo matching approaches. incorrect disparity ranges can affect the performance of the stereo method in both the complexity and the accuracy. Generally, the developed methods assume that the appropriate disparity search range is already established [26].

The subject of using temporal consistency [27–29] in order to obtain more accurate disparity maps has attracted a lot of attention in recent years. Some of the works depend on optical-flow, others choose to use spatio-temporal window to match stereo sequences.

In [30], Zhang et al. extended the spatial window of computing cost to a spatio-temporal window. The spatial window is used to compute the sum of squared difference (SSD) while the spatio-temporal window is used to compute the sum of SSDs. This method showed promising results in static scenes, but did not perform well in dynamic scenes. In [31], the authors presented a novel algorithm that takes a video sequence as input and compute depth map for each frame. Davis et al. [32] have developed a method called space-time stereo similar to the one in [30]. However, the method was designed to perform in geometrically static scenes.

In [33], the authors presented a method of estimating disparity ranges based on a disparity histogram generated with sparse feature matching algorithms SURF. Cai [34] proposed an approach to integrate the optical flow estimation into a block stereo matching algorithm using dynamic programming. In [14], Jiang et al. used features detection, edge motion estimation and motion detection as a method to predict the disparity map in consecutive frames. Dobias et al. [35] proposed a method of predicting the disparity map of the current frame based on the one computed on the preceding frame using estimated motion of the calibrated stereo rig. Vedula et al. [36] computed three dimensional scene flow from two dimensional optical flow using a linear algorithm, from which a three dimensional structure information is deduced.

In [2], the authors presented a method to match edge points of stereo images, in which the temporal information is integrated by finding the relationship between consecutive frames using an algorithm called association. A disparity range is deduced based on both the association and the disparity of the previous frame to compute the current one. [37] expanded on the same idea by computing the initial disparity map by matching edge curves, which is used to obtain disparity ranges together with matching control edge points which are used to guide dynamic programming algorithm in matching the current frame. In [29], the authors presented a fast edge-based stereo matching approach devoted to road applications in which two passes of the dynamic programming algorithm are applied to estimate the final disparity map. Some works proposed to solve the stereo matching problem using convolutional neural networks. Yang et al. [38] suggested a unified model that employs semantic features from segmentation and introduces semantic softmax loss, which helps improve the prediction accuracy of disparity maps. In [39], the authors proposed to encapsulates all convolutional features into unary features descriptor using multi-level context ultra aggregation.

3 PROPOSED STERO MATCHING ALGORITHM

This section details the steps of the proposed stereo matching algorithm, which is depicted in Figure 1. The main contribution of the proposed method consists involving temporal information in the matching process. The matching results of the preceding frame are exploited in the matching of the current one. For clarity purposes, we use the following notations for the rest of the paper:

1. \( I_L^k \) and \( I_R^k \) indicate the left and right stereo images acquired at time \( k \) and \( f_k = (I_L^k, I_R^k) \) is the current stereo pair.
2. \( d_k \) (resp. \( d_{k-1} \)) indicates the disparity map acquired at time \( k \) (resp. \( k - 1 \)).

The computation of the disparity map \( d_k \) illustrated in Figure 1 can be described as follows: First, the so-called ADM is calculated based on the disparity map computed at the preceding frame. The ADM allows to generate possible disparities in each scan-line of the current frame \( f_k \). Second, a dynamic programming based algorithm is applied on the stereo images of the current frame \( f_k \) to derive the disparity map at the current frame.
3.1 The assisting disparity map computation

This section clarifies the steps we follow to obtain the Assisting Disparity Map (ADM). It is an edge based disparity map deduced from the preceding disparity map $d_{k-1}$. The ADM is used to determine the disparity ranges at each image scanline, which guide the dynamic programming search through the disparities obtained at the edge points. The quality of the ADM is an essential element in the process of getting an accurate disparity map.

### 3.1.1 Edge point detection

The first step of computing the ADM is detecting edge curves in the stereo images where the depth discontinuities are more likely to be located. In this work, we use Canny Edge Detector [40] that provides continuous edge curves, and produces more significant edge points, which are crucial to the proposed matching method. Let’s denote by $S_f^m = \{C_f^m\}_{i=1,N}$ the set of edge curves of the image $I_f^m \in \{L, R\}$ such that $f \in \{k-1, k\}$ is the frame index and $m \in \{L, R\}$ is the index of the stereo image, $L$ for the left image and $R$ for right image.

### 3.1.2 Spatio-temporal matching of edge curves

The procedure of deducing the ADM is based on correspondences between edge curves in consecutive frames $f_{k-1}$ and $f_k$ and the disparity map $d_{k-1}$ of the frame $f_{k-1}$. This is achieved through four main steps as shown in Figure 2:

1. The first step is matching edge curves $S_{k-1}^L$ in the image $I_{k-1}^L$ with edge curves $S_{k}^L$ in the image $I_k^L$ by tracking each edge curve $C_{k-1}^{L, i}$ iteratively using Lucas-Kanade optical flow [16, 17].
2. The second step is matching edge curves $S_{k-1}^L$ in the image $I_{k-1}^L$ with edge curves $S_{k}^R$ in the image $I_k^R$ using the disparity map $D_{k-1}$.
3. The third step is matching edge curves $S_{k}^R$ in the image $I_k^R$ with edge curves $S_{k-1}^R$ in the image $I_{k-1}^R$ using the same principle as the first step.
4. The last step deduces the correspondences between edge curves $S_{k}^L$ and $S_{k}^R$. To locate the spatial match of $C_{k}^{L, i}$, we find the match of $C_{k-1}^{L, i}$ in $I_{k-1}^L$, we call it $C_{k-1}^{L, j}$. Second, the edge curve $C_{k-1}^{R, m}$ in the image $I_{k-1}^R$ corresponding to $C_{k-1}^{L, j}$ is found. Third, we search in the image $I_k^R$ for the match of $C_{k-1}^{R, m}$, we call it $C_{k}^{R, n}$. Therefore, the edge curve $C_{k}^{R, n}$ is the match of $C_{k}^{L, i}$. We apply the same steps for all edge curves in the image $I_k^L$ in order to deduce their matches in $I_k^R$. From these correspondences, we can generate the ADM for the current frame $f_k$. An example of the ADM is depicted in Figure 3.

### 3.2 Stereo Matching Algorithm

We detail in this section the proposed algorithm adopted to obtain the disparity map. It is based on dynamic programming. The ADM is used to generate the global and local disparity ranges in each image scan-line. The pairs of matched edge points that create the ADM allow also to guide the dynamic programming algorithm to look for the best matches.
3.2.1 | Disparity range

The choice of minimal and maximal disparity values is a crucial key for the quality of the disparity map. We utilise the same idea as in [2] where the authors compute the v-disparity map [41] from the v-disparity and determine the range of the disparities based on its analysis. The main idea of the v-disparity map consists in accumulating pixels with the same disparity on the same scan-line. Therefore, the v-disparity map in a road scene can be divided into two main parts, i.e. the top part containing obstacles and the bottom one containing the road. The obstacles appear as vertical lines while the road as an oblique lines. In the part representing the road, the disparity in each scan-line \( y_i \) have a value of \( a \times y_i + b \) such that \( a \) and \( b \) are the oblique line equation parameters. Hence, to take into account the uncertainty caused by the computation, the disparity at scan-line \( y_i \) should be between 
\[
\begin{align*}
d_{\text{min}} &= (a \times y_i + b) - \alpha \\
d_{\text{max}} &= (a \times y_i + b) + \alpha
\end{align*}
\]
While the top part representing obstacles, the disparity value should be between 
\[
\begin{align*}
d_{\text{min}} &= d_1 - \alpha \\
d_{\text{max}} &= d_2 + \alpha
\end{align*}
\]
such that \( d_1 \) is the disparity value of the farthest obstacle and \( d_2 \) is the disparity of the closest object and \( \alpha \) is a tolerance value to select. Figure 4 illustrates the process of extracting disparity ranges.

3.2.2 | Local disparity ranges

Since the depth discontinuities are usually located on the edge points, we use the ADM to determine the disparity values of the edge points in the image to use them as local disparity ranges. In each scan-line, a set of local disparity ranges is determined. For each edge point \( e_i \), \( d_i = ADM(e_i) \) is the disparity determined by the ADM. The interval \([d_i - \beta, d_i + \beta] \) is defined as a local range deduced in that point, where \( \beta \) is a tolerance value to select. These ranges are used to guide the dynamic programming algorithm to make the best matches as shown in Figure 5.

3.2.3 | Cost function

We use a cost function based on gray scale, texture, gradient magnitude and orientation information in order to reflect the dissimilarity between two pixels \( p(x,y) \) and \( q(x,y + d) \) in the
same scan-line in the stereo images, which is defined by the following equation:

\[
C(p, q) = \sum_{j} (w_1 \times (f^L(x + i, y + j) - f^R(x + i, y + d + j)) + w_2 \times (f^L_{\text{grad}}(x + i, y + j) - f^R_{\text{grad}}(x + i, y + d + j)) + w_3 \times (f^L_{\text{lap}}(x + i, y + j) - f^R_{\text{lap}}(x + i, y + d + j)) + w_4 \times (f^L_{\text{s}}(x + i, y + j) - f^R_{\text{s}}(x + i, y + d + j)))
\]

(1)

such that \(I(x, y)\) is the gray intensity in the image, \(I_{\text{lap}}(x, y)\) is the census transform, \(I_{\text{grad}}(x, y)\) is the gradient magnitude and \(I_{\text{lap}}(x, y)\) is the gradient orientation. \(w_i\) are weights associated to each component. We believe that the use of these components will better reflect the dissimilarity between pixels even in the areas that contain little texture and distinct features.

### 3.2.4 Dynamic programming

To solve the problem of finding the best match between pixels on left and right images of the same scan-line, we present it as path finding problem on a 2D plane as shown in Figure 5. Let \(L = \{p_1, p_2, \ldots, p_3\}\) (resp. \(R = \{q_1, q_2, \ldots, q_3\}\) the set of pixel of the scan-line of the left (resp. right) image, and \(D = \{d_{\text{min}}, d_1, \ldots, d_{\text{max}}\}\) the set of disparity values integrated from the disparity range explained in the Section 3.2.1. The horizontal axis represents the left scan-line \(L\), while the vertical axis represents the set of disparity values \(D\). The intersection of these lines is called nodes, which indicates candidate matches between the two pixels \(p_i(x, y)\) and \(q_j(x, y + d_k)\). The cost function in 3.2.3 is used to fill the search plane. For each pixel \(p_i\), two paths are computed, one extending from left to \(p_i\) and the other one from right to \(p_i\). The disparity of the path with the minimum cumulative cost is selected as the disparity value for \(p_i\). Each of the paths is forced to pass through the local ranges integrated from the ADM described in 3.2.2, and the nodes outside the local ranges are discarded and noted as invalid nodes. This will correct any failure of the dynamic programming algorithm at any stage. This algorithm is applied for each scan-line of the frame \(f_k\).

Dynamic programming algorithm is known to cause a streaking effect near object boundaries. In order to overcome this problem, we use a disparity map filter based on weighted least squares filter (WLS) [42, 43], which will get rid of errors related the streaking effect due to its good performance and edge preserving smoothing.

### 3.3 Summary of the proposed method

1. **Step 1**: Initialise the system at the frame \(t = 0\) by computing the disparity map \(D_0\) using the dynamic programming algorithm.

2. **Step 2: ADM computation**
   (a) Edge detection of \(I^L\) and \(I^R\) using Canny edge detector.
   (b) Match the edge curves \(S_{k-1}^L\) with the edge curves \(S_k^L\) using Lucas-Kanade optical flow.
   (c) Match the edge curves \(S_{k-1}^R\) with the edge curves \(S_k^R\) using the disparity map \(D_{k-1}\).
   (d) Match the edge curves \(S_{k-1}^R\) with the edge curves \(S_k^R\) using Lucas-Kanade optical flow.
   (e) Deduces the correspondences between edge curves \(S_k^L\) and \(S_k^R\).

3. **Step 3: Stereo Matching**
   (a) Compute Disparity ranges (Local and global) using ADM.
   (b) Solve matching problem using the dynamic programming algorithm.

4. \(t = t + 1\).
5. Repeat from Step 2.

### 4 EXPERIMENTAL RESULTS

The proposed algorithm has been tested on synthetic and real stereo sequences for evaluation purposes and has been compared to recent temporal methods. We also compare it to the same algorithm presented in this paper without using the temporal information (i.e. disparity ranges) to highlight its importance. For the rest of this section, we will refer to the proposed algorithm as DSTM (Dense Spatio-Temporal Matching), and the method without the temporal information as SM (spatial matching).

#### 4.1 Synthetic stereo image sequences

We evaluate the proposed algorithm on the dataset MARS/PRESCAN synthetic stereo images publicly available in [44] containing stereo sequences and their ground-truths. The images are 512x512 in size and have a disparity range of 48 pixels.

From Fig 6, we clearly remark that the disparity computed using SM method shown in Figure 6(c) is more noised, which means that it contains more false matches. On the other hand, the disparity map computed using DSTM illustrated in Figure 6(b) appears to be more homogeneous, meaning that it contains less false matches. For clarity, we utilise false colours ranging from red (small disparity values) to blue (big disparity values), in order to represent the disparity maps.

Table 1 presents the results obtained by matching the frames #293 to #295, using DSTM and SM. The results are expressed in terms of the number of matched pixels (NMP), the percentage of correct matches (PCM) and the number of correct matches (NCM). The improvement that the temporal information yield is very recognisable. DSTM matches more pixels which means that it computes a more dense disparity map, and have a higher percentage of correct matches (PCM), outperforming SM by 6.25%, 3.99% and 4.96% in the frames #293, #294 and #295, respectively.
For further validation of the obtained results, DSTM has been compared to recent temporal matching methods. Fig 7 shows the left image of the frame #294 alongside the disparity maps obtained using DSTM and the spatio-temporal methods proposed in [2, 15, 29, 37]. We note that these methods generate sparse disparity maps, which means that they deal only with edge points which have discriminative features, unlike DSTM that generates a dense disparity map and deals with the entire image including occluded and texture-less areas.

Table 2 presents a comparison of the results obtained by the proposed method and the other estimators in [2, 15, 29, 37] at the frame #294. We remark that DSTM matches 160,671 pixel from which 156,833 are correct, and this is due to the fact that DSTM is a dense matching approach. Moreover, DSTM has the highest percentage of correct matches (97.61%) compared to the other method [2, 15, 29, 37] that match 97.35%, 96.15%, 93.99% and 88.03% respectively, despite the fact that DSTM deals with more difficult large untextured regions.

### 4.2 Real stereo sequences

We also conducted our experiments on real stereo sequences. We utilised colour images of KITTI 2015 dataset [45], which is a real-world dataset with street views from a driving car. It contains stereo sequences with sparse ground-truth disparities obtained using LiDAR sensor. The images size is 376×1240. Figure 8 shows examples of left images from this dataset and Figure 9 their corresponding disparity maps generated using...
**FIGURE 8** Example of left images from the KITTI dataset

**FIGURE 9** The disparity maps computed using:
(a) Ground truth. (b) DSTM. (c) SM

DSTM and SM. We notice that the disparity map computed using SM contains a lot of miss-matches contrary to the disparity map computed using DSTM that seems to be smooth, meaning that it contains less false matches.

We adopt the same metrics described in 4.1 to evaluate the performance of the proposed algorithm on the real stereo sequences. Table 3 shows the results obtained using DSTM compared to SM. As depicted in the table, DSTM provides better results compared to SM. In the disparity maps computed using DSTM, 132,408, 131,978 and 132,044 of the pixels are assigned a level of disparity with a percentage of correct matches of 90.29%, 90.25% and 87.24% in the frames #8, #9 and #10, respectively. On the other hand, the disparity maps computed using SM has 130,331, 129,720 and 130,313 of NMP with only 64.91%, 64.38% and 69.09% of correct matches in the frames #8, #9 and #10, respectively. This means that DSTM is far better at dealing with noise and artifact in real images.

Table 4 presents a comparison of the proposed approach with other ones that are evaluated under KITTI dataset. The results are reported in terms of percentage of stereo disparity outliers. We remark that the proposed method outperforms both SM and TW-SMNet[46]. We also note that the method
SegStereo [38] provide best results. However, the paid cost in terms of computational performance and hardware used for these approaches is very demanding.

4.3 Running time

The proposed algorithm has been implemented using C++ on an HP Intel Core i7-6700 CPU @ 3.40 GHz with 4GB RAM. Since the matching approach deals with each scan-line individually, it can be implemented using several threads. Table 5 illustrates the running time of the proposed method on 1, 4 and 8 CPUs for both the virtual images (frame #294) and the real ones (frame #9). In the virtual image, the ADM computation takes 835 ms, and the stereo matching step takes 13.76 ms for each scan line, which translates to 7.892 s, 2.749 s and 2.030 s when running on 1, 4 and 8 CPUs respectively. For the real image, due to its larger size, 2076 ms is needed to generate the ADM, and about 36.21 ms for computing the disparities of each scan-line. This means that the computation of the entire disparity map takes 15.632 s on 1 CPU, 6.389 s on 4 CPUs and 4.882 s when running on 8 CPUs. The current form of the proposed method is not applicable in real time since the shutter of the camera is triggered at 10 fps. Nonetheless, this issue can be resolved using a GPU with more processors to run both steps.

5 CONCLUSION

The aim of this paper was to present a dense spatio-temporal stereo matching method. The proposed approach takes into account the temporal consistency between adjacent frames, which allows the use the matching results obtained in the previous frame in the computation of the disparity map of the current frame. The proposed method is achieved in three major steps. First, an edge-based disparity map called the assisting disparity map (ADM) is computed. Second, for each scan-line, a global range and a set of local ranges are deduced from the ADM. Third, the obtained ranges are used to guide a two way dynamic programming algorithm to search for the best matches and therefore improve the results. The method proposed has been tested on synthetic and real stereo image sequences and the results obtained show its effectiveness.

ACKNOWLEDGEMENTS

This work is financially supported by the Minister of Higher Education and Scientific Research of Morocco (MESRSFC).

TABLE 5 The running time of DSTM using 1, 4 and 8 CPUs in the virtual and real dataset

|         | ADM | Matching | Total |         | ADM | Matching | Total |         | ADM | Matching | Total |
|---------|-----|----------|-------|---------|-----|----------|-------|---------|-----|----------|-------|
| 1 CPU   |     |          |       | 4 CPUs  |     |          |       | 8 CPUs  |     |          |       |
| #294    | 0.844 | 7.048    | 7.892 | 0.834   | 1.915 | 2.749    | 3.664 | 0.828   | 1.202 | 2.030    |       |
| #9      | 2.053 | 13.578   | 15.632| 2.080   | 4.309 | 6.389    | 8.788| 2.095   | 2.787 | 4.882    |       |

REFERENCES

1. Husain, A.A., et al.: Vehicle detection in intelligent transport system under a hazy environment: A survey. IET Image Processing (2019)
2. El.Ansari, M., et al.: Temporal consistent real-time stereo for intelligent vehicles. Pattern Recognition Letters 31(11), 1226–1238 (2010)
3. Kerkaou, Z., et al.: Edge points-based stereo matching approach for omnidirectional images. Journal of Electronic Imaging 27(5), 053015 (2018)
4. Jiao, J., et al.: Local stereo matching with improved matching cost and disparity refinement. IEEE Multimedia 21(4), 16–27 (2014)
5. Raghavendra, U., et al.: Anchor-diagonal-based shape adaptive local support region for efficient stereo matching. Signal, Image and Video Processing 9(4), 893–901 (2015)
6. Wang, L., et al.: How far can we go with local optimization in real-time stereo matching. In: 3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE, 129–136 (2006)
7. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, IEEE, 2, 508–515 (2001)
8. Boykov, Y., et al.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Analysis Machine Intelligence 23(11), 1222–1239 (2001)
9. Klaus, A., et al.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conf., IEEE, 3, 15–18 (2006)
10. Sun, J., et al.: Stereo matching using belief propagation. IEEE Trans. Pattern Analysis Machine Intelligence 25(7), 787–800 (2003)
11. Kobler, J., et al.: Enhancement of sparse silicon retina-based stereo matching using belief propagation and two-stage postfiltering. Journal of Electronic Imaging 23(4), 043011 (2014)
12. Ohta, Y., Kanade, T.: Stereo by intra-and inter-scanline search using dynamic programming. IEEE Trans. Pattern Analysis Machine Intelligence (2), 139–154 (1985)
13. El.Ansari, M., et al.: A real-time spatio-temporal stereo matching for road applications. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 1483–1488 (2011)
14. Jiang, J., et al.: Disparity prediction between adjacent frames for dynamic scenes. Neurocomputing 142, 335–342 (2014)
15. El.Jaafari, I., et al.: Fast spatio-temporal stereo matching for advanced driver assistance systems. Neurocomputing 194, 24–33 (2016)
16. Bouguet, J.Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation 5(1-10), 4 (2001)
17. Lucas, B.D., et al.: An iterative image registration technique with an application to stereo vision. (1981)
18. Nalpantidis, L., et al.: Efficient hierarchical matching algorithm for processing uncalibrated stereo vision images and its hardware architecture. IET image processing 5(5), 481–492 (2011)
19. Yu, W., et al.: Real time stereo vision using exponential step cost aggregation on gpu. In: Image Processing (ICIP), 2009 16th IEEE International Conference on, IEEE, 4281–4284 (2009)
20. Hosni, A., et al.: Fast cost-volume filtering for visual correspondence and beyond. IEEE Trans. Pattern Analysis Machine Intelligence 35(2), 504–511 (2013)
21. Pham, C.C., Jeon, J.W.: Domain transformation-based efficient cost aggregation for local stereo matching. IEEE Trans. Circuits Systems for Video Technology 23(7), 1119–1130 (2013)
22. Yang, Q.: A non-local cost aggregation method for stereo matching. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 1402–1409 (2012)
23. Mei, X., et al.: Segment-tree based cost aggregation for stereo matching. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 313–320 (2013)
24. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis Machine Intelligence 30(2), 328–341 (2008)
25. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision 47(1-3), 7–42 (2002)
26. Li, R., et al.: Disparity search range estimation based on dense stereo matching. In: Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, IEEE, 753–759 (2013)
27. Gong, M.: Enforcing temporal consistency in real-time stereo estimation. In: European Conference on Computer Vision, Springer, 564–577 (2006)
28. Kerkaou, Z., et al.: Fast spatio-temporal stereo matching method for omnidirectional images. In: 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, 1–6 (2018)
29. El.Jaafari, I., et al.: Fast edge-based stereo matching approach for road applications. Signal, Image and Video Processing 11(2), 267–274 (2017)
30. Zhang, L., et al.: Spacetime stereo: Shape recovery for dynamic scenes. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, IEEE, 2, II–367 (2003)
31. Zhang, G., et al.: Consistent depth maps recovery from a video sequence. IEEE Transactions on Pattern Analysis Machine Intelligence 31(6), 974–988 (2009)
32. Davis, J., et al.: Spacetime stereo: A unifying framework for depth from triangulation. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, IEEE, 2, II–359 (2003)
33. Min, D., et al.: Disparity search range estimation: enforcing temporal consistency. In: Acoustics, Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, IEEE, 2366–2369 (2010)
34. Cai, J.: Integration of optical flow and dynamic programming for stereo matching. IET image processing 6(3), 205–212 (2012)
35. Dobias, M., Sara, R.: Real-time global prediction for temporally stable stereo. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 704–707 (2011)
36. Vedula, S., et al.: Kanade, three-dimensional scene flow. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1991.
37. Mazoul, A., et al.: Fast spatio-temporal stereo for intelligent transportation systems. Pattern Analysis and Applications 17(1), 211–221 (2014)
38. Yang, G., et al.: Segstereo: Exploiting semantic information for disparity estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), 636–651 (2018)
39. Nie, G.Y., et al.: Multi-level context ultra-aggregation for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3283–3291 (2019)
40. Canny, J.: A computational approach to edge detection. In: Readings in Computer Vision, Elsevier, 184–203 (1987)
41. Labayrade, R., et al.: Real time obstacle detection in stereovision on non flat road geometry through “v-disparity” representation. In: Intelligent Vehicle Symposium, 2002. IEEE, IEEE, 2, 646–651 (2002)
42. Min, D., et al.: Fast global image smoothing based on weighted least squares. IEEE Trans. Image Processing 23(12), 5638–5653 (2014)
43. Farbman, Z., et al.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In: ACM Trans. Graphics (TOG), ACM, 27, 67 (2008)
44. van der Mark, W.: Synthetic stereo image data for algorithm evaluation http://stereodatasets.wvandermark.com/. (2008)
45. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
46. El.Khamy, M., et al.: Multi-task learning of depth from tele and wide stereo image pairs. In: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 4300–4304 (2019)

How to cite this article: Kerkaou Z, El Ansari M, Masmoudi L, Lahmyed R. Dense spatio-temporal stereo matching for intelligent driving systems. IET Image Process. 2021;15:715–723.
https://doi.org/10.1049/ipr2.12056