

Temporally Consistent Disparity and Optical Flow via Efficient Spatio-temporal Filtering

Asmaa Hosni*, Christoph Rhemann**, Michael Bleyer, and Margrit Gelautz

Institute for Software Technology and Interactive Systems
Vienna University of Technology, Vienna, Austria
{asmaa,rhemann,bleyer,gelautz}@ims.tuwien.ac.at
<http://www.ims.tuwien.ac.at>

Abstract. This paper presents a new efficient algorithm for computing temporally consistent disparity maps from video footage. Our method is motivated by recent work [1] that achieves high quality stereo results by smoothing disparity costs with a fast edge-preserving filter. This previous approach was designed to work with single static image pairs and does not maintain temporal coherency of disparity maps when applied to video streams.

The main contribution of our work is to transfer this concept to the spatio-temporal domain in order to efficiently achieve temporally consistent disparity maps, where disparity changes are aligned with spatio-temporal edges of the video sequence. We further show that our method can be used as spatio-temporal regularizer for optical flow estimation. Our approach can be implemented efficiently, achieving real-time results for stereo matching. Quantitative and qualitative results demonstrate that our approach (i) considerably improves over frame-by-frame methods for both stereo and optical flow; and (ii) outperforms the state-of-the-art for local space-time stereo approaches.

1 Introduction

Computing disparity maps from two static images is a well-studied problem and many methods have reported impressive results in the past (see e.g. [7] for an overview). However, applying even the best of these methods to *sequences* of stereo pairs in a frame-by-frame manner, often results in temporally inconsistent disparity maps. This temporal inconsistency is perceived as displeasing flickering, when using such disparity maps to visualize a video on auto-stereoscopic monitors. Such displays use disparity maps to render novel views from virtual view points.

The goal of this work is to improve the quality of the reconstruction by additionally exploiting the temporal information available in the input sequence. To understand the basic idea behind our spatio-temporal stereo method, let us first

* Asmaa Hosni was supported by the Vienna PhD School of Informatics.

** Christoph Rhemann and Michael Bleyer received financial support from Vienna Science and Technology Fund (WWTF) under project ICT08-019.

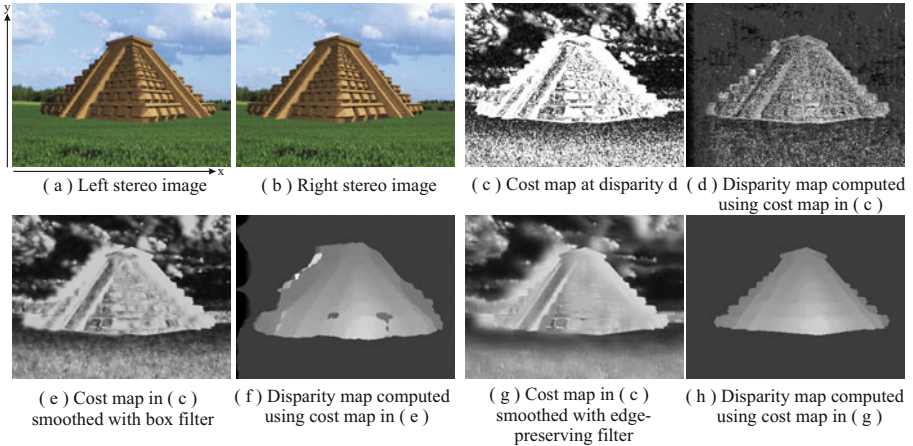


Fig. 1. Single-frame stereo matching. From a stereo pair, captured at a single moment (a),(b) the associated cost maps and computed disparity maps are shown for different filtering techniques (c)-(h). See the text for a detailed explanation.

consider a conventional local stereo approach that computes a disparity map from a static image pair. Such methods typically build upon a Winner-Takes-All framework [7] and first compute the costs for choosing disparity d at each pixel. An example of the resulting cost map for a single disparity level is shown in fig. 1(c). By choosing for each pixel the disparity with the lowest cost gives a noisy disparity map (see fig. 1(d)). The key idea of block matching algorithms is to improve the result by aggregating the costs over a (square) support window. It is known (see e.g. [1],[7]) that this cost aggregation is equivalent to filtering the cost map with a box filter. Fig. 1(e) shows the cost map in fig. 1(c) after smoothing with a box filter. The resulting disparity map in fig. 1(f) is smoother in comparison to the one generated from the raw cost map in fig. 1(d). The implicit assumption of box filtering methods is that all pixels inside the filter kernel have constant disparity. This assumption is violated if the filter kernel overlaps a depth discontinuity, which often coincides with object boundaries. Therefore, the disparity map generated by box filtering (fig. 1(f)) does not preserve the disparity discontinuities well. A solution is provided by adaptive support weight approaches [2],[3],[6],[1] that locally adjust the filter kernel such that it does not overlap the object boundaries. In particular, such approaches smooth the costs with a *weighted* box filter. The weights are chosen such that they are high in regions which are close in color and distance to the central pixel of the filter kernel and low otherwise. The disparity map in fig. 1(h) better preserves the disparity discontinuities, because it was generated from the cost maps filtered with a weighted box filter (fig. 1(g)).

In this work, we transfer the adaptive support weight concept to the spatio-temporal domain: Given multiple frames in time, a spatio-temporal cost volume is generated by stacking the cost maps of the input frames shown in fig. 2(a) (the cost maps are not visualized in fig. 2). A simple approach would smooth

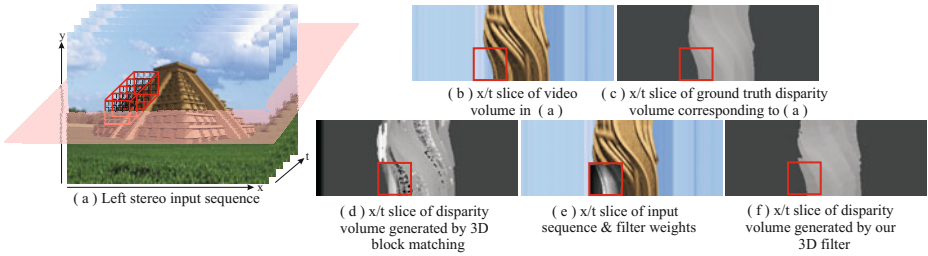


Fig. 2. Spatio-temporal stereo matching. Given a sequence of stereo images (a) the cost volume is smoothed using a 3D box filter illustrated in red in (a) and (b). The resulting disparity map (d) does not preserve disparity discontinuities. Our approach weights the pixels inside the 3D box filter (e) to achieve a result (f) that is aligned with the space-time object boundaries.

the cost volume with a 3D spatio-temporal box filter as illustrated by the red cube in fig. 2(a). This approach assumes that disparities are constant inside this space-time window. This assumption is not met for the red box in fig. 2(b), since the space-time window overlaps an object boundary. Hence, the resulting disparity map (fig. 2(d)) over-smooths the spatio-temporal object boundaries. In our approach, we weight the pixels inside the 3D filter kernel - pixels which belong to the same object as the center pixel receive high weights and low weights if they lie on a different object. Our assumption is that the disparity of an *object* is approximately constant over a small space-time window. The filter weights for the kernel outlined in red in fig. 2(e) are visualized by the intensity values inside the red box: Bright pixels encode high weights and dark pixels encode low weights. We see that the weights nicely adapt to the object outline. As a consequence the object boundaries are well preserved in the resulting disparity map shown in fig. 2(f).

The works most closely related to our approach are the methods of [6] and [1]. The stereo method of [6] uses a fast approximation of the bilateral filter to smooth the disparity costs in the spatio-temporal domain. This method is fast (real-time) but the results cannot compete with the state-of-the-art in stereo matching. [1] showed that state-of-the-art results can be achieved at run times similar to [6], by replacing the bilateral filter with the guided image filter [5]. However, the authors of [1] did not adopt their method to temporal sequences. Therefore, the main contribution of our work is to extend the approach of [1] to the spatio-temporal domain. In particular, we present a space-time stereo method that works in real-time and achieves results that outperform the space-time stereo approach of [6]. This is done by extending the guided image filter [5] to the spatio-temporal domain. Furthermore, we leverage this concept to optical flow estimation to achieve temporally smooth flow fields.

Related Work

In the following, we discuss related work that incorporates temporal smoothness constraints for stereo and optical flow estimation.

Although first attempts to enforce temporal coherence for stereo matching go back as far as [20], relatively little research has been conducted in this domain since then. Many previous approaches simply encourage temporally constant disparity solutions. This is either enforced locally by smoothing the cost volume with a rectangular spatio-temporal support window [19],[18],[17] (as discussed above) or globally by propagating the disparity over temporally consecutive pixels (see e.g. [21],[9]). These methods cannot cope well with considerably large movement of scene objects. To account for faster moving scene objects, previous work proposed to orient the rectangular averaging window in the space-time domain such that linear motions can be handled [17]. Other methods compute the optical flow field between consecutive frames and then smooth the disparity values along the computed flow vectors [22],[23]. In contrast, our method implicitly finds the spatio-temporal neighborhood and thus avoids computing a flow field explicitly. Another approach to overcome the problem of flow estimation is [25], where filter responses that capture the local spatio-temporal structure of the video volume are used as matching primitives. A related field of research aims to compute the three-dimensional scene flow [8], which is the computation of the 3D motion field using scene structure information. In contrast, our work aims to reconstruct a temporally smooth disparity map without recovering three-dimensional flow vectors. Note that avoiding the need to compute optical flow is a considerable advantage. We do need to address the optical flow problem that might due to its large label space (consisting of all 2D vectors) be more challenging than the problem that we are actually trying to solve, i.e. temporal stereo. Obviously, we can also avoid the computational overhead that goes along with optical flow computation.

Analogously to space-time stereo matching, only small amount of works have been devoted to temporal consistent optical flow computation. Most algorithms encourage temporal neighboring pixels to be assigned to the same flow vector [16],[15],[14],[13],[24], which however incorporates the assumption that the flow field remains piecewise constant over time. In [12] the less restrictive assumption of constantly moving objects is encoded by encouraging matching pixels in two consecutive frames to take the same flow vector. Our approach is more related to non-local smoothness terms that assume that the flow vector at a certain pixel is similar to the vectors at self-similar pixels in a larger neighborhood. To the best of our knowledge such non-local smoothness terms have only been applied in the spatial domain yet [11], [10].

2 Proposed Algorithm

In this section we describe our algorithm for generating temporally coherent disparity maps. Section 3 adopts this method as temporal regularizer for optical flow estimation.

Our temporal stereo matching algorithm comprises three major steps: (i) construct a spatio-temporal cost volume for each disparity d ; (ii) smooth each of these cost volumes with a spatio-temporal filter; and (iii) select for each pixel

in the spatio-temporal volume the disparity which holds the lowest costs in its corresponding cost volume. We now discuss each of these steps in detail.

2.1 Cost Volume Construction

In the first step, a cost volume C^d is constructed for each disparity d . This cost volume is a three-dimensional (spatio-temporal) array with axes x, y and t . It stores the costs for choosing disparity d at each voxel $i = (x, y, t)$ (we denote a pixel in a spatio-temporal volume as voxel), where x, y and t are its spatial and temporal coordinates, respectively.

Let I^l and I^r be two spatio-temporal video volumes with axes x, y, t that define a sequence of rectified stereo pairs. Also, let $u = (d, 0, 0)$ be a vector that defines the displacement in x, y and t dimensions. The costs are defined by the correlation of voxel i in I^l with its matching voxel in I^r shifted by vector u :

$$C_i^d = \alpha \cdot M_{i,u}^c + (1 - \alpha) \cdot M_{i,u}^g. \quad (1)$$

Similar to [1], the correlation costs in eq. (1) comprise a color term M^c that is weighted against a gradient term M^g by a factor α . The color term is defined as:

$$M_{i,u}^c = \min (\|I_i^l - I_{i-u}^r\|, \tau_c),$$

where τ_c is a truncation value and $\|I_i^l - I_{i-u}^r\|$ is the summed-up absolute differences in RGB values. Similarly, we define the gradient term as:

$$M_{i,u}^g = \min (\|\nabla_x I_i^l - \nabla_x I_{i-u}^r\|, \tau_g),$$

where ∇_x denotes the gradient in x direction and τ_g is a truncation value.

2.2 Spatio-temporal Cost Volume Smoothing

Once the cost volume is constructed for each disparity, we filter each cost volume in the spatio-temporal domain. In particular, the smoothed cost value \widehat{C}_i^d , associated with disparity d at voxel i is a weighted average of neighboring voxels in C^d :

$$\widehat{C}_i^d = \sum_j W_{i,j} (I^l) C_j^d. \quad (2)$$

The filter weights $W_{i,j}$ depend on the the sequence of input reference frames I^l and have to be chosen such that spatio-temporal edges in I^l are preserved in the filtered output. A possible choice is to use the weights of the bilateral filter. They have the drawback that an exact implementation is slow and approximations come at the loss of quality. Therefore, we follow [1] where the weights of the recently proposed guided image filter [5] have been used. The guided filter has edge-preserving properties similar to the bilateral filter and its exact implementation has a runtime independent of the filter size. Originally, the guided

filter [5] has been defined for the spatial domain only. In this work we extend it to a spatio-temporal kernel, where the filter weights are defined as:

$$W_{i,j} = \frac{1}{|\omega|^2} \sum_{k:(i,j) \in \omega_k} (1 + (I_i^l - \mu_k)^T (\Sigma_k + \epsilon U)^{-1} (I_j^l - \mu_k)). \quad (3)$$

Here, Σ_k and μ_k are the covariance and mean vector computed over the spatio-temporal window ω_k with dimensions $w_x \times w_y \times w_t$, centered at voxel k in the video volume I^l . The number of pixels in this 3D window is denoted by $|\omega|$ and ϵ is a smoothness parameter. I_i^l , I_j^l and μ_k are 3×1 (color) vectors, and the covariance matrix Σ_k and identity matrix U are of size 3×3 .

The guided filter weights do not have to be computed explicitly and can be implemented exactly by applying a sequence of linear operations to the input cost volume [5]:

$$\begin{aligned} a_k &= (\Sigma_k + \epsilon U)^{-1} \left(\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i^l C_i^d - \mu_k \bar{C}_k^d \right). \\ b_k &= \bar{C}_k^d - a_k^T \mu_k. \\ \widehat{C}_i^d &= \bar{a}_i^T I_i^l + \bar{b}_i. \end{aligned} \quad (4)$$

Here, $\bar{C}_k^d = \frac{1}{|\omega|} \sum_{i \in \omega_k} C_i^d$ is the mean of C^d in ω_k and $\bar{a}_i = \frac{1}{|\omega|} \sum_{i \in \omega_k} a_k$ and $\bar{b}_i = \frac{1}{|\omega|} \sum_{i \in \omega_k} b_k$. All summations are 3D box filters and can be computed in $O(N)$ time, where N is the number of image voxels.

2.3 Disparity Selection

After smoothing the cost volumes with the spatio-temporal guided filter, a spatio-temporal disparity volume f^l , which holds the sequence of disparity maps for the left input sequence I^l , is computed by applying the Winner-Takes-All strategy:

$$f_i^l = \operatorname{argmin}_{d \in \mathcal{D}} \widehat{C}_i^d, \quad (5)$$

where \mathcal{D} is the set of allowed disparities.

In addition to the disparity volume for the left video volume I^l , we also compute the disparity volume f^r for the right input video volume I^r in a similar manner by substituting I^l with I^r and vice versa in eqs. (1)-(4). Then we apply left-right consistency checking: A pixel in the left disparity volume f^l is marked as invalid, if the disparity value of its matching pixel in f^r differs by a value > 1 pixel. The invalid pixels are then filled by the lowest disparity value of the closest valid pixels which lie on the same spatial scanline (i.e. a single row in x dimension). This simple filling can generate artifacts in the output disparity maps. To reduce these artifacts, we apply a spatio-temporal weighted median filter on the *filled* regions. This weighted median filtering smoothes the filled

pixels, while preserving the object boundaries. In particular, we use a filter kernel with dimensions $w_x^b \times w_y^b \times w_t^b$ with bilateral filter weights:

$$W_{i,j}^{BL} = \frac{1}{K_i} \exp\left(-\frac{|i-j|^2}{\sigma_s^2}\right) \exp\left(-\frac{|I_i - I_j|^2}{\sigma_c^2}\right), \quad (6)$$

where K_i is a normalization factor and σ_s , σ_c are parameters which adjust the spatial and color dissimilarity, respectively.

3 Temporally Consistent Optical Flow

We now adopt our temporal stereo matching framework to optical flow estimation. Similar to stereo, the goal is to reduce the ambiguity of the solution space by exploiting the temporal coherence of the flow field. The implicit assumption of our algorithm is that the flow vectors are constant within self-similar regions of the video volume. This assumption is usually met for objects that do not quickly change their speed and direction.

Our optical flow algorithm is almost identical to our stereo method. Here, the displacement vector u , used in the terms M^c and M^g in eq. (1), is defined as $u = (a, b, 0)$, where a and b is the flow in x and y directions, respectively. For constructing the cost volume we modify the stereo correlation measure, by additionally using the gradient in y direction. This is done by replacing term M^g in eq. (1) with

$$M_{i,u}^g = \min \left(\|\nabla_x I_i^l - \nabla_x I_{i-u}^r\| + \|\nabla_y I_i^l - \nabla_y I_{i-u}^r\|, \tau_g \right). \quad (7)$$

Once the cost volumes are established, we filter them and obtain the flow field by applying the Winner-Takes-All strategy as in sec. 2.3. As for stereo, we apply left-right consistency checking to determine invalid matches¹. We then follow [1] and fill the invalid matches by applying a 2D (spatial) weighted median filter of size $w_x^b \times w_y^b$ with weights as in eq. (6). Since the weighted median filter overlaps some valid pixels, the flow vectors can be propagated to the invalid regions. To compute sub-pixel accurate flow fields, we simply upscale the input frames using bicubic interpolation as done in [26],[1].

4 Experimental Results

We implemented and tested our proposed temporal stereo and optical flow algorithms on a PC equipped with an Intel Core 2 Quad, 2.4 GHZ CPU and an Nvidia GeForce GTX480 GPU with 1.5GB of memory. We used CUDA for implementing our algorithm on the GPU.

In our test runs, the parameters of our algorithm were set to constant values. We use the following constant parameter settings to generate all results for both stereo matching and optical flow: $\{w_x=w_y, w_t=w_t^b, w_x^b=w_y^b, \epsilon, \alpha, \sigma_s, \sigma_c, \tau_c, \tau_g\} = \{31, 5, 15, 0.001, 0.5, 9, 0.1, 0.028, 0.008\}$. These parameters have been found empirically.

¹ In optical flow literature it is called forward-backward consistency check.

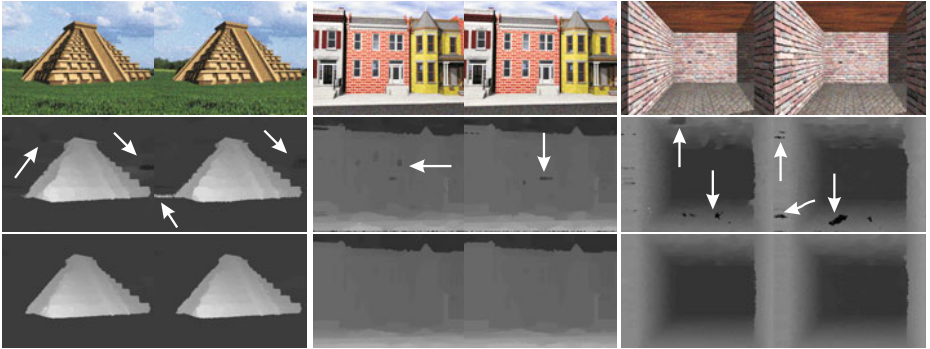


Fig. 3. Temporal vs. frame-by-frame processing. 1^{st} row: Left input frames of stereo sequences are shown. 2^{nd} row: Disparity maps computed by a frame-by-frame implementation show flickering artifacts (arrows point to major artifacts). 3^{rd} row: Our proposed method exploits temporal information, thus can remove most artifacts. Test sequences courtesy of [6].

Table 1. Quantitative stereo comparison. Our method outperforms all competitors in terms of quality and is the fastest method that maintains temporal coherent results (time measured without post-processing). See text for details.

| Algorithm | Time[ms] | Book | | Street | | Tanks | | Temple | | Tunnel | |
|----------------------------|-----------|-------------|------------|-------------|------------|-------------|------------|--------------|------------|-------------|------------|
| | | mean | stdev | mean | stdev | mean | stdev | mean | stdev | mean | stdev |
| DCBGrid [6] | 51 | 52.2 | 2.1 | 32.5 | 2.3 | 36.0 | 6.2 | 39.5 | 1.9 | 25.7 | 11.1 |
| Ours frame-by-frame | 41 | 18.0 | 1.5 | 16.4 | 1.3 | 10.8 | 1.7 | 11.17 | 2.1 | 14.4 | 7.8 |
| Temporal DCBGrid [6] | 90 | 44.0 | 2.0 | 25.9 | 2.0 | 31.4 | 6.1 | 31.7 | 1.8 | 36.4 | 7.9 |
| Ours Temporal | 41 | 10.1 | 1.5 | 12.2 | 1.0 | 8.7 | 1.3 | 6.6 | 1.5 | 17.7 | 8.6 |

4.1 Stereo

We evaluated our stereo method visually on real stereo video sequences as well as quantitatively using a synthetic dataset comprising five stereo sequences with known ground truth disparity, provided by [6]. We compare our spatio-temporal stereo method to both the frame-by-frame and the spatio-temporal method of [6], which we denote as “DCBGrid” and “Temporal DCBGrid”, respectively (we used the authors implementation to generate results for [6]). A further competitor is our approach applied in a frame-by-frame manner, i.e. using a temporal filter window of size 1 ($w_t = w_t^b = 1$ in eq. (3) and eq. (6)). Note that our method applied on a frame-by-frame basis degenerates to the method of [1].

First, we visually compare the results of our spatio-temporal method with the frame-by-frame variant of our approach. Fig. 3 shows results for two consecutive frames of three synthetic sequences. The disparity maps generated by our spatio-temporal method (3^{rd} row in fig. 3) are temporally coherent and exhibit less artifacts than the disparity maps generated by our method applied in a frame-by-frame manner (2^{nd} row in fig. 3).

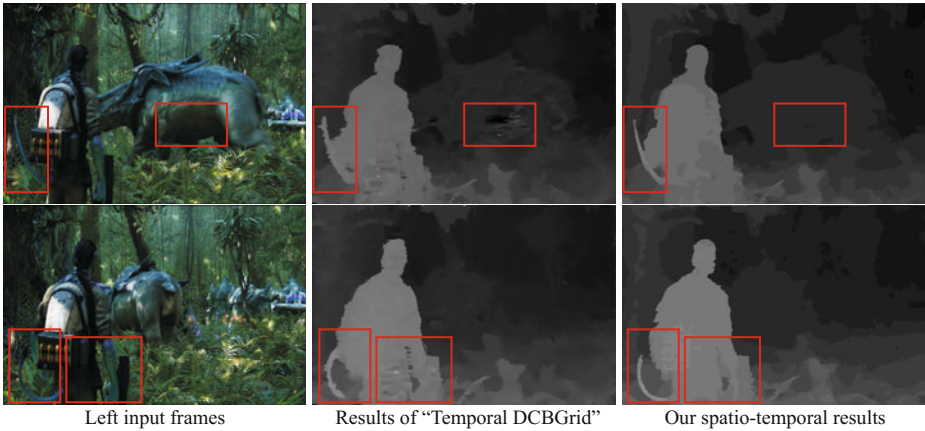


Fig. 4. Visual comparison to “Temporal DCBGrid” [6] Example (1). On these two frames of a movie sequence our temporal regularization could remove artifacts better than the temporal approach of [6] (red boxes mark major differences).

Next, we quantitatively compare our approach to its competitors. To this end, we follow [6] and generate results for the dataset of [6] after adding Gaussian noise² of ($\sigma = 20$) to the input frames. Table 1 shows the mean error (percentage of bad pixels) for the five ground truth sequences as well as their standard deviations. The table shows that our spatio-temporal method outperforms the “Temporal DCBGrid” in terms of mean error and smaller standard deviations³. This indicates that our algorithm gives results of higher quality and is more robust. This is also reflected in the visual comparison shown in figs. 4 and 5. Results for the full video sequences are shown in the supplementary material. Table 1 also reveals that our temporal method is more than two times faster⁴ than the “Temporal DCBGrid”. An important note from this table is that the time consumed by our frame-by-frame method is the same as the time of our temporal method. This is mainly due to the fact that in the 3D box filter the sliding window technique is used in the three dimensions w_x , w_y and w_t .

In another experiment, we evaluate the robustness of the different methods to noise. In particular, we plot the error rates at different noise levels (additive Gaussian noise with σ in the range from 0 to 100) for each of the five sequences of [6] in fig. 6. We see that for a noise level of zero, all methods are close-by. This is because the synthetic scenes are absolutely noise-free (which is in contrast to real-world scenarios), which makes it relatively easy for the different

² This was done in order to provide a more realistic scenario, since the original images are noise-free.

³ The two algorithms have poor performance in the “Tunnel” sequence. A possible explanation is that this is due to that this sequence is highly textured and the filtering process over-smoothes its frames.

⁴ Times were measured on different machines but should give a good indication of the computational complexity.

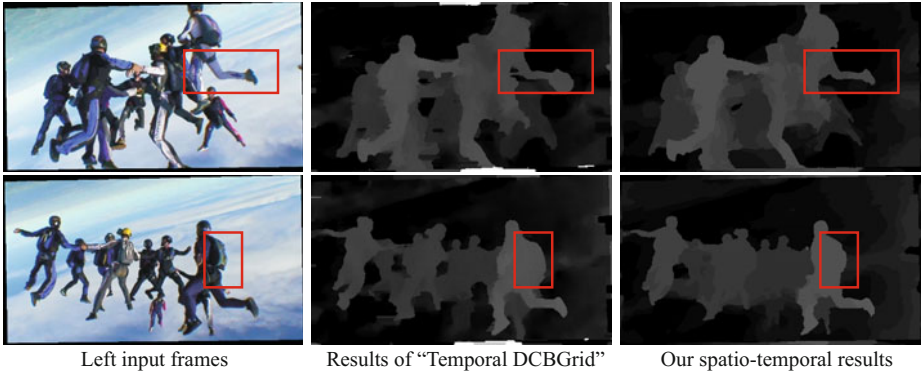


Fig. 5. Visual comparison to “Temporal DCBGrid” [6] Example (2). Our spatio-temporal method can preserve some object outlines better than the approach of [6].

approaches to perform well. However, as the noise level increases the error level for the “DCBGrid” and the “Temporal DCBGrid” increase faster than the error level for our methods. Furthermore, the “Temporal DCBGrid” only slightly improves over its frame-by-frame counterpart (“DCBGrid”). In contrast, the improvement gained by our temporally coherent algorithm over the frame-by-frame implementation of our approach is considerably larger.

4.2 Optical Flow

We follow previous work [24] and test our approach on the “Yosemite” and “Marble” sequences which can be obtained from <http://www.cs.brown.edu/black/images.html> and http://i21www.ira.uka.de/image_sequences/, respectively. We did not test on the Middlebury sequences, because they violate our assumption of a temporally smooth flow field. (Note that also previous work did not test on these sequences for the same reason [24].)

Table 2. Quantitative flow comparison. The average angular error (AAE) for different temporal window sizes on the “Yosemite” and “Marble” sequences is shown.

| Temporal Window Size (w_t) | 1 | 3 | 5 | 7 |
|--------------------------------|-------|-------|-------|-------|
| Yosemite Mean AAE | 3.85° | 2.90° | 2.83° | 2.76° |
| Marble Mean AAE | 6.72° | 5.26° | 4.78° | 4.61° |

We show the Average Angular Error (AAE) for different sizes of the temporal filter window w_t on the “Yosemite” and “Marble” sequences in table 2. We see that an increased temporal window size notably reduces the error. (Note that when using a temporal filter window of size 1 our method degenerates to the approach in [1].) Hence, our temporal processing improves the results of [1]. We

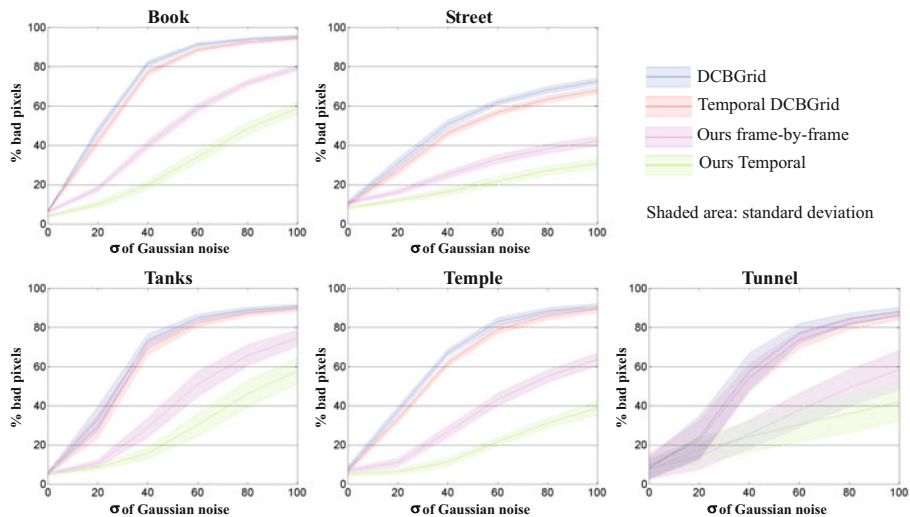


Fig. 6. Robustness to noise. We plot the average error vs. Gaussian noise levels for five ground truth sequences. See the text for a discussion and the supplementary material for a visual comparison. Best viewed in color.

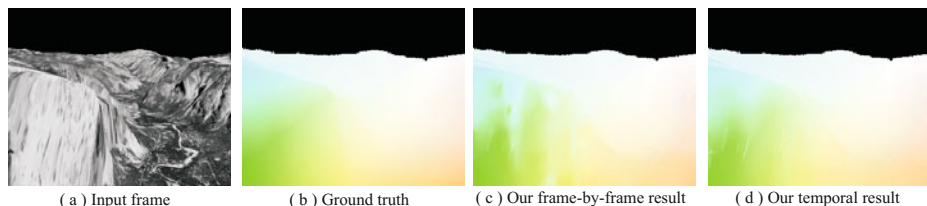


Fig. 7. Effect of temporal smoothing for optical flow. Temporally smoothing the flow field gives a result that is closer to the ground truth than the flow generated without temporal coherency. Color coding of flow maps as on the Middlebury evaluation page.

visually show the effect of the temporal processing for frame 8 of the “Yosemite” sequence in fig. 7. The optical flow map generated with our spatio-temporal method is closer to the ground truth than the one computed by our frame-by-frame method.

5 Conclusion

This paper proposed an efficient technique for computing temporally coherent disparity maps from a sequence of stereo images. The main contribution was to extend the filter-based frame-by-frame stereo approach of [1] to the temporal domain. To this end we adopted the 2D image filter of [5] to the 3D spatio-temporal space. A further contribution was to apply our method to optical flow

estimation. We demonstrated that exploiting temporal information considerably improves the frame-by-frame approach of [1] for both stereo and optical flow estimation and we outperform the current state-of-the-art in local space-time stereo matching.

Future work may concentrate on obtaining a better understanding of the relationship between filtering-based and energy-based optimization. This knowledge may lead to fast and even better filtering approaches than presented in this paper.

References

1. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast Cost-Volume Filtering for Visual Correspondence and Beyond. In: CVPR (2011)
2. Yoon, K.J., Kweon, I.S.: Locally Adaptive Support-Weight Approach for Visual Correspondence Search. In: CVPR (2005)
3. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local Stereo Matching Using Geodesic Support Weights. In: ICIP (2009)
4. Hosni, A., Bleyer, M., Gelautz, M.: Near Real-Time Stereo With Adaptive Support Weight Approaches. In: 3DPVT (2010)
5. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
6. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.A.: Real-time Spatiotemporal Stereo Matching Using the Dual-Cross-Bilateral Grid. In: Daniilidis, K. (ed.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 510–523. Springer, Heidelberg (2010)
7. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV* 47(1/2/3), 7–42 (2002), <http://www.middlebury.edu/stereo/>
8. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-Dimensional Scene Flow. In: ICCV (1999)
9. Leung, C., Appleton, B., Lovell, B.C., Sun, C.: An Energy Minimisation Approach to Stereo-Temporal Dense Reconstruction. In: ICPR (2004)
10. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR (2010)
11. Werlberger, M., Pock, T., Bischof, H.: Motion estimation with non-local total variation regularization. In: CVPR (2010)
12. Salgado, A., Sánchez, J.: Temporal Constraints in Large Optical Flow Estimation. In: Moreno Díaz, R., Pichler, F., Quesada Arenceibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 709–716. Springer, Heidelberg (2007)
13. Brox, T., Bruhn, A., Papenber, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
14. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. In: *JMIV*, vol. 14, pp. 245–255 (2001)
15. Black, M.J., Anandan, P.: Robust dynamic motion estimation over time. In: CVPR (1991)

16. Nagel, H.H.: Extending the 'Oriented Smoothness Constraint' into the Temporal Domain and the Estimation of Derivatives of Optical Flow. In: Faugeras, O. (ed.) ECCV 1990. LNCS, vol. 427, pp. 139–148. Springer, Heidelberg (1990)
17. Zhang, L., Curless, B., Seitz, S.M.: Spacetime Stereo: Shape Recovery for Dynamic Scenes. In: CVPR (2003)
18. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: high-resolution capture for modeling and animation. In: SIGGRAPH (2004)
19. Davis, J., Nehab, D., Ramamoorthi, R., Rusinkiewicz, S.: Spacetime stereo: a unifying framework for depth from triangulation. In: PAMI, vol. 27(2), pp. 296–302 (2005)
20. Jenkin, M., Tsotsos, J.: Applying temporal constraints to the dynamic stereo problem. In: CVGIP, vol. 33, pp. 16–32 (1986)
21. Williams, O., Isard, M., McCormick, J.: Estimating Disparity and Occlusions in Stereo Video Sequences. In: CVPR (2005)
22. Larsen, E.S., Mordohai, P., Pollefeys, M., Fuchs, H.: Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. In: ICCV (2007)
23. Bleyer, M., Gelatuz, M.: Temporally Consistent Disparity Maps from Uncalibrated Stereo Videos. In: ISPA (2009)
24. Zimmer, H., Bruhn, A., Weickert, J.: Optic Flow in Harmony. In: IJCV, vol. 93, pp. 368 – 388 (2011)
25. Sizintsev, M., Wildes, R.P.: Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching. In: CVPR (2009)
26. Steinbrücker, F., Pock, T., Cremers, D.: Large displacement optical flow computation without warping. In: ICCV (2009)