DEPTH ESTIMATION AND DEPTH ENHANCEMENT BY DIFFUSION OF DEPTH FEATURES

Nikolce Stefanoski[1], *Can Bal*[1], *Manuel Lang*[1,2], *Oliver Wang*[1], *and Aljosa Smolic*[1]

[1]Disney Research Zürich [2]ETH Zürich

ABSTRACT

Current trends in video technology indicate a significant increase in spatial and temporal resolution of video data. Recently, a linear-runtime feature diffusion algorithm was presented which aims for fast and accurate processing of such high resolution data. In this paper, we introduce this algorithm from the perspective of image-based depth estimation, expanding upon the algorithm by requiring inter-view consistency in the depth diffusion process. We also discuss different application scenarios and provide an in-depth analysis of the method in this context.

Index Terms— Depth estimation, depth enhancement, depth diffusion, disparity.

1. INTRODUCTION

Depth maps are widely used in many areas, enabling applications such as controller-free gaming [1], depth-aware compositing [2], or depth adjustment in stereoscopic displays. Depth maps will also be a part of new standards currently in development for compressed 3D video [3], which will enable glasses-free stereo.

Depending on the application, the requirements for depth map accuracy are different. While low resolution and low accuracy depth maps are sufficient for some applications, e.g. for human pose estimation in gaming application [1], for some others much higher accuracy is required, e.g. professional content production [2]. Current automatic image-based depth estimation algorithms do not provide sufficiently accurate depth maps in general. For this reason, in professional content production, depth map generation workflow still involves a vast amount of manual interaction. This is not surprising since depth estimation from image data alone is an ill-posed problem. Often additional high-level knowledge about the scene content is required in order to resolve depth ambiguities, and to estimate correct depth values in some parts of a scene such as regions with limited texture.

There are also other factors which limit accuracy of some automatic depth estimation algorithms. Current technology trends in content creation, transmission and display indicate wider adoption of ultra-high spatial and temporal resolutions (2k, 4k, 8k @ 48, 60, 120 fps). Typically complexities of depth estimation algorithms do not scale as the resolution increases. In addition, depth estimation algorithms often work only on image pairs or on short temporal windows and cannot exploit the coherent information available in a complete video volume belonging to multiple views. Consequently, estimated depth maps often suffer from jagged depth edges (Fig. 1(a)) and/or temporal depth inconsistencies.

(a) (b)

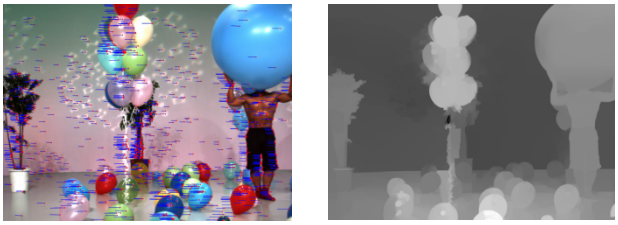
Fig. 1. Images overlaid with associated depth maps: (a) from MPEG's 3D test data set, and (b) estimated with the Feature Flow algorithm. Note the strong alignment of depth edges with texture edges in (b).

Recently, a linear run-time feature propagation method was presented [4], which we will refer to as the *Feature Flow* algorithm. This method estimates the optical flow for a whole video volume and is able to coherently distribute sparse features, like colors, scribbles or depth values, in space and time (Fig. 1(b)). In this work, we expand upon the Feature Flow algorithm for stereo video by requiring bidirectional agreement between left-to-right and right-to-left feature diffusion. In addition we provide an in-depth analysis of how this method performs in the context of depth estimation, something that was missing from the original work.

The rest of the paper is organized as follows: In section 2, we summarize the Feature Flow algorithm. Then in section 3, we discuss applications of the algorithm and introduce an extension, which enforces inter-view consistency during the depth diffusion. In section 4, we report and analyze experimental results and lastly we finalize the paper in section 5 with a brief conclusion.

2. FEATURE FLOW ALGORITHM FOR SPARSE DEPTH DIFFUSION

The Feature Flow algorithm diffuses depth features, which are sparsely defined over a video volume. They are diffused over the whole video volume in space and time by an anisotropic edge-aware diffusion process (Fig. 2). Diffusion stops on color edges in spatial direction; and following the optical flow in temporal direction, it stops at pixel occlusions and disocclusions. Thereby the optical flow is computed simultaneously with the depth by diffusing sparse optical flow vectors.



(a) (b)

Fig. 2. (a) Sparse depth estimates, and (b) diffused depth map.

Sparse depth features are diffused by a process specified by the following general iteration equation:

$$D^{i+1}(x,y,t) \coloneqq \frac{(G_{x.y.t} * D^{i})(x,y,t)}{(G_{x.y.t} * 1_{D^{i}})(x,y,t)} \qquad (1)$$

,where $*$ is a convolution operator, D^0 is a 3-dimensional volume with x, y, and t dimensions that is sparsely populated with depth values, 1_{D^0} is an indicator function having the value 1 for all positions in D^0 where sparse depth values are defined and 0 otherwise, and $G_{x.y.t}$ is a volumetric space and time varying linear filter with finite support (note: if the denominator is zero, then D^{i+1} is defined as zero at the corresponding position). Intuitively, the numerator of the iteration is responsible for the actual diffusion. Depending on the diffusion filter, depth values at new positions may get attenuated with respect to the depth values that were used as input. The denominator takes this into account and normalizes the final depth values for this attenuation, and also restores input depth values used for diffusion.

The Feature Flow algorithm presents an efficient linear-runtime implementation of this diffusion process where the filter $G_{x.y.t}$ is a geodesic filter [4]. A geodesic filter is similar to a bilateral filter [5], however, in contrast it computes inter-pixel distances as geodesic distances in color images [4] where bilateral filter uses Euclidian distance instead. This has the advantage of classifying pixels of similar color as far away when they are separated by a color edge, which stops the diffusion process at color edges. Consequently the diffusion process stops at spatial color edges, and pixel occlusion and disocclusions in temporal direction, which is a highly desired property when diffusing sparse depth values.

Together with the sparse depth D^0, which is used as the input to the diffusion algorithm, often confidence data C is also available. Confidence data describes the reliability of each of the given sparse depth values. The Feature Flow algorithm can take advantage of this information to increase the influence of more reliable depth values on the final depth diffusion result. This is achieved by pre-multiplying all sparse depth values D^0 and indicator values 1_{D^0} with their corresponding confidence before using this data in the diffusion algorithm. This increases the range of diffusion of more reliable depth values and suppresses the influence of less reliable depth values on the final result.

For more details on the implementation of the Feature Flow algorithm the reader is referred to [4].

3. APPLICATIONS AND EXTENSION TO FEATURE FLOW

The main depth-related application areas for the Feature Flow algorithm are depth estimation and depth enhancement. The Feature Flow algorithm uses sparse depth values as input and associated confidence information if it is available. Sparse depth values and associated confidence levels can be acquired from different sources, which depend on the application scenario and the type of data available.

3.1. Depth estimation

Given a stereoscopic video, sparse but reliable disparity and confidence can be estimated from image pairs using state-of-the-art feature tracking and feature matching algorithms. If camera calibration information is available, disparities can then be converted to depth. Nevertheless, Feature Flow can also be applied directly on sparse disparity instead of depth. In our implementation, we estimate disparities using the OpenCV implementation of the Lukas-Kanade feature tracker and combine this data with disparity features obtained with a fast and reliable feature matching method [6].

For multi-view video, when more than 2 views are available and associated camera parameters are given, the additional information can be exploited to estimate more reliable depth features for a given view. In particular, the additional views can provide depth information in regions that are not visible in all views. Additionally the larger number of sparse depth features contributes to the diffusion of more accurate dense depth maps. Experimental results are shown in Sections 4.1 and 4.2.

3.2. Depth enhancement

The Feature Flow algorithm can also be used to enhance depth maps obtained by a different method, e.g. by a conventional image-based depth estimation method or measured by a depth sensor. Sparse or dense depth values can be pro-

vided as input to the Feature Flow algorithm. If confidence information is available, this can also be utilized for explicitly reducing the impact of inaccurate depth values. Corresponding experimental results are shown in Section 4.3

3.3. Inter-view consistency extension to Feature Flow

When applied to stereoscopic or multi-view video, the Feature Flow algorithm diffuses depth values independently in each view. However, if cameras capture the same scene from slightly different positions, which is the usual application scenario, there are dependencies between the depth values of different views. In particular, in a parallel stereoscopic camera setup the depth $D_L(x,y)$ at pixel (x,y) in the left view is equal to $D_R(x+\delta,y)$ in the right view where δ is the corresponding disparity at pixel position (x,y). We extend the Feature Flow algorithm to enforce this property during simultaneous diffusion of depth maps for both views. This is achieved by computing confidence values $C_L(x,y) = C_R(x+\delta,y)$ for each depth value of each view. We define a normalized confidence based on depth differences

$$\bar{C}(x,y) = 1 - \max\{|D_L(x,y) - D_R(x+\delta,y)|,4\}/4 \qquad (2)$$

and then transform these confidences nonlinearly to increase the contribution of strongly matching depth values

$$C_L(x,y) = C_R(x+\delta,y) = 0.3 + 0.7 \cdot \bar{C}(x,y)^5 \qquad (3)$$

The diffusion process for both views is then influenced by this confidence such that the impact of inaccurate depth values on the final diffusion result is reduced. Instead depth values of neighboring more confident pixels are diffused into these regions. That is achieved by pre-multiplying all intermediate depth maps and indicator values in Eq. (1) with the corresponding confidence values before using them for diffusion:

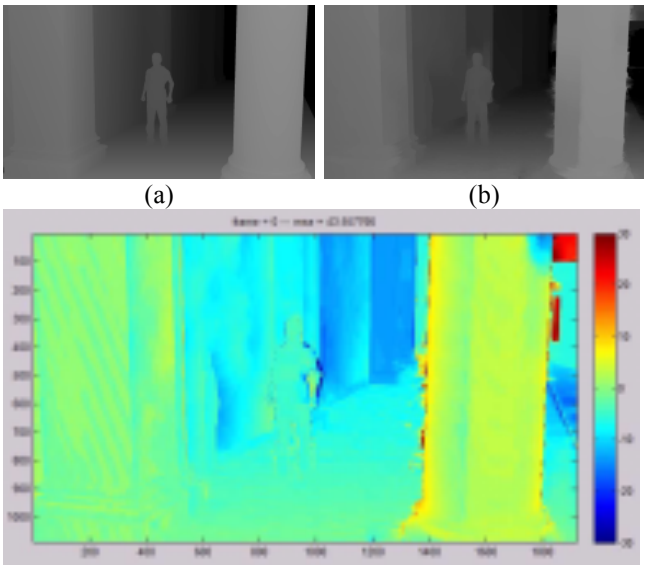
$$D^i_{L,new} := D^i_L \cdot C^i_L \quad \text{and} \quad 1_{D^i_{L,new}} := 1_{D^i_L} \cdot C^i_L \qquad (4)$$

The operator $\cdot$ in Eq. (4) represents a pixel-wise multiplication. A similar update is conducted for the right view simultaneously with the left view. Corresponding experimental results are shown in Section 4.4.

4. EXPERIMENTAL RESULTS

4.1. Ground truth test

In this section we assess the objective accuracy of dense depth maps estimated by diffusion with Feature Flow. We use the deviation from ground truth depth maps in order to measure their accuracy. For this experiment, we use the animated sequence *Undo Dancer,* which is part of the MPEG test set for 3D Video standardization. We use sparse depth values acquired by subsampling the ground truth depth maps by a factor of 8 as the input. In Fig. 3 corresponding depth maps are shown for one frame of the sequence. The results show that the diffusion algorithm is edge-aware, i.e. it is able to estimate sharp depth discontinuities. The root-mean-square error (RMSE) between the depth frames is 6.6.



(a) (b)

(c)

Fig. 3. (a) Ground truth dense depth, (b) depth estimated from sparse ground truth depth, and (c) illustration of depth estimation errors.

4.2. Dense depth estimation by diffusion of estimated sparse depth features

In contrast to the previous ground truth test, in this section we estimate the sparse depth values from multi-view video for a specific view, and diffuse these values with Feature Flow. First, we estimate disparities at good localizable image positions with respect to the other views and convert them to depth. To identify good features we use [7] and employ the disparity estimation methods mentioned in Section 3.1. In Fig. 4 the estimation results are shown for 2, 3, and 5 input views. Note that a larger number of views can also be used to obtain additional depth estimates.

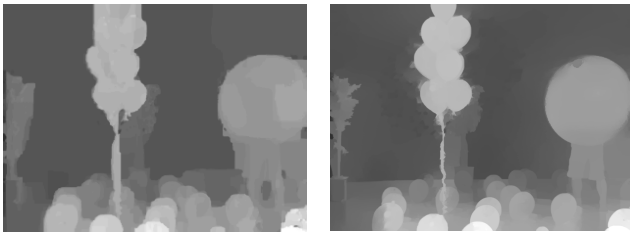
When compared against the ground truth depth map of *Undo Dancer*, we measured RMSEs of 14.4, 8.25, and 8.31 for the depth maps estimated from 2, 3, and 5 input views respectively. There is a saturation when using 3 views and this can be explained by the reduced reliability of the depth features obtained from additional but more distant views. A comparison to the result of Section 4.1 shows that the use of estimated sparse depth leads to an increase of the depth estimation error by 25% in comparison to diffusing error-free sparse depth values.

(a) 2views (b) 3views (c) 5views

Fig. 4. Depth maps estimated by diffusion of sparse depth features, which are estimated from 2, 3, and 5 input views respectively.

4.3. Depth map enhancement

In this section we use the Feature Flow algorithm to enhance existing depth maps, which were estimated by other methods. Fig. 5(a) shows a depth map used by MPEG, which is estimated with method [8]. It suffers from inaccuracies at depth edges and fine structures like the plant on the left or the balloon rope (also see Fig. 1(a)). We apply Feature Flow on sparse depth values obtained by subsampling MPEG depth maps by a factor of 4. As shown in Fig. 5(b), Feature Flow is able to improve the shape of depth discontinuities, which is clearly observable on the fine details of the plant and the round edges of the balloons.



(a) (b)

Fig. 5. (a) Depth map from MPEG's 3D test data set, and (b) corresponding depth map enhanced by Feature Flow.

To demonstrate the benefits of the depth map enhancement, we use the depth maps for depth-aware compositing (Fig. 6). The depth map inaccuracies of the original MPEG depth map lead to occlusions of the logo with background pixels, while the enhanced depth map significantly reduces these errors and keeps edges between the logo and the image content sharp.

(a) (b)

Fig. 6. Depth-aware compositing results with (a) MPEG depth map, and (b) enhanced MPEG depth map.

4.4. Inter-view consistency

To investigate the impact of the inter-view consistency extension (Section 3.3) to the diffusion result of Feature Flow, we estimate the depth maps for both views of a stereoscopic sequence with and without the proposed extension (Fig. 7). The difference image in Fig. 7(c) shows that the extension has its highest impact on regions close to depth discontinuities, which correspond to regions that have no visible correspondences between views due to occlusions. Hence, this shows that enforcing inter-view consistency can help the diffusion of the correct depth values into the occlusion regions.

(a) (b) (c)

Fig. 7. Left depth map estimated without (a) and with (b) inter-view consistency, and (c) the difference image between (a) and (b).

5. CONCLUSION

In this paper we presented an extension to a recently introduced work for linear-runtime feature diffusion. We applied this feature diffusion algorithm to the depth estimation problem, and introduced a novel extension for enforcing inter-view depth consistency when simultaneously estimating depth for two views. We discussed possible application scenarios of the algorithm, and presented and analyzed corresponding experimental results. The results show that sparse depth diffusion is a promising low-complexity method for depth estimation and enhancement of existing depth maps, reaching quality levels suitable for corresponding applications.

6. ACKNOWLEDGEMENT

This research is financially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. ICT-2011-7-287723 (REVERIE project).

7. REFERENCES

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-Time Human Pose Recognition in Parts from a Single Depth Image", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, 2011

[2] A. Smolic, S. Poulakos, S. Heinzle, P. Greisen, M. Lang, A. Hornung, M. Farre, N. Stefanoski, O. Wang, L. Schnyder, R. Monroy, and M. Gross, "Disparity-aware Stereo 3D Production Tools", *Proceedings of European Conference on Visual Media Production (CVMP)*, London, UK, November 2011

[3] J-R. Ohm, D. Rusanovskyy, A. Vetro, and K. Müller, "Work Plan in 3D Standards Development", *JCT3V-B1006, JCT-3V (MPEV|VCEG)*, Stockholm, Sweden, 2012

[4] M. Lang, O. Wang, T. Aydin, A. Smolic and M. Gross, “Practical temporal consistency for image-based graphics applications”, *ACM Transactions on Graphics (SIGGRAPH)*, July, 2012

[5] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, “A Gentle Introduction to Bilateral Filtering and its Applications”, *ACM SIGGRAPH 2007 courses*, 2007

[6] F. Zilly, C. Riechert, P. Eisert, and P. Kauff, “Semantic Kernels Binarized -- A Feature Descriptor for Fast and Robust Matching”, *Proceedings of European Conference on Visual Media Production (CVMP)*, London, UK, November 2011

[7] J. Shi, and C.Tomasi, “Good features to track,” *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on* , pp.593-600, Jun 1994

[8] M.O. Wildeboer, N. Fukushima, T. Yendo, M.P. Tehrani, and M. Tanimoto, “A semi-automatic multi-view depth estimation method”, *Visual Communications and Image Processing 2010*, 77442B-8, 2010