Cost Volume-based Interactive Depth Editing in Stereo Post-processing

Kai Ruhl, Martin Eisemann, and Marcus Magnor
Computer Graphics Lab, TU Braunschweig
Muehlenpfordtstrasse 23, 38106 Braunschweig, Germany
{ruhl,eisemann,magnor}@cg.tu-bs.de [*]

ABSTRACT

In the post production of a stereoscopic visual media production, high-quality depth or disparity maps are essential for a number of workflow components, from initial layer separation, over editing transfers, to the final depth compositing. With automatic depth estimation not being perfect, errors lead to increased manual efforts for the artist, e.g. by requiring additional rotoscoping on the footage. We propose to fix errors in the depth maps instead of in image space, since multiple workflow steps can benefit from improved depth information. Building upon recent advances in discrete real-time stereo estimation algorithms, we guide the artist by integrating the cost volume of a stereo matching estimation into the editing parameters. Our results improve on the good results of automated algorithms and provide an opportunity for user corrections in regions that have only partially correct estimates.

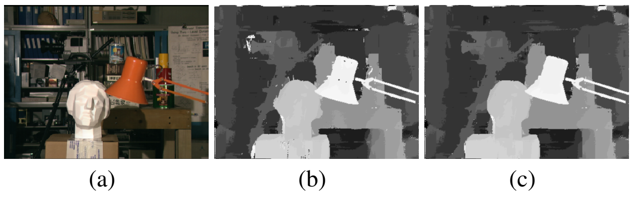

Figure 1: Fully-automated (b) vs. guided (c) depth estimation. The automated approach [11] solves all Lambertian, well-textured, non-occluded areas well. Other areas benefit from approximate user guidance.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Depth cues*; H.5.2 [**Information Interfaces and Representation**]: User Interfaces—*Interaction styles*

Keywords

depth maps, stereo estimation, interactivity, user interface

1. INTRODUCTION

Two schools of thought compete in stereoscopic visual media production: On the one side full stereoscopic capture and editing, aided by tools like e.g. Ocula [22], on the other side single-view capture and generation of the second view with 2D-to-3D conversion, using tools like PFDepth [16]. Both approaches require depth information. In the conversion case, considerable effort has to be spent on rotoscoping and layer assignment, ideally resulting in temporally coherent depth maps used for second eye generation. In the case of stereoscopic capture, the data basis for automated depth map generation via stereo estimation is given, at the cost of increased capturing effort. We concern ourselves with the post-production of stereoscopic footage only.

The quality of depth maps is essential, since they are used for a multitude of purposes. In particular, for each editing session one camera (either left or right; sometimes also a synthesized middle view) is first assigned as principal view. Any grading, insertion of CG and ultimately depth compositing is performed on its frames and subsequently transferred to the other view. Both the selection of layers in the beginning, the transfer of image editing operations and the depth compositing in the end require either high-quality stereo correspondences/depth maps or, barring those, manual corrections in all stages of post-processing. These corrections are usually performed on the captured or intermediate footage, all in image space [15]. Tools for image processing are therefore well developed in order to reduce artist effort.

We propose a complementary approach: Fixing the depth maps instead of fixing the resulting errors in image space. Keeping in mind that working on depth is not as intuitive as working on images, we require only approximate user input in the spirit of the "snapping" behavior found in many common image manipulation programs. We perform a fast initial stereo estimation based on cost volume filtering [11], the top-performing local method in 2011 on the Middlebury index [14], but we do keep the cost volume afterwards. Any artist operations on the depth map can now be performed in "cost blocks" inside that cost volume. In particular, depth estimation can be constrained to consider only depths inside the block, and the z-resolution can be locally increased to obtain smoother results in high-salience areas, thus combining the precision of automated estimation with human scene understanding.

[*]Video and further paper details available under:
`http://graphics.tu-bs.de/publications/ruhl2013cvmp/`

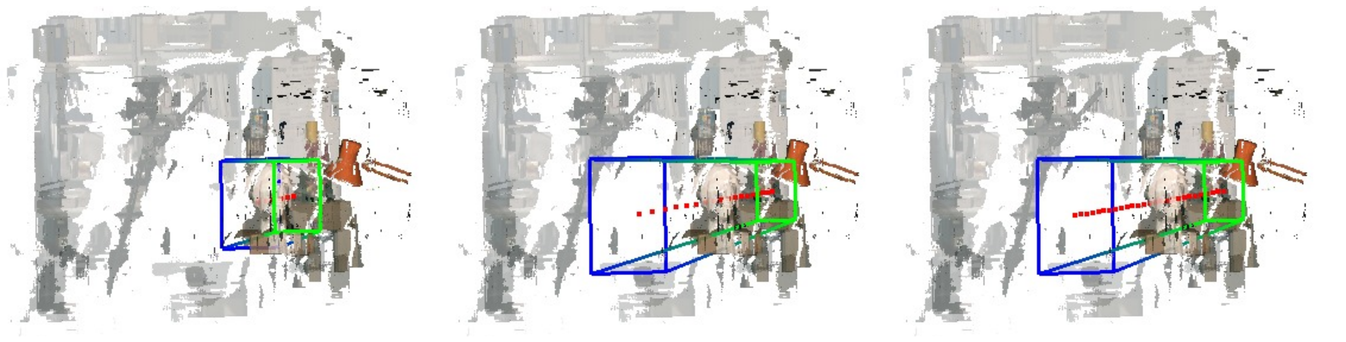
(a) (b) (c)

Figure 2: Cost block within the initial cost volume, visualized as green-blue bounding box, z-layers shown as red dots in the center. (a) cost block with a default size in z-direction (b) the user has increased the z-extent of the cost block (c) the user has increased the number of z-labels within the cost block.

The presented work is only possible thanks to recent advances in near-realtime, GPU-based, local stereo estimation algorithms. Our main contribution is the *interactive editing* of depth maps using *approximate, cost volume-guided*, locally refined depth estimation, useful wherever human scene recognition trumps the capabilities of stereo matching: Occlusions, noise, specularities, lighting differences and other violations of the color constancy assumption.

2. RELATED WORK

Stereo estimation has been an active research area for the last decades and is still improving, with most of the results being compared on the Middlebury stereo data set [14]. Global and local optimizations both abound. Due to the large search range and sharp discontinuities of disparities, discrete methods with a fixed number of labels are more prominent than variational approaches, unlike in e.g. optical flow where movements are usually comparatively smaller (though dedicated long-range optical flow algorithms do exist, e.g. [1]). At the moment, discrete methods outperform continuous ones [14]. Since we consider editing operations, local optimizations which are real-time capable are needed. The recently proposed fast cost-volume filtering approach by Rhemann et al. [11, 7] is the best-performing local method of 2011, and was 9th-ranked overall at that time; it is is based on the guided image filter by He et al. [5], which has linear complexity due to its $O(1)$ boxfilter [3]. Our OpenCL implementation is fast enough for interactivity (e.g. 0.3s on a Tsukuba image pair [14] with a Nvidia GTX 590). Since basically all correspondence estimation algorithms rely on well-behaved scenes, with sufficient texture and Lambertian surfaces which allow unambiguous matches, there are numerous real-world cases where human scene understanding can either support the estimation or correct its errors.

Interactive depth correction and guidance aims at providing the user with intuitive tools to aid the underlying depth estimation algorithm in otherwise difficult and ambiguous cases. In stereo conversion, a 2D video is converted to a 3D video by manually establishing a depth map for the input frames. As "automatic conversion methods are currently not sufficiently robust for general applications" [17], high quality methods are often manual, mostly relying on simple depth painting or adjusting segmented layers of the images, and as a result, are very expensive, with costs of up to $100,000 per minute of converted footage [17]. More sophisticated methods use scribble-based interfaces to draw depth and intelligently interpolate for the remaining pixels [4, 21] or use a set of sparse depth (in)equalities to add depth to cartoons [19].

Given more than a single image per frame, user interaction aids stereo matching by guiding the underlying image correspondence algorithm. Typical ways are specifying sparse ground control points which serve as ground truth to estimate the depth for the remaining pixels [20], providing approximate correspondences which can then be refined by the underlying correspondence algorithm [12, 8, 13], or by removing outliers for a better depth interpolation [2].

Interestingly, user interaction is heavily used in video post-processing tools such as Ocula by The Foundry, e.g. for parallax optimization, color adjustment or detail enhancement. However, they almost always create the necessary disparity maps in an automatic preprocess and provide only relatively simple tools for correction as the assumption is that the precision of the depth map is sufficient or given for synthetic scenes [10] which is not the case for the more complex scenarios we are looking at.

3. COST VOLUME FILTERING

Our initial depth estimation uses a variant of the fast cost volume filtering method [11]. Given left and right views $I_l(\mathbf{x})$ and $I_r(\mathbf{x})$ with pixel coordinates $\mathbf{x} = (x,y) \in \Omega$ and RGB color values in the range $[0,1]$, we strive to attain for each $\mathbf{x}$ an optimal depth $Z_l(\mathbf{x}) \in [d_{\min}, d_{\max}]$, discretized to labels $d \in D = \{d_{\min}, .., d_{\max}\}$ from a set D.

Towards this purpose, we construct a 3-dimensional cost volume $C_l(\mathbf{x},d)$ for the left view I_l. The first two dimensions of C_l are the image size, and the third dimension is the number of depth labels. Each entry within the cost volume is initially a truncated sum of absolute differences (SAD) between the views, using a projection $\pi(\mathbf{x},d)$ from left to right view based on standard epipolar geometry from calibration [18].

$$\begin{aligned} C_l(\mathbf{x},d) = \quad & (1-\alpha)\cdot \quad \min(\tau_1, ||I_l(\mathbf{x})) - I_r(\pi(\mathbf{x},d))||) \\ + \quad & \alpha\cdot \quad \min(\tau_2, \|\nabla I_l(\mathbf{x})) - \nabla I_r(\pi(\mathbf{x},d))\|) \end{aligned} \quad (1)$$

As in [11], we use $\alpha = 0.11$ to favor the color term over the gradient term and $\tau_1 = 0.03$ and $\tau_2 = 0.008$ to favor only very exact matches. With the data term set, we now perform a weighted filtering on C_l

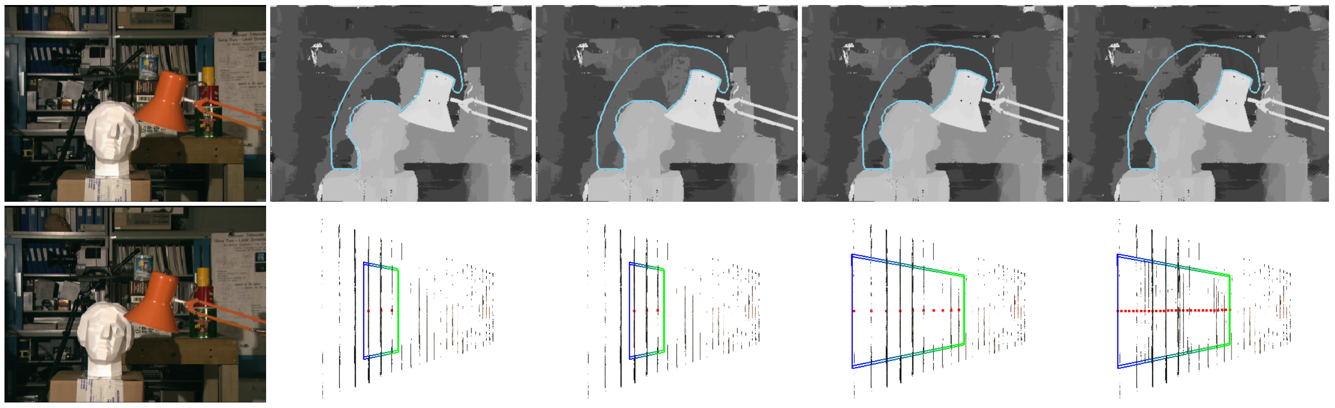

Figure 3: "Tsukuba" scene from the Middlebury data set [14]. (a) left and right view (b) initial depth map [11] and scene layer representation. The selected area has not been properly resolved due to occlusions by bust and lamp (c) re-evaluated cost block. Artifacts around the bust head have vanished, but the book pile (to the left of the lamp) now has wrong depth. (d) increasing the z-extent of the cost block corrects the book pile (e) additionally increasing the number of z-labels refines the region.

to arrive at a smoothed cost volume C'_l:

$$C'_l(\mathbf{x},d) = \sum_{\mathbf{x}' \in N_r(\mathbf{x})} W_{\mathbf{x},\mathbf{x}'}(I_l(\mathbf{x}')) \cdot C_l(\mathbf{x}',d) \quad (2)$$

The filter weights $W_{\mathbf{x},\mathbf{x}'}$ depend on the guidance image I_l [5], similar in spirit to the anisotropic smoothness found in many variational approaches, and are computed on pairs of pixels $(\mathbf{x},\mathbf{x}')$ on a neighbourhood N_r within a filter radius r:

$$W_{\mathbf{x},\mathbf{x}'}(I_l) = \frac{1}{|N_r|} \sum_{k:(\mathbf{x},\mathbf{x}')} (1 + (I_l(\mathbf{x}) - \mu_k)^T (\Sigma_k + \varepsilon U)^{-1} (I_l(\mathbf{x}') - \mu_k)) \quad (3)$$

The mean μ_k and the covariance matrix Σ_k model the local color distribution within the filter window, and εU is a 3×3 diagonal matrix with very small values for numerical stabilization. I_l and μ_k are 3-vectors (the color channels) and Σ_k is a 3×3 matrix of color-channel covariances.

The weights $W_{\mathbf{x},\mathbf{x}'}$ are high when both pixels are on the same side of the mean for correlating color channels and reside within a highly variant region, and low when either the two pixels have different colors or the variance in the region is small (a good gray-image explanation is a also given in [11]). Cost filtering is performed on each depth layer, but not between depth layers since we have no guide in the depth direction.

Runtime is independent of the filter radius r (we use 9–24 depending on image size) when using weighted box filters based on summed area tables, instead of evaluating the weights naively. Our OpenCL implementation uses a tile-based sliding-window variant which works in $O(n)$ on the GPU [6].

Finally, the depth map Z_l is chosen by seeking the depth label with minimal cost per pixel.

$$Z_l(\mathbf{x}) = \arg\min_d C'_l(\mathbf{x},d) \quad (4)$$

4. COST BLOCK EDITING

With the algorithm above (and others [14]) generally generating good initial depth estimates from stereo footage, errors cannot be avoided completely particularly on challenging natural scenes, causing additional artist effort in post-production. Popular causes of artifacts include:

Occluded regions. Objects that are occluded differently in the two views can lose significant overlap, hindering unambiguous matching. In a typical stereo configuration, this happens prominently for any object's left and right edges, which are each only visible in one camera. The closer the object is to the camera, the more pronounced the effect becomes. Automated algorithms cannot hope to recover this error since the information is simply not available. A human user, on the other hand, is able to provide depth information for those non-visible parts by simply guessing the objects's shape.

Ill-textured regions. The majority of stereo algorithms for natural scenes (as opposed to controlled lab settings) rely on the color constancy assumption, which may be violated by lighting or camera sensor differences, noise, specularities, translucent objects, caustics, and so on. This hinders recognition of an object in the other view. Largely uniform or repeating regions in conjunction with different occlusion boundaries in the two views (e.g. columned halls, gratings) are also not solvable with the available information. Again, a human user can assess which objects belong together, and thus distinguish between true and false features.

The question now is how to integrate human scene understanding in a way that minimizes interaction times. Currently, the most common way is to use image editing tools to select a region via rotoscoping or segmentation, and then use stamp, cloning and other tools to assign better depth labels. In our approach, we also start with a lasso selection or mask in 2D image space, but instead of cloning without validation of the resulting depth, we assign a possible range in z-direction, forming a 3-dimensional "cost block" $K_l \subseteq C_l$ as shown in Fig. 2. In the first two dimensions, the cost block is a bounding box around the masked or selected pixels and restricts $\mathbf{x}$ to come from $\Omega' \subseteq \Omega$. In the third dimension, the cost block is centered around the median depth of the selection $med(Z_l(\mathbf{x}'))$ with $\mathbf{x}' \in \Omega'$ (other strategies are also possible) and has some extent that

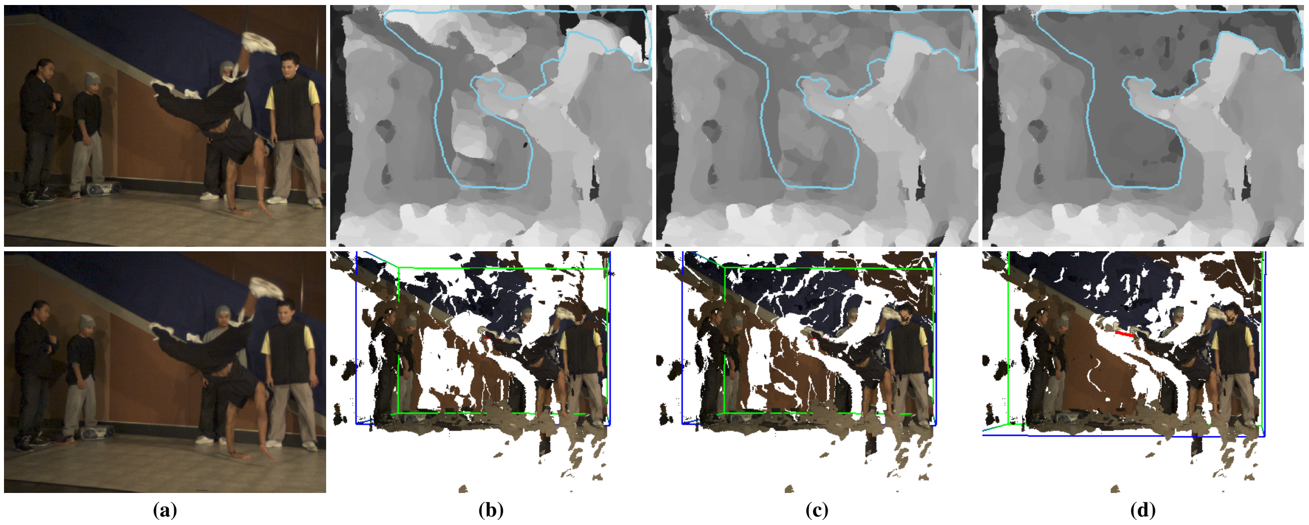
Figure 4: **"Breakdancer" scene from Zitnick et al. [23], a multi-view recording with cameras around 30 cm apart. Here we use only two adjacent cameras. (a) left and right view (b) initial depth map [11] and 3D scene representation. Due to noise and low-textured regions, large parts of the wall are ill-matched. (c) a background region is selected and evaluated with the default z positioning. (d) the cost block is further shifted to the back. Matching the wall improves.**

restricts d to come from $D' \subseteq D$. The initial extent in z-direction can either be a fixed parameter or some percentile of $Z_l(\mathbf{x}')$.

In a 3D view of the scene, Fig. 2, both the current depth estimate and the cost block K_l are visualized. An artist can now shift the cost block along the z-axis until the estimation "snaps" the depth to the most plausible position. With each step, $Z_l(\mathbf{x}')$ is locally re-evaluated for all pixels in the mask, providing visual feedback in real-time. The z-extent of the cost block can be widened if objects in the selected area do not fit into it, or narrowed to eliminate superfluous estimates. As a third option, the depth label subset D' can be subdivided to include more depth labels, even to the point where $|D'| > |D|$. This increases the accuracy of z-values but takes longer to compute when the cost block is large.

It should be noted that using cost blocks does not solve the problem of ill-defined regions in a mathematical sense. Instead, it merely reduces the effect of incorrect cost computation: In a narrowed set of labels d', the cost block $K_l(\mathbf{x}', d')$ merely evaluates to a more plausible depth $Z_l(\mathbf{x}')$, since a search window $I_l(\mathbf{x}')$ has a much lower probability of being matched to a randomly low-cost window $I_r(\pi(\mathbf{x}', d'))$. In the worst case when no support information can be found within filter radius r to be aggregated into the filter weights $W_{\mathbf{x},\mathbf{x}'}$, the final depth Z_l will be essentially random, but still within the bounds of D'. When implausible, this would require the artist to narrow down the z-extent of K_l to a thin slice.

In essence, when thought of in the scope of the entire depth map, the user interaction cuts away large superfluous blocks from the cost volume, rather than refining the stereo matching itself.

5. RESULTS

We test the performance of our method both on a classic Middlebury stereo scene [14] and on more challenging natural scenes involving wider baselines and more low-textured and repeating regions. The latter are multi-view data sets from which we use only two adjacent views. All examples use epipolar geometry instead of rectified footage to avoid re-sampling and support general recording setups. Runtimes are given for an Nvidia 590 GTX graphics card, with all computation performed in OpenCL. Editing times are in the order of seconds to minutes and consist of lasso selections and cost block adjustments.

Fig. 3 shows the Tsukuba scene from the Middlebury stereo evaluation data set [14]. Though well-textured, it contains a number of repeating regions and suffers from poor lighting. Initial depth estimation takes 0.3 seconds with 24 labels on 384×288 frames. The automated cost volume filtering [11] solves the scene generally well, with the exception of occlusions around the bust head, lamp, and camera, and some spurious artefacts due to low lighting. An area behind the lamp and bust is selected with a lasso (b) and re-evaluated. While the occlusion artifacts vanish, the default z-extent has removed the book pile on the right table from the most probable cost volume region (c). Adjusting the cost block z-extent re-includes the depth region (d). Finally, an increase in the number of depth labels refines the layering.

Fig. 4 shows the breakdancer scene by Zitnick et al. [23]. Initial depth estimation takes 1.9 seconds with 91 labels on 1024×768 frames. The wall behind the dancers feature little texture and the recordings are noisy, leading to ambiguous matching (b), which is improved by the default cost block evaluation (c). Shifting the wall further to the back reveals more of its true shape, namely its inclination towards the z-direction (d). In such cases, the number of z-labels has to be increased in order to allow a smooth transition.

Fig. 5 shows the Sassichan1 scene. Initial depth estimation takes 2.2 seconds with 150 labels on 1024×540 frames. The automated cost volume filtering [11] is almost to be considered a failure case due to the wide baseline coupled with many repeating, fine-structured patterns over a large number of depth labels (b). Even with the cost block constraints on the floor, artifacts are reduced but some remain (c). Shifting the back wall deeper removes many spurious artifacts from the front of the volume (d). Further coarse editing, e.g. on the bike stands, improves the result little by little

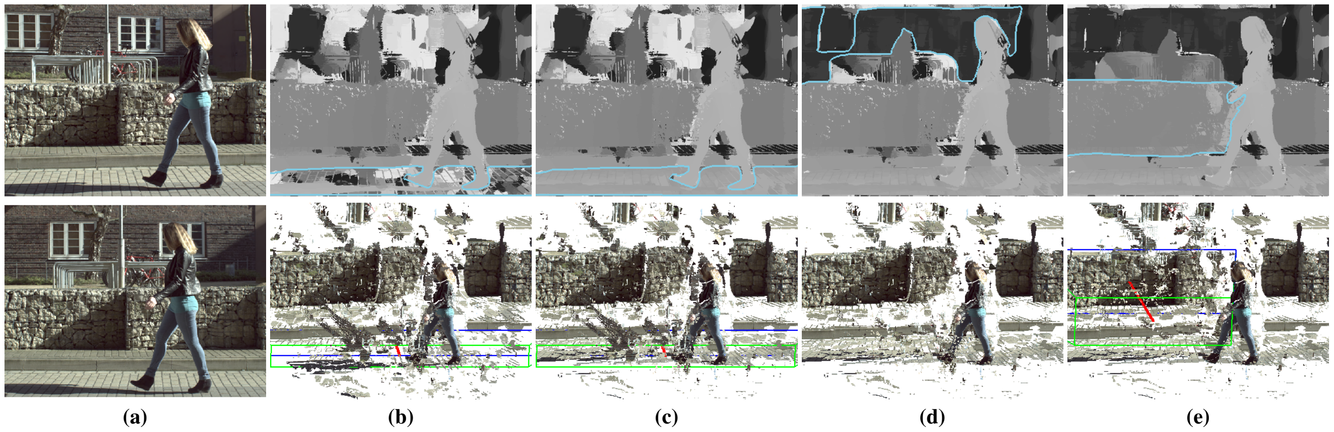

Figure 5: "Sassichan1" scene with 1 m interocular distance. (a) left and right view (b) initial depth map [11] and 3D scene representation, with extremely many ill-matched regions due to repeating patterns, fine structures, and high number of labels. (c) cost block repair for the floor. (d) cost block repair for the background building. (e) cost block repair of further regions.

(e). The stone wall and the back house wall could also be well approximated by a simple plane, but not e.g. the bike stand because it has some z-extent.

Please see the accompanying video for more details.

6. DISCUSSION

The results show that automated cost volume filtering [11] produces estimates ranging from very good for well-behaved scenes such as Tsukuba to near-failure cases for complex natural scenes such as Sassichan1. In all cases, shifting cost blocks around to impose constraints on the cost volume improves the result. Like the stereo matching that underlies the user interaction, more well-behaved scenes benefit more since the cost volume guidance is of higher quality. When the local depth estimation quality degrades too much, the guidance approach comes to its limits, and it is advisable to switch to pure image-based depth map painting.

Computational cost for guidance is generally low, since both the number of pixels and the number of labels is considerably less when compared to the full cost volume. In our experience, increasing the number of labels has little influence on runtime, because it is usually outweighted by the influence of the number of pixels. An interesting alternative would be to use additional variational optimization for depth refinement [9], since its oversmoothing effect can be neutralized by selecting coherent regions. Furthermore, our approach currently supports only frame-by-frame edits. Integrating it into a keyframe-based framework to propagate the depth map corrections would be a further way to save artist effort.

7. CONCLUSION

We presented an interactive approach to edit depth maps created from two synchronized input views. By using cost blocks cut from the stereo matching cost volume as a guide, an artist can approximately indicate the desired depth, and the exact depth is determined via algorithmic refinement. The user interaction effectively cuts away large parts of the cost volume, reducing the probability of erroneous matches. Our approach improves on the results of automated algorithms, and can be used with any real-time local stereo estimation method.

8. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 256941, Reality CG. We would also like to thank Rhemann et al. [11] for making the Cost Volume Filtering source code openly available.

9. REFERENCES

[1] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 0:41–48, 2009.

[2] G. Chaurasia, O. Sorkine, and G. Drettakis. Silhouette-aware warping for image-based rendering. In *Proceedings of the 22nd Eurographics conference on Rendering*, pages 1223–1232, 2011.

[3] F. C. Crow. Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, pages 207–212, New York, NY, USA, 1984. ACM.

[4] M. Guttmann, L. Wolf, and D. Cohen-Or. Semi-automatic stereo extraction from video footage. In *ICCV*, pages 136–142, 2009.

[5] K. He, J. Sun, and X. Tang. Guided image filtering. In *Computer Vision–ECCV 2010*, pages 1–14. Springer, 2010.

[6] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother. Real-time local stereo matching using guided image filtering. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[7] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(2):504 – 511, 2013.

[8] F. Klose, K. Ruhl, C. Lipski, and M. Magnor. Flowlab - an interactive tool for editing dense image correspondences. In *Proc. European Conference on Visual Media Production (CVMP) 2011*, pages 1–8, Aug. 2011.

[9] S. Kosov, T. Thormählen, and H.-P. Seidel. Accurate real-time disparity estimation with variational methods. In *Advances in Visual Computing*, pages 796–807. Springer, 2009.

[10] F. Pitié, G. Baugh, and J. Helms. Depthartist: a stereoscopic 3d conversion tool for cg animation. In *Proceedings of the 9th European Conference on Visual Media Production*, pages 32–39, 2012.

[11] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3017–3024. IEEE, 2011.

[12] D. Ring and A. Kokaram. User-assisted feature correspondence matching. In *Proceedings of the 2009 Conference for Visual Media Production*, pages 214–219, 2009.

[13] K. Ruhl, B. Hell, F. Klose, C. Lipski, S. Petersen, and M. Magnor. Improving dense image correspondence estimation with interactive user guidance. In *Proc. ACM Multimedia 2012*, pages 1129–1132. ACM, 2012.

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

[15] M. Seymour. Art of digital 3D stereoscopic film. *FXguide*, 2008.

[16] M. Seymour. Pixel Farm's PFDepth: exclusive first look. *FXguide*, 2012.

[17] 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. In *Proceedings of the 2011 Conference for Visual Media Production*, pages 165–173, 2011.

[18] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[19] D. Sýkora, D. Sedlacek, S. Jinchao, J. Dingliana, and S. Collins. Adding depth to cartoons using sparse depth (in)equalities. *Computer Graphics Forum*, 29(2):615–623, 2010.

[20] L. Wang and R. Yang. Global stereo matching leveraged by sparse ground control points. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3033–3040, 2011.

[21] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross. Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 47–54, 2011.

[22] L. Wilkes. The role of ocula in stereo post production. *The Foundry, Whitepaper*, 2009.

[23] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. A. J. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, 2004.