Depth Map Design and Depth-based Effects With a Single Image

Jingtang Liao[*] Shuheng Shen[†] Elmar Eisemann[‡]

Delft University of Technology

Abstract

We present a novel pipeline to generate a depth map from a single image that can be used as input for a variety of artistic depth-based effects. In such a context, the depth maps do not have to be perfect but are rather designed with respect to a desired result. Consequently, our solution centers around user interaction and relies on a scribble-based depth editing. The annotations can be sparse, as the depth map is generated by a diffusion process, which is guided by image features. Additionally, we support a variety of controls, such as a non-linear depth mapping, a steering mechanism for the diffusion (e.g., directionality, emphasis, or reduction of the influence of image cues), and besides absolute, we also support relative depth indications. We demonstrate a variety of artistic 3D results, including wiggle stereoscopy and depth of field.

Index Terms: I.1.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

1 Introduction

Representing 3D content on a standard 2D display is difficult. This topic has been of much interest to artists, who learned over centuries how to use effective pictorial cues to enhance depth perception on a canvas. With computer displays, it is possible to add animation to still images to increase depth perception. The *Ken Burns effect* is a simple example that combines zooming and panning effects and is widely used in screen savers. For television and movie productions, this technique can be obtained by a rostrum camera to animate a still picture or object. In its modern variant, the foreground is often separated from the background, which requires a rudimentary segmentation. The resulting parallax effect leads to a strong depth cue, when the viewpoint is changing (Fig. 1). Today, with the help of image-manipulation software, such effects can be easily produced. However, the picture elements are only translated, which is very restrictive and leads to a reduced effectiveness.

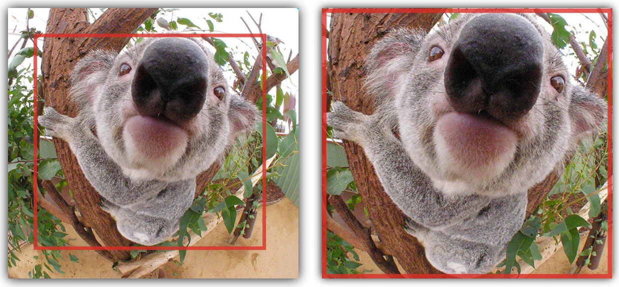

Figure 1: Ken Burns effect. Panning and zooming on still images.(Image source: ©Lone Pine Koala Sanctuary - `www.koala.net`.)

When several views are available, image-based view interpolation [24] is more general. The perceived motion of the objects helps in estimating spatial relationships. Nonetheless, these techniques often require a special acquisition setup or a carefully produced input. Wiggle stereoscopy can be seen as a particular case of view interpolation, which simply loops left and right images of a stereo pair and can result in a striking parallax perception despite its simplicity (Fig. 2). These techniques all avoid special equipment, e.g., 3D glasses, and they even work for people with limited or no vision in one eye.

Alternatively, it is possible to use a single input image and warp it based on a depth map to produce stereo pairs. Yet, computing depth maps for a monocular image is an ill-posed problem. While important advances have been made [8, 12, 26, 27], the methods are not failsafe. Furthermore, many depth-based effects require the possibility for manual adjustments, such as remapping the disparity range of stereoscopic images and video in the production, live broadcast, and consumption of 3D content [13], or to modify a depth-of-field effect in an artistic manner [15], which is why we focus on a semi-automatic solution.

In this paper, we propose a new framework to generate a depth map for a single input image with the goal of supporting artistic depth-based effects to illustrate the spatial information in the image. We build upon the insight that a depth map does not have to be perfect for such applications but should be easily adjustable by a user, as this option allows fine-tuning of the artistic effect. Our results are illustrated with a variety of examples, ranging from depth-of-field focus control to wiggle stereoscopy. Additionally, with such a depth map at hand, it is possible to produce image pairs for 3D viewing without (e.g., via establishing a cross-eyed view) or with specialized equipment (e.g., stereo glasses).

Our approach builds upon the assumption that depth varies mostly smoothly over surfaces and only exhibits discontinuities, where image gradients also tend to be large. In consequence, we follow previous work and require only coarse annotations, such as sparse scribbles [10, 20, 30] or points [21]. These annotations form hard constraints in an optimization system that leads to a diffusion process, taking the image content into account. We focus on the control of this process and our method offers ways to influence the result via local and global constraints. Defining relative depth differences, a non-linear depth diffusion by assigning a strength to scribbles, or privileged diffusion directions are examples. We ensure that all these elements can be formulated in a linear optimization problem

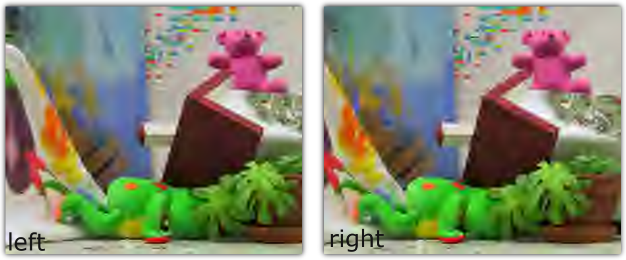

Figure 2: Wiggle stereoscopy. Looping a left/right image pair [28].

[*] j.liao@tudelft.nl
[†] s.shen-1@student.tudelft.nl
[‡] e.eisemann@tudelft.nl

16-19 May, Edmonton, Alberta, Canada

to ensure a fast solving step. We additionally show a selection of effects in our results.

Overall, our work makes the following contributions:

- A fast depth-map creation solution from a single image;
- Various additional tools to refine the depth map;
- A selection of effective effects, including wiggle stereography.

2 Related work

Depth perception helps us perceive the world in 3D using various depth cues, classified into binocular and monocular cues. In an image, we typically encounter monocular cues — depth information that can be perceived with just one eye. Motion parallax [11], size, texture gradient [2], contrast, perspective, occlusion [23], and shadows [4] are examples of these. Motion parallax and occlusion are particularly strong [6]. Parallax arises due to the non-linear displacement relative to the depth when shifting the viewpoint of a perspective projection. In order to add such an effect, one can warp an image based on a depth map, which associates to each pixel the distance to the camera.

A depth estimation for a single image is a well-known problem in computer graphics and computer vision that received much attention. Recent approaches [8, 12, 26, 27] are based on learning techniques. These approaches establish an automatic conversion approach. The quality depends on the variety of the training data set and provided ground-truth exemplars. Additionally, in practice some manual segmentation is needed and the methods are not failsafe, as problematic elements are quite common (e.g., the reflections in a mirror or a flat image hanging on the wall). Even if an accurate depth is obtainable, it is not always optimal for artistic purposes [7, 13], which is our focus.

Depth from defocus (DFD) is another approach where the amount of blur in different areas of a captured image is utilized to estimate the depth [25]. Methods for single DFD from conventional aperture are usually based on such assumptions. Aslantas et al. [1] assumed defocus blur to be the convolution of a sharp image with a 2D Gaussian function whose spread parameter is related to the object depth. Lin et al. [19] designed aperture filters based on texture sharpness. Zhu et al. [31] took smoothness and color edge information into consideration to generate a coherent blur map for each pixel. A disadvantage of single image DFD methods is that they cannot distinguish between defocus in front and behind the focal plane. Coded-aperture setups [16] address this issue by using a specially-designed aperture filter in the camera. Sellent et al. [29] proposed an asymmetric aperture, which results in unique blurs for all distances from the camera. All these coded latter methods require camera modifications and have limitations regarding precision and image quality.

In our approach, the depth map will be designed by the user in a semi-automatic way. Hereby, also artistic modifications are kept possible. Early interactive techniques [5, 18], and their extensions [14], focused on scenes containing objects with straight edges and relied on user-provided point- and straight-line indications to reconstruct a 3D model. In general, the use of edges is a good choice, as many natural scenes consist of piece-wise patches separated by object boundaries. Gerrits et al. [10] introduced a stroke-based user iterative framework in which users can draw a few sparse strokes to indicate depths as well as normals. Their technique optimizes for a smooth depth map in an edge-aware fashion, which is typically applied to photographs containing large planar geometry. Lin et al. [20] focused mainly on recovering depth maps for 2D paintings, where the 2D paintings have to be segmented into areas based on input strokes and the depth values are only propagated locally based on the color difference. Wang et al. [30] proposed a work flow for stereoscopic 2D to 3D conversion, where users draw only a few sparse scribbles, which together with an edge image (computed from the input image) propagate the depth smoothly, while producing discontinuities at edges. Similarly, Lopez et al. [21] used points instead of scribbles to indicate depths and made additional definitions available for the user, such as depth equalities and inequalities, as well as perspective indications. Our work follows similar principles, but offers additional possibilities with the goal of a direct application to artistic depth-based effects. Our work builds upon depth propagation via a diffusion process, similar to diffusion curves [22] and their extensions [3].

3 Our Approach

Our approach is illustrated in Fig. 3; given a single image as input, e.g., a photograph or even a drawing, we seek to create a depth map and show how it can be used as input to various depth-based effects. Consequently, we first describe the depth-map generation via the diffusion process, then discuss additional tools provided to the user (Sec.3.1), before illustrating our implementation of various depth-based effects (Sec.3.2). Finally, we discuss the results (Sec. 4) before concluding (Sec. 5).

3.1 Depth Map Estimation

The basic input by the user are a few depth indications in form of scribbles. These scribbles will be considered hard constraints that should be present in the final depth map. The rest of the depth map will be solved via an optimization procedure. In order to ensure acceptable performance, we cast our problem into a constrained linear system. This initial setup is identical to Diffusion Curves [22], based on Poisson diffusion, except the scribbles take the role of the diffusion curves.

Poisson Diffusion

Given the image $I := \{I_{i,j} \mid i \in 1...w, j \in 1...h\}$, where $I_{i,j}$ are brightness or color values at pixel (i, j), we aim at creating a depth map $D := \{D_{i,j} \mid i \in 1...w, j \in 1...h\}$, given a set of scribbles with associated values $\{S_{i,j} \mid (i, j) \in I\}$ on scribbles, where $I \subseteq \{1...w\} \times \{1...h\}$. The depth map D is then implicitly defined:

$$\Delta D = 0$$
$$\text{subject to:} D_{i,j} = S_{i,j}, \forall (i, j) \in I.$$

where Δ is the Laplace operator. The discretized version for a pixel (i, j) of the first equation is:

$$4D_{i,j} - D_{i+1,j} - D_{i-1,j} - D_{i,j+1} - D_{i,j-1} = 0 \qquad (1)$$

The depth map can, thus, be constructed by solving a constrained linear system. A result is shown in Fig. 4.

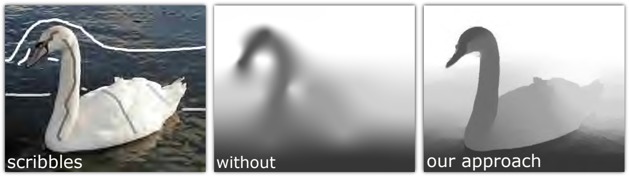

Figure 4: Depth estimation from scribbles. Scribble input (left), only using the scribble input results in a smooth depth map lacking discontinuities (middle), by involving the input image gradients, the depth propagation is improved (right). (Image source: Wikimedia Commons)

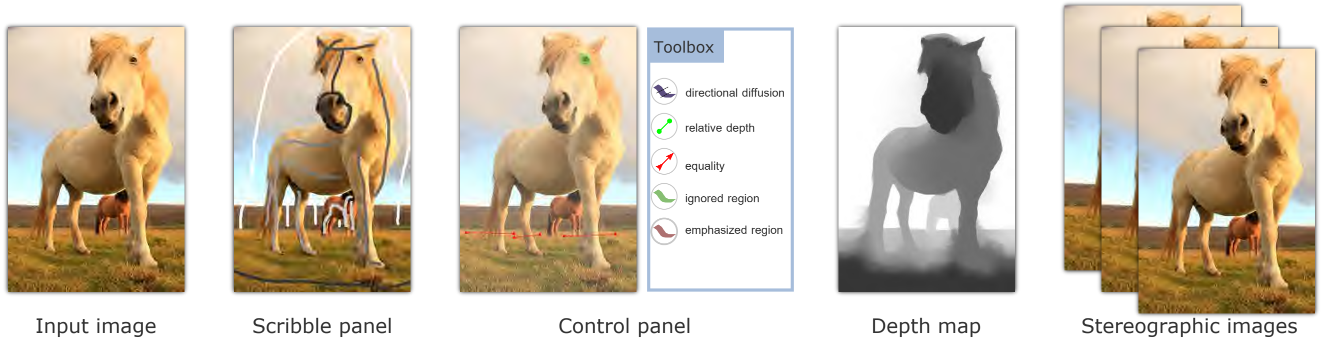

Figure 3: Overview: From left to right, starting from a monocular image, the user draws scribbles, which spread via a diffusion process to define a resulting depth map. The interface allows for constant or gradient-color scribbles, the definition of a diffusion strength, brushes to ignore or emphasize gradients in regions or Bézier curves to direct the diffusion process. Further, relative depth differences and equalities can be annotated. (Image source: ©Robert Postma/Design Pics), used with permission.

Anisotropic Diffusion

Eq. 1 implies that each pixel's depth is related to its four neighbor pixels in an equal way. Consequently, the map is smooth and free of discontinuities. Nonetheless, discontinuities can be crucial for depth effects at object boundaries. Hence, we want to involve the image gradients in the guidance of the diffusion process and, basically, stop the diffusion at object boundaries. To this extent, we will rely on the difference of neighboring input-image pixels to steer the diffusion, transforming the Laplace equation into a set of constraints. For a pixel k and its 4-pixel neighborhood $N(k)$, we obtain:

$$\sum_{l\in N(k)} \omega_{kl}(D_k - D_l) = 0, \tag{2}$$

where ω_{kl} is the first order difference for the two neighboring pixels $\omega_{kl} = \exp(-\beta|I_k - I_l|)$. At the border of an object, ω_{kl} is often close to 0 because the pixel values typically differ. In consequence, the impact of the constraint is reduced, which, in turn, relaxes the smoothness condition. Hence, depth discontinuities will start to occur at boundaries. Fig. 4 (right) shows the effect of integrating the image gradient.

Ignored-gradient Region While object boundaries are useful barriers for the diffusion, some gradients (e. g., shadows, reflections etc.) in the image may introduce unwanted depth discontinuities. For example, Fig. 5 exhibits shadowed areas, which produce strong gradients that lead to artifacts on the floor, although it should actually have been smooth. To avoid such issues, we provide the user with the possibility to use a simple brush to annotate regions where gradients should be ignored. For pixels which were selected in this way, the corresponding diffusion constraint would change back to Eq. 1. Fig. 5 shows a comparison with and without this annotation.

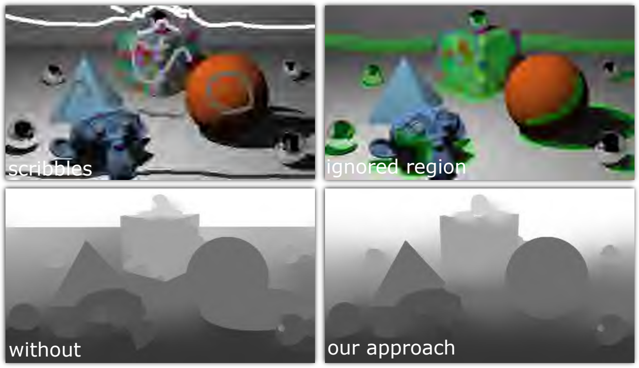

Figure 5: Ignored-gradient region. Shadows etc. introduce unwanted large gradients, which hinder the depth diffusion and lead to discontinuities. Using the ignored-gradient region brush, these gradients can be excluded from the depth derivation. (Image source: [17])

Emphasized-gradient Region Contrary to the previous case, depth discontinuities might also need a boost in other areas. To this extent, we allow the user to emphasize gradients. The gradient of the brushed pixels is enlarged by a scaling factor (two in all examples). This tool is of great use when refining depth maps (Fig. 6), as it helps to involve even subtle gradients when needed.

Directional Guidance While the previous methods stop or accelerate diffusion, its directionality remains unaffected. Still, in some cases, the intended diffusion direction might be relatively clear, e.g., along a winding road to the horizon. In order to integrate a directional diffusion in the linear equation system, we let the user provide a directional vector field and remove the gradient constraints orthogonal to the indicated direction, following [3]. For an arbitrary direction $\boldsymbol{d} := (\cos\theta, \sin\theta)$, the derivative of an image I along direction $\boldsymbol{d}$ is given by $\nabla I\boldsymbol{d}$. In consequence, the constraints for pixel (i, j) are replaced by:

$$\cos\theta\cdot\omega_{ijx}(D_{i+1,j} - D_{i,j}) - \sin\theta\cdot\omega_{ijx}(D_{i,j+1} - D_{i,j}) = 0 \tag{3}$$

where $\omega_{ijx} = \exp(-\beta|D_{i+1,j} - D_{i,j}|)$ and $\omega_{ijy} = \exp(-\beta|D_{i,j+1} - D_{i,j}|)$. Here, the diffusion will then only occur along direction $\boldsymbol{d}$.

To define the vector field, we first ask the user to indicate the region, where to apply the directional guidance with a brush. To specify the directions, the user can then draw Bézier curves. The tangent of a point on the curve is defining the diffusion orientation that is to be used for the underlying pixel. To propagate the information from the Bézier curves to the entire region, we let the direction vector itself be diffused over the marked region using Eq. 1. To avoid singularities, we diffuse the cosine and sine values of the direction and normalize the result after diffusion. Fig. 7 (left, top) shows the curves and brushed region in which the diffusion is guided, as well as the diffused direction information for each pixel of the region (Fig. 7 (right,top)).

It is possible to reduce the directionality by adding an additional constraint for the direction orthogonal to the diffusion direction (i.e., $\boldsymbol{d} := (-\sin\theta, \cos\theta)$). If we do not apply a scale factor to the constraint, the resulting diffusion would go back to a uniform diffusion. The scale factor could be chosen by the user, but we also propose a default behavior based on the image content. The idea is that the user indicates a direction because it is connected to the input image's content. We thus analyze the input image's gradient,

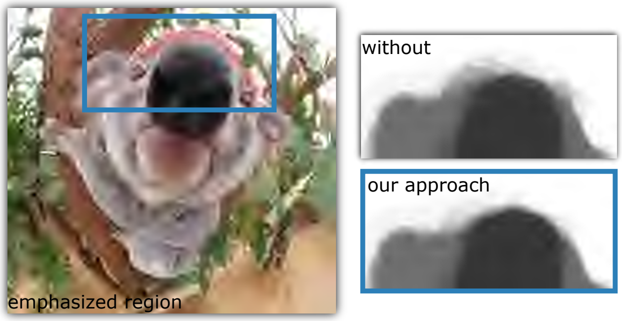

Figure 6: Emphasized-gradient region. Weak gradients can be enhanced to induce depth discontinuities. Here, it ensures a better separation between the foreground and background.

and compute the angle θ between gradient and provided diffusion direction to derive an adaptive scale factor $1-|\cos\theta|$.

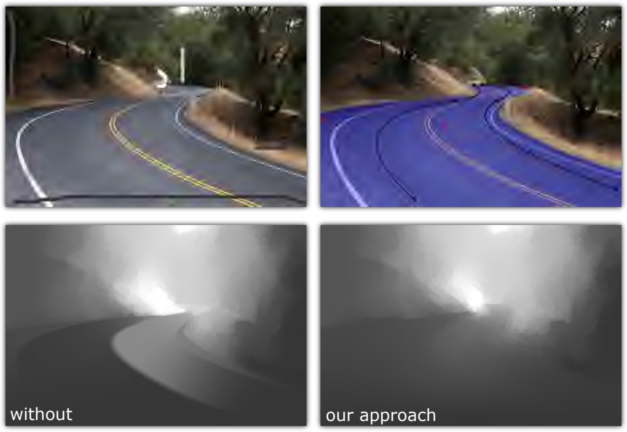

Figure 7: Diffusion guidance. Users brush the region and draw the direct curves to define the direction in which they are interested in. Our approach can direct the diffusion mainly happens in this direction. (Image source: `http://maxpixel.freegreatpicture.com`)

Non-linear Depth Mapping

Perspective projection can result in a non-linear depth mapping, e.g., via foreshortening. For these situations, we want to provide the user with a way to influence the diffusion strength. Following [3], diffusion strength can be added by introducing an additional component to the vector value that is diffused; besides a depth value d, we will have a strength α. For two such elements $(d_1,a_1),(d_2,a_2)$, a mix is assumed to yield:

$$\frac{\alpha_1 d_1+\alpha_2 d_2}{\alpha_1+\alpha_2}. \tag{4}$$

The higher the strength, the higher the influence of the associated depth value on the final result. This equation directly extends to many depth values:

$$\frac{\sum \alpha_i d_i}{\sum \alpha_i} \tag{5}$$

This insight makes it possible to formulate this behavior in our linear optimization system — we now solve for two maps, containing values of type αd and α. Once the diffusion converged, we can divide the first map's values by the second, establishing the result of Eq. 5. Fig. 8 shows the result of assigning different depth strengths to influence the depth-map generation.

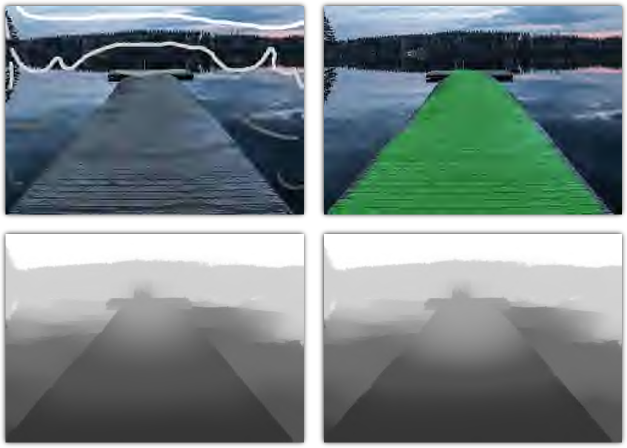

Figure 8: Non-linear depth mapping. Assigning a strength to different scribbles can be used to influence the diffusion speed. (Image source: `https://pixabay.com`)

Equal and Relative Depths

It can be useful to indicate that two objects are located at the same depth, without providing an absolute value. Given our constraint system, this goal can be achieved by adding a constraint of the form $D_k = D_l$, similar to [3]. This possibility is quite useful for images containing symmetric features, as shown in Fig. 9, where pixels on the pillars, which are at the same depth, can be linked. Many pixels can be connected at the same time.

We also introduce a new feature to describe relative depth relationships; let D_1, D_2,D_3 and D_4 be four locations in the depth map. If the user wants the distance of D_1 to D_2 equal to the distance of D_3 and D_4, we can add the constraint $D_1-D_2=D_3-D_4$. For the pillar example, the relative depth indications can be used to ensure the equivalent distances between pillars. Again, this solution can be extended to multiple relative points.

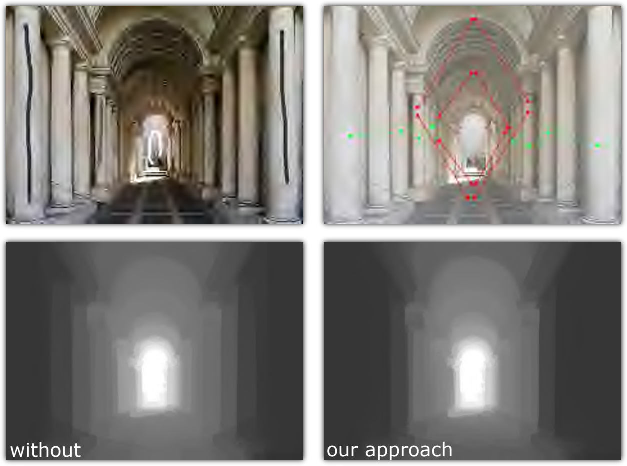

Figure 9: Depth equality and relativity We connect depths from different places together via depth equality and relativity to globally influence the depth estimation. (Image source: wikipedia)

Global Adjustments

Our framework offers the possibility to globally adjust the resulting depth map. We provide the user with a mapping curve, similar to a gamma curve, to specify a non-linear remapping. We use an interpolating spline, adjusted via control points.

3.2 3D Effects

In this section, we illustrate a few of the 3D effects that can be introduced in the input image, when relying on the derived depth map, whose values we assume normalized between zero and one.

Color-based Depth Cues

Given the depth map, we can easily add an aerial perspective to the result. An easy solution is to apply a desaturation depending on the distance as shown in Fig. 10. Alternatively, we can convert the distance to a fog density and apply it as an overlay on the image.

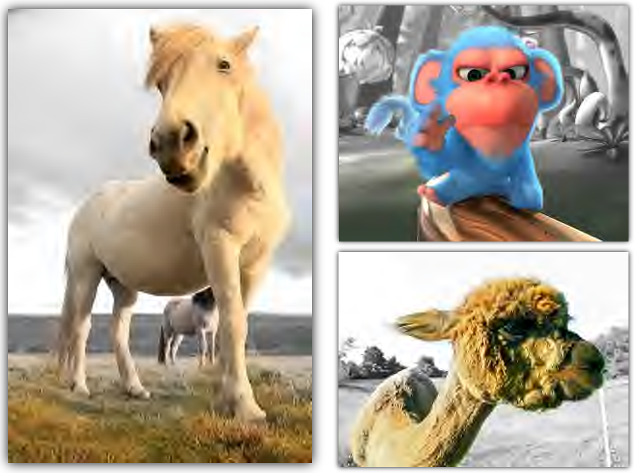

Figure 10: Distance-based desaturation.

Depth-of-Field Effects

It is possible to simulate lens blur to refocus on different parts of the scene. Fig. 11 (right) shows an example.

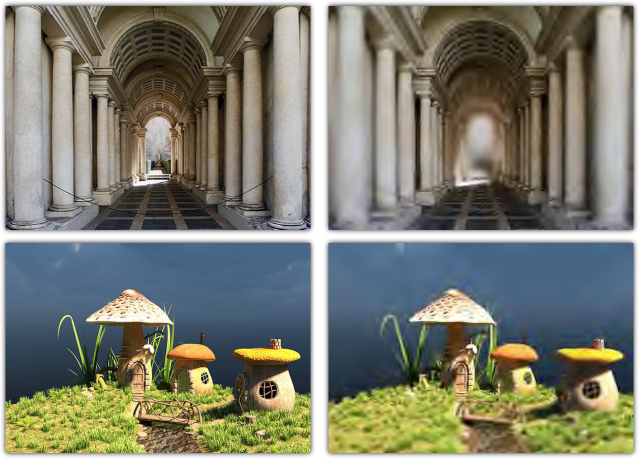

Figure 11: Image refocusing based on the depth values.

Stereographic Image Sequence

When adding motion parallax to the input image, the resulting images can be used as stereo pairs, for wiggle stereoscopy, or even as an interactive application that can be steered with the mouse position. Please also refer to our supplemental material for looping videos, of which a few frames are shown in Fig. 12.

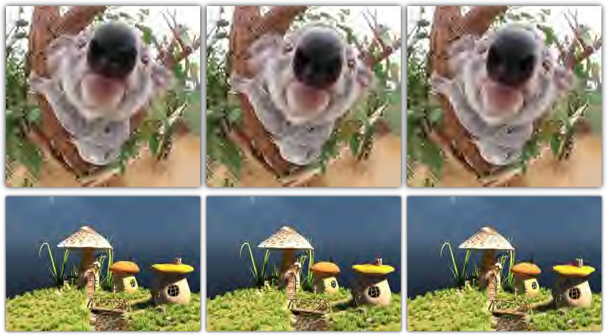

Figure 12: Examples of looping videos.

For a given displacement direction γ and a maximum pixel traversal distance S, the newly-derived image N, in which nearer pixels are shifted more strongly than far-away pixels, is given by:

$$N(i+(1.0-d_{ij})\cos(\gamma)S, j+(1.0-d_{ij})\sin(\gamma)S) := I(i,j)$$

Unfortunately, the definition of N is imperfect, as several pixels may end up in the same location or holes occur (no pixel projects to this location). The first case can be easily solved; as our motion direction does not affect depth, we can, similar to a depth buffer, keep the reprojected pixel with the smallest depth value. To address holes, we rely on a post-processing step. We search from a hole in N along the opposite direction of γ, until we find the first non-hole pixel. Its value is then copied over to the hole location. Fig. 13 shows the comparison with and without hole filling.

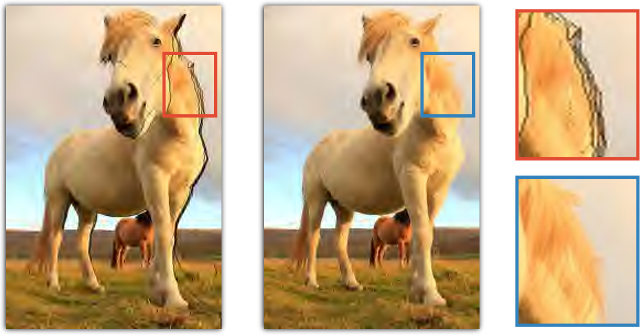

Figure 13: Hole filling. Holes due to reprojection (left) are filled (right).

Artistic Effects

The derived depth map can be used to apply special artistic filters. First, we illustrate the use for movement and show a special rotation, where the radius depends on the distance. Second, there are many depth-based abstraction filters and we show an example, based on the work by Jodeus `http://jodeus.tumblr.com/post/131437406357`, that produces discs from a subset of the pixels to achieve an abstract look (Fig. 14). These effects are best illustrated in the accompanying video.

4 Results

We have implemented our framework in Java on a desktop computer with an Intel Core i7 3.7 GHz CPU. The linear solver is implemented in Matlab and called from within the Java program. To make the

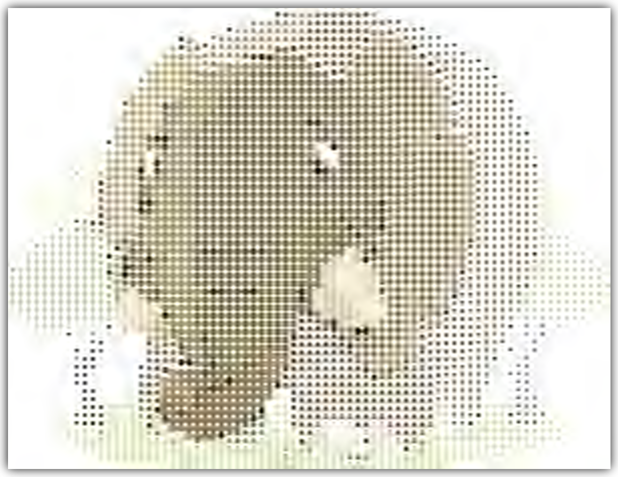

Figure 14: Depth-based abstraction.

solver more efficient, we build up an image pyramid for the input of the solver and solve each layer from low to high resolution, while using the result of the previous layer as the input for current layer. It takes about 1.5 mins to compute a depth map of 600 × 500. Nonetheless, we did not optimize our approach and it could be possible to achieve even real-time rates via a GPU implementation. Furthermore, the approach would lend itself well to upsampling strategies. For now, we provide a small-resolution preview to the user, which is interactive.

We tested our depth estimation on various datasets (e.g., Fig. 15). It works for real photographs, paintings, but also cartoons. All results and all sequences shown in the video have been produced by a user in less than 3 minutes.

5 Conclusion

We presented a pipeline for integrating depth-based effects into a single-image input. We proposed editing tools to facilitate the depth-map creation by influencing a depth-diffusion process. We demonstrated that our solution enables users to generate depth maps very rapidly and presented various examples for depth-based enhancements. In the future, we want to increase performance, which could be achieved via a sparse GPU linear solver.

In the future, we would like to explore using our tools to manipulate depth maps derived from an automated solution, which could reduce the user workload and maintain artistic freedom.

6 Acknowledgments

We are thankful to the anonymous reviewers for their invaluable feedbacks. We would also like to show our gratitude to Wikimedia Commons, `http://maxpixel.freegreatpicture.com`, `https://pixabay.com/`, Blender for providing various copyright free images used in our paper. We are also immensely grateful to Robert Postma/Design Pics and Lone Pine Koala Sanctuary for their permission to use the horse and koala images.

References

[1] V. Aslantas. A depth estimation algorithm with a single image. *Optics express*, 15(8):5024–5029, 2007.

[2] R. Bajcsy and L. Lieberman. Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 5(1):52–67, 1976.

[3] H. Bezerra, E. Eisemann, D. DeCarlo, and J. Thollot. Diffusion constraints for vector graphics. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 35–42. ACM, 2010.

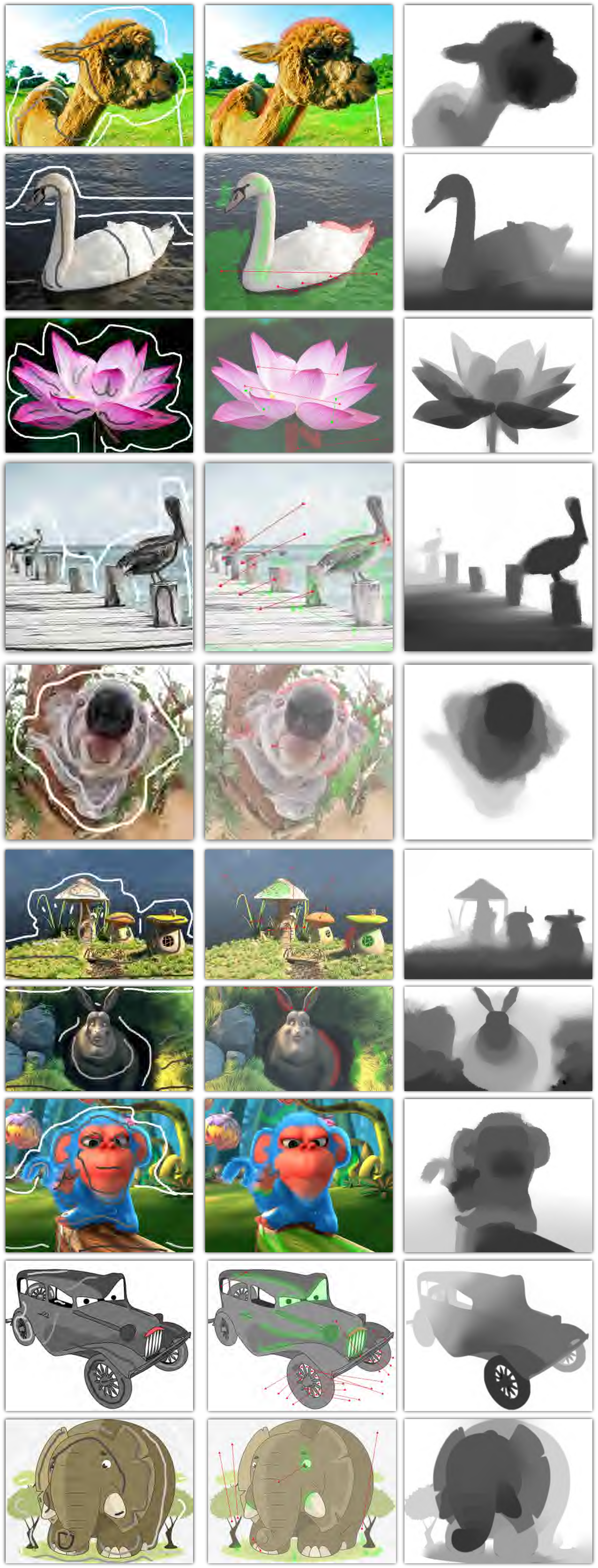

Figure 15: Examples. We support a wide variety of inputs including real photographs and cartoon images. Image source: from top to bottom, row 1, 2, 3, 4, 6, 10 are from `https://pixabay.com/`; row 9 is from [9]; row 7, 8 are from ©Blender open source movie *Big buck bunny* and *Monkaa*, respectively.

[4] S. Bruckner and E. Gröller. Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1344–1351, 2007.

[5] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[6] J. E. Cutting. Potency, and contextual use of different information about depth. *Perception of space and motion*, p. 69, 1995.

[7] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, and H.-P. Seidel. A perceptual model for disparity. In *ACM Transactions on Graphics (TOG)*, vol. 30, p. 96. ACM, 2011.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pp. 2366–2374, 2014.

[9] E. Eisemann, S. Paris, and F. Durand. A visibility algorithm for converting 3d meshes into editable 2d vector graphics. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 28:83:1–83:8, July 2009.

[10] M. Gerrits, B. D. Decker, C. Ancuti, T. Haber, C. Ancuti, T. Mertens, and P. Bekaert. Stroke-based creation of depth maps. In *2011 IEEE International Conference on Multimedia and Expo*, pp. 1–6, July 2011. doi: 10.1109/ICME.2011.6012006

[11] P. Kellnhofer, P. Didyk, T. Ritschel, B. Masia, K. Myszkowski, and H.-P. Seidel. Motion parallax in stereo 3d: model and applications. *ACM Transactions on Graphics (TOG)*, 35(6):176, 2016.

[12] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1330–1337. IEEE, 2012.

[13] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross. Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. Graph.*, 29(4):75:1–75:10, July 2010. doi: 10.1145/1778765.1778812

[14] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2136–2143. IEEE, 2009.

[15] S. Lee, E. Eisemann, and H.-P. Seidel. Real-time lens blur effects and focus control. In *ACM Transactions on Graphics (TOG)*, vol. 29, p. 65. ACM, 2010.

[16] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007.

[17] J. Liao, B. Buchholz, J. M. Thiery, P. Bauszat, and E. Eisemann. Indoor scene reconstruction using near-light photometric stereo. *IEEE Transactions on Image Processing*, 26(3):1089–1101, March 2017. doi: 10.1109/TIP.2016.2636661

[18] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Computer Graphics Forum*, vol. 18, pp. 39–50. Wiley Online Library, 1999.

[19] J. Lin, X. Ji, W. Xu, and Q. Dai. Absolute depth estimation from a single defocused image. *IEEE Transactions on Image Processing*, 22(11):4545–4550, 2013.

[20] Y.-H. Lin, M.-H. Tsai, and J.-L. Wu. Depth sculpturing for 2d paintings: A progressive depth map completion framework. *J. Vis. Comun. Image Represent.*, 25(4):670–678, May 2014. doi: 10.1016/j.jvcir.2013.12.005

[21] A. Lopez, E. Garces, and D. Gutierrez. Depth from a Single Image Through User Interaction. In A. Munoz and P.-P. Vazquez, eds., *Spanish Computer Graphics Conference (CEIG)*. The Eurographics Association, 2014. doi: 10.2312/ceig.20141109

[22] A. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, and D. Salesin. Diffusion curves: A vector representation for smooth-shaded images. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, vol. 27, 2008.

[23] S. E. Palmer, J. L. Brooks, and K. S. Lai. The occlusion illusion: Partial modal completion or apparent distance? *Perception*, 36(5):650–669, 2007.

[24] J. H. Park and H. W. Park. Fast view interpolation of stereo images using image gradient and disparity triangulation. *Signal Processing: Image Communication*, 18(5):401–416, 2003.

[25] A. P. Pentland. A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, (4):523–531, 1987.

[26] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pp. 1161–1168, 2005.

[27] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009.

[28] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003.

[29] A. Sellent and P. Favaro. Which side of the focal plane are you on? In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pp. 1–8. IEEE, 2014.

[30] 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*, pp. 47–54. ACM, 2011.

[31] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar. Estimating spatially varying defocus blur from a single image. *IEEE Transactions on image processing*, 22(12):4879–4891, 2013.