Depth Annotations
Designing Depth of a Single Image for Depth-based Effects

Liao, Jingtang; Shen, Shuheng; Eisemann, Elmar

DOI
10.1016/j.cag.2017.11.005

Publication date
2018

Document Version
Accepted author manuscript

Published in
Computers & Graphics

Citation (APA)
Liao, J., Shen, S., & Eisemann, E. (2018). Depth Annotations: Designing Depth of a Single Image for Depth-based Effects. *Computers & Graphics*, *71*, 180-188. https://doi.org/10.1016/j.cag.2017.11.005

Depth Annotations: Designing Depth of a Single Image for Depth-based Effects

Jingtang Liao, Shuheng Shen, Elmar Eisemann

Delft University of Technology, The Netherlands

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. 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. In case that a depth estimate is available from an automatic solution, we illustrate how this information can be integrated in form of a depth palette, that allows the user to transfer depth values via a painting metaphor. We demonstrate a variety of artistic 3D results, including wiggle stereoscopy, artistic abstractions, haze, unsharp masking, and depth of field.

Keywords:

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. On a computer display, it is also possible to add animation for the purpose of an increased 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.

When several views are available, image-based view interpolation (Park and Park, 2003) 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 (Eigen et al., 2014; Lai et al., 2012; Saxena et al.,

[*] Corresponding author.
Email address: j.liao@tudelft.nl (Jingtang Liao)

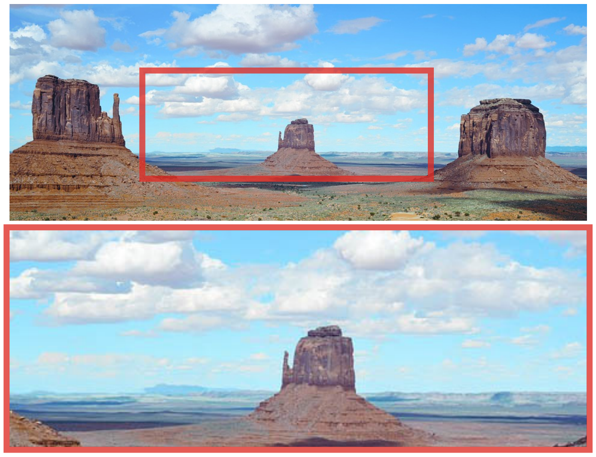


Figure 1: Ken Burns effect. Panning and zooming on still images. (Image source: `http://maxpixel.freegreatpicture.com`.)

2005, 2009), 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 production, live broadcast, and consumption of 3D content (Lang et al., 2010), or to modify a depth-of-field effect in an artistic manner (Lee et al., 2010), which is why we focus on a semi-automatic solution. We will show that a depth estimate, if available, can be beneficial as a starting point for our interactive depth-map design.

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).

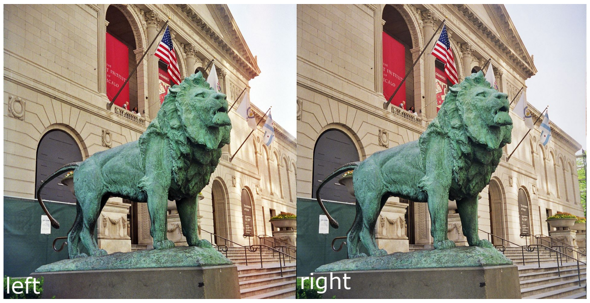


Figure 2: Wiggle stereoscopy. Looping a left/right image pair (Image source: Wikimedia Commons).

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 (Gerrits et al., 2011; Lin et al., 2014; Wang et al., 2011) or points (Lopez et al., 2014). 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, such as defining relative depth differences, a non-linear depth diffusion by assigning a strength to scribbles, or privileged diffusion directions. We ensure that all these elements can be formulated in a linear optimization problem to ensure a fast solving step. We additionally show a selection of effects in our results.

This article presents an improved and extended version of (Liao et al., 2017a). Besides all relevant aspects of previous work, we introduce several new contributions. The original contributions include:

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

In this work, we also present the following extensions:

- A new depth design tool, in form of a depth palette if an estimated depth map is available.
- Additional depth-based effects, such as unsharp masking, haze, or new artistic abstractions.

- An extended discussion of the presented techniques and new experiments.

Furthermore, we describe new interface decisions to ease the creation of the depth map and facilitate the choice of adequate depth values.

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 (Kellnhofer et al., 2016), size, texture gradient (Bajcsy and Lieberman, 1976), contrast, perspective, occlusion (Liao et al., 2017b), and shadows (Bruckner and Gröller, 2007) are examples of these. Motion parallax and occlusion are particularly strong (Cutting, 1995). 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.

Depth estimation for a single image is a well-known problem in computer graphics and computer vision that received much attention. Recent approaches (Eigen et al., 2014; Lai et al., 2012; Saxena et al., 2005, 2009; Liu et al., 2015; Karsch et al., 2014) are based on learning techniques. They enable an automatic conversion from a photo to a depth map. Nonetheless, 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 (Lang et al., 2010; Didyk et al., 2011), 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 (Pentland, 1987). Methods for single DFD from conventional aperture are usually based on such assumptions. Aslantas (2007) 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. (2013) designed aperture filters based on texture sharpness. Zhu et al. (2013) took smoothness and color edge information into consideration to generate a coherent blur map for each pixel. Shi et al. (2015) inferred depth information from photos by proposing a non-parametric matching prior with their constructed edgelet dataset, based on small small-scale defocus blur inherent in an optical lens. Their method is limited to photos in their original resolution and does not resolve ambiguities due to smooth edges. A general disadvantage of single-image DFD methods is that they cannot distinguish between defocus in front and behind the focal plane. Coded-aperture setups (Levin et al., 2007) address this issue by using a specially-designed aperture filter in the camera. Sellent and Favaro (2014) proposed an asymmetric aperture, which results in unique blurs for all distances from the camera. All these latter coded 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 (Criminisi et al., 2000; Liebowitz et al., 1999), and their extensions (Lee et al., 2009), focused on scenes containing objects with straight edges to reconstruct a 3D model by geometric reasoning and finding the best fitting model to line segments. In general, the use of edges is a good choice, as many natural scenes consist of smooth patches separated by object boundaries. Gerrits et al. (2011) 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. (2014) 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. (2011) proposed a workflow 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. (2014) 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. Tools for the definition of equalities and inequalities (Yücer et al., 2013; Sŷkora et al., 2010) can help reduce the amount of user intervention. 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(Orzan et al., 2008) and their extensions (Bezerra et al., 2010).

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* (Orzan et al., 2008), based on Poisson diffusion (Pérez et al., 2003), 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 $\{S_{i,j} \mid i \in 1..w, j \in 1..h\}$ on these scribbles. 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 (middle). It can be seen that the colors on scribbles smoothly diffuse across the whole image. The absolute depth values defined by the scribbles are useful to roughly associate depth ranges to different objects or parts of the scene. This is common in practice (Mendiburu, 2009) where a coarse layout of the scene depth is defined in the preprocess of the 3D design.

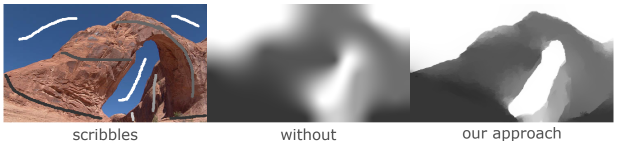


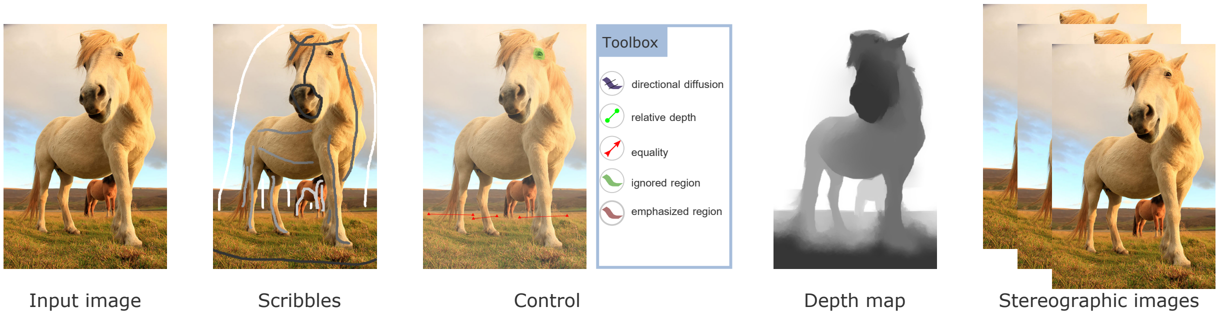
Figure 4: Depth estimation from scribbles. Scribble input (left), only using the scribble input results in a smooth depth map lacking discontinuities (middle), by involving the input image gradients, the depth propagation is improved (right). (Image source: `www.pixabay.com`)

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 (Perona and Malik, 1990). 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:

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

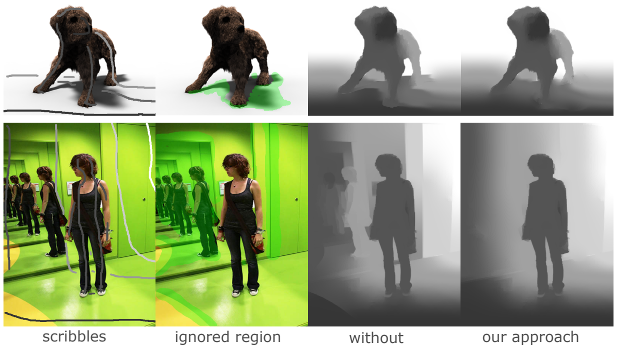
where ω_{kl} is the first order difference for the two neighboring pixels $\omega_{kl} = \exp(-\beta|I_k - I_l|)$. At the border of an object, ω_{kl} is often close to 0 because the pixel values typically differ. In consequence, the impact of the constraint is reduced, which, in turn, relaxes the smoothness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



Input image Scribbles Control Depth map Stereographic images

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.

boundaries. Fig. 4 (right) shows the effect of integrating the image gradient.



scribbles ignored region without our approach

Figure 5: Ignored-gradient region. Shadows and reflections 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. (Top image: courtesy of Erik Sintorn; bottom image: Flickr - salendron)

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 (top row) exhibits shadowed areas, which produce strong gradients that lead to artifacts on the floor, although it should actually have been smooth. For automated methods (Liu et al., 2015), a user might also want to tweak the resulting depth map. For example, reflections from a mirror in Fig. 5 (bottom row) might lead to artifacts, which can be addressed with an interactively designed depth map. To this extent, 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.

Emphasized-gradient Region. Contrary to the previous case, depth discontinuities might also need a boost in other areas. Consequently, we also allow the user to emphasize gradients. The gradient of the brushed pixels is enlarged by a scale 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. As illustrated in Fig. 6, there is no clear boundary at the highlighted (red and blue rectangles) locations. With this tool, the depth discontinuities at these image areas could be well pronounced.

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 Bezerra et al. (2010). For an arbitrary direction $\boldsymbol{d} := (\cos\theta, \sin\theta)$, the derivative of an image I along direction $\boldsymbol{d}$ is given by

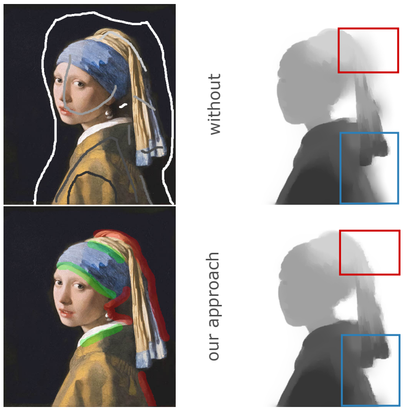


Figure 6: Emphasized-gradient region. Weak gradients can be enhanced to induce depth discontinuities. Here, it ensures a better separation between the foreground and background. (Image source: ”Girl with a Pearl Earring” by Johannes Vermeer)

$\nabla \boldsymbol{Id}$. In consequence, the constraints for pixel (i, j) are replaced by:

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

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

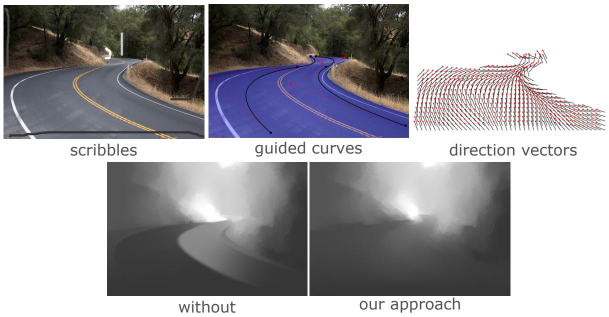


Figure 7: Diffusion guidance. A user brushes the region and draws the direct curves to define the direction in which he or she is interested in. Our approach can direct the diffusion mainly happens in this direction. (Image source: `http://maxpixel.freegreatpicture.com`)

Depth Diffusion Strength

Perspective projection can result in a non-linear depth mapping, e.g., via foreshortening. Furthermore, surfaces might not always be mostly planar but exhibit a convex or concave bent surface. For these situations, we want to provide the user with a way to influence the diffusion strength. Following (Bezerra et al., 2010), 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 $(d_1, \alpha_1), (d_2, \alpha_2)$, 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. Fig. 8 demonstrates a simple example with two input scribbles, a darker scribble on the left and lighter scribble on the right. We obtain a result where the two values uniformly spread across the image when using equal strength (Fig. 8 (left)). When selecting a bigger influence on the right part by assigning a higher strength to the left scribble its influence

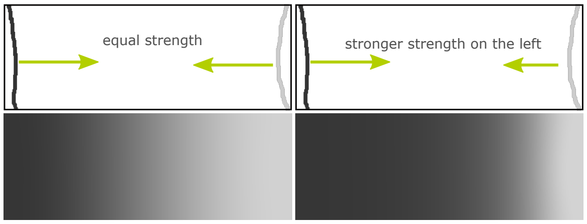


Figure 8: Scribble strength. Equal strength (middle); strength of left scribble stronger than the one on the right.

on the result is increased (Fig. 8 (right)). 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. 9 shows the influence of the diffusion strength for different values.

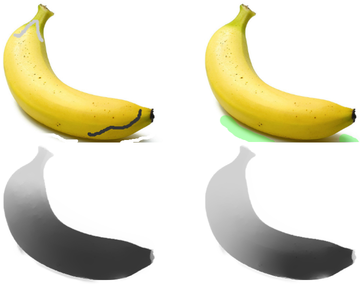


Figure 9: Non-linear depth mapping. Assigning a strength to different scribbles can be used to influence the diffusion speed.

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 Bezerra et al. (2010). This possibility is quite useful for images containing symmetric features, as shown in Fig. 10, where pixels on the pillars, which are at the same depth, can be linked. There are also cases in which it may be hard for a user to choose adequate depth values for scribbles. Fig. 11 shows an example, in which drawing scribbles with absolute values for each gap inside the wheels would be very difficult, as the correct value depends on the background. With our tool, we can link the background to other regions. It is worth noting that many pixels can be connected at the same time.

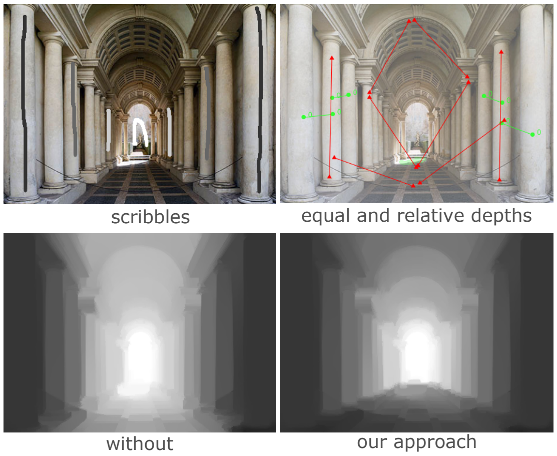


Figure 10: Depth equality and relativity We connect depths from different places together via depth equality and relativity to globally influence the depth estimation. (Image source: wikipedia)

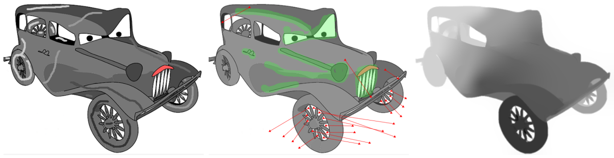


Figure 11: Equal constraints. Connecting depth from different places via depth equality can reduce the user interventions. (Image source: Eisemann et al. (2009))

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 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.

Depth Palette

With the advent of single-image depth estimation, automated approaches (Liu et al., 2015; Karsch et al., 2014) can provide useful information to initiate the depth map design. Unfortunately, there can be multiple depth inconsistencies or noise, as shown in Fig. 12 (top row, left) highlighted (red rectangles) regions. Additionally, the resulting depth might not be adequate for artistic purposes (Wang et al., 2011). Hence, directly using the resulting depth as an input for the 3D effects, e.g., wiggle stereography, could cause visible artifacts (please refer to supplementary video). However, the initial depth maps can serve as a good starting point for the depth-map design. Similar to a color or normal palette, a user can transfer depth values directly from the reconstruction. For this purpose, a position is chosen in the reference depth image. The selected depth value can then be used to draw a scribble with the corresponding value, which will generate a corresponding hard constraint. While drawing the scribble, the value can either be held constant or the values of the corresponding pixels from the reference could be transferred.With only a few depth transfers, it is possible to improve the depth-map quality using our solution.

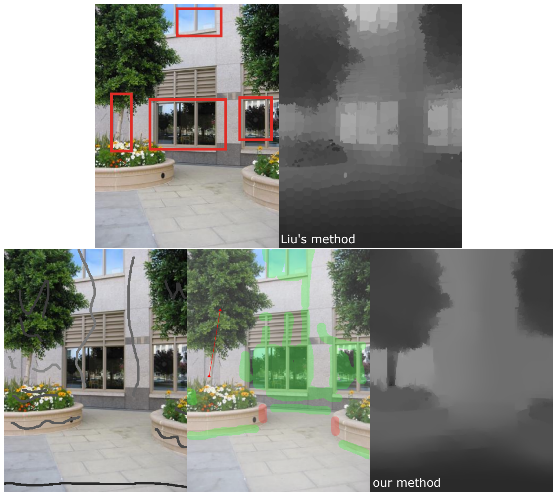


Figure 12: Depth palette. Using the result of automated methods (Liu et al., 2015) as depth palette can ease the depth creation.

Additional Interface Elements

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 nonlinear remapping. We use an interpolating spline, adjusted via control points. A result is illustrated in Fig. 13 (left), where the depth appearance of the scene is globally influenced to obtain Fig. 10. Global adjustments are particularly useful for stereo-based effects, as they allow the user to influence the global disparity range. In this context, we provide a simple user interaction to control the 3D effect on the canvas. Instead of defining the stroke values by choosing from a palette, the user can also simply drag the mouse to indicate a disparity baseline that then corresponds to a depth value that is automatically transferred to the stroke. This process makes it easy to control warping effects, in case the depth map is used to derive a stereo pair. Please also refer to supplementary video.

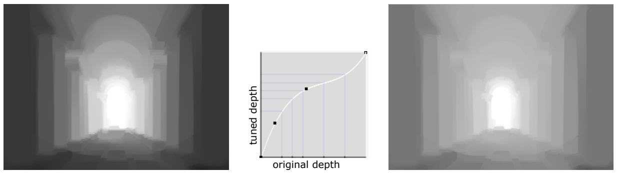


Figure 13: Depth adjustment. Depth map can be globally adjusted using a mapping curve.

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. 14. Alternatively, we can convert the distance to a fog density and apply it as an overlay on the image (Willis, 1987).

Depth-of-Field Effects

It is possible to simulate lens blur to refocus on different parts of the scene. Fig. 15 (right) shows an example.

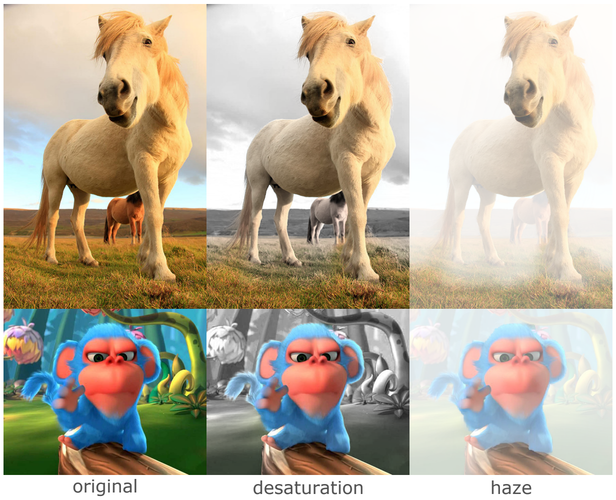


Figure 14: Distance-based desaturation and haze.

Unsharp Masking

Textures and colors in images can be enhanced by utilizing unsharp masks. Contrary to what its name may indicate, unsharp masks are used to sharpen the images. An unsharp mask is created by subtracting a low-pass filtered (usually Gaussian filter) copy from the original image. The mask is then added back to the original image to get a local contrast enhancement. While typically applied for color images, the involvement of depth enables us to well separate different elements from each other (Luft et al., 2006). Note that even when colors are similar (color of the puppy's hair and the background), involving the depth map makes sure that the depth difference becomes more evident.

Luft et al. (2006) proposed a depth-based unsharp-masking method. Assuming that a depth map D is available, the unsharp-masking process is applied to the depth buffer: $\Delta D = G * D - D$, with $G * D$ is the convolution of a Gaussian filter. The resulting high frequency ΔD is then used to alter the original image I to achieve a sharpening or a local contrast enhancement: $I^{'} = I + \Delta D \cdot \lambda$, where λ is a user defined gain parameter. Thus, the greater the spatial difference, the higher the local enhancement.

We found that in some cases, distant elements receive an overly strong enhancement. In consequence, we propose an adaptive gain value and Gaussian kernel size. Based on the observation in Ritschel et al. (2008) that unsharp masking can be performed in 3D instead of image space, we propose a hybrid approach. We adapt the kernel size depending on the depth map values, i.e., the farther away, the smaller the kernel size. Specifically, we define the kernel size as: $\delta_{adapt} = \delta(1-0.5D)$, with δ being 2% of the image diagonal. Moreover, we apply a bilateral filter instead of a Gaussian filter, to ensure that elements from different depths do not mix and, hereby, keep the contrast of edges. To avoid oversatuation, all operations are executed in CIELAB color space.

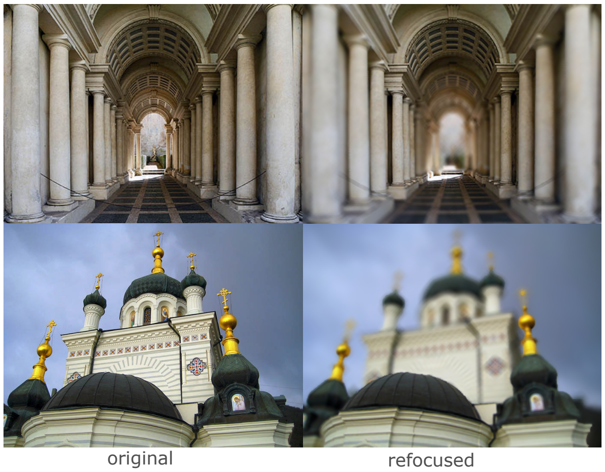


Figure 15: Image refocusing based on the depth values.

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. 17.

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

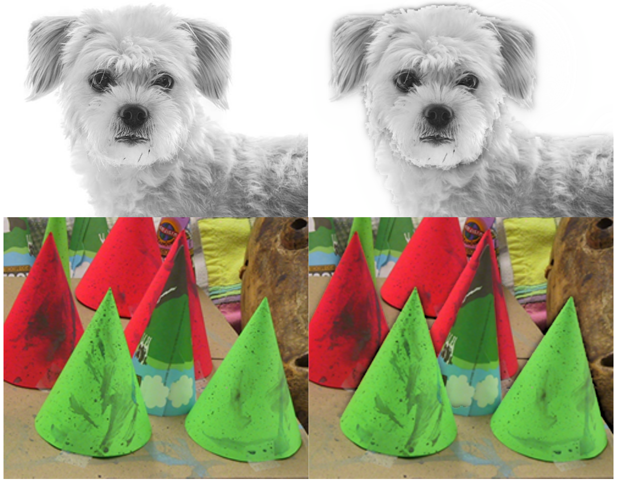


Figure 16: Unsharp masking using a depth buffer. It can enhance the depth arrangement in the scene and make a dull appearance more interesting.

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. 18 shows the comparison with and without hole filling. Note that our hole filling method is not suitable for big motions.

Figure 17: Examples of looping videos, please refer to the supplemental animations.

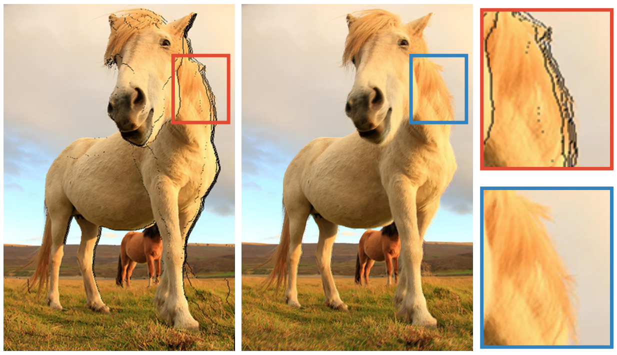


Figure 18: Hole filling. Holes due to reprojection (left) are filled (right).

Artistic Effects

Besides changing the viewpoint, the derived depth map can also be used to apply 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`. Here, discs are used to replace a subset of the pixels to achieve an abstract look (Fig. 19). These effects are best illustrated in the accompanying video.

4. Results and Discussion

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 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 the current layer. It takes about 30 seconds 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. 20). It works for real photographs, paintings, but also cartoons. All results and all sequences shown in

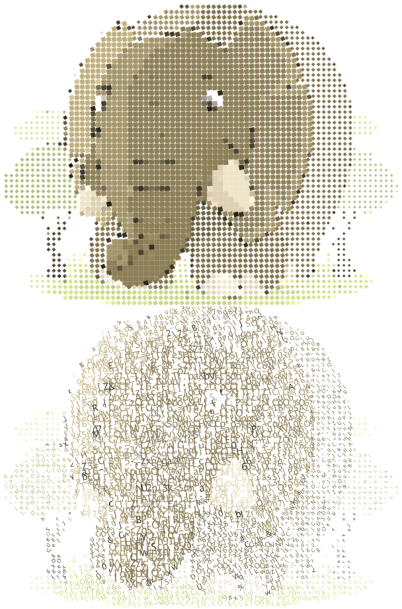


Figure 19: Example of different depth-based abstractions.

the video have been produced by a user in less than 3 minutes.

We did not conduct a user study to investigate the effectiveness of our tools. In practice, we received positive feedback from three test users. Nonetheless, expertise in image editing is definitely an advantage. This is similar to novice users applying advanced tools in software, such as Photoshop or Gimp. A certain amount of training is also helpful to gain familiarity. Increasing user friendliness further could be an interesting direction for future work.

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.

It would also be interesting to apply our method for animations. One possible solution might be to design depth

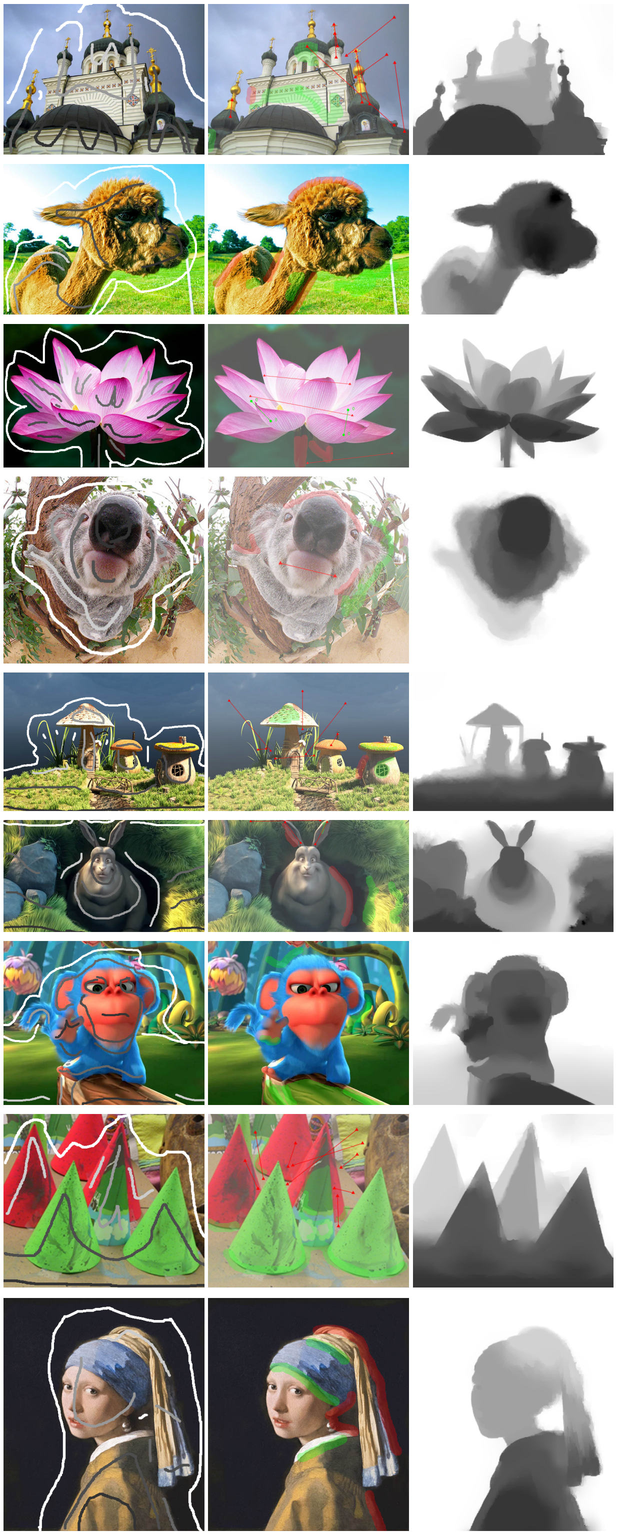


Figure 20: Examples. We support a wide variety of inputs including real photographs, paintings and cartoon images. Image source: from top to bottom, row 1, 2, 3, 5 are from `https://pixabay.com/`; row 4 is from Lone Pine Koala Sanctuary; row 6, 7 are from ©Blender open source movie *Big buck bunny* and *Monkaa*, respectively; row 8 is from (Scharstein and Szeliski, 2003); row 9 is "Girl with a Pearl Earring" by Johannes Vermeer.

maps for several key frames and propagating the annotations, similar to rotoscoping (Agarwala et al., 2004).

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

Park, JH, Park, HW. Fast view interpolation of stereo images using image gradient and disparity triangulation. Signal Processing: Image Communication 2003;18(5):401–416.

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

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

Saxena, A, Chung, SH, Ng, AY. Learning depth from single monocular images. In: Advances in Neural Information Processing Systems. 2005, p. 1161–1168.

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

Lang, M, Hornung, A, Wang, O, Poulakos, S, Smolic, A, Gross, M. Nonlinear disparity mapping for stereoscopic 3d. ACM Trans Graph 2010;29(4):75:1–75:10. URL: http://doi.acm.org/10.1145/1778765.1778812. doi:10.1145/1778765.1778812.

Lee, S, Eisemann, E, Seidel, HP. Real-time lens blur effects and focus control. In: ACM Transactions on Graphics (TOG); vol. 29. ACM; 2010, p. 65.

Gerrits, M, Decker, BD, Ancuti, C, Haber, T, Ancuti, C, Mertens, T, et al. Stroke-based creation of depth maps. In: 2011 IEEE International Conference on Multimedia and Expo. 2011, p. 1–6. doi:10.1109/ICME.2011.6012006.

Lin, YH, Tsai, MH, Wu, JL. Depth sculpturing for 2d paintings: A progressive depth map completion framework. J Vis Comun Image Represent 2014;25(4):670–678. URL: http://dx.doi.org/10.1016/j.jvcir.2013.12.005. doi:10.1016/j.jvcir.2013.12.005.

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

Lopez, A, Garces, E, Gutierrez, D. Depth from a Single Image Through User Interaction. In: Munoz, A, Vazquez, PP, editors. Spanish Computer Graphics Conference (CEIG). The Eurographics Association. ISBN 978-3-905674-67-5; 2014,doi:10.2312/ceig.20141109.

Liao, J, Shen, S, Eisemann, E. Depth map design and depth-based effects with a single image. In: Proc. of Graphics Interface (GI). 2017a,.

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

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

Liao, J, Eisemann, M, Eisemann, E. Split-depth image generation and optimization. Computer Graphics Forum 2017b;36(7).

Bruckner, S, Gröller, E. Enhancing depth-perception with flexible volumetric halos. IEEE

Transactions on Visualization and Computer Graphics 2007;13(6):1344–1351.

Cutting, JE. Potency, and contextual use of different information about depth. Perception of space and motion 1995;:69.

Liu, F, Shen, C, Lin, G. Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, p. 5162–5170.

Karsch, K, Liu, C, Kang, SB. Depthtransfer: Depth extraction from video using non-parametric sampling. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2014;.

Didyk, P, Ritschel, T, Eisemann, E, Myszkowski, K, Seidel, HP. A perceptual model for disparity. In: ACM Transactions on Graphics (TOG); vol. 30. ACM; 2011, p. 96.

Pentland, AP. A new sense for depth of field. IEEE transactions on pattern analysis and machine intelligence 1987;(4):523–531.

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

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

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

Shi, J, Tao, X, Xu, L, Jia, J. Break ames room illusion: depth from general single images. ACM Transactions on Graphics (TOG) 2015;34(6):225.

Levin, A, Fergus, R, Durand, F, Freeman, WT. Image and depth from a conventional camera with a coded aperture. ACM transactions on graphics (TOG) 2007;26(3):70.

Sellent, A, Favaro, P. Which side of the focal plane are you on? In: Computational Photography (ICCP), 2014 IEEE International Conference on. IEEE; 2014, p. 1–8.

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

Liebowitz, D, Criminisi, A, Zisserman, A. Creating architectural models from images. In: Computer Graphics Forum; vol. 18. Wiley Online Library; 1999, p. 39–50.

Lee, DC, Hebert, M, Kanade, T. Geometric reasoning for single image structure recovery. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE; 2009, p. 2136–2143.

Yücer, K, Sorkine-Hornung, A, Sorkine-Hornung, O. Transfusive weights for content-aware image manipulation. In: VMV. 2013, p. 57–64.

Sỳkora, D, Sedlacek, D, Jinchao, S, Dingliana, J, Collins, S. Adding depth to cartoons using sparse depth (in) equalities. In: Computer Graphics Forum; vol. 29. Wiley Online Library; 2010, p. 615–623.

Orzan, A, Bousseau, A, Winnemöller, H, Barla, P, Thollot, J, Salesin, D. Diffusion curves: A vector representation for smooth-shaded images. In: ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008); vol. 27. 2008,URL: http://maverick.inria.fr/Publications/2008/OBWBTS08.

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

Pérez, P, Gangnet, M, Blake, A. Poisson image editing. In: ACM Transactions on Graphics (TOG); vol. 22. ACM; 2003, p. 313–318.

Mendiburu, B. Chapter 5 - 3d cinematography fundamentals. In: Mendiburu, B, editor. 3D Movie Making. Boston: Focal Press. ISBN 978-0-240-81137-6; 2009, p. 73 – 90. URL: https://www.sciencedirect.com/science/article/pii/B9780240811376000057. doi:https://doi.org/10.1016/B978-0-240-81137-6.00005-7.

Perona, P, Malik, J. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence 1990;12(7):629–639.

Eisemann, E, Paris, S, Durand, F. A visibility algorithm for converting 3d meshes into editable 2d vector graphics. ACM Trans Graph (Proc of SIGGRAPH) 2009;28:83:1–83:8. URL: `http://graphics.tudelft.nl/Publications-new/2009/EPD09`.

Willis, P. Visual simulation of atmospheric haze. Computer Graphics Forum 1987;6(1):35–41. URL: `http://dx.doi.org/10.1111/j.1467-8659.1987.tb00343.x`. doi:`10.1111/j.1467-8659.1987.tb00343.x`.

Luft, T, Colditz, C, Deussen, O. Image enhancement by unsharp masking the depth buffer. ACM Trans Graph 2006;25(3):1206–1213. URL: `http://doi.acm.org/10.1145/1141911.1142016`. doi:`10.1145/1141911.1142016`.

Ritschel, T, Smith, K, Ihrke, M, Grosch, T, Myszkowski, K, Seidel, HP. 3d unsharp masking for scene coherent enhancement. In: ACM Transactions on Graphics (TOG); vol. 27. ACM; 2008, p. 90.

Scharstein, D, Szeliski, R. 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,.

Agarwala, A, Hertzmann, A, Salesin, DH, Seitz, SM. Keyframe-based tracking for rotoscoping and animation. ACM Transactions on Graphics (ToG) 2004;23(3):584–591.