

Virtual Passepartouts

Tobias Ritschel

Krzysztof Templin

Karol Myszkowski

Hans-Peter Seidel

MPI Informatik



Figure 1: Adding a virtual passepartout to a 3D scene (left) improves the 3D separation between foreground and background objects (center) using a “looking-through-a-window” metaphor of a 2D medium, such as a computer monitor (right).

Abstract

In traditional media, such as photography and painting, a cardboard sheet with a cutout (called *passepartout*) is frequently placed on top of an image. One of its functions is to increase the depth impression via the “looking-through-a-window” metaphor. This paper shows how an improved 3D effect can be achieved by using a *virtual passepartout*: a 2D framing that selectively masks the 3D shape and leads to additional occlusion events between the virtual world and the frame. We introduce a pipeline to design virtual passepartouts interactively as a simple post-process on RGB images augmented with depth information. Additionally, an automated approach finds the optimal virtual passepartout for a given scene. Virtual passepartouts can be used to enhance depth depiction in images and videos with depth information, renderings, stereo images and the fabrication of physical passepartouts.

CR Categories: I.3.m [Computer Graphics]: Miscellaneous

Keywords: matting, passepartout, depth

1 Introduction

Designing a picture’s frame can have a tremendous impact on the perception of the image as a whole [Mitchell and Roberts 1996]. Besides the material and geometry of the physical frame, the key design decision is the selection of the visible part of a (virtual) world.

Color images augmented with depth information (RGBZ images), such as produced by rendering, stereo reconstruction and 3D sensors have already become common in production and will soon become widespread for everyday users. RGBZ offers a range of new opportunities in image processing and stylization, which goes beyond the possibilities of plain RGB.

The definitive version is available at <http://diglib.eg.org/>.

Here, we propose a framework to design “virtual passepartouts” that enhance the depiction of RGBZ images when reproduced using traditional 2D media, e.g., printed paper or non-3D screens. Our framework allows to manually or automatically derive passepartouts that clip an RGBZ image in 3D space, such that their 2D reproduction provides additional occlusion cues (Fig. 1). Therefore, the frame is perceived as a “window” inside the medium, e.g., a hole cut in a piece of paper, but its placement is also related to the three-dimensional organization of the scene.

2 Related Work

Our approach draws inspiration from classic artistic practice, human perception, contemporary computer graphics, and stereo 3D cinematography.

Art Ever since the early renaissance [Alberti 1991] the metaphor of an image as a window into another world has inspired many artists. Selecting the appropriate frame for this “window” has been a challenge to artists for centuries [Mitchell and Roberts 1996]. Besides “classic” frames (mostly rectangular or elliptical), the use of “virtual” passepartouts where the virtual world extends towards the viewer and beyond the window’s frame has been explored (Fig. 2). The concept of virtual passepartouts is applicable to painting, photography and computer-generated content. However, designing a virtual passepartout without a framework such as ours imposes considerable manual work: The artist has to cut away all RGB pixels in 2D according to a 3D passepartout definition manually. Trying out a handful of different designs in an explorative fashion becomes infeasible. To our knowledge, there is no automatic way to suggest a virtual passepartout for a given RGBZ image. Our framework includes a simple metric that quantifies how much occlusion cues are gained by inserting a virtual passepartout and how much image information is lost from cutting. This metric can serve as an optimization function to automatically suggest virtual passepartouts.

Perception Occlusion is one of the strongest depth cues [Cutting and Vishton 1995; Palmer 1999]. An object that is occluding is perceived closer and complete; a partially occluded object is perceived more distant and to extend behind the occluding object. This effect is believed to contribute to the famous moon illusion [Palmer et al.

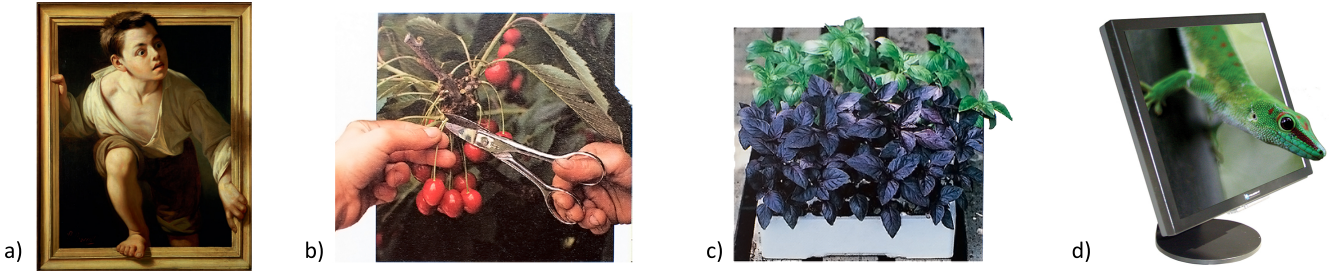


Figure 2: The use of virtual passepartouts in painting, photography and computer graphics. a) “Escaping Criticism” by Pierre Borell del Caso (1874): A classic trompe-l’oeil making use of a virtual passepartout. b) and c): Typical photography-based virtual passepartout illustrations from “Fruit and Vegetable Gardening” by Jim Arbury et al. (2002); . d) A virtual passepartout to convey the functionality of a 3D screen.

2007] where an object such as the moon appears larger when it is occluded by the horizon.

Objects can appear more three-dimensional, farther away and more extended when seen through a hole (the Schlossberg illusion [Ames 1925]). The illusion is related to depth perception when an image is framed [Gooch and Gooch 2005]. To our knowledge, perception of such framed images has not yet been studied explicitly, in particular, in the presence of local frame occluders which belong to the scene (see Fig. 1). We introduce the term “virtual passepartout” to characterize such image-frame configurations and the underlying depth illusion that is induced by introduced that way occlusions.

Occlusion configurations have received attention in computer vision, e. g., based on line drawings [Waltz 1975] or natural images [Ren et al. 2006]. By adding a virtual passepartout to a 3D scene, pairwise T-junctions are produced both between the edges of the foreground objects and the passepartout itself as well as between the passepartout edges and the middleground objects. The passepartout edges terminate at the T-junctions, which is a strong indication that the passepartout is behind the foreground object. This impression is additionally enhanced by the perceived continuation of the occluded contour (the amodal completion [Kellman and Shipley 1991]).

Apart from the strong depth cues that are caused by explicit occlusions, passepartout edges, which separate textures of different blurriness, may induce relative depth cues that are interpreted by the visual system as depth-dependent observer’s accommodation [Mather and Smith 2004]. For example a blurred wooden frame overimposed on sharp picture should appear closer to the viewer, in particular when the frame edges are blurred as well, which stresses their belongingness to the frame [Marshall et al. 1996].

Computer graphics The metaphor of a rendered image being a window into a virtual world [Sutherland 1965] is well established in computer graphics. The design of frames was extensively studied in a computer graphics context by Gooch and Gooch [2005]. Given an image and a classic clipping region they seek to select matte colors and styles to enhance the sense of depth. The “Tour into pictures” [Horry et al. 1997] and “photo pop-ups” [Hoiem et al. 2005] project photos to coarse 3D geometry which then allows to change the viewpoint to a certain extent, however classic clipping is used. Images with depth information approximated based on the RGB image configuration [Lopez-Moreno et al. 2010] or acquired with a time-of-flight camera [Richardt et al. 2012] were used for relighting and color stylization, but no RGBZ pixels were clipped. Explosion [Bruckner and Groller 2006] or cut-away illustrations [Burns and Finkelstein 2008] make a scene more understandable by removing parts of the scene and revealing its interior structures. Finally, the problem of finding the optimal passepartout (virtual or not) is similar to viewpoint selection [Vázquez et al. 2001].

Stereo 3D cinematography The floating stereoscopic window, which is created by adding black masks around the space with 3D objects, is an important tool in 3D cinematography that enables control over the placement of story space within the theater space [Mendiburu 2009]. For example, tilting the window horizontally brings one side of the scene closer to the audience. Alternatively, the out-of-screen effect can be exaggerated by pushing the window behind the screen plane, or “the-sky-is-falling” effect can be achieved by leaning the top of the window towards the audience. These techniques are conceptually similar to those that are viable with virtual passepartouts, however the floating stereoscopic window affects depth perception by manipulating the binocular disparities of the screen border, whereas our method relies on shape and occlusion cues only. Attempts to combine both approaches have been made [Gardner 2011].

3 Virtual Passepartouts

Our approach is based on the definition of virtual passepartouts (Sec. 3.1), that can be used either in a manual (Sec. 3.3) or an automated approach (Sec. 3.4).

3.1 Definition

We will differentiate between *classic* and several forms of *virtual* passepartouts. Classic passepartouts define a region in image space that clips (i. e., replace with “paper”) all locations outside that region. Formally, an operator

$$p_c(c, m, s)(\mathbf{x}) := \begin{cases} c(\mathbf{x}), & \text{if } s(\mathbf{x}) = \mathbf{I} \\ m(\mathbf{x}), & \text{otherwise,} \end{cases} \quad (1)$$

maps an input color image $c(\mathbf{x}) \in \mathbb{R}^2 \rightarrow \mathbb{R}$, a passepartout matte image $m(\mathbf{x}) \in \mathbb{R}^2 \rightarrow \mathbb{R}$ using a passepartout shape function $s(\mathbf{x}) := \mathbb{R}^2 \rightarrow \{\mathbf{I}, \mathbf{O}\}$ to a new color image with a virtual passepartout effect. The matte image m can be a simple constant e. g., white, or a pattern such as paper or wood. The shape function s return 0 if the pixel is outside and should be clipped and 1 if it is inside and should remain visible.

Taking a different approach, a virtual passepartout (Fig. 3, right) clips parts of the world in such a way, that additional occlusion cues are provided by letting foreground elements of the scene “escape” the image towards the viewer. To this end, we again define an arbitrary region in image space, that clips all pixels outside that region that belong to the background but we exclude locations that belong to the foreground from clipping:

$$p_v(c, m, g, s)(\mathbf{x}) := \begin{cases} c(\mathbf{x}), & \text{if } s(\mathbf{x}) = \mathbf{I} \text{ or } g(\mathbf{x}) = \mathbf{F} \\ m(\mathbf{x}), & \text{otherwise,} \end{cases} \quad (2)$$

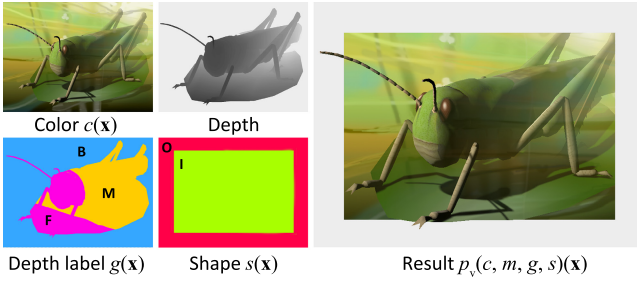


Figure 3: The color c , depth, depth label g and shape function s that compose a virtual passepapout result $p_v(c, m, g, s)$.

where $g(\mathbf{x}) := \mathbb{R}^2 \rightarrow \{F, M, B\}$ is a function that labels an image space location \mathbf{x} as F if it belongs to the foreground, as M if it belongs to the middle ground and as B if it belongs to the background. The choice of three vs. two depth labels is motivated by psychophysics and art history as discussed in Section 4.

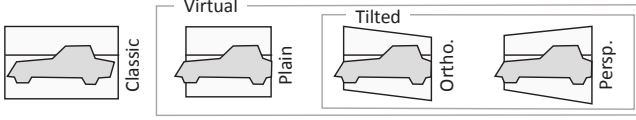


Figure 4: Different types of passepapouts.

Passepapout shapes Different definitions of s are possible, but in practice, three forms are important: *Plain* (Fig. 4, middle-left) rectangular passepapouts. In *tilted* (sheared) virtual passepapouts (Fig. 4, middle-right) the edges are not orthogonal anymore. In *Orthographic* passepapouts, opposite edges are parallel but not aligned to the principal axes anymore. *Perspective* virtual passepapouts (Fig. 4, right) are subject to slight perspective foreshortening. In Sec. 3.3 a system to manipulate the shape is presented and Sec. 3.4 proposes an approach to optimize for the shape of the passepapout to find the best pop-out effect.

Foreground-background separation A separation into foreground, middle ground and background is input to our algorithm. In the simplest case, the separation can be provided by the user e. g., by painting, eventually complemented by image segmentation. If the image is complemented by depth information, more advanced separation is possible. In Sec. 3.4, we propose a simple automatic procedure to label depth.

3.2 Implementation

Implementing p_v as a post-process using a GPU is straightforward. A fragment shader is used to process all pixels of the image in parallel. The image c , the passepapout matte image m , the passepapout shape image s and the labeling g are input textures to this shader. The resulting fragment program can execute in few milliseconds for HD content on a recent GPU and is easy to integrate into existing software.

3.3 Manual control

Virtual passepapouts can be controlled using a simple user interface with real-time feedback. Two modes are used to manipulate a virtual passepapout: a *shape* mode to specify the shape s of the cut-out in the image space and a *depth* mode to specify the depth labeling g .

Shape mode The shape of non-tilted passepapout can be controlled for each edge independently by specifying its position with respect to the original image. For tilted virtual passepapouts the user manipulates the shape of an orthographic or perspective projection in the image space (Fig. 4).

Depth mode In depth mode, the assignment of labels to different depth values in an RGBZ image can be controlled.

The simplest mode controls the far plane (i. e., the plane separating the foreground and the middle ground) using a single depth value. This mode is sufficient, if the scene has a simple organization that separates the image into foreground, middle ground and background easily. In most cases, assuming a background at 0.9 and a middle ground at 0.5 is a good initial guess.

In the advanced mode, the far plane can be *slanted* (Fig. 5) around the x -axis (not to be confused with “tilted”, Fig. 4). This is most

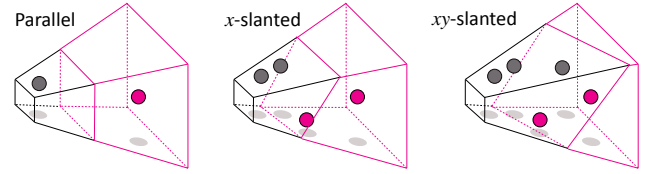


Figure 5: Parallel, x -slanted and xy -slanted far planes.

useful, when a ground plane is visible in the image and a simple classification into foreground / background is not possible based on depth values alone. In some cases an additional slant also along the y -axis can be used for stylization.

3.4 Automatic virtual passepapouts

Besides manual control as introduced in the previous section, we propose to automatically create virtual passepapouts given an RGBZ image. A user can specify constraints and preferences and manipulate the image e. g., by moving the camera or playing an animation, and the passepapout is adapted and rendered in real-time.

The automatic procedure consists of three steps: First, the user specifies the type of passepapout (plain, orthographic, perspective), second the system optimizes for the best labeling g , and finally for the optimal shape s . The key question is of course, what is a good labeling, and what is a good shape?

Automatic labeling We found a good labeling to cut the scene into one large chunk of background with similar depth values and two equally-sized chunks of foreground and middle ground. First, the depth is normalized to fall into the unit interval by linear re-scaling. Next, all pixels with a depth larger than 0.9 are considered background. For the remaining pixels, a cumulative depth histogram is created, inverted and all pixel with depth values in the first half are labeled foreground and all remaining ones are labeled middle-ground.

More advanced, a plane that best separates the points into separate clusters could be fit. This approach is similar to the separating plane found for support vector machines in machine learning [Cortes and Vapnik 1995].

Automatic shape Finding the optimal shape functions s is restricted to the family of projectively transformed convex quads

$$s(\mathbf{x}) := \begin{cases} 1, & \text{if } \mathbf{x} \in T([s_{\min}, s_{\max}] \times [t_{\min}, t_{\max}]) \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where T is a given projective transformation of a plane, such as identity, a shear, a rotation or a perspective transformation (cf. Fig. 4). We call each shape parameter $u \in \{s_{\min}, s_{\max}, t_{\min}, t_{\max}\}$ a “cut” and optimize for it independently. A good cut is found, if a significant, but not too large piece of foreground overlaps the passepartout, or a significant, but not too large piece of middle ground is overlapped by the passepartout. For a usual 1280×1024 screen “significant” is a feature with a size of $a \approx 15$ pixels squared, i. e., approximately foveal. The background is ignored, i. e., nothing is won by cutting background. Also, cuts that go too deep into the picture should be omitted, as well as cuts that intersect the geometry. Note that this formulation allows foreground to overlap the passepartout and background to be cut by the passepartout along the same passepartout edge.

The following simple heuristic procedure achieves this behavior. For now, assume we want to find the cut $u = s_{\min}$ (the other cuts are performed in an analogous way and independently) of a plain passepartout, i. e., $T = \text{identity}$. First, we count the number of foreground pixels in front of the passepartout and middle ground pixels behind the passepartout for all pixels in each column. This is, what is gained by cutting along this column. The sums can be computed on the GPU for columns in parallel and only the result is read back to the CPU. Next, we loop over all columns and find the one, where a cut would result in a gain that is the closest to the optimal gain. All cuts that result in an intersection are skipped. A cut is considered to cause an intersection, if a foreground and a middle ground pixel come closer than a defined threshold (e. g., 0.05 of the maximal cut size). The search is stopped, if the cut size is larger than a threshold b , typically equal 0.25. If no cut was found, u is set to 0.

To generalize to arbitrary transformations T , the image is first deformed by T^{-1} and the above procedure is repeated (cf. Fig. 6). This is similar to rotating the image to allow separable anisotropic, non-axis-aligned filters for segmentation used by Arbelaez et al. [2011].

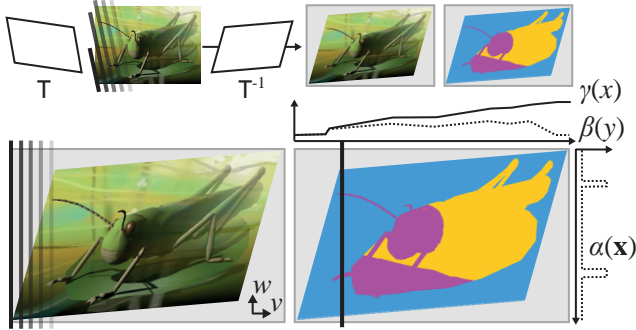


Figure 6: Automatic virtual passepartouts (see text for details).

For selecting the best cut formally let

$$\alpha(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) = F \text{ or } g(\mathbf{x}) = M \\ 0, & \text{otherwise} \end{cases}$$

be the gain at location \mathbf{x} . The vertical plot right in Fig. 6 illustrates a vertical slice of the two-dimensional function α . Let $\mathbf{v} \in \mathbb{R}^2$ be the direction of the cut and $\mathbf{w} \in \mathbb{R}^2$ the direction orthogonal to it, we define the gain *along* a cut (horizontal plot on the right of Fig. 6)

$$\beta(y) = \int_0^1 \alpha(T(y\mathbf{v} + x\mathbf{w}))dx,$$

and *up to* a cut as (horizontal plot on the right of Fig. 6)

$$\gamma(x) = \int_0^x \beta(y)dy.$$

The best cut u for the direction \mathbf{v} finally is

$$\operatorname{argmin}_{u \in [0 \dots b]} |\gamma(u) - a|.$$

Discussion We experimented with a separation into only two depth labels (foreground and background, i. e., figure-ground). Such a labeling however, could not faithfully reproduce the artistic effect. A three-fold labeling of space for example into “personal”, “action”, “vista” space is common in psychophysics [Cutting and Vishton 1995, Fig. 1] or art history.

Our metric is greedy and does not account for the combined effect of cutting from multiple directions. Especially for large cuts, accounting for their combined optimum could lead to different results. However, treating all cuts separately allows the problem to be split into simple partial sums that can be evaluated using a GPU easily. The optimization is also greedy, in such that it first finds the labeling, and then the shape. A joint optimization remains as future work. Our metric to compute the gain and loss of information is admittedly simple and excludes all higher order information. Also no composition rules are included in the optimization [Liu et al. 2010]. Instead of counting the rows or columns lost, the saliency in the lost area could be accumulated. We experimented with this approach but we did not find it useful enough to justify the additional complexity. In future work, the optimization should also include picking the transformation [Vázquez et al. 2001], which by now is specified beforehand.

4 Results

Several results of manually adding virtual passepartouts to RGBZ images of different modality are presented in Fig. 7. Please see the supplemental video for the interactions required to produce the results. Results of our optimization for a 3D scene, which is viewed from different perspectives, can be found in Fig. 8 and for photos in Fig. 9. In the last case we created the depth buffers manually.

Light transport between the virtual world and the medium is a classic ingredient found in traditional trompe-l’oeil painting (Fig. 2a). This further bridges the real and the virtual world making the medium disappear. The same concept can be applied to virtual passepar-

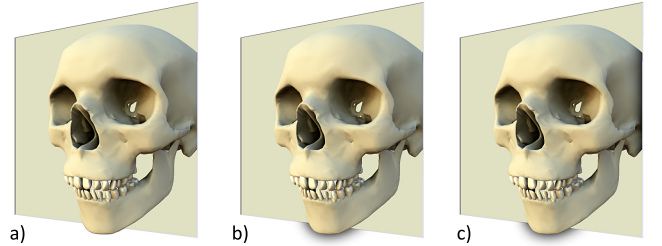
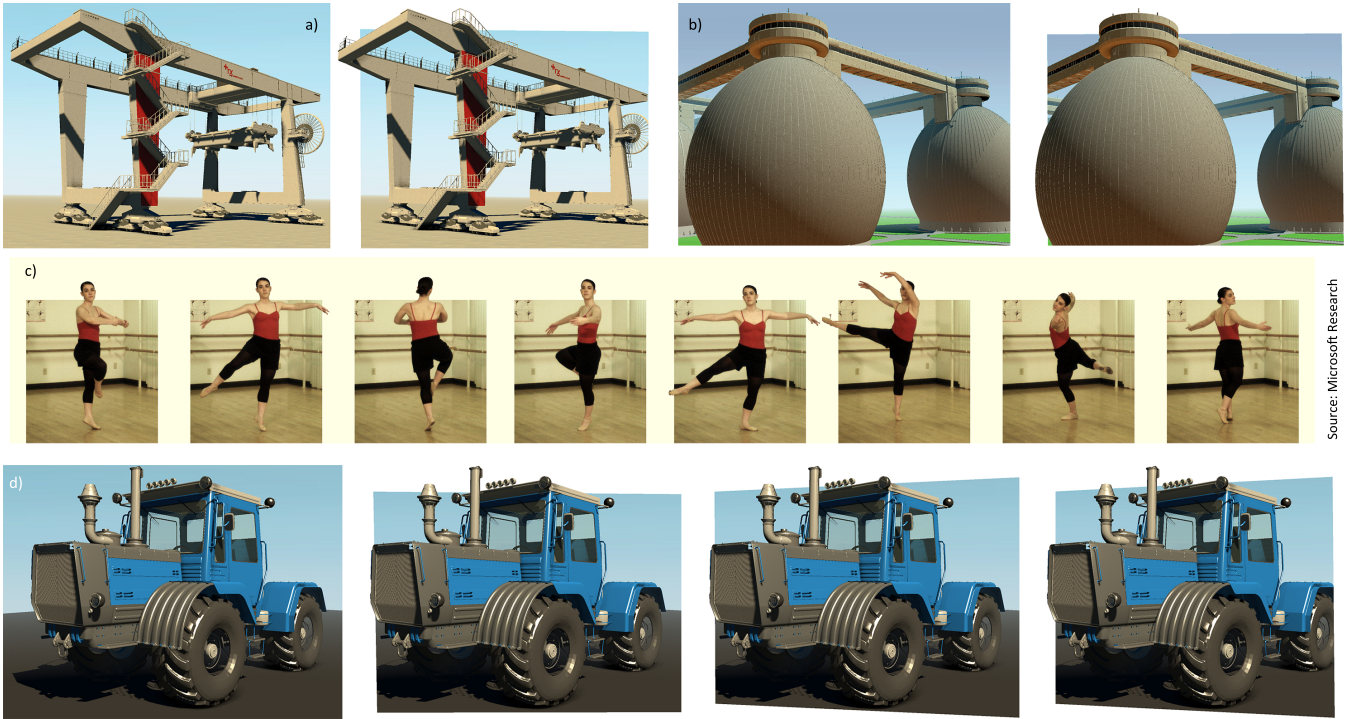


Figure 10: Light transport between the virtual world and the passepartout medium: a) Virtual passepartout. b) Shadow from the skull cast onto the passepartout as well as the c) shadow cast onto the skull by the right side of the passepartout.

tout, e. g., by dropping a shadow from the virtual object onto the passepartout (Fig. 10).

Our approach can be applied to depth produced from computer vision. In Fig. 11, we applied the approach to disparity maps from



Source: Microsoft Research

Figure 7: Input images (left) and results (right) using manual virtual passepartouts. a)–b) real-time rendered 3D geometry. c) frames from a video using virtual passepartout. d) incrementally applying virtual passepartouts (none, simple, perspective, perspective).

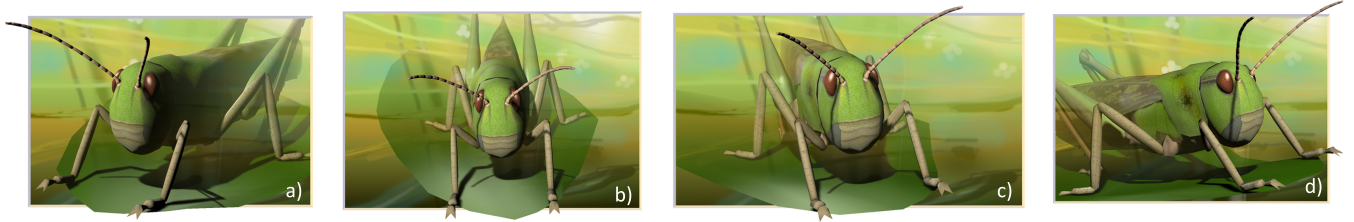


Figure 8: Automatic virtual passepartouts in a 3D scene. Starting from an initial viewpoint with all cuts a), a user moves the camera. In b), cuts are only performed vertically and not horizontally. On c), the left cut is not performed, whereas all cuts are possible in d).

the Middlebury stereo dataset. A morphological opening operation followed by a closing operation (both 8 pixels in size) was used to remove some misclassified pixels.

Our approach can be used for stereoscopic 3D media as well. In the anaglyph Fig. 12, the object appears three-dimensional, yet the occlusion by our virtual frame enhances the effect of actually coming out of the paper. The result was obtained manually, and the passepartout was placed in the screen plane.

Finally, virtual passepartouts can be used to fabricate postcards, flyers or posters with an improved depth impression. To this end – instead of not printing or not setting a pixel – the passepartout is physically cut from the paper as demonstrated in Fig. 13.

5 Discussion

Our pipeline is not applicable to all types of images, but only to the images that already have a potential to use virtual passepartouts. If the foreground object extends beyond the natural image boundaries (e. g., the wing of the airplane in Fig. 9f), the impact of the presented technique is limited. In such a situation we deal with two counter-

acting forces: the virtual passepartout makes the foreground object appear closer to the viewer, but at the same time an impression of distance is introduced by the image boundary. In our opinion the resulting image does not reveal the full potential of the technique. When the clipped part is relatively small (e. g., the seagull’s feet in Fig. 9c), the outcome is even confusing.

On the other hand, the foreground object cannot be *too far* from the periphery: A scene with only one nearby object in the center and large areas of distant objects around it will be unsuitable or at least require overly intrusive cutting, totally changing the image composition. Another class of scenes where usage of the virtual passepartout is limited, are the scenes with visible ground.

6 Conclusion

In this work, we proposed a framework to design “virtual passepartouts” that enhances the depiction of RGBZ images when reproduced using traditional 2D media, e. g., printed paper or non-3D screens. The frame is perceived as a “window” in the medium, e. g., a hole cut in a piece of paper, whereas its placement relates to the three-dimensional organization of the scene – it occludes farther parts of

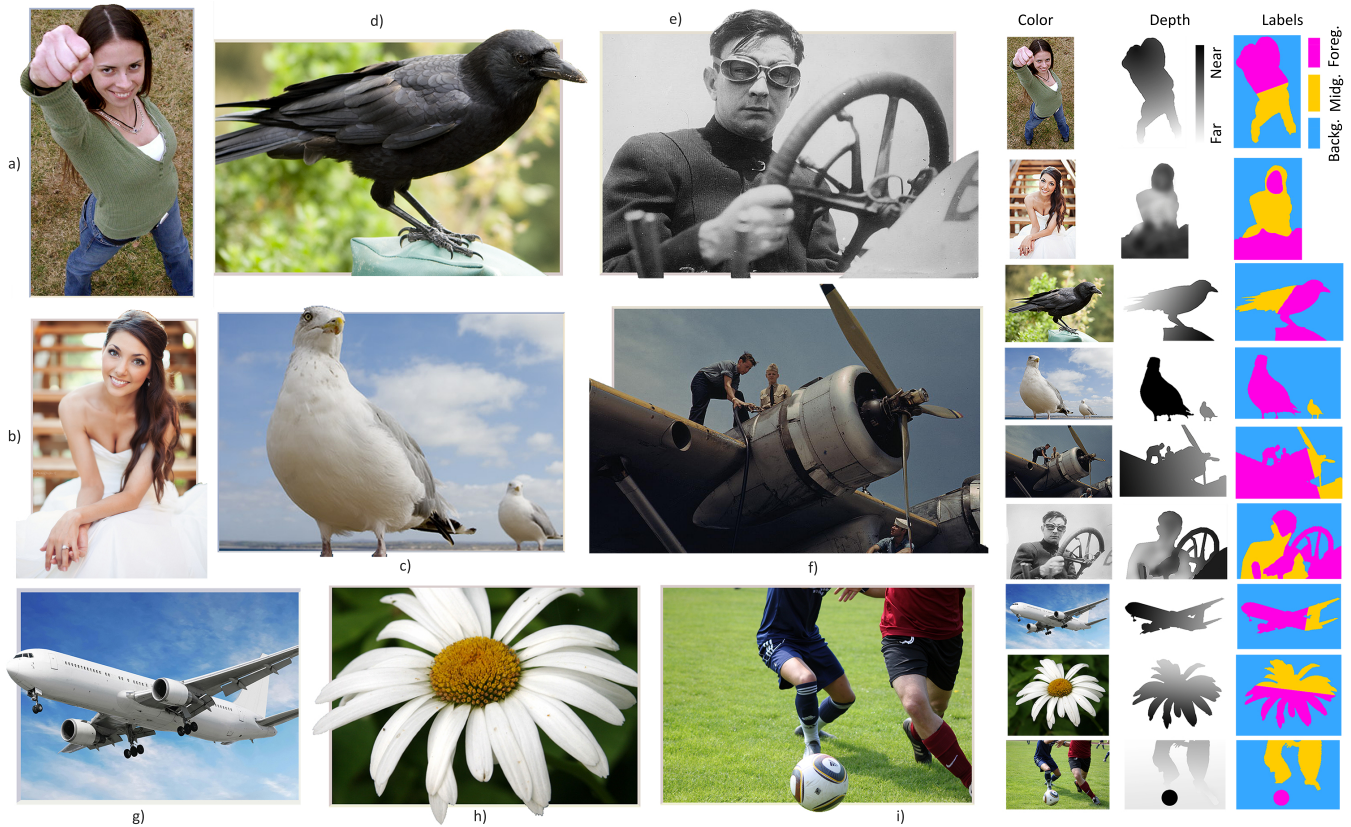


Figure 9: Automatic virtual passepartouts. a) Hand and head escape the frame; the leg is cut; an easy and successful example. b) Head and skirt are closest and escape the frame. The middle ground is not cut and would not be required in this example. c) The front seagull appears closer, but the legs are cut in the original, which makes it less attractive. An artist might also have cut the legs. d) A difficult case, as the crow is shallow, that works well however. g) The labeling of the right wing as middle ground, that is occluded, is unfortunate. If it was labeled foreground, the effect became stronger, as it also would have escaped the frame. h) A successful example resulting a composition that is also popular in professional flower photos. i) Floating objects are easy to handle successfully.

Photos: a) spiked-stock, deviantart.com b) www.leninglass.com c) www.angiemuldowney.co.uk d) Camaryn-Wallpaper, deviantart.com i) Thorsten Bogdenand, pixelio.de

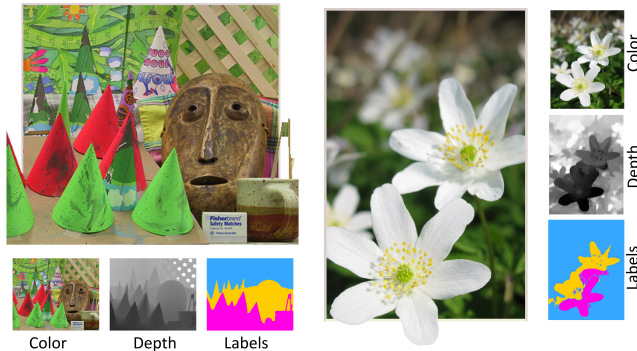


Figure 11: Virtual passepartout for depth produced by computer vision algorithms, here using multi-view stereo (left, Middlebury) and depth-from-defocus by Zhuo and Sim [2009] (right).

the scene, but at the same time it can be occluded by objects closer to the viewpoint. This contributes to the impression that the scene is “detached” from the medium and looks more three-dimensional. Our framework allows to manually or automatically derive passepartouts, that provide additional occlusion cues (Fig. 1).



Figure 12: An anaglyph stereo-based virtual passepartout.
Model: AIM@SHAPE

In future work, it would be worthwhile to include higher order composition rules and optimization over all parameters of our model. More generally, other perceptual illusions such as the Schlossberg’s or Cornsweet-Craik-O’Brien’s illusion could be directly introduced in a 3D scene configuration to create improved perception, i. e., to make an object appear bigger, denser, and so forth. Also, including occlusions into automatic scene arrangement, viewport selection or automatic camera path design is a potential future direction.

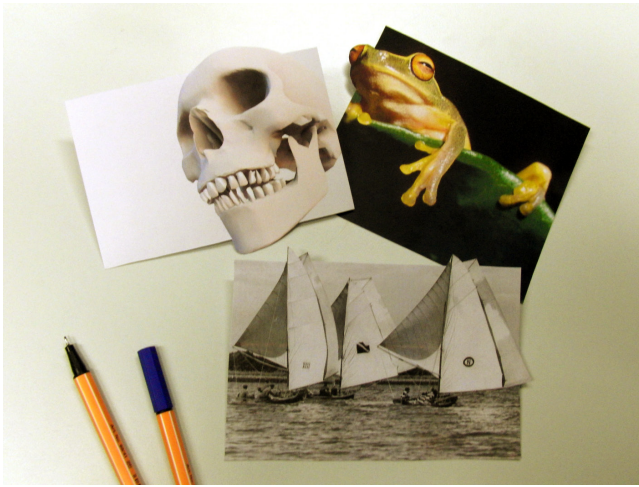


Figure 13: Fabricating a virtual passepartout by cutting paper.

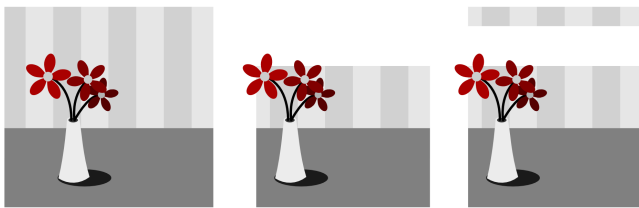


Figure 14: Scenes where the ground is visible (left) are often problematic. The ground at the bottom edge is closer to the viewer than the flower. The desired placement of the virtual passepartout (middle) is such, that the ground is masked out, whereas the flower is not. This, however, requires the far plane to be strongly slanted, and results in an intersection with the wall (right).

References

- ALBERTI, L. 1991. *On painting*. Penguin Classics.
- AMES, J. 1925. The illusion of depth from single pictures. *JOSA* 10, 2, 137–147.
- ARBELAEZ, P., MAIRE, M., FOWLKES, C., AND MALIK, J. 2011. Contour detection and hierarchical image segmentation. *IEEE PAMI*, 99, 1–1.
- BRUCKNER, S., AND GROLLER, M. E. 2006. Exploded views for volume data. *IEEE Trans. Vis. Comput. Graphics* 12, 1077–84.
- BURNS, M., AND FINKELSTEIN, A. 2008. Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM Trans. Graph. (Proc. SIGGRAPH ASIA)* 27, 5, 124:1—124:9.
- CORTES, C., AND VAPNIK, V. 1995. Support-vector networks. In *Machine Learning*, vol. 20, 273–297.
- CUTTING, J., AND VISHTON, P. 1995. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. *Perception of space and motion* 5, 69–117.
- GARDNER, B. R. 2011. The dynamic floating window: a new creative tool for 3d movies. In *Proc. SPIE* 7863, 78631A.
- GOOCH, A., AND GOOCH, B. 2005. Enhancing perceived depth in images via artistic matting. *Proc. CAe*, 83–90.
- HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph.* 24, 577–84.
- HORRY, Y., ANJYO, K.-I., AND ARAI, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proc. SIGGRAPH*, 225–232.
- KELLMAN, P. J., AND SHIPLEY, T. F. 1991. A theory of visual interpolation in object perception. *Cognitive Psychology* 23, 2, 141 – 221.
- LIU, L., CHEN, R., WOLF, L., AND COHEN-OR, D. 2010. Optimizing photo composition. *Computer Graphic Forum (Proc. Eurographics)* 29, 2, 469–478.
- LOPEZ-MORENO, J., JIMENEZ, J., HADAP, S., REINHARD, E., ANJYO, K., AND GUTIERREZ, D. 2010. Stylized depiction of images based on depth perception. In *Proc. NPAR*, 109–118.
- MARSHALL, J. A., MARSHALL, J. A., ARIELY, D., BURBECK, C. A., ARICLY, T. D., ROLLAND, J. P., AND MARTIN, K. E. 1996. Occlusion edge blur: A cue to relative visual depth. *Intl. J. Opt. Soc. Am. A* 13, 681–688.
- MATHER, G., AND SMITH, D. 2004. Combining depth cues: effects upon speed of performance in a depth-ordering task. *Vision Research* 44, 557–562.
- MENDIBURU, B. 2009. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Elsevier.
- MITCHELL, P., AND ROBERTS, L. 1996. *Frameworks: form, function & ornament in European portrait frames*. P. Mitchell.
- PALMER, S. E., BROOKS, J. L., AND LAI, K. S. 2007. The occlusion illusion: Modal completion or apparent distance? *Perception* 36, 5, 650–69.
- PALMER, S. 1999. *Vision science: Photons to phenomenology*, vol. 1. MIT press Cambridge, MA.
- REN, X., FOWLKES, C., AND MALIK, J. 2006. Figure/ground assignment in natural images. *Proc. ECCV*, 614–627.
- RICHARDT, C., STOLL, C., DODGSON, N., SEIDEL, H.-P., AND THEOBALT, C. 2012. Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos. *Computer Graphics Forum (Proc. Eurographics)* 31, 2.
- RITSCHER, T., TEMPLIN, K., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2012. Virtual passepartouts. In *Proc. NPAR*. To appear.
- SUTHERLAND, I. E. 1965. The ultimate display. In *Proceedings of the IFIP Congress*, 506–508.
- VÁZQUEZ, P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2001. Viewpoint selection using viewpoint entropy. In *Proc. VMV*, 273–80.
- WALTZ, D. 1975. Understanding line drawings of scenes with shadows. In *The psychology of computer vision*, McGraw-Hill, P. Winston, Ed., 19–92.
- ZHUO, S., AND SIM, T. 2009. On the recovery of depth from a single defocused image. In *Proc. CAIP*, 889–897.