A Viewer-Centric Editor for 3D Movies

Sanjeev J. Koppal ■ *Harvard University*

C. Lawrence Zitnick, Michael F. Cohen, Sing Bing Kang, and Bryan Ressler ■ *Microsoft Research*

Alex Colburn ■ *University of Washington*

A proposed mathematical framework is the basis for a viewer-centric digital editor for 3D movies that's driven by the audience's perception of the scene. The editing tool allows both shot planning and after-the-fact digital manipulation of the perceived scene shape.

Stereoscopic movies provide slightly different points of view to each eye. The scene points' differing positions (that is, the disparity) creates the illusion of 3D depth (see Figure 1). For many films, the potential to create a visually stunning experience outweighs the extra work needed to overcome the challenges of creating stereo. Stereo requires new planning and postprocessing tools that leverage recent advances in stereo-vision systems. Besides regular 2D film-editing parameters such as field of view (FOV) and camera position, additional degrees of freedom exist, such as camera vergence and interocular distance. Each of these operations has perceptual implications.

Our viewer-centric editing interface provides filmmakers with new degrees of freedom specific to stereo. Our editing technique concentrates on the audience's experience rather than mere manipulation of camera parameters. This is possible because of a mathematical framework that explains previously recorded perceptual effects and abstracts away the camera-centric calculations usually necessary in 3D movies. Stereo-cinematographers can, therefore, concentrate on the desired visual experience while our tool automatically converts the edits into camera parameters. They can use these parameters to render new stereo frames or plan future shots at the same scene.

An Overview of Our Approach

Before the shot, a director obtains rough video or still photographs of the scene. Our interface then offers a digital dry run of the scene by depicting how the audience will perceive the rough cuts. If the predicted stereo experience isn't what the director envisioned, he or she can change the shot plan using new camera parameters calculated by our editing tool. This stereo preview ensures that the stereo rig's configuration will be correct when the real shooting takes place, saving time and money.

After the shooting, our editor can digitally enhance or remove stereo effects, using a variety of tools. These include tools for changing the horizontal image translation or FOV, or modifying the *proscenium arch* (the perceived depth of the screen's edge, also called the floating window). We also allow small changes in the (virtual) camera positions by dollying the camera (moving it backward and forward) and varying its interocular distance (its baseline). Any position shift requires rerendering using precomputed image disparities.

In 3D movies, scene transitions might cause visual discomfort if the shots aren't designed carefully. This is because the human visual system requires time to adjust to drastic changes in visual cues.[2] So, our tool lets users cross-dissolve stereo parameters, such as the horizontal image translation, around shot boundaries even when the shots are hard cuts. This technique can produce more comfortable shot transitions.

 Published by the IEEE Computer Society

The Geometric Framework

A 3D movie experience is the result of a complex combination of factors such as camera parameters, viewing location, projector-screen configuration, and psychological factors. The experience can range from pleasant to distracting or even cause eyestrain (for example, due to long exposures to large disparities). The communities of 3D filmmakers and photographers have learned over the years the various heuristics for avoiding or deliberately enhancing well-known stereo effects. (For a brief look at the history of 3D movies and at current stereo-editing technology, see the related sidebars.)

Table 1 lists the major effects, their representative heuristics, and the heuristics' geometric explanations. We can exploit the geometric explanations to enhance or remove the effects.

Considerable research deals with modeling these distortions.[6–8] We use a unique framework that abstracts the camera-projector-screen-viewer geometry as ratios, allowing easy user manipulation. This editing setup also suggests a geometric interpretation of the major stereo effects. We investigate a rectified stereo setup (see Figure 2) and assume that you can represent the eyes as pinhole cameras with parallel optical axes (as George Wald validated[9]). This approach is most similar to that of Robert Held and Martin Banks,[10] who suggested that geometry is a conservative predictor of what humans can actually fuse. However, whereas Held and Banks investigated the geometric setup's relationship to actual perception, we exploit it to model a movie audience's visual experience.

We assume that several parameters associated with the viewer's experience are known. These include the screen width (S_w), the viewer's distance from the screen (S_z), and the distance between the viewer's eyes (B_e). We assume that all parameters share the same units and that the world coordinates are centered between the viewer's eyes. So, the left and right eyes' positions are $\{-B_e/2, 0, 0\}$ and $\{B_e/2, 0, 0\}$. Let the left and right image widths be W. We use $S_r = S_w/W$ to map pixel locations to a physical-screen location. (Table 2 lists the symbols used in this article.)

Let a corresponding pair of points across the left and right images be $\mathbf{p}_L = (c_L, r_L)$ and $\mathbf{p}_R = (c_R, r_R)$. Because we assume both images are rectified, $r_L = r_R$. After projecting both images onto the screen, we have the corresponding screen locations $\mathbf{p}_{L_S} = (c_{L_S}, r_{L_S})$ and $\mathbf{p}_{R_S} = (c_{R_S}, r_{R_S})$ (see Figure 2). We specify $\mathbf{p}_{L_S}$ and $\mathbf{p}_{R_S}$ in pixels.

When placing the images on the screen, you can take two approaches. Small screens typically use a *vergent configuration*, in which the image centers are at the screen's center. Larger screens commonly use a *parallel configuration*, in which the assumed interocular distance of the eyes offsets the image centers. The following equations are the same for both, except where noted.

The image disparity is $d = (c_R - c_L)$. The screen disparity, $d_S = c_{R_S} - c_{L_S}$, is equal to d for the vergent configuration or to $d_S = d + B_e/S_r$ for the parallel configuration. Using d_S, we can compute the

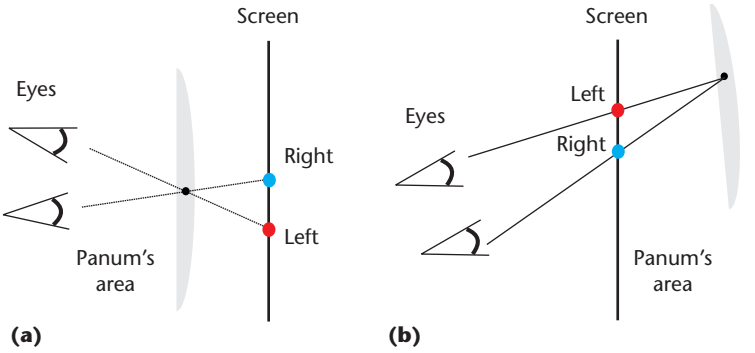


Figure 1. Depth perception (a) in front of and (b) behind a screen. During a 3D movie, the eyes converge on a point, which gives absolute depth to the viewer. In an area around this point (called Panum's area[1]), the brain merges the two images to form a single image, thus perceiving relative depth.

Table 1. Major stereoscopic effects.

Effect	Heuristic or commonly held belief	Geometric explanation
Cardboarding	Keep object "roundness" more than 20 percent.[3]	Camera focal length (f_c) > eye focal length (f_e)
Pinching	Match the eye-camera field of view (FOV).[4]	$f_c < f_e$
Gigantism	A narrow camera baseline causes this effect.[5]	Camera baseline (B_c) < eye baseline (B_e)
Miniaturization	Avoid hyperstereoscopy.[1]	$B_c > B_e$

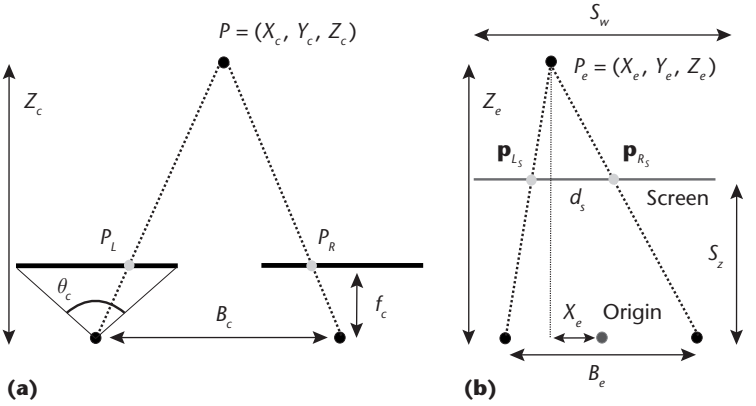


Figure 2. Rectified (a) cameras and (b) eyes. Here we show the disparities created when a rectified stereo pair views a point *P*. Point P_e is the perceived location of *P* when viewed by the eyes on the right. Table 2 explains the symbols in this figure.

Table 2. Symbols used in this article.

Variable	Geometric meaning
(X_c, Y_c, Z_c)	Real-world coordinates of point P
(X_e, Y_e, Z_e)	Perceived coordinates of P
$\mathbf{p}_L = (c_L, r_L)$	Left image coordinates of P (similarly for $\mathbf{p}_R$)
$\mathbf{p}_{Ls} = (c_{Ls}, r_{Ls})$	Left screen coordinates of P (similarly for $\mathbf{p}_{Rs}$)
$\mathbf{p}_{Lb}$	Left screen coordinates at the screen base (similarly for $\mathbf{p}_{Rb}$)
B_e	Eye baseline
B_c	Camera baseline
d	Image disparity
d_S	Screen disparity
f_c	Camera focal length
H	Image height in pixels
K_x	The viewer's horizontal shift
S_z	Viewer screen distance
S_w	Screen width
S_r	Pixel-screen mapping
V_c	Horizontal shift (vergence)
V_{c0}	Original vergence position
θ_c	Camera FOV
θ_{c0}	Original camera FOV
α_θ	Ratio change for FOV
B_{c0}	Original baseline
α_B	Ratio change for baseline
Z_s	Dollying (forward camera shift)
Z_{s0}	Original dolly position
α_Z	Ratio change for the dolly
W	Image width

perceived depth Z_e in both cases. Using similar triangles to equate the base ratios $d_S S_r/B_e$ and the height ratios $(Z_e - S_z)/Z_e$ (see Figure 2), we get

$$Z_e = \frac{B_e S_z}{B_e - d_S S_r}. \quad (1)$$

Similarly, we compute the perceived x-coordinate from the viewer's perspective, X_e, from the two similar right triangles created by projecting P_e along the z dimension (see Figure 2). We equate the base ratios

$$\frac{X_e - S_r\left(c_{Ls} - \frac{W}{2}\right)}{X_e + \frac{B_e}{2}}$$

and the height ratios $(Z_e - S_z)/Z_e$, to get

$$X_e = \frac{Z_e}{S_z}\left[S_r\left(c_{Ls} - \frac{W}{2}\right) + \frac{B_e}{2}\right] - \frac{B_e}{2}. \quad (2)$$

We compute the perceived y-coordinate similarly.

Explaining the Effects

To explain the major stereo effects, we assume an initial configuration in which the camera and eyes have the same FOV and interocular distance. In this situation, the eyes see exactly what the cameras see, and no distortion exists (see Figure 3a). We call this state the *initial case*.

Cardboarding and pinching. Changing the FOV stretches the world in the x and y directions, changing all the parameters in the previous equations. This makes it difficult to depict the perceived behavior simply from the formulas.

Instead, to illustrate the effect, we use a simpler example. We ignore the projector's effect, and the viewer's eyes directly see the image created by the camera. This causes a flattening effect called *cardboarding* for narrower FOVs (see Figure 3b) and causes *pinching* for wider FOVs (see Figure 3c).

Gigantism and miniaturization. Let's start again with the initial case, in which $B_e = B_c$. Without loss of generality, assume we can change B_e instead of B_c, because the change between the two is relative. If we decrease B_e, the denominator in Equation 1 decreases, increasing the depth Z_e. A more direct relationship decreases X_e in Equation 2. This holds both when $Z_e/S_z > 1$ and when the perceived image is in theater space ($(Z_e/S_z) \leq 1$), which causes the signs in Equation 1 to reverse.

This results in *miniaturization*: the viewer perceives the scene as more miniaturized or "toy-like" (see Figure 3d). The opposite effect, *gigantism*, occurs when the camera's interocular distance is smaller than the eyes' interocular distance (see Figure 3e).

Horizontal and vertical viewer motion. We can easily extend the previous math for the viewer's vertical (forward-backward) motion because that implies a new value for S_z. Horizontal (sideways) viewer motion doesn't change Z_e because the motion is parallel to the screen. It does, however, result in a skew-like distortion of the scene shape due to X_e changing. Using K_x as the viewer's horizontal shift, we add the corrective term $(-K_x(Z_e - S_z))/S_z$ to X_e in Equation 2.

Perspective distortion. When a viewer rotates his or her head while moving around the theater space, perspective distortion causes a *keystone* effect. Although conventional movies have largely ignored this effect, it adds vertical disparity to stereo con-

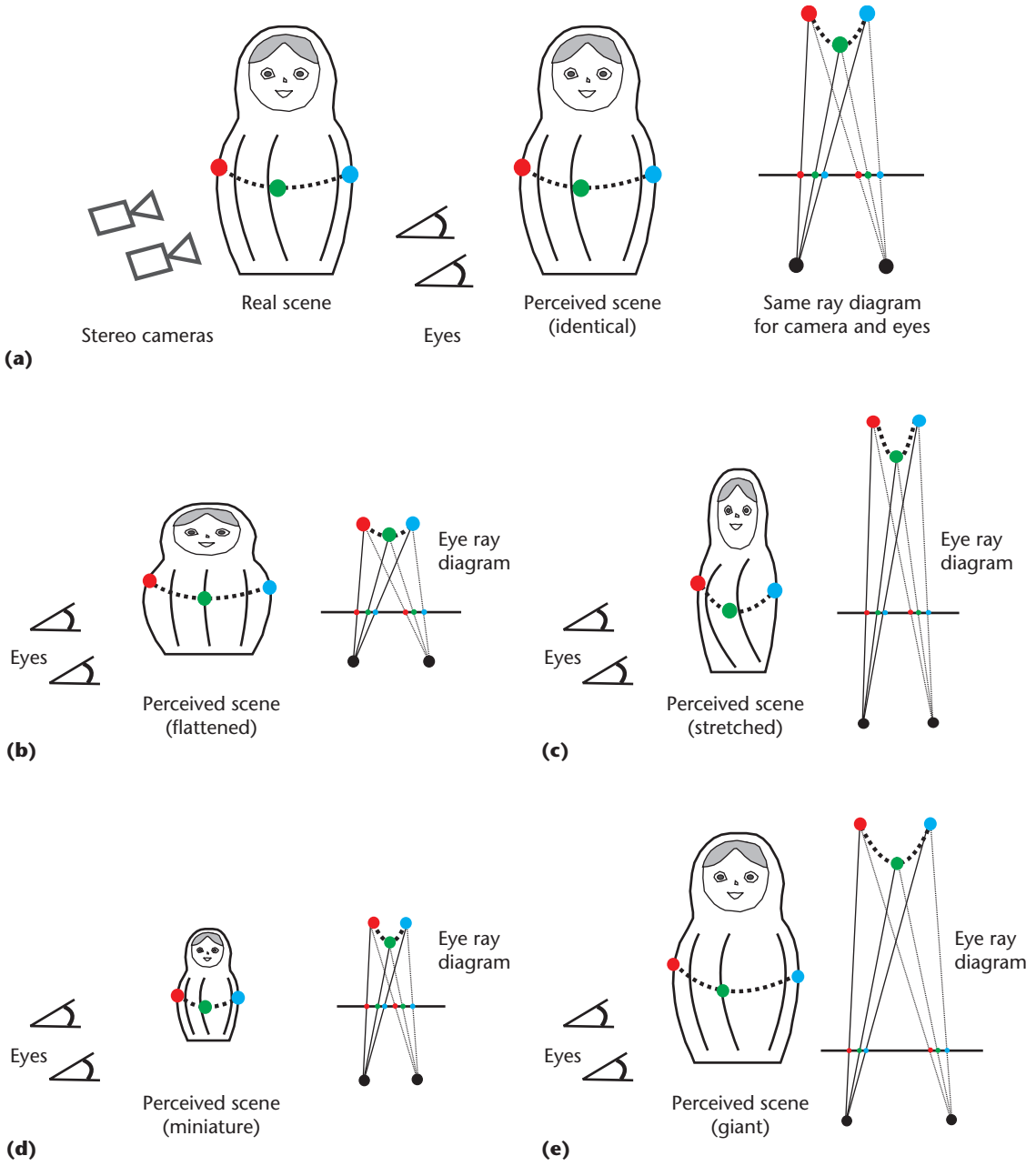


Figure 3. Geometric explanations of well-known 3D movie effects. (a) When the cameras' and eyes' internal parameters (field of view and interocular distance) are the same, the perceived scene is identical to the real world. Any difference between cameras and eyes causes distortions. (b) Narrowing the field of view (FOV) flattens the scene (cardboarding). (c) Widening the FOV elongates the screen (pinching). (d) Increasing the baseline decreases the scene size (miniaturization). (e) Decreasing the baseline increases the scene size (gigantism).

tent. Despite eyestrain, binocular fusion in 3D movies is typically robust to such changes.[1,10]

We let the user decide the amount of tolerable strain by providing the vertical disparity d_V between two corresponding points $\mathbf{p}_{L_S}$ and $\mathbf{p}_{R_S}$, $d_V = (h_L - h_R)$. Assuming the viewer's height is equal to the screen's center, h_L is $\left(r_{L_S} - (H/2)\right)\left(f_e / Z_{p_L}\right)$, where H is the image height in pixels. Z_{p_L}, the Euclidian distance along the x and z dimensions between the left eye and $\mathbf{p}_{L_S}$, is

$$\sqrt{\left(K_x - \frac{B_e \cos\theta}{2}\right)^2 + S_r\left(c_{L_S} - \frac{w}{2}\right)^2 + \left(S_z + \frac{B_e \sin\theta}{2}\right)^2}.$$

We use similar equations for h_R; the sidebar "Viewer's Perspective Distortion" provides their derivation.

Current Stereoscopic-Editing Technology

Because 3D movies have begun significantly affecting studio revenues, researchers and software companies have developed a variety of editing tools. Our tool differs from others in that we designed it solely with the viewer's experience in mind.

Noncommercial Tools

Many editing tools provide significant control but don't model viewer characteristics, such as eye position and parameters, and only directly manipulate the disparity map or the raw images.[1–3] Others are for shot planning on location rather than for postproduction. These include Florian Maier's Stereoscopic Calculator and Robert Mueller and his colleagues' system for easier shooting of live-action 3D movies.[4] Like us, Kenichiro Masaoka and his colleagues use a bird's-eye view of the scene.[5] However, they don't allow user interaction with the reconstructed point cloud for rerendering images. This characteristic lets our editing tool enable the creation of new effects, such as a 3D Hitchcock zoom (for more on this, see the section "Parameter coupling" in the main article).

Commercial Tools

Many commercial editing tools are available, but their inner workings are proprietary and can't be easily compared with our interface. However, our tool is different in that it allows control of the camera position in 3D (unlike Tweak's RV), including dollying the camera forward (unlike the Foundry's Ocula plug-in for Nuke). Quantel's tool provides a broad swath of controls for stereoscopic content. However, these tools' fundamental primitive is to adjust image and camera parameters to achieve the desired 3D view.

In contrast, our tool centers on user interaction with a point cloud that correctly depicts the viewer's 3D experience. The rendered images follow as a by-product of this. In this sense, our viewer-centric tool complements other camera-centric tools. Finally, our editing framework allows blending of all the different stereo parameters over the transitions between cuts, whereas other software is limited to a few of these parameters.

Formats and Glasses

Many formats exist for simultaneously displaying the two stereo images to your eyes. They multiplex the stereo pair either in time (using fast projectors and displays), space (with alternate rows or columns belonging to different images), or wavelength (through the red, green, and blue color channels). As we explain in the main article, we used a polarized color display.

References

1. C. Wang and A. Sawchuk, "Disparity Manipulation for Stereo Images and Video," *Stereoscopic Displays and Applications XIX*, Proc. SPIE, vol. 6803, 2008.
2. M. Suto, "StereoMovie Maker," 2006; http://stereo.jpn.org/eng/stvmkr.
3. T. Kawai et al., "Development of Software for Editing of Stereoscopic 3-D Movies," *Stereoscopic Displays and Virtual Reality Systems IX*, Proc. SPIE, vol. 4660, 2002, pp. 58–65.
4. R. Mueller, C. Ward, and M. Husak, "A Systematized WYSIWYG Pipeline for Digital Stereoscopic 3D Filmmaking," *Stereoscopic Displays and Applications XIX*, Proc. SPIE, vol. 6803, 2008.
5. K. Masaoka et al., "Spatial Distortion Prediction System for Stereoscopic Images," *J. Electronic Imaging*, vol. 15, no. 1, 2006, pp. 013002-1–013002-12.

User-Controlled Parameters

Our editing interface lets the user change the viewer's perception of the scene by varying the four parameters we mentioned earlier: camera FOV (θ_c), the camera's interocular distance (B_c), horizontal image translation (V_c), and the dolly (Z_s).

V_c is similar to changing the cameras' angle of vergence. Assuming the cameras rotate along the y-axis and are rectified in a specific manner, a change in vergence will cause a horizontal image shift.

Changes in θ_c and V_c require resizing and shifting the images, respectively. However, manipulating the interocular distance and dolly require rerendering the scene. This is because changing the interocular distance and dolly result in camera translation, which must account for scene parallax.

We compute the new pixel positions on the basis of the four parameters. We change these values in the order corresponding to a cameraman performing the same changes during video capture: Z_s, B_c, θ_c, and V_c.

Whereas users directly manipulate V_c, they manipulate the other parameters as ratios of the original camera parameters θ_{c0}, B_{c0}, and Z_{s0}:

$$\tan(\theta_c/2) = \alpha_\theta \tan(\theta_{c0}/2), \quad (3)$$

$$B_c = \alpha_B B_{c0}, \quad (4)$$

and

$$Z_s = \alpha_Z Z_{s0}. \quad (5)$$

By definition, $V_{c0} = 0$. From Equations 3 through 5, α_θ scales the image about its center, α_B is the camera baseline's relative change, and α_Z is the "normalized" dolly using the unit distance Z_{s0}. We compute Z_{s0} as a function of the viewer-to-screen

A Brief History of 3D Movies

Charles Wheatstone was arguably the first to discover stereopsis; he defined disparity in terms of differences in subtended angles.[1] Following this research, David Brewster built the first viewing device, called the stereoscope, in 1844. Hermann von Helmholtz and Wilhelm Rollman brought about the anaglyph (red-cyan) format.[2] John Norling introduced the polarized method in the US; Raymond Spottiswoode introduced it in Britain in the 1940s.[2] These and other inventions fueled a boom in 3D movies in the 1950s, which was followed by a checkered run of popularity up to the present.[3]

Along with the rise of 3D movies, many related areas of research and engineering have undergone development. Much research has concentrated on human perception in 3D movies, such as improving the viewing experience[4–6] and understanding dizziness and other physiological effects.[7] 3D displays (including autostereoscopic displays) now let computer users display color 3D movies on the desktop.[7–9] Another area is display technologies that use high-speed mirrors and projectors.[10,11] VR[12–14] and human-computer-interaction applications also exist. Finally, considerable research has focused on finding transmission and encoding protocols and portable display solutions for 3D television.[15,16]

References

1. C. Wheatstone, "On Some Remarkable, and Hitherto Unobserved, Phenomenon of Binocular Vision," *Philosophical Trans. Royal Soc. of London*, vol. 128, 1838, pp. 371–394.
2. L. Lipton, *Foundations of the Stereoscopic Cinema*, Van Nostrand Reinhold, 1982.
3. E. Sammons, *The World of 3D Movies*, 1992; www.3d.curtin.edu.au/cgi-bin/library/sammons.cgi.
4. R. Held and M. Banks, "Misperceptions in Stereoscopic Displays: A Vision Science Perspective," *Proc. 5th Symp. Applied Perception in Graphics and Visualization*, ACM Press, 2008, pp. 23–32.
5. H. Kim et al., "Reconstruction of Stereoscopic Imagery for Visual Comfort," *Stereoscopic Displays and Applications XIX*, Proc. SPIE, vol. 6803, 2008.
6. M. Siegel and S. Nagata, "Just Enough Reality: Comfortable 3-D Viewing via Microstereopsis," *IEEE Circuits and Systems for Video Technology*, vol. 10, no. 1, 2000, pp. 387–396.
7. M. Lambooij, W. Ijsselsteijn, and I. Heynderickx, "Visual Discomfort in Stereoscopic and Autostereoscopic Displays: A Review of Concepts, Measurement Methods, and Empirical Results," *Stereoscopic Displays and Applications XVIII*, Proc. SPIE, vol. 6490, 2007, pp. 64900I.1–64900I.11.
8. T. Tessman, "Perspectives on Stereo," *Stereo Displays and Applications*, Proc. SPIE, vol. 1256, 1990, pp. 22–27.
9. A. Schwerdtner and H. Heidrich, "The Dresden 3D Display," *Stereoscopic Displays and Virtual Reality Systems V*, Proc. SPIE, vol. 3295, 1998, pp. 203–210.
10. A.C. Traub, "Stereoscopic Display Using Rapid Varifocal Mirror Oscillations," *Applied Optics*, vol. 6, no. 6, 1967, pp. 1085–1087.
11. A. Jones et al., "Rendering for an Interactive 360° Light Field Display, *ACM Trans. Graphics*, vol. 26, no. 3, 2007, article 40.
12. G.C. Burdea and P. Coffet, *Virtual Reality Technology*, John Wiley & Sons, 2003.
13. H. Rheingold, *Virtual Reality*, Simon and Schuster, 1992.
14. W. Robinett and J. Rolland, "A Computational Model for the Stereoscopic Optics of a Head-Mounted Display," *Stereoscopic Display and Applications II*, Proc. SPIE, vol. 1457, 1991, pp. 140–160.
15. W. Matusik and H. Pfister, "3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 814–824.
16. H. Isono et al., "50-Inch Autostereoscopic Full-Color 3D TV Display System," *Stereoscopic Displays and Applications*, Proc. SPIE, vol. 1669, 1992, pp. 176–185.

distance as reprojected in camera space. Assuming the viewer and camera have the same FOV, this distance is $S_w/(2\tan(\Theta_{c0}/2)$. Scaling by the relative distance between B_{c0} and B_e, we get

$$Z_{s0} = \frac{B_{c0}S_w}{2B_e \tan\left(\frac{\theta_{c0}}{2}\right)}.$$

Casting user-controlled quantities as ratios is useful when camera parameters are hard to quantify or are unknown. If the users desire only post-production effects, the camera parameters aren't needed. However, to plan a shot, you must know the original camera parameters. Our key assumption in using ratios is that by directly manipulating the stereo effect, we're indirectly changing the camera parameters that caused it. This is supported by the linearity of Equation 2 and Equations 3 through 5 in the four parameters. For example, we'll scale the scene in a manner inversely proportional to the camera interocular ratio α_B. So, we're addressing gigantism and miniaturization by changing the scene shape, which is equivalent to changing the camera baseline.

We use Equations 1 and 2 to compute the original X_e and Z_e coordinates before any manipulations using the original screen column location c_{Ls} and screen disparity d_S for pixel $\mathbf{p}_{Ls}$. Applying the changes in the camera's interocular distance and the dolly, we find a new set of 3D perceived coordinates $\bar{X}_e$ and $\bar{Z}_e$:

Viewer's Perspective Distortion

If we know Z_{p_L} and Z_{p_R}, the Euclidian distances between the left and right eyes and the screen coordinates of point P, we can get the maximum vertical disparity. (For an explanation of the other symbols used in this sidebar, see Tables 1 and 2 in the main article.) The height h of $\mathbf{p}_{L_S}$ is $S_r\left(r_{L_S} - (H/2)\right)$ and is identical (owing to rectification) for $\mathbf{p}_{R_S}$. The projected heights of these screen images of P ($\mathbf{p}_{L_S}$ and $\mathbf{p}_{R_S}$) on the retina of the left and right eyes are h_{left} and h_{right}. We scale these by S_r:

$$h_{\text{left}} = \frac{1}{S_r}\left(S_r\left(r_{L_S} - \frac{H}{2}\right)\right)\frac{f_e}{Z_{p_L}} = \left(r_{L_S} - \frac{H}{2}\right)\frac{f_e}{Z_{p_L}}$$

and

$$h_{\text{right}} = \left(r_{R_S} - \frac{H}{2}\right)\frac{f_e}{Z_{p_R}}.$$

The distortion d_v is the difference between the two heights, which should be 0 in the case of no distortion:

$$d_v = h_{\text{left}} - h_{\text{right}}.$$

The x-coordinates of $\mathbf{p}_{L_S}$ and $\mathbf{p}_{R_S}$ are c_{L_S} and c_{R_S}. Consider from Figure A the right triangle ($\mathbf{p}_{L_b}$, L, $B1$). This triangle's base is

$$(B1, L) = \left(K_x - (B_e/2)\cos\theta\right) + S_r\left(c_{L_S} - (W/2)\right).$$

Its height is $(B1, \mathbf{p}_{L_b}) = S_z + (B_e/2)\sin\theta$. So,

$$Z_{\mathbf{p}_L} = \sqrt{\left(K_x - \frac{B_e}{2}\cos\theta + S_r\left(c_{L_S} - \frac{W}{2}\right)\right)^2 + \left(S_z + \frac{B_e}{2}\sin\theta\right)^2}.$$

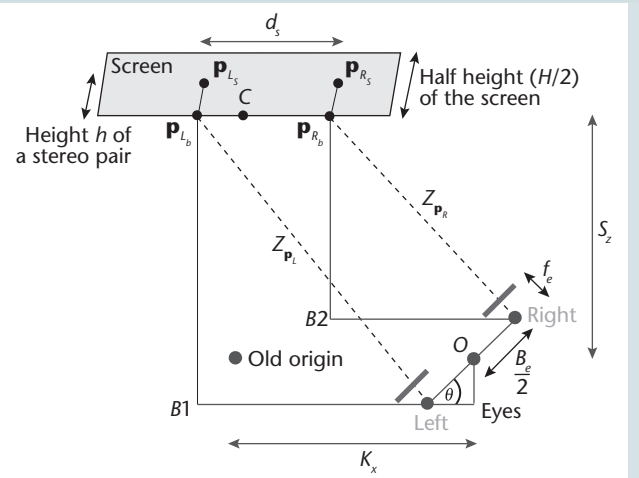


Figure A. Perspective distortion. Except for points $\mathbf{p}_{L_S}$ and $\mathbf{p}_{R_S}$, all other points and lines are on the "middle" plane parallel to the ground. The viewer has moved horizontally by K_x and tilted his or her head by θ to the x-axis. We project the height h of $\mathbf{p}_{L_S}$ onto the left eye's retina and the height h of $\mathbf{p}_{R_S}$ onto the right eye's retina. The difference of the two projected heights gives us the maximum vertical disparity. (For an explanation of the symbols used in this figure, see Table 2 in the main article.)

Now consider from Figure A the right triangle ($\mathbf{p}_{R_b}$, R, $B2$). This triangle's base is

$$(B2, R) = \left(K_x + (B_e/2)\cos\theta - S_r\left(c_{R_S} - (W/2)\right)\right).$$

Its height is $(B1, \mathbf{p}_{R_b}) = S_z - \frac{B_e}{2}\sin\theta$. So,

$$Z_{\mathbf{p}_R} = \sqrt{\left(K_x + \frac{B_e}{2}\cos\theta - S_r\left(c_{R_S} - \frac{W}{2}\right)\right)^2 + \left(S_z - \frac{B_e}{2}\sin\theta\right)^2}.$$

$$\bar{Z}_e = \frac{Z_e + S_z\alpha_Z - S_z}{\alpha_B},$$

$$\bar{X}_e = \frac{X_e}{\alpha_B}.$$

Next, we can project the transformed point onto the movie screen to find a new set of screen coordinates $(\bar{c}_{L_S}, \bar{r}_{L_S})$ and screen disparity $\bar{d}_S$:

$$\bar{c}_{L_S} = \frac{\left(2\bar{X}_e S_z - B_e\bar{Z}_e + B_e S_z\right)}{2\bar{Z}_e S_r} + \frac{W}{2}.$$

We can similarly compute the value of $\bar{c}_{R_S}$, after which we can compute the new disparity $\bar{d}_S = \bar{c}_{R_S} - \bar{c}_{L_S}$. We then apply our FOV and horizontal-image-translation changes to find the new screen coordinates (c'_{L_S}, r'_{L_S}) and warped screen disparity d'_S:

$$c'_{L_S} = \alpha_\theta\left(\bar{c}_{L_S} - \frac{W}{2}\right) + \frac{W}{2} - \frac{V_c}{2},$$

$$d'_S = \alpha_\theta\bar{d}_S + V_c,$$

and

$$c'_{R_S} = c'_{L_S} + d'_S.$$

The previous three equations assume a vergent configuration. For a parallel configuration, we would also have to shift the images in the x direction (by $B_e/2S_r$) before and after scaling.

The Editing Tool

Whereas a human fuses a stereo pair to perceive depth, a stereo algorithm can reconstruct the scene from the same images. Our interface's key contribution is that the user directly manipulates the world's shape as a viewer perceives it. This is enabled by a top-down, bird's-eye view of the perceived scene's point cloud, (see Figure 4). To automatically generate the image disparities and render a new set of stereo images given the edited parameters, we use C. Lawrence Zitnick and his colleagues' algorithm.[11] (We shot all the stereo examples in this article using the stereo rig in Figure 5. The large rig baseline is only for presentation purposes.)

Editing with the Box Widget

A box widget lets users easily manipulate the world's perceived shape. As Figure 4 shows, the interface overlays the box on the perceived scene points. The user manipulates various parts of the box to effect specific changes. For examples of these changes, see our accompanying videos at www.koppal.com/stereoscopy.html.

When this box is exactly a square, there's zero distortion for the viewer. The box's shape summarizes the stereo effects in the rendered images. For example, cardboarding or pinching corresponds to a flattening or elongation (respectively) of this square. The user can change the perceived scene shape (and subsequently rerender new stereo images) by manipulating the box in the following ways.

Adding or enhancing cardboarding and pinching. Users can change the FOV by dragging the purple dot on the box's side; this also changes the original camera focal length. The box's distortion mirrors the pinching that occurs with wider FOVs.

Figure 6 shows a scene whose FOV has been

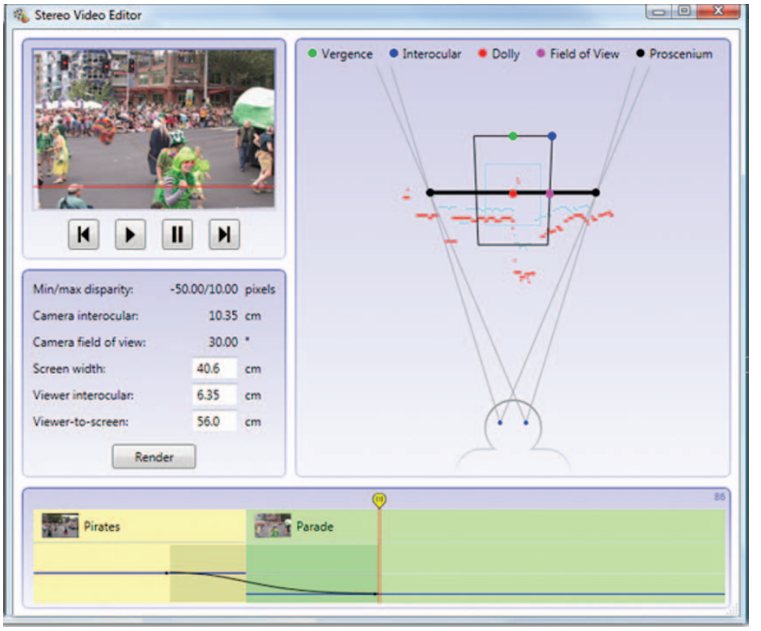


Figure 4. Our stereo-editing tool's interface. It offers a bird's-eye view of the scene in the upper-right section. In this view, users can control the stereo parameters. At the bottom, the timeline allows cross-fading of stereo parameters across scene transitions. Users interact with the point cloud through changes in horizontal image translation, the FOV, the dolly, and the interocular distance. In addition, they can adjust the perceived screen edge depths using the proscenium arch.

(a) **(b)**

Figure 5. Our stereo rig. (a) A close-up view. (b) The rig placed among the audience during a parade. To collect the data in our edited movies, we used two Canon HD XLH1 cameras that we synchronized and placed on a metal stereo rig that allows normal pan-tilt motion. (The large baseline is only for presentation purposes.)

Figure 6. A sequence in which we used our editing tool to decrease an image's FOV. Doing this caused flattening.

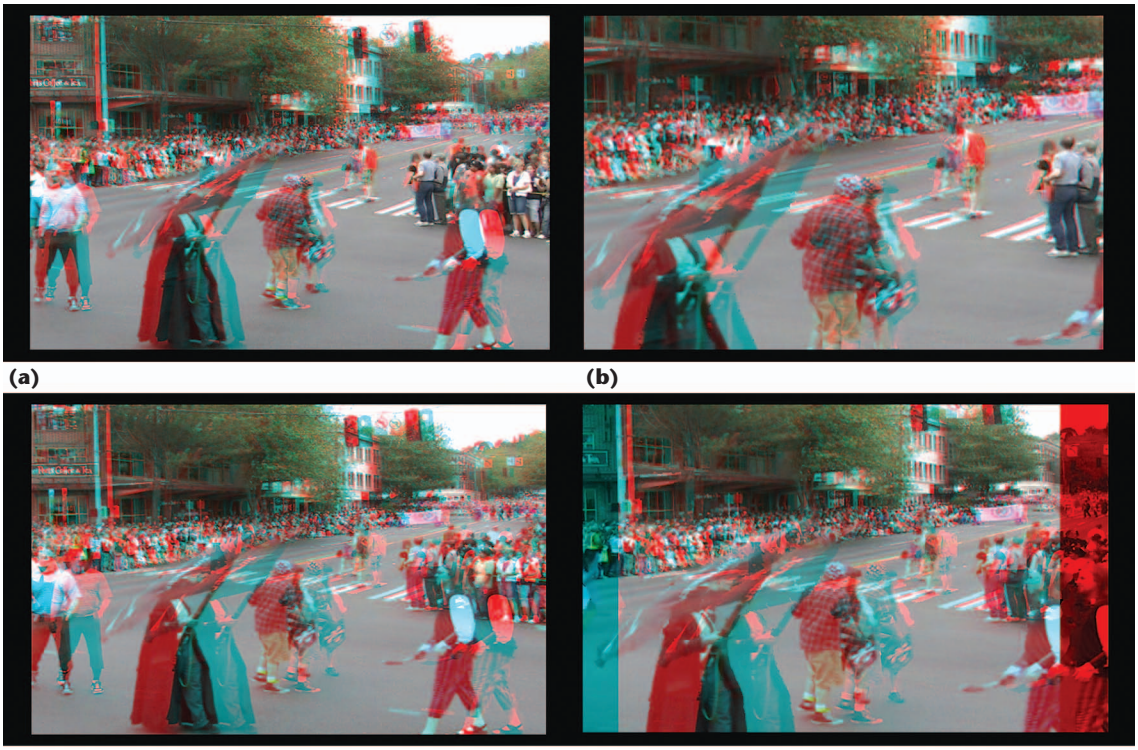


Figure 7. Sample output of our viewer-centric editing tool. (a) The input stereo pair. (b) A decrease in FOV. (c) A modified horizontal image translation. (d) A virtual camera moving forward (a dolly maneuver) as the proscenium arch (the perceived depths of vertical window edges) changes. All the results display parallax changes.

digitally decreased. The foreground tree appears flat in the final image. Figure 7a shows an input stereo pair; Figure 7b shows another example of changing the FOV.

Although we developed our 3D movies in polarized format, we present the images here in anaglyph format for convenience. However, this format might contain compression artifacts, such as bleeding between colors, that weren't present during development or in our user studies. You can obtain anaglyph glasses from a store such as 3D Glasses Direct (www.3dglassesdirect.com). They must have red for the left eye and cyan for the right eye.

Translating images left and right. Recall that the parts of the scene with zero disparity appear to be on the screen. Changing the horizontal image translation changes the parts of the scene that appear to be on the screen. Users translate the images by moving the green dot at the box's top up or down. This shifts the left and right stereo frames in the *x* direction. This action distorts the 3D scene shape nonuniformly. The flag holder in Figure 7c was shifted out of screen space and now appears closer.

Translating a scene forward or backward. The user dollies the camera (changes the camera-scene distance) by dragging the red dot in the square's center. As the scene gets closer to the viewer, the virtual cameras move closer to the scene. The dolly causes no distortions to the box widget or point cloud because it accounts for parallax effects (which are depth dependent). The extent to which you can dolly depends on the stereo data's quality. Although small shifts are possible, they might considerably change the stereo experience (see Figure 7d).

Scaling the perceived scene size. By dragging the blue dot on the box's corner, the user can scale the scene to make it appear larger or smaller. This effect changes the camera baseline and is identical to miniaturization and gigantism. In Figure 8, the interocular distance decreases, and the figures appear larger than life, relative to the viewer.

Parameter coupling. Our system lets users combine different camera parameters to create new stereo effects. One example is the 3D equivalent of the Hitchcock zoom pioneered in the film *Vertigo*. We demonstrate our own form of 3D Hitchcock zoom by coupling our editing tool's dolly, FOV, and image translation parameters. For a parallel configuration, only the dolly and FOV must be coupled. Figure 9 shows how we applied this zoom to a scene in which the foreground girl's size is stabilized in the image, and her position is stabilized in 3D.

(a) (b)

Figure 8. Decreasing the interocular distance. (a) The original image. (b) The image with the decreased baseline. This action makes the figures appear larger than life, relative to the viewer.

(a) (b) (c)

Figure 9. Our 3D Hitchcock zoom: (a) flat, (b) normal, and (c) stretched. Our editing tool created this effect by coupling its dolly, FOV, and image translation parameters.

(a) (b)

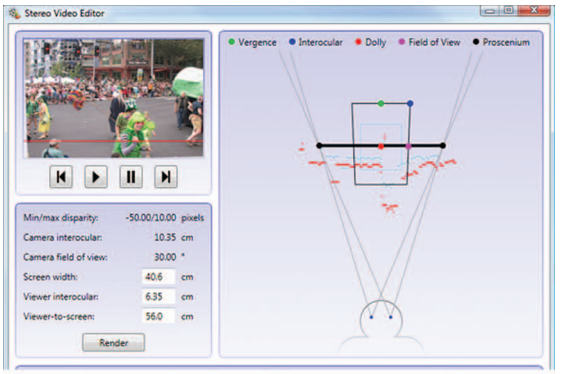
Figure 10. A scene (a) without and (b) with the proscenium arch, which removes objects at the FOV's edge. When the proscenium arch is appropriately positioned, objects near the image's edge, such as the statue, become easier to fuse.

Shifting the proscenium arch. In many stereo shots with objects appearing in front of the screen, there are regions on the screen's edges that only one eye can see. These areas appear inconsistent with the scene edges and can cause eyestrain. The proscenium arch simply blacks out part of the stereo frame to move the screen's perceived edge closer to the viewer.

Our interface has black markers for the image's left and right vertical edges. Users adjust the markers' lengths by moving them along the line of sight. Figure 10 shows an image with aligned edges. When the proscenium arch is appropriately positioned, objects near the image's edge, such as the statue, become easier to fuse.

Planning a Capture Session

Shooting a 3D movie is difficult precisely because it's challenging to imagine how the audience's experience will differ from the director's vision. Our interface addresses this problem by providing a way to plan the shot, given rough takes of the



(a)

(b)

(c)

Figure 11. Shot planning. (a) Our interface. (b) The shot used for planning. (c) The shot taken with the estimated parameters. The user plugs the images into our editor to perceive a bird's-eye view of the scene.

scene or still images. Figure 11a shows shot planning with our interface; Figures 11b and 11c show a scene of two people in front of trees. We assume this sort of still "prototype" shot requires little or no effort. Users can plug these images into our editor, and, as the figure shows, they'll perceive a bird's-eye view of the scene.

Given this setup, the director might wish to change some aspect of the scene. For example, the director might wish the viewers to perceive the two people as being closer together. By adjusting the point cloud in our interface's top-down view, users can change the original camera parameters. The interface then outputs the desired camera parameters as ratios of the parameters used to generate the rough takes.

The examples in this article use real imagery, which requires computer vision technology to compute depth. A more direct use of our tool would be with synthetic imagery, in which you can directly extract the depths from the 3D model. Our stereo-editing tool could then be integrated with the geometric modeler so that users could edit the stereo parameters and change to the 3D scene in the same application.

Creating Postproduction Effects

An important capability in a movie-editing tool is cutting between shots, because many times the story is told by switching between contrasting scenes.[12] Recent trends in film and TV have tended toward multiple cuts a minute and many cuts per second. The former is now common in prologues of crime dramas such as *CSI* and *24*.

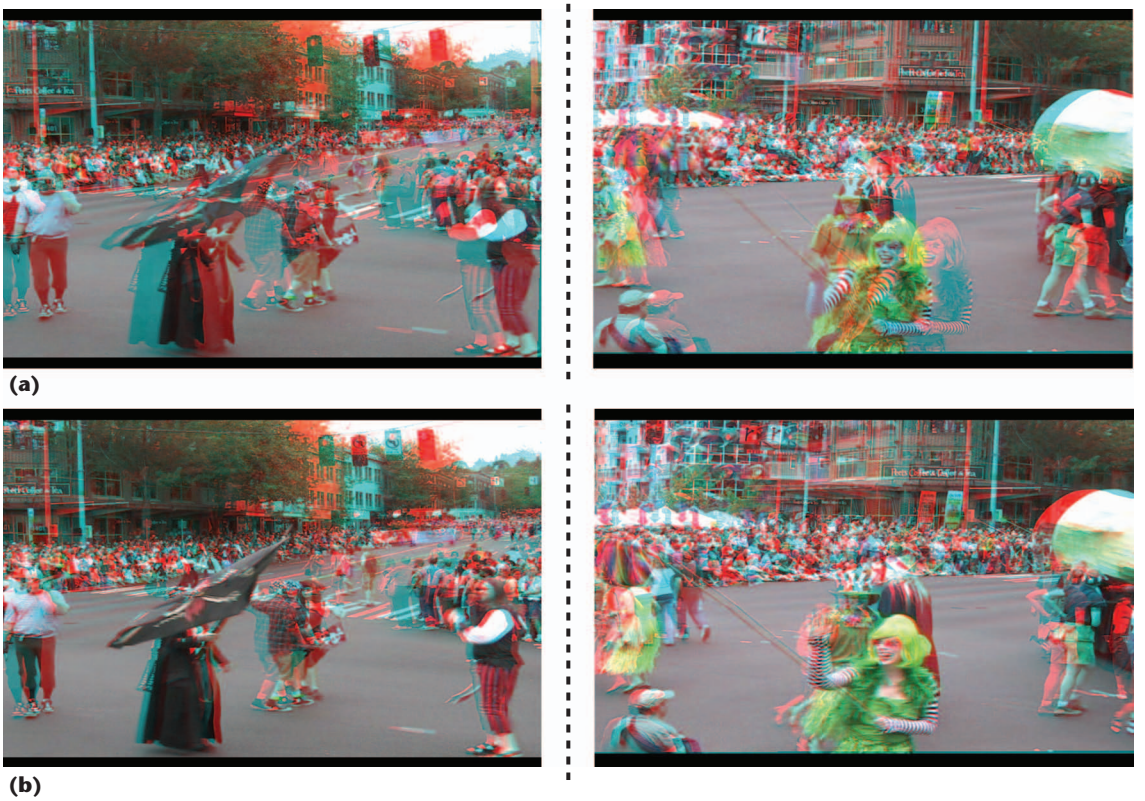
For stereo content, the potential for visual discomfort in these cases is large because there's a lag time in fusing scenes at differing depths. One way to mitigate this issue is to blend the horizontal image translation during a cut so that the objects of interest have the same depth at the cut. The subtle shifting of image translation before and after the cut can be done without the audience noticing it.

Figure 12 summarizes such a cut. In the first scene, the flag is shot with negative disparity and appears to be behind the screen. In the next clip, the green-haired girl appears in front of the screen, resulting in a jarring jump as the viewer quickly readjusts. Using our editor, users can select an object in the previous and next clip and select a disparity, as in Figure 4. The user can blend the global disparities before and after the cut so that at the cut, the two objects have identical disparities. This produces a more visually pleasing transition.

Another application of horizontal image translation exploits its two properties:

- Viewers usually don't notice global disparity changes through image shifts.
- Full image fusion occurs after a short time lag.

If the scene cuts back and forth faster than this time lag (which might vary from person to person), we hypothesize that viewers will first fuse objects with disparities similar to those of the currently fused area. So, directing the audience's attention seems possible, as in Figure 13. Anecdotal



(a)

(b)

Figure 12. Easing the transition between two clips by crossfading the horizontal image translation to zero at the cut. (a) A direct transition. (b) Cross-fading the horizontal image shift.

evidence from a small set of users confirmed that by adjusting the areas of similar disparity using horizontal image translation, we could shift the audience's attention across the scene.

A Discussion on Usability

Our interface is driven by two stereo-editing innovations. The first is the bird's-eye view, which offers users the choice of working without stereo glasses. This is possible because our editing tool shows what the audience will perceive in theater space. Users can try different theater dimensions and even change the interocular distance and viewer position to see exactly what a specific viewer would perceive. Users can also quickly view any of the rendered scenes in stereo format. The second innovation is the box widget, which allows an intuitive depiction of stereo distortions, without users having to understand projective geometry's intricacies.

We created our interface's other building blocks from widely accepted technologies. For example, the timeline showing the current clip is also in many successful editing programs, such as Adobe Premiere and Apple Final Cut Pro. Furthermore, our editing tool allows real-time manipulation of the clips. Although rendering times depend on the back-end used, users can obtain low-resolution rough cuts in a few seconds.

So, we believe our editing tool eases editing of 3D movies. However, we leave a user study on its usability for the future. Instead, in the next section, we explore the more fundamental questions

(a) (b) (c)

Figure 13. Transitioning between a (a) clip and a (b) background and (c) foreground version of that clip with different horizontal image translations. These actions direct the user's attention to the foreground or background, respectively.

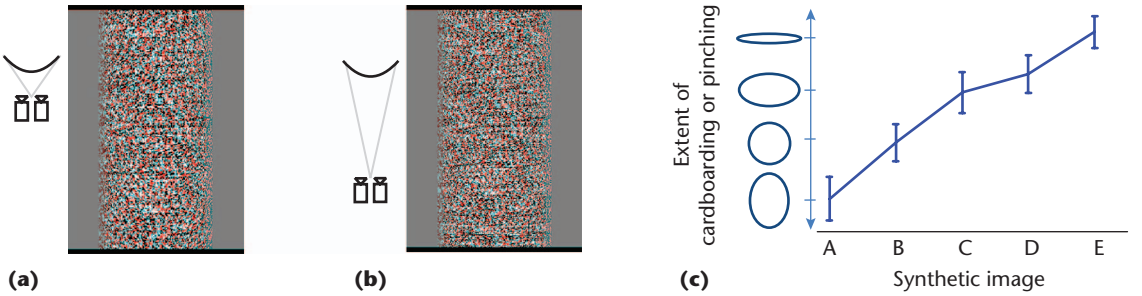


Figure 14. Images and results for the user study of the FOV-and-dolly parameter. (a) Close-in dolly, wide FOV (pinched). (b) Far dolly, narrow FOV (cardboarded). (c) The mean response across 30 users. Participants ranked the cardboarding or pinching of five synthetic images by matching the icons shown on the graph's *y*-axis. Cardboarding or pinching occurred when the FOV and dolly varied.

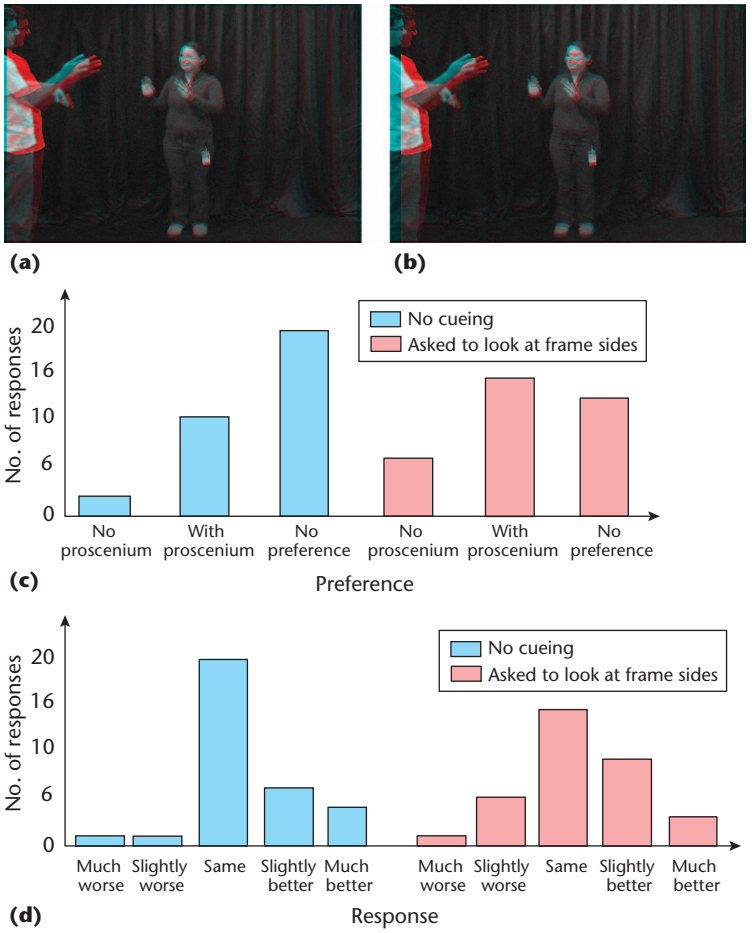


Figure 15. Images and results for the user study of the proscenium-arch parameter. (a) In the raw video, the left eye sees more of the man. (b) The proscenium arch corrects this. Participants (c) indicated which video sequence they preferred and (d) compared the proscenium video to the original. The results indicated that the proscenium arch positively affected the participants' experience.

of whether users perceive the created effects and how the frame rendering affects quality.

Perceptual Studies for the Stereo-Geometry Model

Predicting how a stereo-editing tool's changes will affect viewer perception is challenging. We performed user studies to validate our ideas of how the audience reacts to certain types of editing. We studied the four parameters we described earlier (dolly and FOV, proscenium arch, horizontal image translation, and interocular distance) under controlled scenarios.

We started with 32 potential participants: 23 men and 9 women from a variety of cultural backgrounds. We determined that two of them had stereo blindness, so we excluded them from the studies. To reduce bias, we randomly permuted the experiments' order. We showed short 3D movie clips (no audio) in color on a Hyundai P240W 3D monitor to users wearing polarized glasses. Our multiple-choice questions contained ternary (yes/no/maybe) and Likert-scale (rating from best to worst) answers.

We performed chi-squared statistical-significance tests on our studies. We assumed uniform random distribution (33 percent each for ternary questions and 20 percent for Likert-scale questions). For all our experiments, $p \leq 0.01$.

We investigated the following four questions.

Question 1

Does changing the camera FOV cause cardboarding and pinching? Our participants viewed five synthetic stereo images of a cylinder, with varying distances between the camera and cylinder (see Figures 14a and 14b). We kept the object's image size constant by appropriately adjusting the focal length. The participants ordered these cylinders by cross section, selecting from the icons in Figure 14c. Of the users' rankings, 84.2 percent agreed with the predictions of the model linking cardboarding and FOV. Their responses' mean values closely matched the actual cross-section order.

Question 2

Does the proscenium arch improve viewing comfort? Participants viewed videos of two people in conversation. In the raw video (see Figure 15a), the

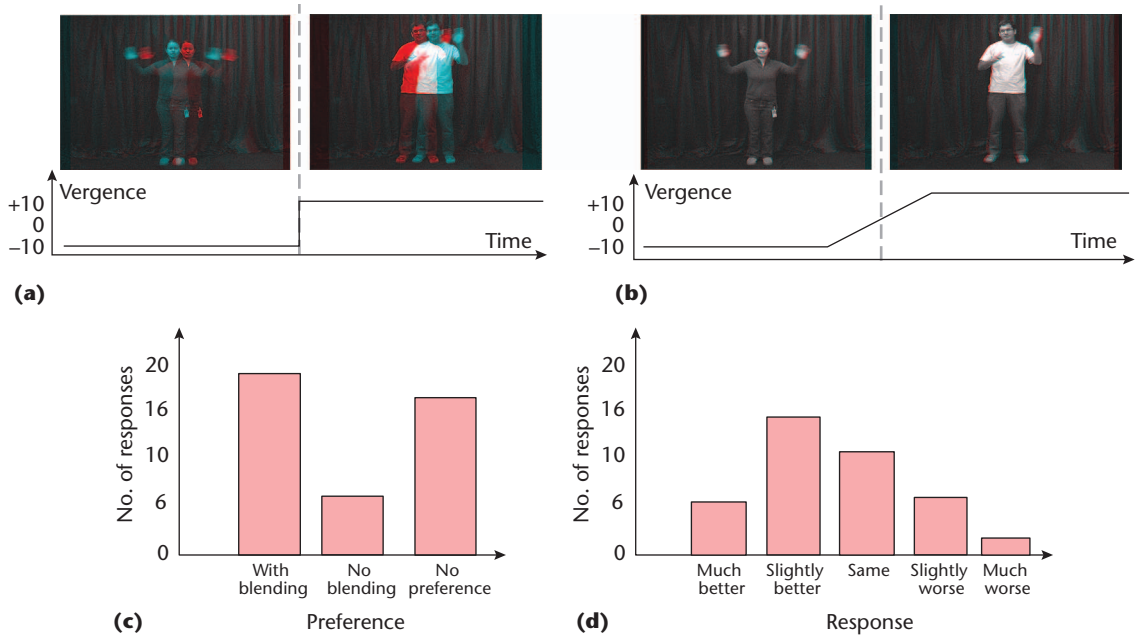


Figure 16. Images and results for the user study of the horizontal-image-translation parameter. (a) A direct transition. (b) A blended transition. (c) A graph indicating which video sequence the participants preferred. (d) A graph describing the participants' comparison of the blended video with the original video. The results indicate that the image translation cross-fade eased the scene transition.

right camera sees more of the man than the left. In the edited video (see Figure 15b), we adjusted the proscenium arch to make the left and right views more even.

The experiment had two stages. First, the participants rated the two videos. Of the participants, 93.1 percent didn't prefer the raw video. We then told them to again compare the videos, paying attention to the screen edges. As we expected, a similar percentage of the participants (93.3) preferred the video with the proscenium arch. The graphs' shift to the right in Figures 15c and 15d indicates these results.

Question 3

Does cross-fading image translations ease scene transitions? Participants watched two videos in succession. The first was of a woman waving, with negative disparity (the woman appeared to be in front of the screen). The second was of a man waving, with positive disparity (the man appeared to be behind the screen). In Figure 16a, the transition is abrupt. In Figure 16b, we shifted the left and right frames before and after the cut so that the two people's depths are at zero disparity at the cut.

The participants then rated the scene transition. Figure 16c indicates that most participants either had no preference or preferred the blended video. Supporting this, Figure 16d indicates that a vast majority (85.7 percent) of the participants preferred the edited transition.

Question 4

Does rendering with a new interocular distance affect quality? A stereo-editing tool might render images by changing the interocular distance using a stereo reconstruction of the scene. How would the result compare to the original, if it was available?

Figures 17a and 17b compare a pair of such videos for a toy train moving on a circular track. We chose this scene because it's repeatable and the stereo reconstruction has artifacts. Figure 17c shows that if the original were available, many participants would prefer it (although many participants had no preference). However, a finer-grained question (see Figure 17d) shows that most participants (75.5 percent) chose the middle three ratings: slightly better, identical, or slightly worse. We included "slightly worse" because, in practice, shooting the same scene at many different interocular distances is expensive and inconvenient. So, this result is positive because, in postproduction, the actual footage with edited baselines won't be available. We tried to remove stereo artifacts by blurring the rendered view and found qualitative evidence that some participants couldn't detect any differences, supporting similar previous research.[13]

Although our user studies exhibited strong trends, we must provide some caveats. For example, the participants might not represent the general 3D movie audience, so the results might

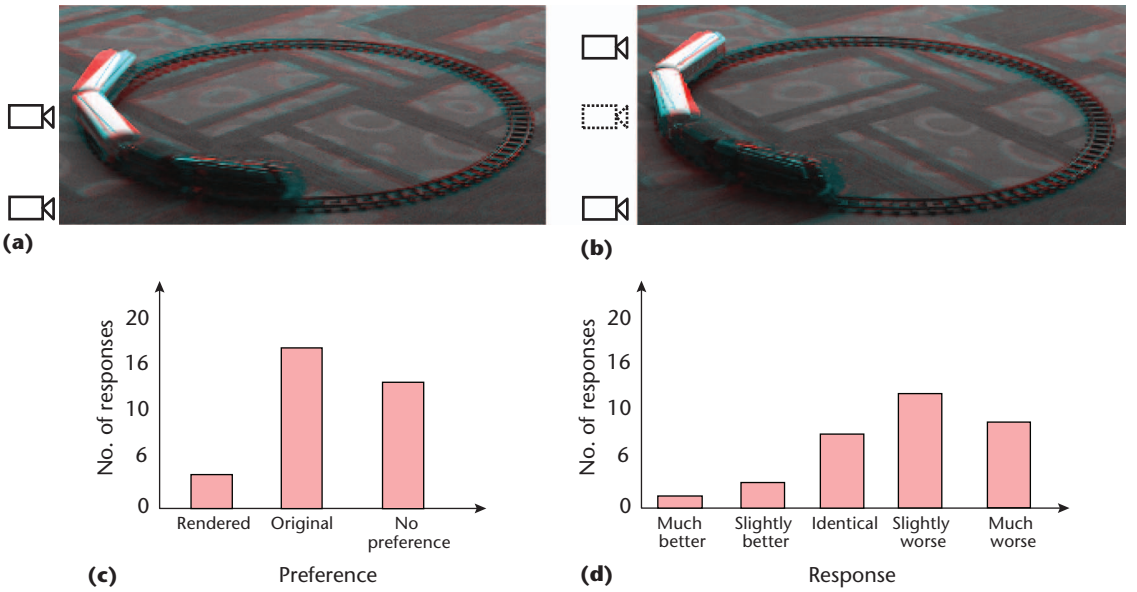


Figure 17. Images and results for the user study of the interocular-distance parameter. (a) A real stereo image of a train scene. (b) The rendered left image from a virtual camera. (c) A graph indicating which video sequence the participants preferred. (d) A graph describing the participants' comparison of the blended video with the original video. Participants considered the videos with a rendered left video stream to be as good as or only slightly worse than the real stereo video.

not accurately reflect how people experience 3D movies. This is because each clip was short and, in some cases, we cued the participants (for example, in the second part of the proscenium-arch experiment, we asked them to notice the sides). However, these studies suggest you can reasonably predict certain perceived effects, and they support our goal of building a stereo-editing tool.

References

1. L. Lipton, *Foundations of the Stereoscopic Cinema*, Van Nostrand Reinhold, 1982.
2. E. Levonian, "Stereoscopic Cinematography: Its Analysis with Respect to the Transmission of the Visual Image," MA thesis, Univ. of Southern California, 1954.
3. M. Empey and R. Neuman, "Stereoscopic Depth as a Storytelling Tool," presentation at 2008 Conf. Animation, Effects, Games, and Interactive Media (FMX 08), 2008.
4. B. Clark, "The 10 Commandments of 3D Cinematography," blog, 30 Dec. 2007; http://forums.digitalcinemasociety.org/showthread.php?t=44.
5. P. Streather, "Barry, with Massive Respect, a Few Comments!" blog, 30 Dec. 2007; http://forums.digitalcinemasociety.org/showthread.php?t=44.
6. K. Masaoka et al., "Spatial Distortion Prediction System for Stereoscopic Images," *J. Electronic Imaging*, vol. 15, no. 1, 2006, pp. 013002-1–013002-12.
7. V. Grinberg, G. Podnar, and M. Siegel, "Geometry of Binocular Imaging," *Stereoscopic Displays and Virtual Reality Systems*, Proc. SPIE, vol. 2177, 1994, pp. 56–65.
8. A. Woods, T. Docherty, and R. Koch, "Image Distortions in Stereoscopic Video Systems," *Stereoscopic Displays and Applications IV*, Proc. SPIE, vol. 1915, 1993, pp. 36–48.
9. G. Wald, "Eye and Camera," *Scientific American*, Aug. 1950, pp. 32–41.
10. R. Held and M. Banks, "Misperceptions in Stereoscopic Displays: A Vision Science Perspective," *Proc. 5th Symp. Applied Perception in Graphics and Visualization*, ACM Press, 2008, pp. 23–32.
11. C. Zitnick et al., "High-Quality Video View Interpolation Using a Layered Representation," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 600–608.
12. D. Arijon, *Grammar of the Film Language*, Focal Press, 1976.
13. L.B. Stelmach et al., "Human Perception of Mismatched Stereoscopic 3D Inputs," *Proc. 2000 Int'l Conf. Image Processing* (ICIP 00), vol. 1, IEEE Press, 2000, pp. 5–8.

Sanjeev J. Koppal *is a research associate at Harvard University's School of Engineering and Applied Science. His research interests span all combinations of lights, cameras, and computers for application in vision and graphics, including microsensors, novel cameras and illumination, physics-based vision, digital cinematography, 3D cinema, active vision, image-based rendering, appearance modeling, and 3D reconstruction. Koppal has a PhD from Carnegie Mellon University's Robotics Institute. Contact him at sjkoppal@seas.harvard.edu.*

C. *Lawrence Zitnick* *is a researcher at Microsoft Research's Interactive Visual Media group. His latest research includes object recognition, spatial and temporal video interpolation, and computational photography. Zitnick has a PhD in robotics from Carnegie Mellon University. Contact him at larryz@microsoft.com.*

Michael F. Cohen *is a principal researcher at Microsoft Research, where he has worked on image-based rendering, animation, camera control, and artistic nonphotorealistic rendering. He received the 1998 Siggraph Computer Graphics Achievement Award for his contributions to the radiosity method for image synthesis. Cohen has a PhD from the University of Utah. Contact him at michael.cohen@microsoft.com.*

Sing Bing Kang *is a principal researcher at Microsoft, working on image and video enhancement and image-based modeling. Kang has a PhD in robotics from Carnegie Mellon University. He's an associate editor for* IEEE Transactions on Pattern Analysis and Machine Intelligence *and an associate coeditor in chief for the* IPSJ Transactions on Computer Vision and Applications. *Contact him at singbing.kang@microsoft.com.*

Bryan Ressler *is a principal research software development engineer at Microsoft, where he applies innovative user interface ideas to Web browsing and search. Ressler has a bachelor's in information and computer science from the University of California, Irvine. Contact him at bryanr@microsoft.com.*

Alex Colburn *is a graduate student in the University of Washington's Computer Science and Engineering department, where he works on computer graphics and computer vision. Colburn has an MS in computer science from the University of Washington. Contact him at alex@colburn.org.*