Stereoscopic 3D Line Drawing

Yongjin Kim
POSTECH

Yunjin Lee
Ajou University

Henry Kang
University of Missouri at St. Louis

Seungyong Lee
POSTECH

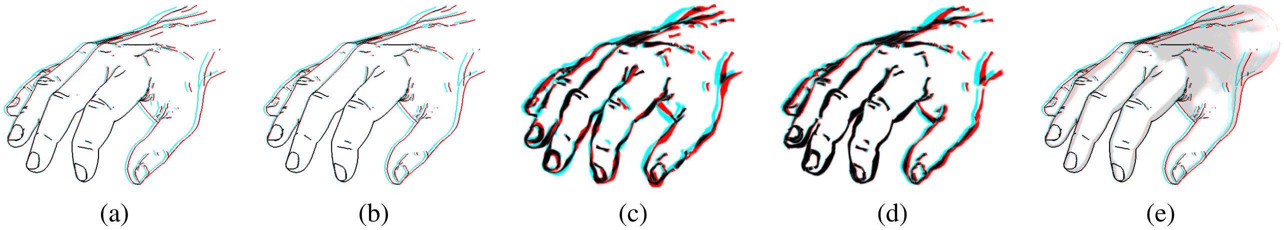


Figure 1: *Stereo line drawing. (a) each-eye-based lines, (b) our lines, (c) each-eye-based stylization, (d) our stylization, (e) our result combined with toon shading. Binocular rivalry occurs in several regions (e.g., wrinkles) in (a) and (c), whereas our method eliminates such artifacts in (b) and (d) by using stereo-coherent lines and stylization. Stereo images in this paper can be viewed using red/cyan 3D glasses.*

Abstract

This paper discusses stereoscopic 3D imaging based on line drawing of 3D shapes. We describe the major issues and challenges in generating stereoscopic 3D effects using lines only, with a couple of relatively simple approaches called each-eye-based and center-eye-based. Each of these methods has its shortcomings, such as binocular rivalry and inaccurate lines. We explain why and how these problems occur, then describe the concept of stereo-coherent lines and an algorithm to extract them from 3D shapes. We also propose a simple method to stylize stereo lines that ensures the stereo coherence of stroke textures across binocular views. The proposed method provides viewers with unique visual experience of watching 2D drawings popping out of the screen like 3D.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

Keywords: non-photorealism, line drawing, line stylization, stereoscopy, binocular rivalry, stereo coherence

Links: DL PDF WEB VIDEO

1 Introduction

Stereoscopic 3D imaging is a way to provide a compelling 3D visual experience with just two images obtained from the respective viewpoints representing our left and right eyes. The stereo anaglyph (fusion) of these two images, when observed through 3D glasses, generates stereoscopic 3D view of the scene as the binocular disparity between the two images activates 3D depth perception. This causes the characters and objects in the scene to seemingly pop out of or sink into the screen, creating a more immersive environment for the audience. With the rapid advances of the stereo capture/display devices, stereo imaging is now ubiquitous in a variety of areas such as movies (e.g., *Avatar*), video games, scientific visualization, medical imaging, and remote operations.

Most of the research on stereo imaging has been limited to generating photorealistic 3D effects. Indeed, a fusion of photorealistic stereo image pair could help make the viewing experience feel even more real. On the other hand, if the source images are non-photorealistic, such as paintings, the stereoscopic fusion should create an illusion of being in a non-realistic, painted world [Northam et al. 2012]. Now, what if the stereo image pair consists solely of *lines*? It is well known that human brain has a powerful mechanism to reconstruct the original 3D shapes from just a small number of lines [Rusinkiewicz et al. 2008], but little is known about how to deal with lines in a stereoscopic setting.

In this paper, we address the problem of *stereo line drawing*, that is, stereoscopic 3D line drawing of objects. We discuss the unique challenges of stereo line drawing, then present tailored solutions to meet them. We also show that the fusion of stereo line drawings provides viewers with surreal visual experience of being in a hand-drawn, stereoscopic 3D world. Stereo line drawing naturally combines the benefits of both line drawing and stereo imaging, i.e., concise depiction of shapes and immersive 3D effects. Stereo line drawing may also help facilitate stereoscopic extension of other non-photorealistic rendering (NPR) styles, such as toon shading [Lake et al. 2000], as the stereoscopic contours can guide 3D perception of object interiors.

Combining the two principles however is not a trivial task due to the largely view-dependent nature of line drawing. Line drawing of smooth objects typically involves view-dependent lines such as contours and suggestive contours [DeCarlo et al. 2003], whose surface points called *line generators* vary across views. This means that stereo line drawing must deal with two different view-dependent line generators for left and right eyes (Fig. 2). However, with lines being the sole source of information, it is essential that these line generators be *stereo-coherent*, meaning each line segment from one view has a fusible counterpart in the other view. Otherwise it may suffer from the stereoscopic artifact known as binocular rivalry [Alais and Blake 2005; Blake and Tong 2008].

We demonstrate this with a couple of straightforward methods for stereo line drawing; *each-eye-based* and *center-eye-based*. Each-eye-based approach separately extracts line generators from each

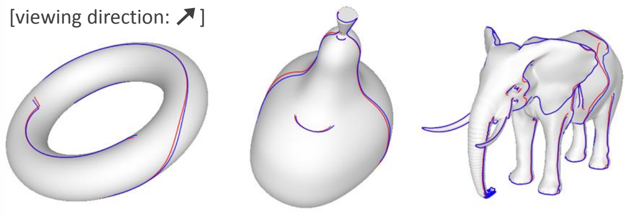


Figure 2: *View-dependent geometrical features (contours + suggestive contours) observed by the left eye (pink) and the right eye (blue) do not coincide with each other. In each figure, the view directions are from lower left to upper right.*

view, possibly resulting in stereo-incoherent lines and thus binocular rivalry. Center-eye-based approach on the other hand extracts a single, shared set of line generators from the center of two eyes to avoid binocular rivalry. However, drifting away from the original eyes inevitably ends up with wrong line generators and in the worse case could lead to distortion of stereo-fused shapes.

To address the aforementioned limitations, we present a novel, rigorous definition of *stereo-coherent lines* that leads to accurate extraction of lines for each view with no binocular rivalry. Inspired by an epipolar curve on a surface with moving eyes [Giblin and Weiss 1995], we examine the continuity of each line generator point on the surface along its epipolar curve (where the epipolar plane intersects the surface). We show that the lines constructed by passing this continuity test are guaranteed to be accurate and also free of binocular rivalry. We also propose an algorithm to extract these stereo-coherent lines while excluding lines that cannot be stereo-fused. Another important issue in stereo line drawing is stylization of lines as it could also lead to binocular rivalry if separately handled in each view. We solve this problem by parameter propagation between views using the full correspondence between stereo-coherent line pairs.

The main contributions of this paper are summarized as follows.

- Introduction of concept, method, and analysis of stereoscopic 3D line drawing
- A theoretical foundation as to what constitutes stereo-coherent lines, which leads to accurate 3D stereo line drawings free of binocular artifact
- An image-space algorithm for computing stereo-coherent lines based on epipolar curves
- A technique to ensure stereo-coherent stylization of lines

2 Related Work

One of the key issues in line drawing is what lines to draw, which may vary depending on the definition of lines (see [Rusinkiewicz et al. 2008] for an extensive survey). Generally speaking, there are two types of lines: *view-independent* and *view-dependent*. View-independent lines (e.g., creases) refer to static features on the object surface defined solely by the local surface geometry. View-dependent lines (e.g., contours) on the other hand dynamically relocate themselves on the surface as the viewpoint changes. The primary focus of this paper is on handling view-dependent lines (contours in particular) in stereoscopic setting, which requires establishing coherence between two different sets of lines that are generated from the binocular viewpoints.

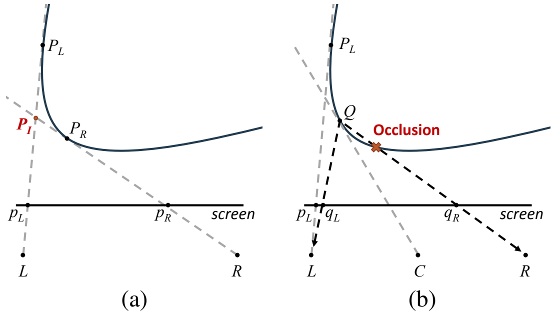


Figure 3: *Contours from different eye positions. (a) off-surface 3D position of a stereo-fused contour. (b) inconsistency between real silhouette and the one observed from the center eye.*

The basic concept of stereo imaging in the context of binocular human vision was introduced by Wheatstone [1838], where he established the relationship between the binocular disparity of two projections and the perceived depth of the original 3D point. *Binocular rivalry* states that when two mutually incoherent images are presented to left and right eyes, respectively, the visual perception randomly alternates between the two images and thus fails to properly reconstruct the original 3D shape [Blake and Tong 2008]. In this case, the two images may differ in various stimuli, such as color, lightness, contour, form, and size. Binocular rivalry has garnered a lot of attention over the years especially in the area of visual perception [Alais and Blake 2005]. In our work, we propose a solution to fix the binocular rivalry that occurs when the two line drawings have mutually incoherent line pairs.

Another related area is non-photorealistic stereo rendering, although, to the best of our knowledge, stereo line drawing from 3D models has not been attempted yet. Stavrakis et al. presented a series of artistic stereo rendering algorithms and frameworks for stereo painting [Stavrakis and Gelautz 2004], stereo edge detection and stylization [Stavrakis and Gelautz 2005], and stereo artwork [Stavrakis et al. 2005]. Stavrakis [2008] also proposed a stereoscopic line drawing method that takes a pair of stereo photographs as input and thus relies on image processing techniques in conjunction with depth-from-stereo. For stereoscopic 3D visualization of anatomic structures, Tokunaga et al. [2010] used simple non-photorealistic shading styles. Richardt et al. [2010] performed a pilot user study on stereo watercolor rendering, and Northam et al. [2012] presented a stereo image stylization and painting method that ensures consistency between the binocular renderings.

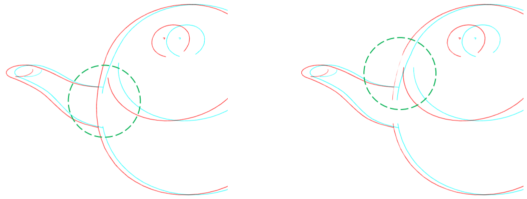
There are a variety of rendering techniques for drawing lines extracted from 3D shapes, focusing on stylization [Northrup and Markosian 2000; Kalnins et al. 2002], visibility test [Cole and Finkelstein 2010], and temporal coherence [Kalnins et al. 2003; Bénard et al. 2010; Bénard et al. 2012]. In this paper, we build on existing stylization methods to provide stereo coherence for stylized lines between two views. We do not however address temporal coherence of stylized stereo lines.

3 Basic Approaches

We first introduce a couple of simple approaches to stereo line drawing and discuss their relative strengths and limitations.

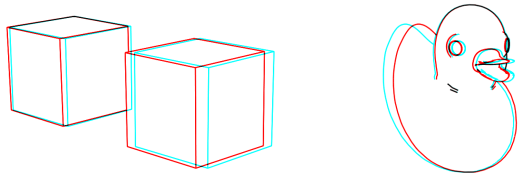
3.1 Each-eye-based stereo line drawing

What makes stereoscopic line drawing unique and different from normal stereoscopy is that there are only lines and no interior con-



(a) each-eye-based (b) center-eye-based

Figure 4: *Limitation of basic approaches. (a) Binocular rivalry occurs between the spout and the body. (b) The silhouette of the left upper part is partially occluded by the body interior.*



(a) (b)

Figure 5: *Missing lines due to visibility changes. (a) A part of the right vertical edge of the back cube is visible only from the left eye (cyan line). (b) The back of the duck is partially occluded by the head, and slightly more visible from the left eye.*

tent. Without interior information, all we can do is to find matching line pairs to establish correspondence between two views. When these lines are view-dependent, the situation can get a bit complicated. Let P_L and P_R be surface points on some view-dependent line generators observed from left and right eyes, respectively, and p_L and p_R be their projections (Fig. 3a). Note that this matching pair of stereo line points actually represent different points on the surface. Therefore, the reconstructed 3D point by stereo fusion is neither P_L nor P_R, but the intersection P_I of the two respective view rays. Fig. 3a shows that the intersection point can be off the surface, which however does not pose a problem for stereopsis, since the projection of this point onto each view coincides with that of the silhouette (boundary of the shaded object). Once the collection of such intersection points form a stereoscopic 3D outline, the interior could be automatically filled as it would for normal line drawing [Cole et al. 2009], but in this case, interpolating the depth of its outline. This is why something as simple as line drawing is still good enough to produce a stereoscopic 3D effect, and a unique one at that, because of its concise, hand-drawn look.

However, the view-dependent nature of some lines, combined with the binocular disparity, could prevent them from having matching counterparts in the other view. Since we have no information other than lines, this could present a serious threat to successful stereo fusion and depth perception. Fig. 4a shows an example, where some portion of a line is detected only from the right eye (red) but not from the left eye (cyan). When the stereo counterpart of a line is missing like this, it not only fails to recover its depth cue, but also may induce a visual artifact of binocular rivalry, thus disturbing the process of stereo fusion.

It should be pointed out that the visibility may also cause some lines to be missing in the other view (Fig. 5). However, the visibility issue arises in any normal stereoscopic imaging, which in general does not cause noticeable visual artifact. In fact, occlusion itself is known to help the human visual system in enhancing the sense of depth [Nakayama and Shimojo 1990], and depth discontinuity cues provided by occlusion have been used for stereo matching computation [Geiger et al. 1995].

3.2 Center-eye-based stereo line drawing

A simple way to resolve binocular rivalry would be to use a single, shared eye for extracting lines and then use them for both views. In our experiment, we use the center eye (center of the two eyes). Once we have extracted lines from the center eye, we project them onto each view to generate a stereo line drawing. Therefore, all the lines in one view should have matching counterparts in the other view, eliminating the possibility of binocular rivalry.

However, lines obtained in this way are not quite accurate in that their line generators are off from where they are supposed to be (Fig. 3b). This error could lead to distortion (such as shape change or shrinkage) of a stereoscopically perceived shape as the projections of line points from the center eye do not coincides with the object silhouettes in left and right views. In addition, when the line points are projected to left and right views, visibility test can remove them because they are hidden by object interior (e.g., the contour point Q in Fig. 3b), as illustrated in Fig. 4b. This visibility issue could be resolved by depth offsetting, but determining the required offset in a robust way is non-trivial.

4 Stereo-Coherent Lines

In order to find lines that are both accurate (originated from accurate line generators) and stereo-coherent (free of binocular rivalry), we build on the each-eye-based approach since the only accurate lines are ones obtained from the original eye locations. We first determine line generators separately from each eye location, then project them back onto each view. As shown earlier, some segments of these projected lines are fusible while some are not. Our main task is therefore to identify and eliminate line segments that fall in the latter group.

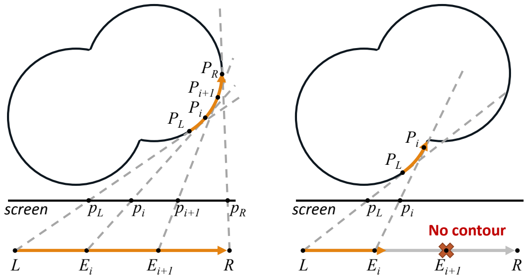
4.1 Definition

The stereo fusion of left and right projections occurs amongst points in the same scanline on the screen. Each scanline is contained in an associated epipolar plane in the stereoscopic setting, where a single big screen is used for both projections (Fig. 6). Let L and R denote left and right eyes, respectively, whereas P_L and P_R their corresponding line points. Suppose we have an imaginary viewpoint E that moves along the baseline between two eyes (Fig. 6a). As E moves from L to R, its line point P_E should also move from P_L to P_R along a curve on the intersection between the surface and the epipolar plane. This curve is called an *epipolar curve* [Giblin and Weiss 1995] and has been used to define epipolar parameterization [Cipolla and Blake 1992] for reconstructing a 3D surface from multiple occluding contours with moving viewpoints. Now if the epipolar curve is continuous from P_L to P_R, we call P_L *epipolar-slidable*. If on the other hand the epipolar curve gets cut short before it reaches R, P_L is not epipolar-slidable.

We may parameterize the moving viewpoint E on the baseline as

$$E(t) = (1 - t)L + tR.$$

Formally, a line point P_L observed from L is epipolar-slidable if there exists a stereo counterpart P_R such that the epipolar curve between P_L and P_R is continuous throughout $t \in [0, 1]$. Note that if P_L is epipolar-slidable, so is its counterpart P_R. A non-epipolar-slidable line point (either P_L or P_R) means that there is no matching counterpart in the other view, therefore no stereo fusion is possible and, worse still, it may induce binocular rivalry. This observation



(a) Path exists (b) Path does not exist

Figure 6: *Epipolar-slidability. (a) (P_L, P_R) becomes an epipolar-slidable pair. (b) No epipolar-slidable pair exists for P_L.*

leads us to define a *stereo-coherent line* as a pair of left and right lines whose line points are epipolar-slidable.

4.2 Computation

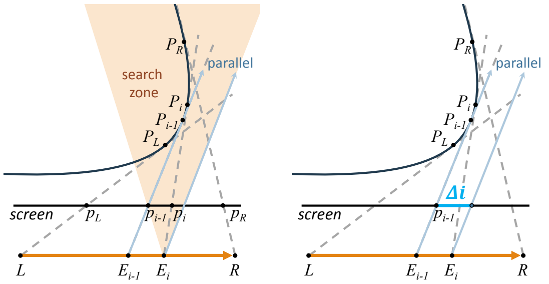
Our overall process for computing stereo-coherent lines is as follows. (1) Render each-eye-based lines on each view. (2) For each point on a line, examine its epipolar curve (from L to R) to check epipolar-slidability. (3) Erase the point if not slidable. (4) Repeat (2)-(3) from R to L. We first explain the process conceptually in the object space, then detail how to implement it in image space.

In the object space, along the baseline between the two eyes, we can first regularly sample viewpoints, denoted E_i. We then follow the epipolar curve starting from P_L. Given the current line point P_{i-1}, we check if there exists, on the epipolar curve, a line point P_i observed from the next sample viewpoint E_i. If one exists, we proceed to the next sample viewpoint and repeat the process until we reach P_R. If such a line point does not exist, however, it means that the epipolar curve is cut short and so P_L is not epipolar-slidable.

We find that P_i must be searched within the vicinity of P_{i-1}, bounded by the two view rays emanating from E_i (see Fig. 7a). This stems from the observation that all the view rays between L and R cross one another (so the stereo fusion is possible), and that the projections move rightward as we follow the epipolar curve from left to right (unless there is an inflection point, which however violates epipolar-slidability anyway). Depending on the object shape, on rare occasions more than one line point may be found within this triangular zone, which however can be resolved by increasing the sampling rate of viewpoints to reduce the search zone.

The object-space approach may require complicated processing and heavy computation (e.g., for finding intersections) in 3D. We thus develop a simpler and more efficient solution where we examine the epipolar-slidability on the image plane. Given the equation $E = (1-t)L + tR$, let p_L and p_R denote the *line pixels* (projections of line points) for L and R, and p_i be the line pixels for E_i. Since an epipolar curve is projected onto a horizontal scanline, the idea is to check for p_L, whether every p_i is found within the search interval Δ_i until we reach p_R as we move from left to right on the scanline. Here, Δ_i is an interval on the scanline, which is the projection of the search zone in the object space, as shown in Fig. 7b. Algorithm 1 summarizes our image-space approach, where in each iteration we use the lines on the image viewed from E_i to find p_i.

We use Algorithm 1 to produce all of our experimental results. Although our analysis is concentrated on contours, our concept of stereo-coherent lines based on epipolar-slidability can be applied to arbitrary definitions of view-dependent lines. In our experiments, we extract contours and suggestive contours together, and a contour can be matched to a suggested contour, and vice versa, when we check the epipolar-slidability. Note however that view-independent lines are inherently stereo-coherent and should be handled separately from view-independent lines in stereo line drawing.



(a) object-space approach (b) image-space approach

Figure 7: *Search area for finding the next line point from E_i. (a) search zone set by $\overline{E_i p_{i-1}}$ and a line emanating from E_i and parallel to $\overline{E_{i-1} p_{i-1}}$. (b) search interval Δ_i on the scanline bounded by p_{i-1} and the intersection of the scanline with a line emanating from E_i and parallel to $\overline{E_{i-1} p_{i-1}}$.*

Algorithm 1 Image space algorithm (L to R)

```
Render lines seen from $L$ and store them as texture $T_0$
Mark each line pixel $p_L$ slidable, and record its pixel location
for each $E_i$, $i = 1, \ldots, N$ do
    Render lines from $E_i$ onto a texture $T_1$
    for each line pixel $p_L$ marked as slidable do
        Read the position of $p_{i-1}$ from $T_0$
        Set the interval $\Delta_i$ using $E_i$, $E_{i-1}$, and $p_{i-1}$
        On $T_1$, find the next line pixel $p_i$ in $\Delta_i$
        If $p_i$ exists, record its position at $p_L$ on $T_0$
        Otherwise mark $p_L$ on $T_0$ as non-slidable
    end for
end for
```

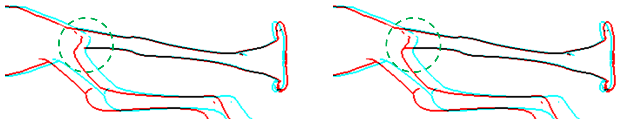
4.3 Degenerate cases

We note that there are a few degenerate cases [Cipolla 1998] which we can handle as follows. When the epipolar plane is the tangent plane of the surface, there should be a single line point found on one view and the matching one on the other. If a line is formed on an epipolar plane (e.g., by touching the side of a cylinder), it is still not a problem for our algorithm since the entire line will map on a scanline. Even when a line is formed along a view ray so that it appears as a point in that view but as a line in the other view, our Algorithm 1 will find the correspondence. When the view ray is tangent to a contour generator (e.g., at the crossing of a contour and a suggestive contour), its line point is still drawn if a matching line point can be found.

When two or more unrelated lines are overlapped in image space, we give priority to the closer one via hidden line removal. However this approach may lead to loss of visible line segments that should have been matched with hidden lines in the other view (Fig. 8a). One way to resolve this issue is to incorporate hidden lines when checking the stereo coherency, then remove them later (Fig. 8b).

5 Stereo-coherent Line Stylization

In stereo line drawing, as in normal line drawing, line stylization would be useful for controlling the look, feel, and mood of the



(a) with hidden line removal (b) without hidden line removal

Figure 8: *Problem that may be caused by hidden line removal when determining stereo-coherent lines. (a) Since a part of the right leg near the body is hidden in the left view (cyan), the corresponding part in the right view (red) cannot be matched and is erased. (b) If we include hidden lines in checking the stereo coherency, the unmatched part in (a) can survive.*

rendering. Stylized lines could also make the subject more recognizable [Goodwin et al. 2007]. One obvious challenge in stylizing stereo lines is that we need to make sure the two lines in each stereo pair have the matching look in every detail, otherwise it could disrupt their stereo fusion. Even if we use stereo-coherent lines only, this issue remains. Typically, stylization of lines is conducted in image space by linking, parameterizing, and texture-mapping them [Northrup and Markosian 2000; Kalnins et al. 2002]. If we go through this process separately on each view, the resulting stereo images would appear to have mutually incoherent stroke textures since the image-space linking and parameterization is inherently view-dependent.

Therefore, we perform linking and parameterization in a single (left) view then propagate the information to the other. Given the set of stereo-coherent lines obtained using Algorithm 1, we perform stroke linking based on their proximity in position and orientation in the left view. We also consider their positional proximity in 3D to make sure distant line segments remain separated, even when they are projected nearby. After linking, we parameterize each linked path to set up for stroke texture mapping. This linking/parameterization information on the left view is then mapped to the lines on the right view by referencing the stereo-correspondence buffer (generated by Algorithm 1) that records the correspondence of line pixels across binocular views. Finally, we map stroke textures onto the parameterized stroke paths in each view.

6 Experimental Results

Our stereo line drawing system was implemented using OpenGL 3.2 and GLSL 1.5. We used a pixel shader to accelerate the inner loop of Algorithm 1 using GPU. All experiments were performed on a PC with a 2.66 Ghz Intel Core i7 CPU 920, 6 GB memory, and an NVidia Quadro FX3800. All images shown in the paper and the supplementary material were rendered with image resolution of 960×480. The default number of sample viewpoints is 6. Without stylization, our unoptimized system renders stereo line drawings at about 3 fps with meshes of up to 30,000 vertices, where most computation is consumed by multiple line renderings for sample viewpoints. When stylization is on, our system runs at about 1.5 fps with similar meshes, where the computation time is proportional to the number of vertices because of the on-the-fly extraction and linking of visible line segments [Northrup and Markosian 2000].

Fig. 9 compares each-eye-based (top) and our stereo-coherent (bottom) stereo line drawings. Note that the use of stereo-coherent lines effectively eliminates binocular rivalry observed in each-eye results, and therefore provides more comfortable (less distracting) stereoscopic viewing experience.

Fig. 10 shows stylized stereo line drawings generated by our system. The left image in each pair illustrates what happens when stylization is done naively on each view. As can be observed through 3D glasses, stroke textures in this case are highly incoherent between left (cyan) and right (red) renderings and thus cause severe binocular rivalry. In contrast, our stereo-coherent stylization method using parameter propagation eliminates the possibility of such artifacts and works robustly with a variety of stroke textures.

Fig. 11 shows more examples of stereo line drawings and stylization results generated by our stereo-coherent approach. Figs. 11d, 11e, 11h, and 11i demonstrate that our method can be combined with stereo shading, providing additional shape and depth cues as well as a different look and feel during stereoscopic 3D (S3D) viewing. All rendering results in this paper contain only contours and suggestive contours as view-dependent lines, except that Figs. 11c, 11d, 11h, and 11i include view-independent creases as well.

7 Discussion and Future Work

In our analysis of stereo-coherent lines, we used contours as the representative of view-dependent lines. While we believe the underlying principle of our algorithm is general enough to cover other types of view-dependent lines (e.g., suggestive contours [DeCarlo et al. 2003] and apparent ridges [Judd et al. 2007]), such extension would be benefited by theoretical analysis on the search areas for finding matching pairs of general view-dependent lines as well as on their degenerate cases. In our experiments, we used combination of contours and suggestive contours as our view-dependent line set. As shown in the rendering results, this combination works quite well since suggestive contours are contours at nearby viewpoints. More experiments are needed to confirm that the mixture of other view-dependent lines could generate nice results in a similar way.

We can consider a few improvements of our image-space algorithm for obtaining stereo-coherent lines. The epipolar-slidability that is confirmed through multiple sample viewpoints generally provides accurate matching for line pixels in left and right views. If needed, we can increase the accuracy of the matching by comparing the disparity of a matched pair of line pixels with their corresponding 3D depths. In our current implementation, we use a fixed number of sample viewpoints for testing epipolar-slidability. Our algorithm can be made more efficient if we could determine the minimum number of sample viewpoints needed for specific input objects and scenes. Another idea to improve the performance is to detect the parts of the input model which have high probability of inducing binocular rivalry in a preprocessing step, then perform the epipolar-slidability test only on those detected regions.

Another logical next step would be to address temporal coherence of stylized stereo 3D line drawing under arbitrary camera motion. Existing methods for temporally-coherent line drawing [Kalnins et al. 2003; Bénard et al. 2010; Bénard et al. 2012] are not directly applicable as they do not ensure the stereo coherence between binocular views when moving from one frame to the next. Considering that our epipolar-slidability handles coherence of view-dependent lines across binocular views, one might be able to extend this concept to time domain across arbitrary view changes. However, the extension would not be straightforward because we have to handle animated objects together with view changes.

We conducted a preliminary perceptual study on comparing each-eye-based and center-eye-based approaches, and the results were as expected: The majority of the subjects were able to experience convincing 3D effects from lines only. Also, many of them reported some form of discomfort, particularly with each-eye-based stylization examples, such as weird flickering, blurriness, and distraction. Although we are yet to conduct a comprehensive user study including our stereo-coherent approach, we have witnessed no such artifacts from the same set of examples when processed via Algorithm

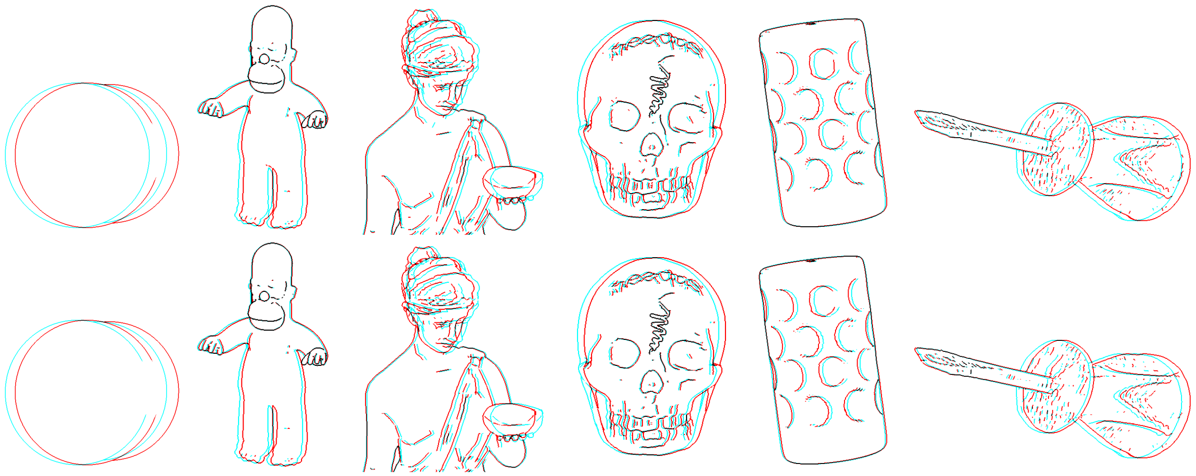


Figure 9: *Stereo line drawing (must be viewed through red/cyan 3D glasses otherwise may not be clear) by the each-eye-based method (above), and our stereo-coherent line method (below). The each-eye-based method may have binocular rivalry artifact at some regions where stereo pairs cannot be formed. Our stereo-coherent line method does not induce binocular rivalry and provides comfortable stereo viewing.*

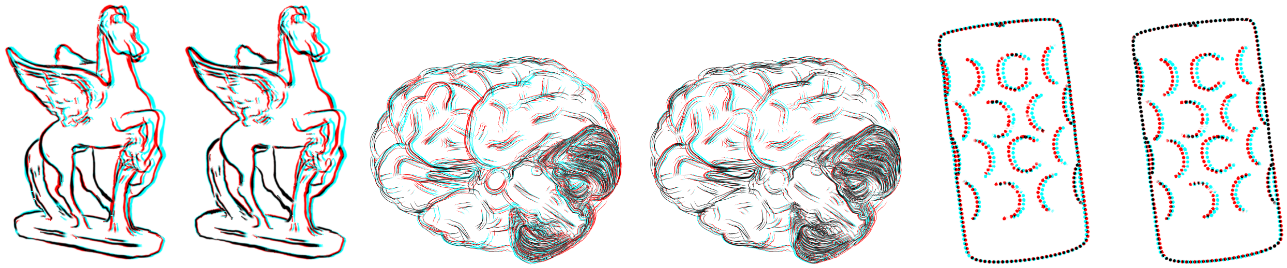
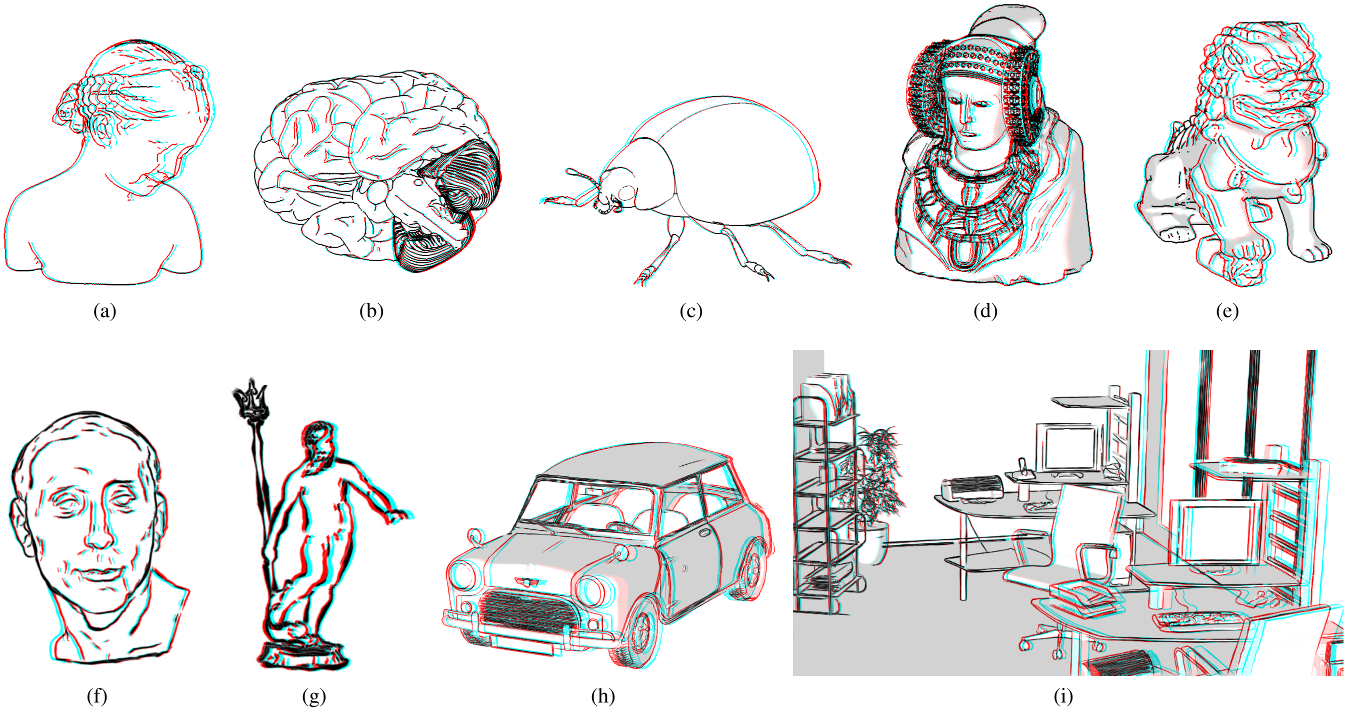


Figure 10: *Stereo coherence of stylized lines. (left) stylization at each-view. (right) our stereo-coherent stylization.*



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Figure 11: *Stereo line drawings (above) and stylization results (below)*

1 and stereo-coherent stylization in Sec. 5. The potential items in the future user study may include: (1) monoscopic vs. stereoscopic line drawing; (2) comparisons of each-eye-based, center-eye-based, and our stereo line drawing methods; (3) stylized vs. non-stylized stereo lines; (4) stereo line drawing vs. stereo shading; (5) the effect of stereo line drawing + stereo shading, etc.

The underlying principle of our technique, i.e., the epipolar-slidability, may be applicable to solving other stereo rendering problems as well, such as one involving sharp shading highlights. Another interesting (but potentially more challenging) future research direction would be to resolve the situation of lines missing stereo counterparts by 'reconstructing' them in the other view while ensuring proper disparity and stereo coherence.

Acknowledgements

We would like to thank all the reviewers for their constructive comments. The models used for line drawing examples are courtesy of the AIM@SHAPE Shape Repository, Amazing3D, Princeton Graphics Group, Oyonale, the Utah 3D Animation Repository, and Headus. This work was supported by Basic Science Research Program of NRF (2012-0008835, NRF-2012R1A1A2042837), IT/SW Creative Research Program of NIPA (NIPA-2012-H0503-12-1008), and Cooperative R&D Program of SMBA (C0004960).

References

Alais, D., and Blake, R., Eds. 2005. *Binocular Rivalry*. MIT Press.

Bénard, P., Cole, F., Golovinskiy, A., and Finkelstein, A. 2010. Self-similar texture for coherent line stylization. In *Proc. Non-photorealistic Animation and Rendering*, 91–97.

Bénard, P., Lu, J., Cole, F., Finkelstein, A., and Thollot, J. 2012. Active strokes: coherent line stylization for animated 3d models. In *Proc. Non-photorealistic Animation and Rendering*, 37–46.

Blake, R., and Tong, F. 2008. Binocular rivalry. *Scholarpedia 3*, 12, 1578.

Cipolla, R., and Blake, A. 1992. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision 9*, 2, 83–112.

Cipolla, R. 1998. The Visual Motion of Curves and Surfaces. *Philosophical Trans. Royal Society London A 356*, 1740, 1103–1121.

Cole, F., and Finkelstein, A. 2010. Two fast methods for high-quality line visibility. *IEEE Trans. Visualization and Computer Graphics 16*, 707–717.

Cole, F., Sanik, K., DeCarlo, D., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S., and Singh, M. 2009. How well do line drawings depict shape? *ACM Trans. Graphics 28*, 3, 28:1–28:9.

DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., and Santella, A. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph. 22*, 3, 848–855.

Geiger, D., Ladendorf, B., and Yuille, A. 1995. Occlusions and Binocular Stereo. *International Journal of Computer Vision 14*, 211–226.

Giblin, P. J., and Weiss, R. S. 1995. Epipolar curves on surfaces. *Image and Vision Computing 13*, 33–44.

Goodwin, T., Vollick, I., and Hertzmann, A. 2007. Isophote distance: a shading approach to artistic stroke thickness. In *Proc. Non-photorealistic Animation and Rendering*, 53–62.

Judd, T., Durand, F., and Adelson, E. H. 2007. Apparent ridges for line drawing. *ACM Trans. Graphics 26*, 3, 19:1–19:8.

Kalnins, R. D., Markosian, L., Meier, B. J., Kowalski, M. A., Lee, J. C., Davidson, P. L., Webb, M., Hughes, J. F., and Finkelstein, A. 2002. WYSIWYG NPR: Drawing strokes directly on 3D models. *ACM Trans. Graphics 21*, 3, 755–762.

Kalnins, R. D., Davidson, P. L., Markosian, L., and Finkelstein, A. 2003. Coherent stylized silhouettes. *ACM Trans. Graphics 22*, 3, 856–861.

Lake, A., Marshall, C., Harris, M., and Blackstein, M. 2000. Stylized rendering techniques for scalable real-time 3D animation. In *Proc. Non-photorealistic Animation and Rendering*, 13–20.

Nakayama, K., and Shimojo, S. 1990. Da vinci stereopsis: depth and subjective occluding contours from unpaired image points. *Vision Research 30*, 1811–1825.

Northam, L., Asente, P., and Kaplan, C. S. 2012. Consistent stylization and painterly rendering of stereoscopic 3D images. In *Proc. Non-photorealistic Animation and Rendering*, 47–56.

Northrup, J. D., and Markosian, L. 2000. Artistic silhouettes: a hybrid approach. In *Proc. Non-photorealistic Animation and Rendering*, 31–37.

Richardt, C., Kyprianidis, J. E., and Dodgson, N. A., 2010. Stereo coherence in watercolour rendering. Poster presentation at *Non-photorealistic Animation and Rendering 2010*.

Rusinkiewicz, S., Cole, F., DeCarlo, D., and Finkelstein, A., 2008. Line drawings from 3D models. *ACM SIGGRAPH 2008 Course Notes*.

Stavrakis, E., and Gelautz, M. 2004. Image-based stereoscopic painterly rendering. In *Proc. 15th Eurographics Workshop on Rendering Techniques*, 53–60.

Stavrakis, E., and Gelautz, M. 2005. Computer generated stereoscopic artwork. In *Proc. Eurographics Workshop on Computational Aesthetics in Graphics Visualization and Imaging 2005 (CAe 2005)*, 143–149.

Stavrakis, E., Bleyer, M., Markovic, D., and Gelautz, M. 2005. Image-based stereoscopic stylization. In *Proc. International Conference on Image Processing (ICIP 2005)*, 5–8.

Stavrakis, E. 2008. *Stereoscopic Non-Photorealis Rendering*. PhD thesis, Vienna University of Technology, Vienna, Austria.

Tokunaga, D. M., Corrêa, C. G., Nakamura, R., Nunes, F. L. S., and Tori, R. 2010. Non-photorealistic rendering in stereoscopic 3d visualization. In *Proc. ACM SIGGRAPH 2010 Posters*, 131:1–131:1.

Wheatstone, S. C. 1838. Contributions to the physiology of vision - part the first. on some remarkable, and hitherto unobserved phenomena of binocular vision. *Philosophical Transactions 128*, 371–394.