Self-Animating Images: Illusory Motion Using Repeated Asymmetric Patterns

Ming-Te Chi Tong-Yee Lee
National Cheng-Kung University*, Taiwan

Yingge Qu Tien-Tsin Wong
The Chinese University of Hong Kong

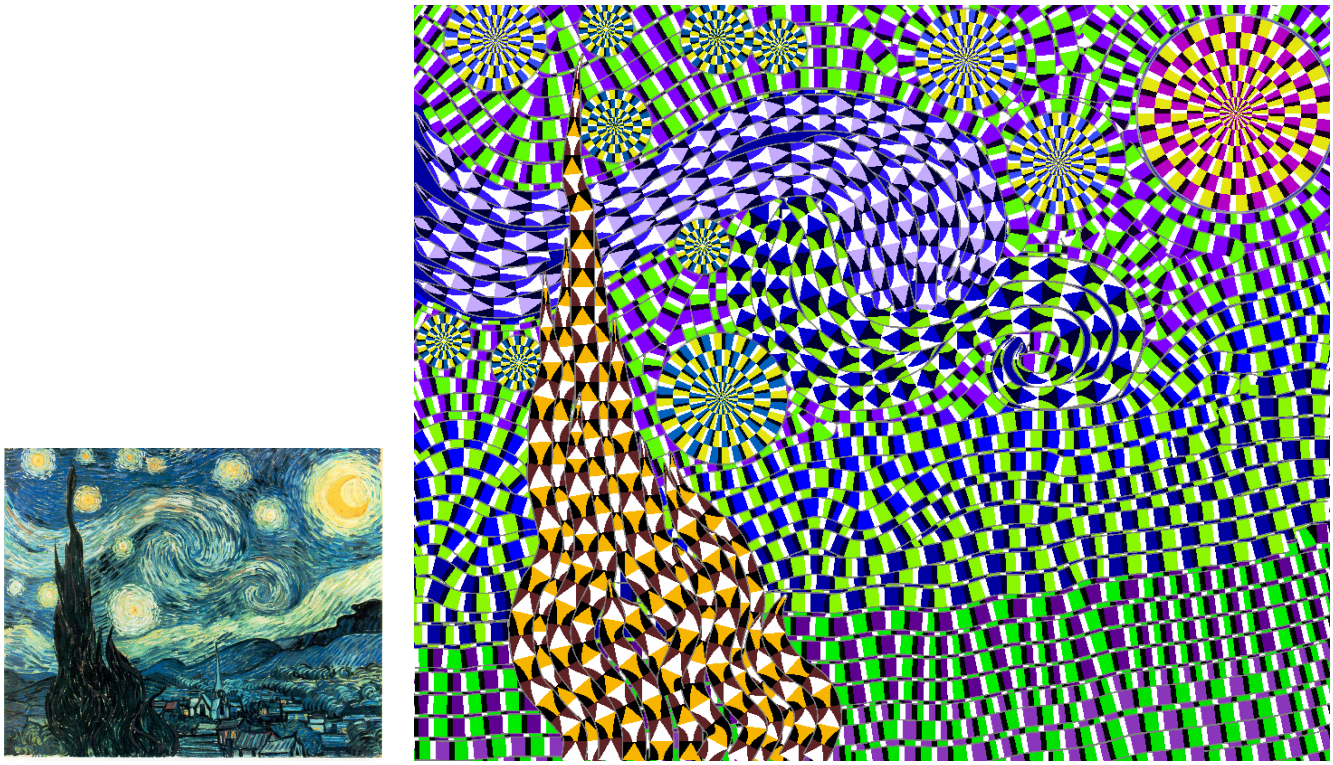


Figure 1: *A self-animating image inspired by Van Gogh's "Starry Nights."*

Abstract

Illusory motion in a still image is a fascinating research topic in the study of human motion perception. Physiologists and psychologists have attempted to understand this phenomenon by constructing simple, color repeated asymmetric patterns (RAP) and have found several useful rules to enhance the strength of illusory motion. Based on their knowledge, we propose a computational method to generate self-animating images. First, we present an optimized RAP placement on streamlines to generate illusory motion for a given static vector field. Next, a general coloring scheme for RAP is proposed to render streamlines. Furthermore, to enhance the strength of illusion and respect the shape of the region, a smooth vector field with opposite directional flow is automatically generated given an input image. Examples generated by our method are shown as evidence of the illusory effect and the potential applications for entertainment and design purposes.

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

Keywords: Illusory motion, Repeated Asymmetric Pattern (RAP)

*Project Web Page: http://graphics.csie.ncku.edu.tw/SAI/

ACM Reference Format
Chi, M., Lee, T., Qu, Y., Wong, T. 2008. Self–Animating Images: Illusory Motion Using Repeated Asymmetric Patterns. *ACM Trans. Graph. 27*, 3, Article 62 (August 2008), 8 pages. DOI = 10.1145/1360612.1360661 http://doi.acm.org/10.1145/1360612.1360661.

Copyright Notice
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, fax +1 (212) 869-0481, or permissions@acm.org.
© 2008 ACM 0730-0301/2008/03-ART62 $5.00 DOI 10.1145/1360612.1360661
http://doi.acm.org/10.1145/1360612.1360661

1 Introduction

Illusory motion is a fascinating physiological and psychological phenomenon. It refers to the phenomenon that *still images* composed of certain colors and patterns lead to the human perception of *motion*. The information gathered by the eye is processed by the brain to give a perception that does not tally with a physical measurement of the stimulus source. For example, "Rotating Snake" is a remarkable motion illusion painting created by Kitaoka [2003]. The circular snakes appear to rotate "spontaneously," although the image is static. His study [Kitaoka 2006b] also indicates that the two most illusive color combinations are blue-yellow and red-green. As reported in [Kitaoka 2005], 5% of people cannot perceive this kind of illusory motion. Scientists [Conway et al. 2005; Backus and Oruc 2005; Murakami et al. 2006] have attempted to explain this phenomenon and have found useful rules. The design of artful and illusory patterns is their main concern. Illusion demonstration is indeed fun to view and is potentially useful for entertainment and advertising purposes.

In spite of these scientific studies, illusion art is still confined by a limited choice of color pairs and simple geometric shapes. Most existing work is done manually. There is no existing work automatically converting a given image to one that possesses illusory motion. In this paper, we propose a computational method to achieve such a goal. Based on existing psychological knowledge on illusory motion, our method automatically optimizes for the effect of illusory motion. Our major contributions can be summarized as:

- We propose a *streamline-based RAP placement* technique (Section 4) to generate illusory motion for a given static vector field. This streamline based illusion can help faithfully

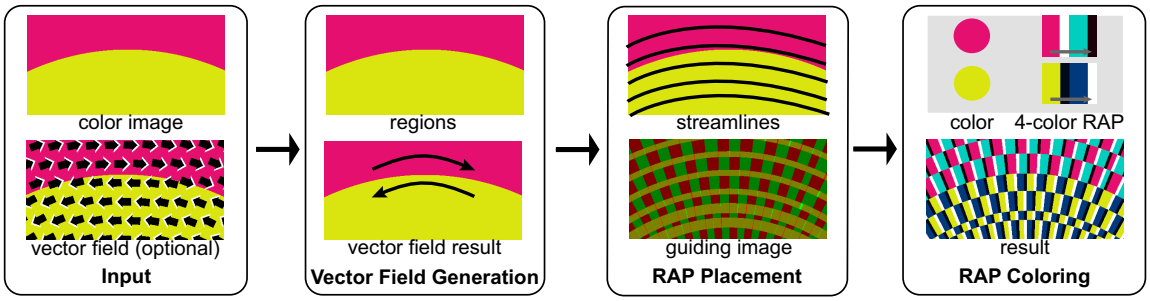


Figure 2: *An overview of generating a self-animating image.*

convey the raw region shapes of the target image.

- We also propose a novel heuristic to determine the *four-color combination* (Section 5) for the RAP, which is used to render the streamlines that strengthen the illusion. Therefore, the RAP is no longer confined to the limited color combinations.
- Finally, we propose *illusion-strengthening vector-field generation* methods for images without a vector field (Section 6).

Figure 1 demonstrates our illusion result converted from Van Gogh's "Starry Night," in which the illusive flow conveys the natural region shapes of the original painting. Our coloring scheme automatically selects a rich set of colors in contrast to tedious manual selection.

An overview of our method is summarized in Figure 2. The user starts by employing a color image as input. The vector field generation step generates opposite directional flow to express the region shapes and to guarantee the illusion effect. The step is optional if a vector field is provided by the user. Following the vector field, streamlines are rendered along the vector field, and an optimization is performed to position the primitive illusion pattern on the streamline. Finally, we determine the most illusive color combination according to the color of input image, and colorize the RAP to generate the self-animating image.

2 Previous Works

Illusory Motion Our research has been inspired by many previous approaches in the study of illusory motion. These works proposed different illusory patterns to enhance the perceived motion. However, many findings were obtained based on observations or rules-of-thumb that are still not fully verified in theory. In 1979, Fraser and Wilcox [1979] first reported "anomalous motion illusion" characterized by apparent motion in a still image. Faubert and Herbert [1999] pointed out that illusory motion in peripheral vision can be generated using "sawtooth" luminance patterns from dark to light. Later, Kitaoka and Ashida [2003] suggested that the combinations of black, dark-gray, white and light-gray in "sawtooth" luminance patterns can produce a very strong peripheral drift illusion. Kitaoka published a very strong illusory design called "Rotating Snakes" on his web page [Kitaoka 2003]. Recently, Kitaoka [2006a] roughly classified even further the optimized Fraser and Wilcox illusion into four categories. Later, by modifying the colors, he presented a color version of the original "Rotating Snakes" that produces a stronger illusion than the original gray version. He also pointed out that color can, at times, enhance illusion. His experiments indicate that certain color combinations, say blue-yellow or red-green, are more effective in enhancing illusion [Kitaoka 2006b].

Self-Animating Images The research in producing motion using a single image has motivated only a few works in computer graphics. In 1979, Shoup [1979] proposed a simple color table animation technique to animate the colors of objects and areas within an image. Freeman et al. [1991] used a quadrature pair of oriented filters to vary the local phase, thereby making patterns in a still image appear to move without really animating them. Masuch [1999] used speedlines, fading contour repetitions, and motion arrows to create the sensation of motion in motionless pictures. Later, by taking advantage of several existing illusory motion phenomena, Gossett and Chen [2004] attempted to create illusory motion by self-animating line textures. Wei [2006] proposed a fully automatic method to visualize vector fields using the tile-based RAP. Unlike these simple applications of illusory motion, we propose a computational approach to maximize the strength of illusory motion.

3 Optimized Fraser-Wilcox Illusion

Our work is based on the classification of the optimized Fraser-Wilcox illusion [Kitaoka 2006a] as shown in Figure 3. For completeness, we briefly describe each category below. A detailed explanation can be found in [Kitaoka 2006a].

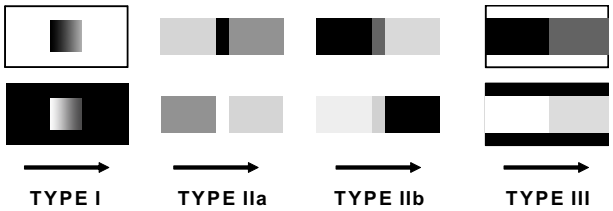


Figure 3: *Types of optimized Fraser-Wilcox illusions excerpted from [Kitaoka 2006a]. Arrows indicate the direction of the perceived motion.*

- TYPE I: The direction of the perceived motion depends on whether a luminance gradient is surrounded by a lighter or darker field.
- TYPE IIa and IIb: Both types are characterized by a line separating two flanks of different luminance or dark-and-white fields. The direction of perceived motion depends on the color combination of the separating line and the two flanks.
- TYPE III: This is characterized by two flanks of similarly shaded fields that are enclosed by either a lighter or darker shaded field. The direction of the perceived motion depends on the color contrast between the flanks and the enclosing field.

In the study of perception, a stimulus can be made from any of the above four types. From our experience, a single stimulus does not produce strong motion. Sometimes it is difficult to perceive the motion direction as well. However, when many stimuli are placed together, we can easily perceive strong motion. In this paper, we focus on TYPE IIa which can generate the strongest illusion among the four types of illusory patterns. The four-intensity combination of this type is in the order of B-DG-W-LG (black, dark-gray, white and light-gray). Each four-intensity pattern is also called a repeated asymmetric pattern (RAP) [Backus and Oruc 2005]. In the following sections, we first study how the basic four-color (Black-Blue-White-Yellow) stimulus is used to generate interesting illusory motion for a given static vector field. Then, the four-gray (B-DG-W-LG) pattern is extended to other four-gray patterns (Section 5.1) and then finally to four-color patterns (Section 5.2).

4 Streamline-Based Illusion

We propose a novel computational approach for generating Fraser-Wilcox illusion using the RAP placement. Given a vector field, we position the RAPs along the vector flow to generate streamlines having a motion that is consistent with the vector field. Optimization of RAP placement is also introduced for strengthening the illusion.

4.1 Streamline Placement

Given a vector field, a naïve approach for RAP placement is to randomly generate seed points and then place the RAP on the integrated streamline path along the vector field. The problem with this method is that short and irregular streamlines may result. Note that it is found that long streamlines lead to stronger illusion than the short ones because they carry more RAPs. The overlap among short streamlines generated by the above naïve method may significantly reduce the strength of illusion.

Instead, we want to identify major long streamlines to express the vector field. Long streamlines guarantee continuity and ensure undisturbed RAP flow, resulting in better illusory motion. We adopt Mebarki's method [2005] to generate long and evenly-spaced streamlines. We parameterize each streamline along the path in order to texture it with RAPs. We divide the streamline into several segments with equal length, controlled by *SegLen*. The length of each color in the RAP segment conforms to the ratio 1:2:1:2 in order to satisfy the requirement of the TYPE II optimized Fraser-Wilcox illusion. An array $RAPColor[6] = \{black, blue, blue, white, yellow, yellow\}$ is first prepared. Next, we can color the streamline as:

$$Color = RAPColor[\ \lfloor (i \bmod SegLen) \times (6/SegLen) \rfloor\ ] \quad (1)$$

where i is the distance from the starting point of the streamline.

4.2 Fragment Placement Optimization

The pattern surrounding the RAP segment also affects the strength of the illusion. Kitaoka and Ashida [2003] pointed out that the peripheral drift illusion can be further enhanced by fragments of stimuli. Figure 4 demonstrates the importance of fragment placement. In Figure 4(a), arbitrary placement of RAP segments in adjacent streamlines may cause the overall layout to look like a single patch, and does not produce much motion illusion. In contrast, the RAP segments of adjacent streamlines in Figure 4(d) are properly placed to strengthen the illusion effect.

To improve the RAP placement on the streamline, the following optimization problem is formulated. According to the four-intensity pattern in TYPE II, a RAP segment can be segmented into lighter and darker fragments using the middle-gray color. Let X be the fragment set of all RAP stimuli, $N(x)$ is the set of neighborhood fragments of x in X. $L(z)$ is the intensity of fragment z.

$$E_{\text{fragment}} = \sum_{x \in X} \sum_{y \in N(x)} (L(x) - L(y))^2 \quad (2)$$

The goal of optimization is to maximize E_{fragment}, i.e., the difference in intensity between neighboring RAP fragments from neighboring streamlines. We propose an image-based method to measure the placement difference between two neighboring streamlines in order to maximize Equation (2).

A guiding image is first rendered. Figure 4 shows how this guiding image is derived from the initial stimulus (top row). For clarity, we illustrate our idea using two streamlines in (b) with the guiding image in (c). First, all segments in streamlines start with equal lengths. Then, the lighter part of a RAP segment is set as middle-red and the darker part as middle-green. Hence, each RAP segment contains a pair of middle-red and middle-green fragments. The width of the streamline can be determined to ensure that RAPs on neighboring streamlines touch or even overlap each other. The red-green streamlines are then drawn one-by-one with intersected regions being blended together. They are blended in a way that same-color regions overlap to give higher intensity. Hence, light red or green areas indicate poor placement, since intensities on adjacent streamlines are similar. On the other hand, yellow areas signify large differences in intensity and, therefore, have better placement. The more yellow pixels are obtained, the more optimized the placement is.

With this guiding image, solving the optimization problem of Equation (2) is now a matter of maximizing the number of yellow pixels. The fragments are essentially in a darker-to-lighter alternative order when placed on a streamline. During the optimization, the placement is adjusted by two parameters, the starting fragment colors (darker or lighter), and the length of each fragment. To avoid the fragment from being too small or too large, we constrain the length of a fragment to the range of $[0.4(SegLen), 0.65(SegLen)]$. The length of the fragment is initalized to $0.5(SegLen)$. We start with a randomly selected streamline and its nearest neighbor (Figure 4(b)), and the goal is to differentiate colors between any two neighboring fragments from these two streamlines. In particular, as shown in Figure 4(e), the starting fragment on the target streamline (the upper one) is adjusted to a lighter fragment (yellow and white in this example), according to the colors of its neighbor on the reference streamline (i.e. blue and black). Its fragment length is also adjusted to differentiate colors with its neighbor. Such adjustment is then applied to all other streamlines to maximize the size of the yellow region in the guiding image (Figure 4(f)). Once the placement of the fragments is optimized, we can then texture the streamline with RAPs with the optimized parameters. Figures 4(d) and (e) show the result after optimization.

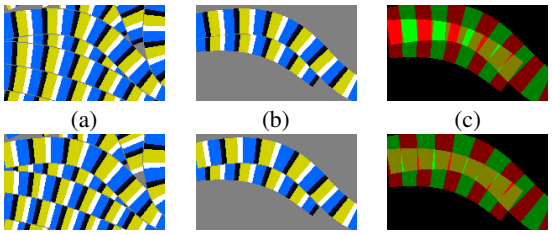


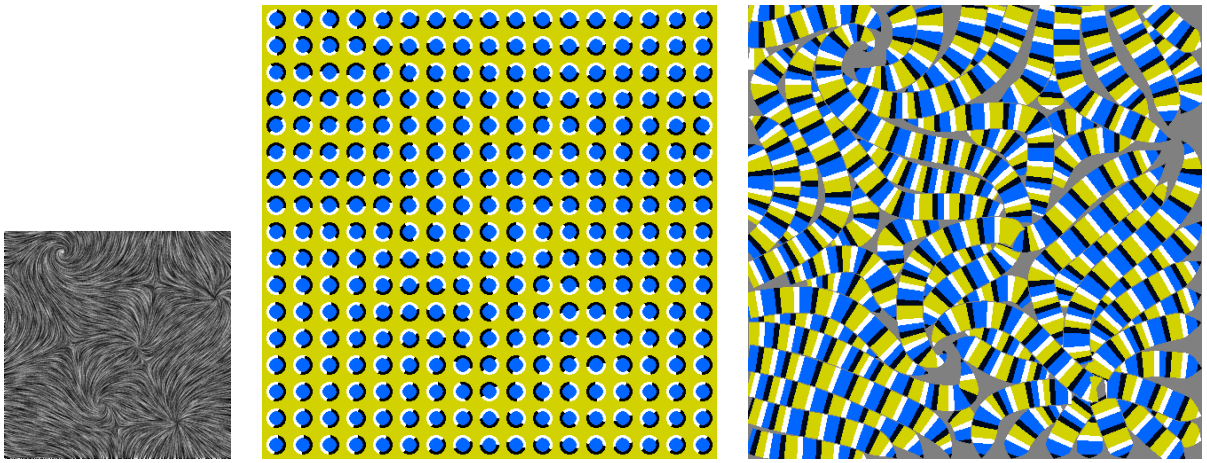
Figure 4: *Fragment offset optimization. Top row: before optimization. Bottom row: after optimization.*

Figure 5 shows an example of the streamline-based RAP placement. The input vector field in Figure 5(a) is rendered by the LIC method [Cabral and Leedom 1993]. The proposed method can express one more piece of information that compares to LIC, namely the flow direction.

Finally, we slightly separate the streamlines with a middle-gray boundary (width of 1-2 pixels). Without such a boundary, neighboring RAPs may accidentally merge and change the pattern. We also use tapering to smoothly reduce the width at the two ends of each streamline to give a stylish result.

5 Color Combination in RAP

Traditionally, the colors of RAP are manually selected. The choice of colors is usually very limited. In this section, we propose heuristics to extend the RAP color combination from the limited color set to a wide range of color. Given an input color C_0, we want to find the other three colors to fit in RAP with the optimized illusion which is closely related to the perceived lightness of stimuli. Since the CIE $L^*a^*b^*$ (CIELAB) color space aspires to the perceptual uniformity and its L^* component closely matches the human perception of lightness, we therefore first convert C_0 to CIELAB(L_0, a_0, b_0) to find the other three colors.



(a) (b) (c)

Figure 5: *RAP placements. (a) The input vector field, (b) tile-based result, and (c) our streamline-based result.*

5.1 Determining Four-Gray RAPs

First, we consider C_0 as gray and its CIELAB value as (L_0, a_0=0, b_0=0). Based on the discovery of Kitaoka and Ashida [2003], a basic B-DG-W-LG configuration is denoted as $RAP_0 = \{0, 25, 100, 75\}$ using their lightness values in CIELAB color space, and such order can be rotated circularly to create another three RAP_i, $i = \{1, 2, 3\}$. They all belong to the TYPE II stimulus because their lightness contrasts are not changed. In the following pseudocode, we use a greedy nearest-first approach to determine a four-gray RAP (L_0-L_1-L_2-L_3). Their lightness values are then stored in a table LL[4]. For a TYPE II stimulus, the existing B-DG-W-LG combination is a good choice for a RAP and their lightness values are stored in a table L[4]={B=0, DG=25, W=100, LG=75}.

FOURGRAYRAPFINDING(L_0, L[])
1 Find $i_{nearest}$, s.t. $\arg\min ||L_0 - L[i_{nearest}]||$
2 $LL[0] = L_0$;
3 **for** $i \leftarrow 1$ **to** 3
4 **do** $LL[i] = L[(i + i_{nearest}) \bmod 4]$;
5 **return** LL[];

The basic idea behind our greedy method is very simple. We first pick up one, say RAP_i^*, of the above four RAP_i, $i = \{0, 1, 2, 3\}$ such that the difference between L_0 and the first element of RAP_i^* is minimal. Then, we replace the first element of RAP_i^* with L_0. In other words, we wish to create a new RAP that is very similar to RAP_i^*. This new RAP is a TYPE II stimulus. Finally, we convert L_1, L_2, and L_3 with (a_i=0, b_i=0) to their gray values and hence a four-gray RAP is determined.

Readers may intuitively think about modifying L_1, L_2, and L_3 with some offsets based on the difference between L_0 and the first element of RAP_i^*. However, from our evaluation, this simple extension may only work in some cases. The magic combination of (B=0, DG=25, W=100, LG=75) is still the best choice for illusory motion.

5.2 Determining Four-Color RAPs

Up to now, little work has addressed how color influences the illusory motion perceived in a still image. Since we cannot find related studies or theory in existing literature, we propose a heuristic method based on the following observations.

Observations Kitaoka[2003] designed eight sets of four-color RAPs (C_0, C_1, C_2, C_3) to demonstrate his "Rotating Snakes." We convert each set of four-color RAP into CIELAB and consider their L components only. Interestingly, both (L_0, L_1, L_2, L_3) and B-DG-W-LG have the same direction of the perceived motion. For each set of four-color RAP, its perceived motion direction is mainly determined by C_1 and C_3. We plot a vector $\overrightarrow{C_1C_3}$ for each RAP and consider their a and b components only in Figure 6. Then, we have the following observations. When C_1 and C_3 are at opposite quadrants of the coordinate system (i.e., a, b are opposite in signs) in Figure 6, a RAP (C_0, C_1, C_2, C_3) is observed to generate stronger motion than other patterns. We also observe that the magnitude of the perceived motion seems to be proportional to the distance between C_1 and C_3 in CIELAB.

Based on the above observations, CIELAB seems to be an appropriate color space for studying the illusory motion. Given any color C_0, we determine C_1, C_2 and C_3 such that a four-color RAP (C_0, C_1, C_2, C_3) can generate an illusory motion with the following pseudocode.

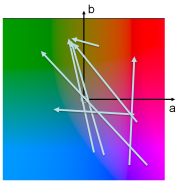
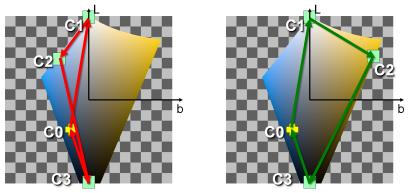


Figure 6: $\overrightarrow{C_1C_3}$ *of the four-color RAPs in CIELAB.*



(a) before rotation (b) after rotation

Figure 7: *Rotating C_2 by 180 ° to make C_0 and C_2 in different opposite quadrants to enhance the perceived motion.*

FOURCOLORRAPFINDING(C_0, L[], *degree*)
1 $LabC_0 = rgb2Lab(C_0)$;
2 $LL[] =$ FOURGRAYRAPFINDING($LabC_0.L$, L[]);
3 $LabC_1 = Lab(LL[1], LabC_0.a, LabC_0.b)$;
4 $LabC_2 = Lab(LL[2], LabC_0.a, LabC_0.b)$;
5 $LabC_3 = Lab(LL[3], LabC_0.a, LabC_0.b)$;
6 **if** $i_{nearest} == 0$ or 2
7 **then** $Rotate(LabC_2, degree, L_{axis})$;
8 $Rotate(LabC_3, degree, L_{axis})$;
9 **else** $Rotate(LabC_1, degree, L_{axis})$;
10 $Rotate(LabC_2, degree, L_{axis})$;
11 **for** $i = \{1, 2, 3\}$
12 **do** $Find\_gamut\_boundary(LabC_i)$;
13 $C_i = Lab2rgb(LabC_i)$;
14 **return** C_1, C_2, C_3;

To ease the explanation of the pseudocode, we plot the CIELAB color space in 2D with $a = 0$ as shown in Figure 7. The computation is carried in CIELAB (line 1). Then, in line 2, we obtain a specific lightness table LL for the input C_0 using the greedy nearest-first approach in Section 5.1 and the input table $L[4] = \{0, 25, 100, 75\}$, as the standard RAP of (B, DG, W, LG). After determining the lightness in table LL and the chromatic value of C_0, we can generate $LabC_1$, $LabC_2$, and $LabC_3$. According to our observations, both $\{C_0, C_2\}$ and $\{C_1, C_3\}$ should be at opposite quadrants in order to give a strongly perceived motion. Therefore, in lines 6-10, we rotate them along the L axis in the 3D CIELAB color space to change their chromatic values without changing their L components, placing them at opposite quadrants. Figure 7 gives an example of rotating C_2 by $degree = 180$. In most cases, this setting helps to increase the illusion.

In lines 11-13, we correct $LabC_1$, $LabC_2$, and $LabC_3$ before converting them to the RGB color space. This correction is required for two reasons. First, the CIELAB color gamut is a distorted 3D cube [Lindbloom 2007]. When we assign C_0's chromatic value to $LabC_i$ for $i = \{1, 2, 3\}$, $LabC_i$ may be outside the CIELAB color gamut. We move it back to the boundary point of the color gamut in two steps: 1) we first compute an isoluminant plane with $L = LabC_i.L$, and 2) along the vector from ($LabC_i.L$, $LabC_i.a$, $LabC_i.b$) to ($LabC_i.L$, 0, 0), we compute the intersection at ($LabC_i.L$, a, b) with CIELAB color gamut. The second reason is that the magnitude of the perceived motion appears to be proportional to the lengths of $\overrightarrow{C_0C_2}$ or $\overrightarrow{C_1C_3}$ in CIELAB color space. We heuristically move them to the boundary of the CIELAB color gamut with the same luminance of $LabC_i.L$, thereby increasing the circumference of a four-sided shape ($LabC_0$, $LabC_1$, $LabC_2$, $LabC_3$) as shown in Figure 7 (b).

Although we can generate a color combination with an illusion effect for various colors, some combinations are weaker than others, especially when the input color carries middle luminanace, i.e. $L = 50$. The input colors near white or black belonging to TYPE IIb also have a weaker effect. Therefore, for a better effect in TYPE IIa, we can shift the input color to a lighter area while keeping the same hue and enhancing the input color saturation.

The color RAP selection is useful when we want to colorize a streamline with varying colors. We can divide the streamline into several segments and the average color of the region underlying each segment is used to find the RAP combination for that segment. To produce the strongest illusion, we reorder the colors of the RAP in the streamline to follow the lightness change of the TYPE IIa four-gray pattern in the order of (B-DG-W-LG). Note that, manually selecting a proper color combination for a strong illusion effect is difficult and tedious, especially for an image with a complex structure. Figure 8 shows three examples for designing a colorful streamline illusion. Mistakes in the lightness order (Figure 8(a)) or the hue difference (Figure 8(b)) may easily happen and may therefore reduce the illusory effect. The proposed method automates the color selection and strengthens the illusion (Figure 8(c)).

6 Opposite Directional Flow Generation

So far, we assume that the vector field is given. If an image is given as an input without a vector field, we have the freedom to generate a vector field for a self-animating image. Other than the RAP placement and color selection, the contrast in *direction* can also strengthen the illusion. By observing Kitaoka's results, a stronger illusion occurs when the motion directions of neighboring regions are opposite to each other. It may be that, when looking at a fixed point, an illusion occurs in the peripheral vision where the opposite directional flow can be more easily perceived and even enhanced. Another possible reason could be that the opposite direction may easily lead to saccades and could, therefore, evoke the illusion.

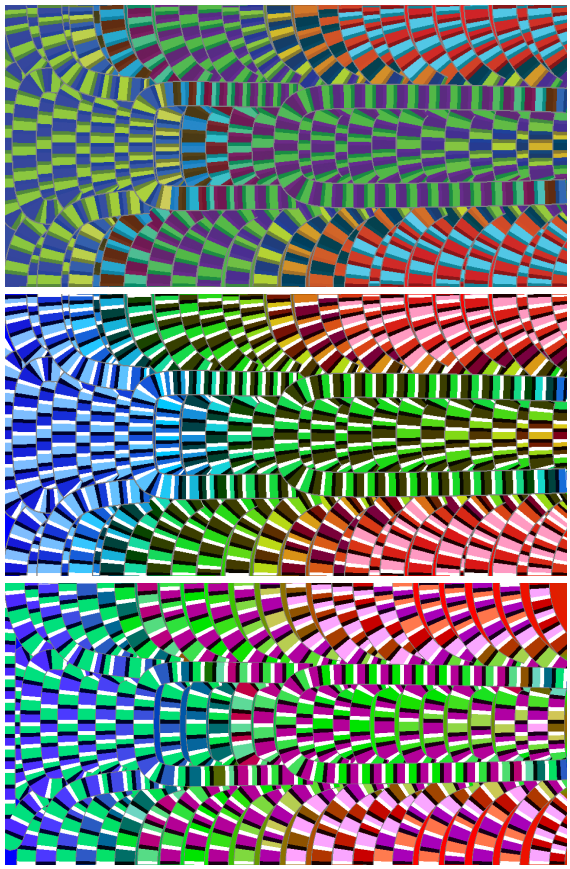


Figure 8: *Color streamline-based illusion. (a): Mistake in lightness order. (b): Small hue difference leads to weak motion. (c): The proposed method.*

Based on this, we introduce two illusion-strengthening vector-field generation methods given an input image.

6.1 Gradient Vector Flow

Given an image containing regions with arbitrary shapes, we now aim at constructing a vector field in which the neighboring regions have opposite directions. There are two kinds of opposite directions, one is parallel to the boundary and the other is perpendicular to the boundary. Both conditions can be generated based on the initialization from the gradient vector flow (GVF) method [Xu and Prince 1997] . The GVF is calculated by applying a generalized diffusion to the gradient components of an image edge map so it can assure a smooth vector field. The smoothness property of GVF allows us to directly apply our streamline-based illusion (Section 4). However, the GVF cannot guarantee the opposite direction requirement. Therefore, we propose two modifications to GVF in order to generate the perpendicular and parallel opposite directional vector fields.

For the perpendicular type, the idea is to generate GVF for each region individually. To do so, we need to segment the input image and then compute the GVF for each region independently. In this way, vectors inside a region tend to point away from the boundary in its perpendicular direction. Therefore, vectors in two neighboring regions are opposite to each other because they share the same boundary. The parallel type can be easily achieved by rotating the vectors clockwise by 90 degrees. The rotation keeps the vectors of the neighboring regions opposite to each other, while keeping them parallel to the boundaries.

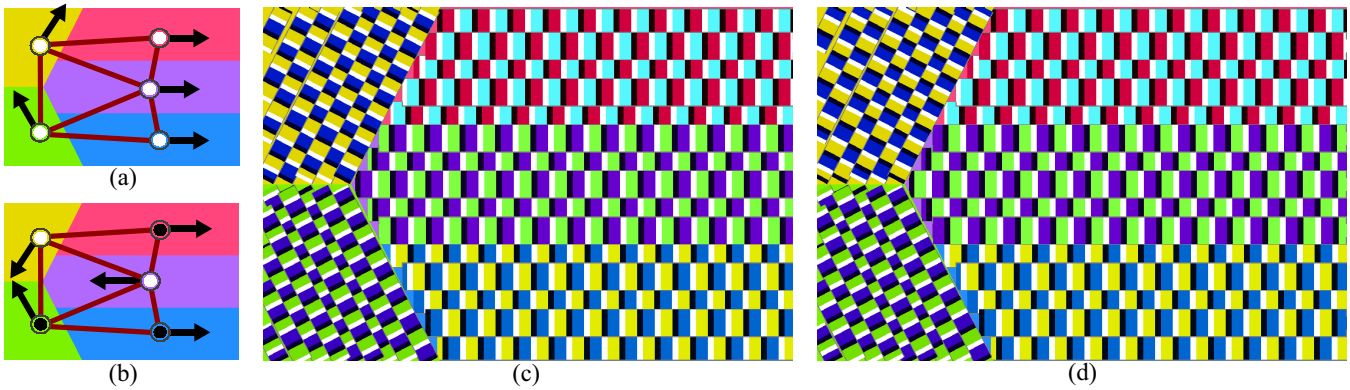


Figure 9: *Graph-based vector field generation. (a) The graph and direction before optimization. (b) After optimization. (c) An un-optimized vector field result. (d) An optimized vector field result.*

6.2 Graph-based Vector Field Generation

The method described above is intuitive and robust. Its distinctive advantage is that the generated vectors follow the region boundaries. However in some cases (such as the arrow pattern in Figure 9), the user may expect the vectors to flow in a consistent direction along the region skeleton, instead of swirling along the region boundaries, and keeping the opposite directions of neighboring regions.

To do so, we need to maximize the appearance of neighboring regions containing opposite flow vectors. We also try to maximize the length of the boundary with opposite vectors on its two sides. Hence, we formulate it as an optimized graph coloring problem. Each region is represented as a node in the graph and an edge represents the connection (shared boundary) between two regions. For an edge $e_{i,j}$ connecting nodes i and j, we use the difference between two region skeleton directions of regions i and j as the energy of the connecting edge, e.g. $E(e_{i,j}) = \arccos(Dir_i \cdot Dir_j)$, where Dir_i means the direction of the skeleton in region i. The larger the inner angle between Dir_i and Dir_j, the larger the opposite directional effect. By accumulating the energy on each edge in the graph, we can measure the directional relation among all regions as:

$$E_{graph} = \sum_{i \in R} \sum_{j \in N_r(i)} w_b \arccos((C_i \cdot Dir_i) \cdot (C_j \cdot Dir_j)) \qquad (3)$$

$$C_i,\ C_j = \begin{cases} +\mathbf{1}, & \text{black}, \\ -\mathbf{1}, & \text{white}. \end{cases} \qquad (4)$$

where R is the node set, $N_r(i)$ is the neighboring region of i with the shared boundary. We modulate the $E(e_{i,j})$ by w_b, which represents the length of the shared boundary between two connected regions so the energy favors a longer connected area.

To maximize the direction energy E_{graph}, we can only change the direction of a region skeleton by multiplying by -1. Therefore, the energy maximization is reduced to a two-coloring graph problem. We use C_i to indicate the color state in Equation (4): black is for preserving the direction and white is to invert the direction. In most cases, we can find the best solution by maximizing Equation (3). By modifying the vector directions in the image according to the optimized graph, we can generate the maximized opposite flow result.

We use Figure 9 for illustration. In the initial step (a), the right three regions are in the same direction. The illusory motion is therefore weak (shown in (c)). According to Equation (3), these three regions in the graph are optimized with different colors (shown in (b)). Figure 9(d) shows the corresponding result in which the direction of two regions are inverted to give a stronger illusion. Notice that this coloring result is already modulated by w_b, which considers the contribution from the length of the shared boundaries.

7 Applications and Discussion

Our method can be easily extended to generate various interesting results, including tile placement, illusory toon-shading, and TYPE III illusions. Readers can perceive stronger motion from the images shown in this paper by enlarging those images on display. Glance around the images and do not stare at a fixed place of the image too long. In our experiments, the best viewing distance to screen is roughly half the width of the screen.

Tile placement Figures 11 and 14 show an illusory Tai-Chi pattern and SIGGRAPH logo produced by our system respectively. Our method can be naturally extended to generate tile-based illusions. The idea is the same: drawing the four color RAPs according to the vector field. Figure 10(a) gives an illustration of a single tile pattern configuration. We generate a tile-based result by placing primitive tiles surrounded by an outer boundary, then we rotate the tiles along the input vector field and colorize them with the color RAP combination from the input image. Additionally, the consistent distance between neighboring tiles is maintained to provide a uniform illusory motion, and arrow patterns are used to enhance the motion direction.

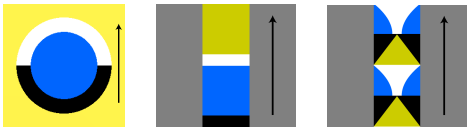


Figure 10: *Basic configuration of (a) tile, (b) toon-map, and (c) TYPE III streamline. The arrow indicates the illusion direction.*

Illusory toon-shading We also attempt to apply the four-color RAP as a toon-map for shading a 3D surface. The "exaggerated shading" technique [Rusinkiewicz et al. 2006] is adopted here to capture the illuminance. The illuminance is mapped to the four-color RAP map instead of the original toon-map, as in Figure 12. The illusion can also indicate the direction of the light source. We can further strengthen the illusion by using an opposite direction illusion pattern on the background.

TYPE III streamline Kitaoka recently proposed a TYPE III illusion pattern for central vision. For example, in Figure 10(c), each lighter color diamond pair is surrounded with darker colors and vice versa. Therefore, it meets the TYPE III condition. Although the TYPE III pattern may be weaker than TYPE II and only effective along straight streamlines, it has advantages in perceiving the illusion even when you stare at it. Replacing TYPE II RAPs with TYPE III RAPs, we can easily achieve the illusion in central vision. We demonstrate a TYPE III streamline example for an ocean flow in Figure 13.

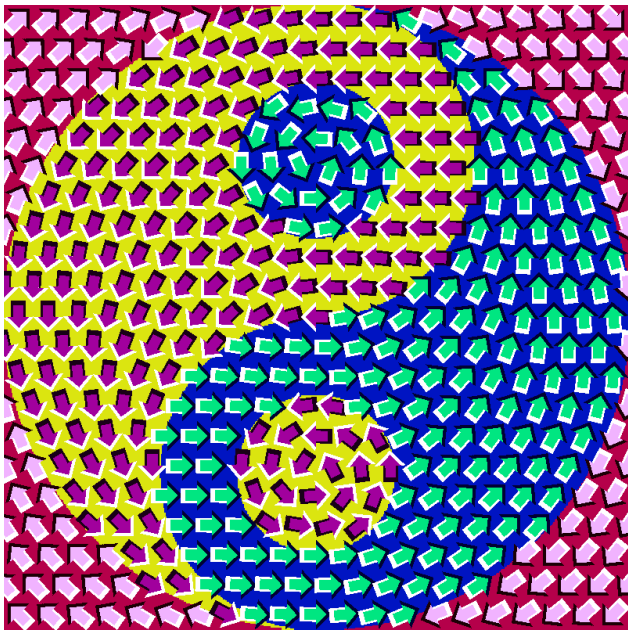


Figure 11: *Illusory tiles on Tai-Chi.*

Applications A new stylization of Van Gogh's "Starry Night" is demonstrated in Figure 1. In the original painting, Van Gogh used high contrast colors and long strokes to express the sensations of flowing motion. Based on the composition of this painting, we generate the illusion effect by applying the following steps. First, we segment the image into several regions and fill the region with its average color, which, in turn, is used to determine the nearest color with a better illusion (Section 5.2). Second, we determine the opposite directional flow field to represent the input image. Finally, we texture the optimized streamlines by colorizing with the selected RAPs to create a self-animating image for the "Starry Night" as shown in Figure 1. Finally, in Figure 14, we show an example of an advertising design using the proposed method.

Discussion and Limitations The three essential elements, include the RAP placement, the coloring of RAP, and the opposite directional flow contribute to the strength of the illusion. If any one of these requirements cannot be satisfied, the illusion may be weakened. The comparison for missing elements in a suitable color combination is demonstrated in Figure 8, and the optimization of directional flow is shown in Figure 9. Another important issue is the resolution of the individual RAP elements making up the self-animating images. Each individual perceives the strength of the illusion differently depending on the size of the RAP elements. Generally speaking, when these motion-inducing elements are too small, the evoked motion illusion vanishes. This can be easily verified, when looking at the same illusion pattern from a farther viewing distance. Therefore, our method could not well express the motion flow when the input images contain too many small regions.

The strength of the illusion primitive is also important. In the optimized Fraser-Wilcox illusion, we mainly use TYPE II, and extend to the TYPE III in Section 7. Although TYPE I can be easily applied on the streamline, it is not used because it has the weakest illusion among the four categories. In Figure 5, we compare the streamline and tile based methods. The Fraser-Wilcox illusion is produced by the edge with asymmetric contrast. In a tile-based RAP, not all the surrounding edges are perpendicular to the vector field, therefore reducing the strength of the illusion. This interruption may make it difficult to express the flow. In contrast, all the edges in the streamline are perpendicular to the vector field to give a consistent illusion direction and therefore strengthen the illusion.

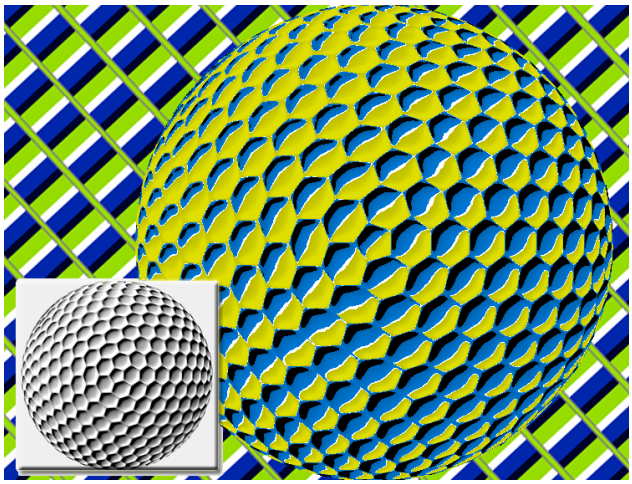


Figure 12: *Illusory toon-shading on the golf ball. Sub-image is made by Xshade(http://xshade.cs.princeton.edu/).*

8 Conclusion

Based on existing findings on illusory motion, we propose a computational method to generate self-animating images from given input images. We can deal with a more general four-color RAP. Images with a strong motion illusion are produced. We also provide algorithms to generate vector fields that can strengthen the illusion when no vector field is given. Various extensions and applications are demonstrated. Illusory motion is still an open and challenging cross-disciplinary field of research. Many unsolved problems still exist; for example, it is very challenging to generate a self-animating image that is faithful to every single detail of the input image. In addition, a metric to measure perceived motion in an image will be very useful to quantify the illusion.

Acknowledgments

We would like to thank all reviewers for their valuable suggestions to improve the paper. We are grateful for the great discovery of Akiyoshi Kitaoka. Thanks to Zhanping Liu and Han-Wei Shen for providing access to their flow dataset. This work is supported by the Landmark Program of the NCKU Top University Project under Contract B0008, the National Science Council, Taiwan under NSC-96-2628-E-006-200-MY3, and the Research Grants Council of the Hong Kong Special Administrative Region, under RGC Earmarked Grants (Project No. CUHK417107).

References

BACKUS, B. T., AND ORUC, I. 2005. Illusory motion from change over time in the response to contrast and luminance. *J. Vis. 5*, 11 (12), 1055–1069.

CABRAL, B., AND LEEDOM, L. C. 1993. Imaging vector fields using line integral convolution. In *Proceedings of ACM SIGGRAPH 1993*, ACM Press / ACM SIGGRAPH, New York, NY, USA, 263–270.

CONWAY, B. R., KITAOKA, A., YAZDANBAKHSH, A., PACK, C. C., AND LIVINGSTONE, M. S. 2005. Neural basis for a powerful static motion illusion. *J. Neurosci. 25*, 23 (June), 5651–5656.

FAUBERT, J., AND HERBERT, A. M. 1999. The peripheral drift illusion: A motion illusion in the visual periphery. *Perception 28*, 617–621.

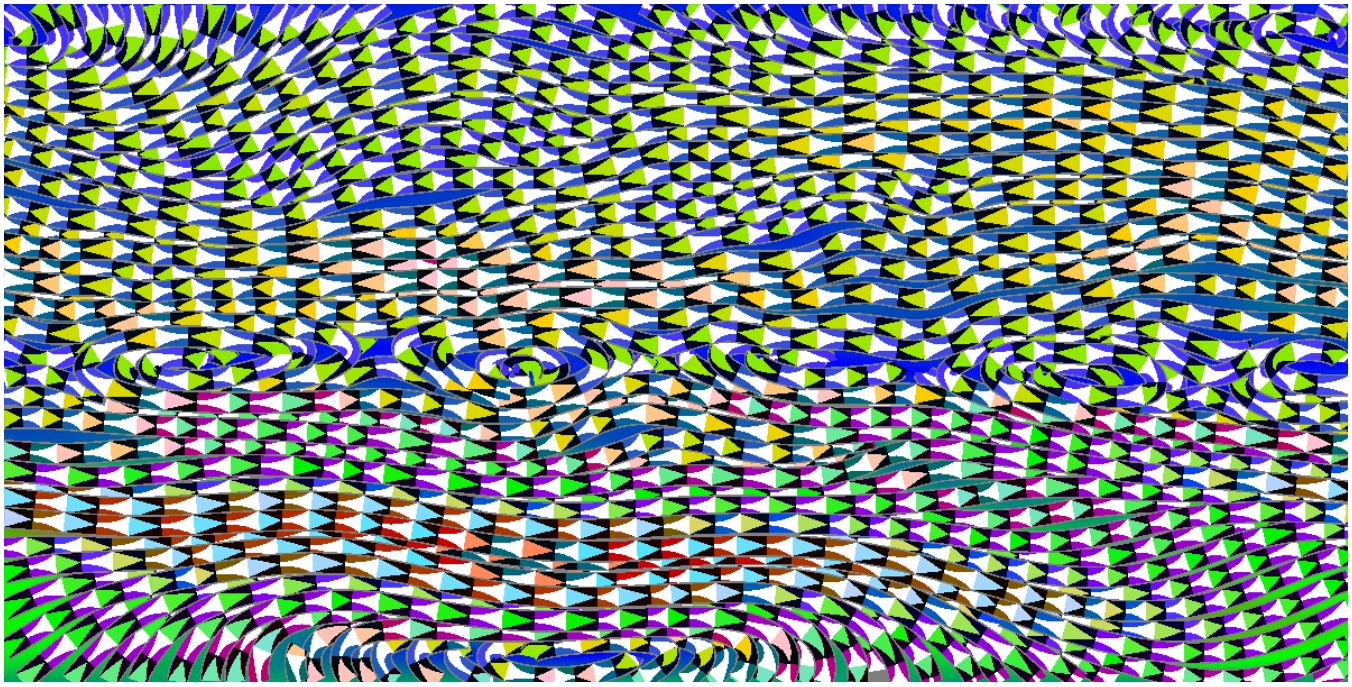


Figure 13: *An example of TYPE III streamlines on showing an ocean flow.*

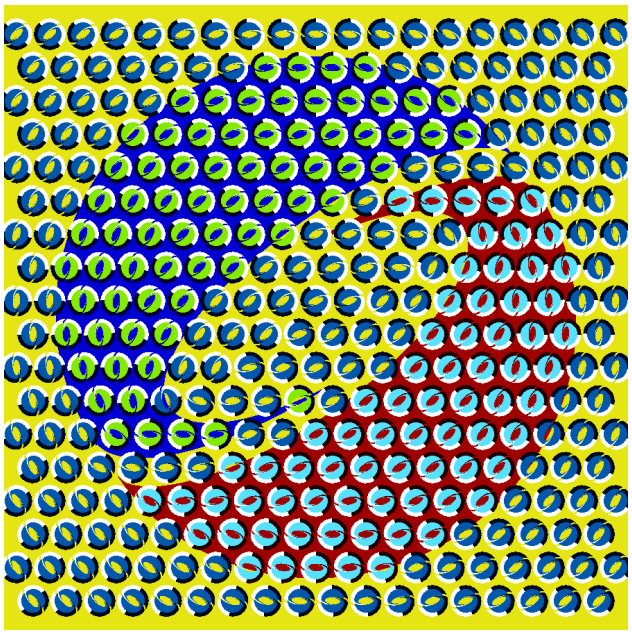


Figure 14: *Advertising design using the SIGGRAPH logo.*

FRASER, A., AND WILCOX, K. J. 1979. Perception of illusory movement. *Nature 281*, 565 – 566.

FREEMAN, W. T., ADELSON, E. H., AND HEEGER, D. J. 1991. Motion without movement. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, ACM, vol. 25, 27–30.

GOSSETT, N., AND CHEN, B. 2004. Self-animating line textures. Tech. rep. `http://www.dtc.umn.edu/~gossett/publications/`.

KITAOKA, A., AND ASHIDA, H. 2003. Phenomenal characteristics of the peripheral drift illusion. *VISION (Journal of the Vision Society of Japan) 15*, 261–262.

KITAOKA, A., 2003. Rotating snakes. `http://www.psy.ritsumei.ac.jp/~akitaoka/rotsnakee.html`.

KITAOKA, A. 2005. *Trick Eyes Graphics*. Tokyo: Kanzen.

KITAOKA, A. 2006. Anomalous motion illusion and stereopsis. *Journal of Three Dimensional Images (Japan) 20*, 9–14.

KITAOKA, A. 2006. The effect of color on the optimized fraser-wilcox illusion. *the 9th L'ORE'AL Art and Science of Color Prize*, 1–16.

LINDBLOOM, B., 2007. Lab Gamut Display. `http://brucelindbloom.com/LabGamutDisplay.html`.

MASUCH, M. 1999. Speedlines: depicting motion in motionless pictures. In *SIGGRAPH '99: ACM SIGGRAPH 99 Conference abstracts and applications*, ACM Press, New York, NY, USA, 277.

MEBARKI, A., ALLIEZ, P., AND DEVILLERS, O. 2005. Farthest point seeding for placement of streamlines. In *Visualization, 2005. VIS 05. IEEE*, 479–486.

MURAKAMI, I., KITAOKA, A., AND ASHIDA, H. 2006. A positive correlation between fixation instability and the strength of illusory motion in a static display. *Vision Research 46*, 2421–2431.

RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated shading for depicting shape and detail. *ACM Transactions on Graphics 25*, 3, 1199–1205.

SHOUP, R. G. 1979. Color table animation. In *Computer Graphics (Proceedings of ACM SIGGRAPH 79)*, ACM Press, 8–13.

WEI, L.-Y. 2006. Visualizing flow fields by perceptual motion. Tech. Rep. MSR-TR-2006-82, Microsoft Research, June.

XU, C., AND PRINCE, J. 1997. Gradient vector flow: A new external force for snakes. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '97)*, IEEE, 66–71.