**ABSTRACT**

Trick art is a genre of artistic expression that can give not only beautifulness but also entertainment. Lenticular image (or printing) is one skill of trick art that shows different image depend on the viewer’s viewpoint by controlling refraction of light. The object, seen by viewer in some view angle, goes shimmering when the viewer change the view angle. In computer graphics area, other trick arts were simulated, but simulation of lenticular image is not well progressed. So in this paper, we do simulate the lenticular Image. For this, we first set the position of input object image on the given background image depend on the similarity of gradient. And then blend the object into background image. Lastly, we cover the lenticular lens on the image that is the composition of original background image and blended image. Finally, we can generate the 3D scene which simulates the lenticular images.

**KEY WORDS**

Object Positioning, Lenticular, Simulation

---

**1. Introduction**

Trick Art is a new art genre that makes a person see an image differently by work applying some refraction and reflection of light, perspective, or shade into paintings. So it can give viewers interests or artistic entertainments. Lenticular image (or printing) is one such form of trick art. It permits the viewers to see, or not an object according to their viewing angle, applying the trick by controlling the refraction of light. Figure 1 shows an example of a lenticular image, in which the man and woman’s clothe that can be seen from the right cannot be seen when the view is turned into the left.

The simplest way to express a lenticular image is to take two images and split them vertically into strips of the same width. Placing pieces from each image alongside one another in turn so that those from the first image face slightly to the left and those from the second face right (forming a zigzag when viewed from above), we have a lenticular image that looks different when seen from left or right. It is an easily expression method for lenticular printing. However, such images have one particular problem: when observed from the front, both images can be seen in their separate, striped form. To remedy this, we can use a lenticular lens. This method also uses two images that are split vertically. After attaching the pieces together in turn, an array of semi-cylindrical lenses is placed over the image which image can be seen depends on the viewing angle as light is refracted or scaled through this lens. This technique is used in 3D display effects, as it allows the overlapping of two images from the front. The attributes of various lenses, such as their thickness and refractive index, are widely studied in the field of computer graphics. Attributes of various lenses, like thickness, refractive index, etc. are studied. However, studies into the simulating of lenticular printing based on this lens are not advanced.

In this paper, we propose an algorithm to express the trick art effect of lenticular printing with one object image and one background; Object will be set on background image. Our algorithm consists of two steps. First, the object is situated on the background. We want to synthesize the object in the background image naturally, so we find the best position for the object based on the gradient of color between the two. Next, we use a Poisson blending method[5] to position the object smoothly. The second step is to apply a lenticular lenses with the original background image and the blended image obtained from

---

**Figure 1. Example of Lenticular Image( left to right ), “The costume of painter kiss” by Junsung Bae**
the first step. We split the two images vertically into predefined widths, and alternately combine the pieces one-by-one to form a single stripe-patterned image. Next, we cover the combined image with a lenticular lens whose attributes are pre-calculated and simulate the lens by applying a raytracing method. Finally, we can express the trick art effect.

Our paper makes a number of contributions. We can reproduce the lentcular image using a computer simulation. This can reduce the time and cost for artist to working their art. Secondly, we propose the smooth positioning object to background. This method can be used various ways connected with trick art method later.

2. Related Work

There are various studies into computer graphics skills used to cause optical illusions with refraction, reflection, perspective, or shade. [9] combined low- and high-frequency images into one hybrid image. Humans can then see two images from the one hybrid image. [6] applied an ideal high pass filter and a Gaussian low pass filter to two images and combined them into one. The resulting image also causes humans to see two images depending on the viewing distance. By applying the RGBZ depth, [7] suggested an approach that extends the 2D passepartous technique to 3D. This gives the impression of an object jumping out of the screen. [8] used a splat texture, which allows the image to be recognized by a human but cannot be distinguished by automatic computer algorithms. However, different from these, our paper concerns the simulation of trick art by a lenticular lens, which gives a view-dependent image.

The skills required to set the object smoothly onto the background are alpha blending and Poisson blending [5]. Alpha blending not only considers RGB values, but also an alpha value that controls the transparency. Two image scan thus be blended by controlling the alpha value. Poisson blending synchronizes the object’s gradient with the background image’s gradient. After transforming the given background image into a line drawing, we generate the hidden-picture puzzles based on an object database. Numerous techniques for finding the best position for the object have been proposed. [1] extracts the feature points of the object and the background in order to compare their similarity. In [2], one or more objects are hidden in the original image recursively, generating the artistic result. Using a matching method based on the Hausdorff distance, the best position for each object is found so as not to invade the original image’s silhouette. [3] calculates the object’s luminance according to the background image, and expresses the camouflage effect after texture synthesis. The technique in [4] finds the best position based on the direction and intensity of edges, and hides the object in the background by a transformed Poisson blending method. Our algorithm is based on [4], and we use gradient information in the object positioning step.

There are various papers analyzing lenticular lenses. [10] explains the method behind visualizing a 2D image in 3D without the use of special glasses, and explains the basis of the lenticular lens. [10] defines the width of the lens pitch, and describes the calculation of the pitch size and other lens attributes, as well as the viewing angle range of a lenticular lens. Similarly, most research [11, 12] has introduced and analyzed lenticular lenses with respect to 3D display techniques, but there is little research on the simulation of lenticular images.

Therefore, in this paper, after we set the object in the background following the method of [4], we simulate the lenticular lens to generate a view-dependent image, thus expressing one of the skills of trick art.

3. System

3.1 Overview

Figure 2 shows an overview of the system described in our paper. Our proposed system generates a 3D lenticular
image scene from the background image where the object is located, using the given object and background images. Our algorithm consists of two steps: Object Positioning and Lens Simulation.

In the Object Positioning Step, after finding the best position for the object image, we blend it into the background. In this process, we build an Energy function based on the gradient. Once this function has been applied to find the optimum position, we use the Poisson blending method from [5] to reduce the degree of difference between the object and the background.

In the Lens Simulation Step, we analyze a real lenticular lens and use a ray-tracing method to simulate it in the computer graphics arena. At this point, we use the original background image and the blended background image from the object positioning step.

3.2 Object Positioning

To blend the given object into the background automatically, we must find the best position for the object in the background. When object image O and background image H are given, we use gradient information. If we consider gradient information alone, then some problems may be encountered. In particular, the position where the gradients of two edges are exactly opposite may be selected as a good position (see Figure 3).

When we choose the best position based on gradient, we can reduce the distortion of the object, because the object’s gradient is not significantly changed by Poisson blending. Given images H and O, we extract the gradient maps GH and GO. These consist of parameters (\(\nabla x\), \(\nabla y\), \(m\)), which give the gradient direction of the x-axis, the gradient direction of the y-axis, and the magnitude of each. Next, we build the energy function \(E\), which compares the similarity in gradient between point \(p\) in the object and point \(q\) in the background. In this function, \(S\) is the area of mask \(\Omega\).

\[
E = \frac{1}{S} \sum_{p \in \Omega} G_H(q) \cdot G_O(p)
\]

\[
S = \text{AREA}(\Omega)
\]

We wish to find the point \(q\) that maximizes \(E\). After finding the best position on the background, we use Poisson blending to insert the object. Figure 4 shows this process flow. (a) is the given object and background image. (b) is the visualization of each point’s energy value on the background. We normalize the values in the range 0–255 so that (c) shows the result of the object positioning.

![Figure 3. Problem of comparing edges only](image)

![Figure 4. Process of Image Positioning](image)
3.3 Lens Simulation

In this step, we simulate a view-dependent lenticular image by analyzing the lenticular lens, before finding and applying a suitable method of simulation. We pre-define certain lens attributes, and position this lens over the stripe-patterned image resulting from the combination of the original background and the blended image. Finally, the lenticular image is generated depending on the viewing angle.

3.3.1 Concept of lenticular lens

A lenticular lens is an array of semi-cylinders of regular width that cause light to be refracted. The lens thus causes humans to see only part of the image, depending on their viewpoint. Figure 5 visualizes this principle for ease of understanding. Because lenticular lenses allow people to see a different set of pixels depending on their viewpoint, they can be used to express various effects, such as 3D display, flip, zoom, motion, and morphing. In the case of 3D display, a lenticular lens can transform a flat image to a stereotactic image using Binocular Disparity, which controls perspective. In the case of other effects, images that cannot be seen from some viewpoints will appear when the viewpoint is rotated. In this paper, we simulate the flip effect of a lenticular lens.

Lenticular lenses have a number of attributes, including the pitch, mean width, radius, refractive index, and thickness. Generally, lenses with a pitch of 335.65 μm, radius of 190.5 μm, thickness of 457 μm, and refractive index of 1.557 are used in commercial printing, and our algorithm is based on this specification.

3.3.2 Simulation

To simulate a lenticular lens, we combine the blended image with the original background image, as in Figure 6. If the pixel size is defined by a pitch of N, then setting smaller values of N will improve the quality of the result. Figure 6 shows an image with a pixel size of 6, which means that each image is split into 3-pixel vertical pieces.

After we cover the combined image in Figure 6 with the lens, we observe a different image according to changes in the viewing angle. We use a ray-tracing method, which can easily express the tracking of an eye. Because, in general, the human eye takes in light from a unit face rather than a unit point, ray-tracing from a single viewpoint cannot mimic the human eye. This causes the problem that the tracking of light into the lenticular lens cannot be simulated. As we can see in Figure 6, the refraction does not occur in front of the scene. Thus, we must use a distributed ray-tracing method to simulate the lens. If the number of viewpoints increases, we obtain the correct result of lenticular image observation, although at the cost of an increase in computation time. We set the number of viewpoints P, and control this parameter while applying the ray-tracing method.

4. Results and Discussion

The implementation described in this paper used the MFC library in Microsoft’s Visual Studio 2008 for 32-bit Windows to generate results with the proposed algorithm. We also used OpenCV 2.42 to generate the blended images. The computer hardware used for implementation and testing was a standard PC, Intel Core i5-2500 CPU(3.30GHz) with 4.00GB RAM.

With a 448 ×298 background image and a 128 ×128 object image, it took 30 s to process the image positioning algorithm. Furthermore, it took 2 s to generate one lenticular scene with a 4-pixelpitch. The total processing time is controlled by the image size, pitch size, and the number of distributed camera positions.

Figure 7 shows the whole process of our algorithm. (a) shows the object image and (b) the background image. (c) is the blended image after the image positioning step.
Figure 7. Result Overview
and (d) is the combined stripe-patterned image used for simulating the lenticular image. Covering (d) with a lenticular lens produces the angle-dependent images (e)–(j). (e) shows the scene from a -26° viewing angle, (f) is from -12°, (g) is from 0°(front), (h) is from 9°, (i) is from 18°, and (j) is from 48°. As we can see from the result, the rabbit that is visible in (e) is more clear in (f). We can still see the rabbit in the front of the scene. Turning further to the right, we can see the ‘noise’ of vertical lines, which is the silhouette of the rabbit. The rabbit disappears as we move to the right in scene (i). Furthermore, we can see the rabbit again in (j) by turning further right from (i). Thus, whether we can see the rabbit depends on the viewing angle.

Figure 8 shows the result of changing the parameters, pitch size, or number of viewpoints. As you can see from the (a) and (b), if we increase the pitch size, we get the low-resolution scene. The quality of result is watchable when the pitch size is 4, but N = 8 generates bad quality to see. (c) and (d) shows the result depend on the number of viewpoints. The scene with 9 viewpoints is poor than the see. (e) and (d) are making lenticular image with given two images. As we can see from those two, when two similar images are given, we can make lenticular image easily.

5. Conclusion and Future Work

Our paper has suggested a simulation algorithm for lenticular imaging. Our method automatically blends the background image with a synthesized image of the object and the background. Finally, we obtain a view-dependent lenticular image.

Our method has a number of advantages. First, we can find the best position for the object on the background image using the similarity of the gradient and a blending technique. The blended image has a natural feel, because we consider not only edge similarity but also image inside transition. Secondly, by simulating a lenticular lens, we generate a view-dependent image. This optical illusion reproduces the new genre of trick art, which entertains observers. Finally, our algorithm has the possibility of being improved by the implementation of a ray-tracing method, which can be easily achieved by parallel programming.

To improve our results, we could consider some additional details. For example, we can take an aesthetic approach to the quality of the results. The proposed algorithm requires one object and one background image as input. Although we find the best position for the object on the background, the quality of the blended image may be poor if the object is significantly different from the background. To solve this problem, we could take many background images, and build the energy function by considering the aesthetic value of each image with respect to the object. Secondly, we could improve the simulation time. If we can generate a lenticular printing scene in real time, we could make an application with a head-tracking method. This would allow us to handle the image directly, enhancing their enjoyment of the trick art. Lastly, we could apply a photo map instead of ray-tracing to improve the quality of the simulation result.

Acknowledgements

This work was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012 (Grants No.C0004960), and by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (no. 20100018445).

References


Figure 8. Another Result controlling parameters