

Pixel Art with Refracted Light by Rearrangeable Sticks

Yonghao Yue¹, Kei Iwasaki², Bing-Yu Chen³, Yoshinori Dobashi⁴ and Tomoyuki Nishita¹

¹The University of Tokyo ²Wakayama University ³National Taiwan University ⁴Hokkaido University

¹{yonghao,nis}@nis-lab.is.s.u-tokyo.ac.jp ²iwasaki@sys.wakayama-u.ac.jp ³robin@ntu.edu.tw ⁴doba@ime.ist.hokudai.ac.jp

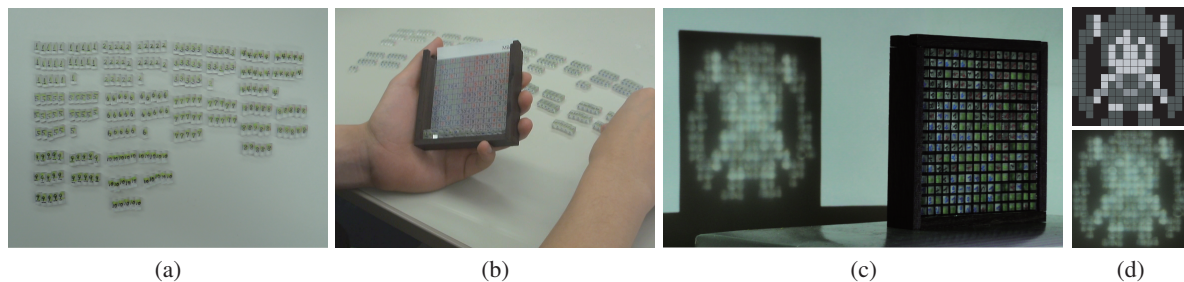


Figure 1: (a) Our fabricated transparent sticks. (b) Arranging the sticks and putting them in a container by a user. (c) When the user puts the arranged sticks between a projector and a screen, a pattern on the screen can be observed due to the refracted light. (d) Top: the desired pattern (please see Acknowledgments for the details), bottom: the pattern on the screen.

Abstract

Pixel art is a kind of digital art that through per-pixel manipulation enables production of a diverse array of artistic images. In this paper, we present a new way for people to experience and express pixel art. Our digital art consists of a set of sticks made of acrylate resin, each of which refracts light from a parallel light source, in certain directions. Artistic users are able to easily rearrange these sticks and view their digital art through the refracted light projection on any planar surface. As we demonstrate in this paper, a user can generate various artistic images using only a single set of sticks. We additionally envision that our pixel art with rearrangeable sticks would have great entertainment appeal, e.g., as an art puzzle.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

Pixel art (a type of digital art) represents game characters or other general images using a carefully chosen set of pixels, each of which has a finite planar dimension and which is usually immediately recognizable visually. Pixel art enables rich power of expression, although it usually consists of only a small number of pixels. Thus, the graphics used in classic video games or in GUI icons are mostly based on pixel art [KL11].

The basic concept of pixel art is to tile pieces of ‘picture elements’ that are the same size. We can find other art formats or toys that share similar concepts, such as 3D pixel art

(e.g. [QBL,Jea10]), tile art, ‘Post-it’ art (e.g. [Liu09]), jigsaw puzzles, or Lego building blocks. These art-forms or toys, in common with pixel art, stimulate human creativity, and people often enjoy creating their desired patterns or shapes by themselves.

On the other hand, the creation of desirable patterns or shapes, is itself an interesting research topic in the field of computer graphics. Many methods, possibly utilizing specialist devices or the manufacture of various objects, have been proposed for reproducing a number of specialized settings, such as the 6D light fields [FRSL08], the desired shadows [MP09], shadings [AM10], sub-surface scatter-

ing properties [DWP*10, HFM*10], and hologram patterns (e.g. [RRMG10]). Additionally, the redirection of light to generate desired patterns is a powerful approach that enables rich power of expression. Such approaches include methods of computing reflective or refractive surfaces for producing caustics [WPMR09, FDL10, PJJ*11], and a method for a high contrast projector using light reallocation [HSHF10].

In this paper, we bring the expressive power of pixel art and the light redirection approach together, and present a new approach to producing creative and enjoyable pixel art, as shown in Figure 1. Our method consists of a set of transparent sticks made of acrylate resin (refractive index: 1.49), each of which refracts light in a certain direction. By arranging these sticks and placing them appropriately between a parallel light source (e.g. sun-light or a projector) and a viewing-surface (screen), one can observe the resulting pixel art on the screen.

By bringing pixel art and the light refraction together, people can enjoy the surprise in that they cannot tell what happens by just taking a look at the sticks because the resulting pattern is only visible when the sticks are appropriately placed between the light source and the screen. In this sense, our art can be used as a tool for steganography.

In our system, the number and types of the sticks are fixed in advance. A user first inputs an image of the desired pixel art. Then, our system solves an optimization problem to determine the layout of the sticks so that the light redirected by the sticks forms the pixel art on the screen. According to the layout, the user arranges the sticks manually and inserts them into a container. Finally, the user can enjoy the pixel art by actually placing the container between the light source and the screen.

Using only a single set of sticks, the user can produce various patterns of pixel art by rearranging the sticks. This concept is similar to that of Lego building blocks, and it helps to stimulate imagination and creativity. We believe that the work that is needed to arrange the sticks is not a burden to the user's creativity, since the amount of effort is small thanks to the nature of pixel art: a small number of pixels is sufficient to enable rich power of expression.

Our method can be closely related to the method presented by Papas *et al.* [PJJ*11]. In their method, they used a NC (Numerical Controlled) machine to fabricate a plate made of acrylate resin, whose surface has micro-facets. With this plate, one can produce smooth caustics that are highly detailed, but the user needs to create individual plates for individual caustics patterns. Our method has some advantages in the terms of the entertainment aspect and the cost for production can be reduced, since the user can produce various patterns using only a single set of sticks.

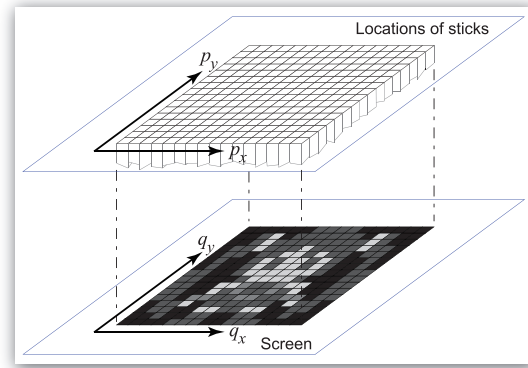


Figure 2: An illustration of the problem settings. We arrange the sticks like a grid.

2. Our Approach

2.1. Problem Settings

As shown in Figure 2, we use a set of two integer 2D orthogonal coordinates to describe the locations of the sticks (p_x, p_y) and the screen pixels (q_x, q_y). We assume that the desired pixel art consists of a set of square pixels, and that their sizes are all the same. As shown in Figure 3, the shape of a stick differs slightly from a parallelepiped. The difference is that one of its faces is slightly tilted in order to redirect light via refraction. The redirected (refracted) light will point to a pixel on the screen (Figure 4). Unless otherwise stated, we let the light that is incident onto the sticks all contribute to one of the pixels in the pixel art. It is also possible to let some of the light refract toward the outside of the resulting pattern. The base size of the stick is the same as the size of a pixel, which could be e.g. 5mm \times 5mm or 10mm \times 10mm. The height of a stick is sufficiently large (e.g., 10mm) for the user to handle it easily.

Because the base shape of a stick is square, if we rotate the stick along its height axis by 90° , 180° or 270° , the refracted light will point to different pixels. Therefore, allowing for the rotation, a single stick can potentially redirect the light to 4 locations. As shown in Figure 4, if the shift or translation in the locations s_x and s_y satisfy $s_x \geq 1$ and $s_y \geq 0$ (i.e., the redirected location is in the gray region), we state that the stick is in its basic orientation. When two sticks in their basic orientations and at the same location redirect the light to different locations, we state that their types are 'different'.

We basically assume that the number of pixels in the desired pattern is the same as the number of arranged sticks. Additionally, we also assume that the pixel resolution, measured as the number of pixels, is square in shape, though, our method can handle more general settings beyond these two assumptions.

On one hand, it is possible to provide the user a large num-

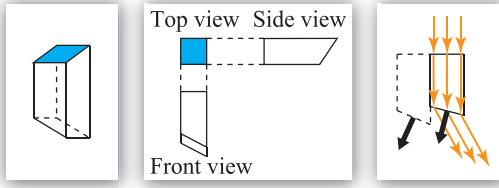


Figure 3: A stick is shown on the left, while a 2D diagram of the stick is shown in the center. We call the blue square face the ‘base’ of the stick. On the right we show a 2D illustration of the light transfer (orange arrows) related to a stick (shown as the solid black lines). The light is assumed to enter the stick perpendicularly from the base and transmit through the stick. Then, the light gets refracted at the tilted face. As the light refracts in the ‘opposite’ direction to the normal (bold black arrow) of the tilted face, it will usually not be blocked by the neighboring sticks (shown as the dashed black lines), or else the amount of blocked light is negligible.

ber of sticks in advance, and then let the user arrange the sticks using only a small subset of them. On the other hand, from an entertainment point of view, it is interesting if there are no superfluous sticks after the user arranges the sticks. Therefore, we basically consider that the number of sticks the user has in advance is equal to the number of arranged sticks, and thus is also equal to the number of pixels in the desired pattern. By keeping the number of sticks smaller, it is easier for the user to find the necessary sticks during the arrangement. The user can, of course, use additional sets of sticks (e.g. colored sticks) for extending power of expression.

2.2. Constraints

For simplicity, we first consider only the intensity of light and ignore the color. We formulate the problem as an optimization problem as follows. We use the binary variables

$$\gamma_{ij,k} = 0 \text{ or } 1, \quad (1)$$

to denote all of the possible ways of placement of the sticks. The integer subscript $i = 1, \dots, N_T$ denotes the type of the stick, and N_T is the number of types. The integer subscript $j = 1, 2, 3, 4$ indicates the basic orientation (i.e., 0°) and the 90° , 180° and 270° rotated orientations, respectively. The integer subscript $k = 1, \dots, N_P$ denotes the location of the sticks, and N_P is the number of pixels. $\gamma_{ij,k} = 1$ means that at the location k , we place a stick whose type is i and its orientation is according to j .

For later use, we also introduce alternative representations for the subscripts i and k . For the type of a stick, we alternatively represent it by the shift of the stick i.e., s_x and s_y

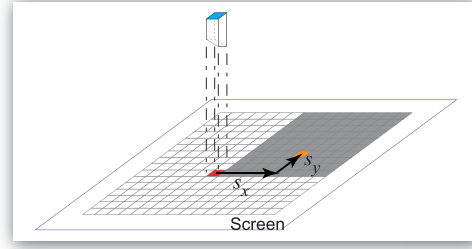


Figure 4: The relation between a stick (shown above) and the screen location of the redirected light (shown as orange). The blue region shows the base of the stick. If the light is not redirected, it will reach the red pixel on the screen. We define the difference (s_x, s_y) between the red pixel and the orange pixel as the shift, or translation, of the stick.

(Figure 4). For the location of a stick, we alternatively represent it using the integer coordinates p_x and p_y (Figure 2).

When we place a single stick at a location, we have the following constraint,

$$\sum_{i=1}^{N_T} \sum_{j=1}^4 \gamma_{ij,k} = 1. \quad (2)$$

For each stick type i , the number of the sticks is limited to M_i . Thus, we have

$$\sum_{j=1}^4 \sum_{k=1}^{N_P} \gamma_{ij,k} \leq M_i. \quad (3)$$

The constraint for making the resulting desired pattern can be formulated as follows. Let us use s_x and s_y as described above instead of i , and p_x and p_y instead of k . Then, if we denote the desired pattern at the screen pixel (q_x, q_y) as $c_{(q_x, q_y)}$, we have

$$\sum_{\substack{s_x, j + p_x = q_x \\ s_y, j + p_y = q_y}} \gamma_{(s_x, s_y)j, (p_x, p_y)} = c_{(q_x, q_y)}, \quad (4)$$

where $s_{x,j}$ and $s_{y,j}$ represent the shift after considering the rotation, i.e.,

$$(s_{x,j}, s_{y,j}) = \begin{cases} (s_x, s_y) & j = 0 \\ (-s_y, s_x) & j = 1 \\ (-s_x, -s_y) & j = 2 \\ (s_y, -s_x) & j = 3 \end{cases}. \quad (5)$$

However we cannot always reproduce the desired pattern exactly. This can be caused by an unachievable desired pattern, e.g., the total intensity of the desired pattern exceeds the total available incident light, or it requires a setting for which the number and types of sticks are limited. We believe that, in practice, the introduction of a small amount of modification to the desired pattern is acceptable. Thus, we represent the amount of modification using variables $\epsilon_{(q_x, q_y)}$, and use



Figure 5: Left: the input pattern (we prepared the pattern according to the Eurographics logo). Center and right: the resulting patterns using the objective functions shown as Eq.(8) and Eq.(9), respectively. Note that by using Eq.(9), the black regions and the smoothness of the pattern is better preserved. The computation time is set to 1 minute.

the following constraint instead,

$$\sum_{\substack{s_x, j + p_x = q_x \\ s_y, j + p_y = q_y}} \gamma_{(s_x, s_y)j, (p_x, p_y)} = c_{(q_x, q_y)} + \epsilon_{(q_x, q_y)}. \quad (6)$$

Because the intensity of the pixels on the screen should not be negative, we have the following constraint,

$$c_{(q_x, q_y)} + \epsilon_{(q_x, q_y)} \geq 0. \quad (7)$$

2.3. Objective Function

The objective of the optimization routine is to make the modification smooth and small enough to retain the desired pattern. To measure the amount of modification, we consider the quadratic norm of the modification $\sum \epsilon_{(q_x, q_y)}^2$ and that of the gradient of the modification $\sum |\nabla \epsilon_{(q_x, q_y)}|^2$. By making $\sum \epsilon_{(q_x, q_y)}^2$ and $\sum |\nabla \epsilon_{(q_x, q_y)}|^2$ small, the modification would be small and smooth.

Although we could use the following simple objective function,

$$\operatorname{argmin}_{\{\gamma_{i,j,k}\}, \{\epsilon_{(q_x, q_y)}\}} (w_a \sum_{q_x, q_y} \epsilon_{(q_x, q_y)}^2 + w_b \sum_{q_x, q_y} |\nabla \epsilon_{(q_x, q_y)}|^2), \quad (8)$$

where w_a and w_b are ‘weights’, we found it more useful to allow the weights for each pixel to change as follows;

$$\operatorname{argmin}_{\{\gamma_{i,j,k}\}, \{\epsilon_{(q_x, q_y)}\}} \sum_{q_x, q_y} (w_{a,(q_x, q_y)} \epsilon_{(q_x, q_y)}^2 + w_{b,(q_x, q_y)} |\nabla \epsilon_{(q_x, q_y)}|^2). \quad (9)$$

For the default values of these weights, we use

$$w_{a,(q_x, q_y)} = 1.0 / (c_{(q_x, q_y)}^2 + \delta), \quad (10)$$

where $\delta = 0.001$ is a constant, and

$$w_{b,(q_x, q_y)} = 1.0 / (|\nabla c_{(q_x, q_y)}|^2 + \delta). \quad (11)$$

Using these default weights, the intensity and/or gradient of the desired pattern will be better preserved if their values are lower. This property is important, as the ‘black’ regions and ‘smooth’ regions will be well preserved (see Figure 5). The user can increase or decrease the weights for stronger or weaker preservation if needed.

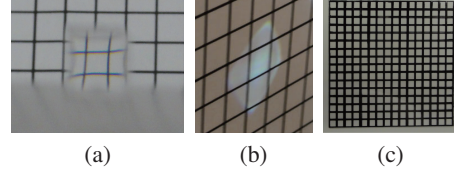


Figure 6: (a) The refraction pattern of the stick. We can see that the face is slightly distorted near the edges. (b) The resulting pattern on the screen. The light leaks to the neighbor pixels. (c) In our prototype, we attach to the container a transparent filter with a black grating pattern to avoid the leak of light.

2.4. Solving the Optimization Problem

In summary, we minimize Eq.(9) subject to the constraints shown as Eqs.(1), (2), (3), (6), and (7). Due to the constraint shown as Eq.(1), our optimization problem becomes a class of mixed integer programming problem, which is NP-hard. We have found that an approximated solution will usually give a good result. In our prototype, we used the IBM ILOG CPLEX Optimizer [CPL]. The user can set the time allowed for solving the problem (e.g., a few minutes to 10 minutes), then the solver utilizes a branch-and-bound algorithm and progressively refines the solution during the allocated time.

2.5. Fabricating the Sticks

We calculate the normals of the tilted faces from the shifts of the sticks according to Snell’s law. The main considerations in fabricating the sticks are; 1) how to manufacture flat surfaces for the base and the tilted face; 2) how to make these surfaces face the desired orientation; and 3) how to make the deviation in the sizes of the bases small. The first and second considerations are important for accurate redirection of light, while the third consideration is important for arranging the sticks as a grid.

In our prototype, we created a single mold that can produce 10 different types of sticks, with base sizes of 5mm \times 5mm. The number 10 comes from the following reason. When we cast the hot acrylate resin into the mold, it will flow following the path created in the mold, and it will reach each of the cavities for the sticks sequentially. Since we do not want the hot acrylate resin to cool too much before it reaches all the required areas, there is a limit to the number of sticks that can be cast in a mold at any one time.

As the accuracy of production in our prototype is not very high, there are slight distortions near the edges of the sticks as shown in Figure 6(a). As a result, the redirected light will leak to the neighbor pixels (Figure 6(b)). To avoid this problem, we attach to the container a filter as shown in Figure 6(c). We believe that we can improve the production process of the mold and the casting process of the acrylate resin to avoid this problem in the future.

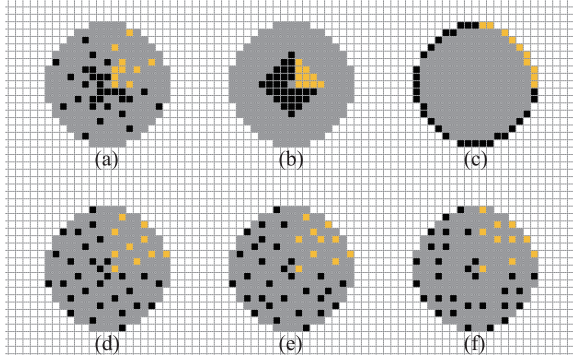


Figure 7: We show the types of the sticks by the locations of the redirected light. A stick is assumed to be located at the center of each gray circle, which shows the range in consideration. The orange pixels show the locations of the redirected light when the sticks are in their basic orientations. The black pixels show the locations of the redirected light when the sticks are rotated by 90° , 180° or 270° . (a) to (f) show 6 different configurations of types.

2.6. Choosing the Types of Sticks

Focusing now on the creation of a mold, we should consider the case where the number of types of sticks is 10. In our prototype, we choose 10 different shifts within the range where the distance of a shift is no larger than 8 times of the width of the stick, as shown in Figure 7.

There are many possibilities in choosing the types of stick. We first chose 10 types of stick with the smallest different shifts (in pixels), as shown as Figure 7(b). We found that this selection is not ideal if the desired patterns contain a wide black region. This is because the redirected light cannot diverge far from the incident light, and thus cannot avoid the black region. On the other hand, by choosing the 10 types of sticks that give the largest shifts (Figure 7(c)) we can create various desired patterns fairly easily. One disadvantage is that when the shifts are larger, the shape of the redirected light becomes much more distorted. Thus, we then tested others configurations of stick-types, this time including both sticks that produced a large shift along with sticks that caused a small shift in different distributions, as shown in Figures 7(a), (d), (e) and (f). The ability of these different bundles to create the desired patterns seemed to be similar. We chose the set shown as Figure 7(a), with the intention of making the distortion small.

2.7. The Layout Map

To identify the type of a stick, we first align the stick in its basic orientation and its tilted face in the near side. Then, we draw a number with different colors on the faces of the stick for better identification of the type and the orientation of the stick. Black, red, green and blue colors are used to

indicate the basic orientation (*i.e.*, 0°) and 90° , 180° and 270° rotated orientations, respectively.

We show the layout of the stick by using a map which we call layout map (see, *e.g.*, Figure 10). In the map, the type of the stick for each location is shown using the number. According to the color of the number, we align the stick so that the number with the color will face upward.

2.8. Using Color Sticks

Our method can be extended to handle colors. We can create a ‘colored’ stick by simply painting the color onto the tilted face or by pasting a color filter onto the tilted face using instant glue (cemedine). Note that by using the cemedine, the transparency and the refraction index is then retained pretty well, since the dominant component of the cemedine is cyanoacrylate. In our prototype, instead of actually creating colored sticks, we attach to the container a color filter created by printing the color pattern onto a transparency sheet using a standard ink-jet printer.

To work with colored sticks, we can extend Eq.(6) as follows.

$$\begin{cases} \sum_{\substack{s_x, j + p_x = q_x \\ s_y, j + p_y = q_y}} T_{(p_x, p_y)}^{(r)} \gamma_{(s_x, s_y)j, (p_x, p_y)} = c_{(q_x, q_y)}^{(r)} + \epsilon_{(q_x, q_y)}^{(r)} \\ \sum_{\substack{s_x, j + p_x = q_x \\ s_y, j + p_y = q_y}} T_{(p_x, p_y)}^{(g)} \gamma_{(s_x, s_y)j, (p_x, p_y)} = c_{(q_x, q_y)}^{(g)} + \epsilon_{(q_x, q_y)}^{(g)} \\ \sum_{\substack{s_x, j + p_x = q_x \\ s_y, j + p_y = q_y}} T_{(p_x, p_y)}^{(b)} \gamma_{(s_x, s_y)j, (p_x, p_y)} = c_{(q_x, q_y)}^{(b)} + \epsilon_{(q_x, q_y)}^{(b)} \end{cases}, \quad (12)$$

where the superscripts (r) , (g) and (b) indicate the red, green and blue components, respectively, and T indicates the transmittance of the filter (the color and the intensity of the light source can be included as well, if needed). Although we can mix different colors to produce new colors, multiple sticks are needed for representing a single mixed color pixel, possibly resulting in a reduced power of expression. As an alternative, we can use a few set of sticks with basic colors. To obtain the layout, we first solve the optimization problem for the intensities, then the colors of the sticks are chosen to be the corresponding pixel colors.

3. Results

In this section, all of the results are generated by rearranging a single set of sticks. The types are introduced in Section 2.6. The number of sticks of each type is 25 or 26, and the total number of the sticks is 256. The resolutions of the input patterns are all in 16×16 . For computation, we used a notebook PC with an Intel(R) Core(TM)2 Duo U9600 CPU (1.6GHz) and 4GB main memory, and set 10 minutes for solving the optimization problem. The default weights are used for all the tests. We let users arrange the sticks, and the times they took for arranging them varied between 12 and 23 minutes.

We show the results of simulation and the results created

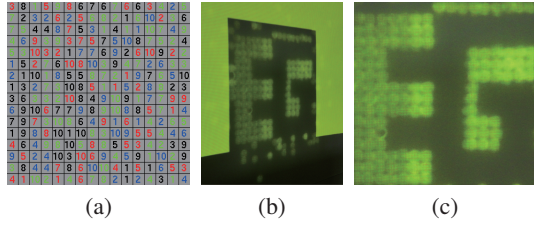


Figure 8: The input pattern is the same as the one used for Figure 10(c). Using the layout map as shown in (a), we can redirect the light toward the outside of the resulting pattern as shown in (b). Then, the intensities of the pixels in the 16×16 region (c) match that of the input pattern perfectly. If we do not allow the light redirection toward the outside of the resulting pattern, the result will be Figure 10(c).

by the sticks fabricated by using a mold. We show the results of monochromatic patterns in Figures 10(a) to (c), and colored patterns in Figures 10(d) to (g). In Figure 10(c), we projected yellow color from the projector. In the colored examples shown in Figures 10(d) to (g), we attached color filters to the container instead of actually coloring the sticks. In Figures 10(a) to (d), the total intensities of the input patterns are smaller than the total intensity of the incident light. As a result, some pixels in the simulated results become brighter. Conversely, in Figure 10(g), the total intensity of the input pattern is larger than that of the incident light. As a result, some pixels in the simulated results become darker. Nevertheless, we believe that these results still look good. In Figures 10(e) and (f), the total intensities of the input patterns are the same as the total intensity of the incident light, and we were able to obtain the results that perfectly match the input patterns.

We can replace some of the transparent sticks with opaque sticks. Additionally, we can fix the locations of these opaque sticks so that we can observe a pattern on the arranged sticks, as shown in Figure 9. By calculating the layout for the rest of the sticks, we can also make this set produce a (different) hidden pattern on the screen. This usage of our art-form can be regarded as a kind of steganography. To realize this art-form, we simply set $\gamma_{ij,k} = 0$ for the locations k where we want to place the opaque sticks, and delete the corresponding constraints shown as Eq.(2).

As mentioned before, we may consider the option to redirect the light toward the outside of the resulting pattern. By considering this option, the modification can be reduced if the total intensity of the desired pattern is less than that of the incident light (Figure 8). On the other hand, it is more constrained and interesting if we discard this option, since in this case all of the incident light is used. Thus, we did not use this option other than the example shown in Figure 8. We can easily disable this option by omitting some of the vari-

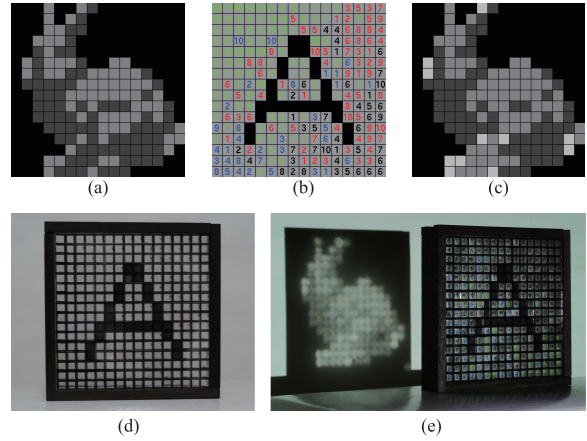


Figure 9: (a) The input pattern. (b) The layout map. (c) The simulated result. (d) Opaque (black) sticks are introduced so that we can observe the pattern ‘A’ directly from the arranged set of sticks, while (e) transparent sticks produce a different pattern on the screen.

ables $\gamma_{ij,k}$ that redirect light toward the outside of the desired pattern.

4. Limitations

We show three typical failure cases in Figure 11. Figure 11(a) shows a toy example where the input pattern contains a large black region. Due to the capability in redirecting light, it is not possible to redirect all the light to the lower-right corner using the set of the sticks. To handle input patterns like this example, we need to use the sticks that can redirect light to farther locations. We would like to note that an input pattern usually does not contain such a large black region.

For general input patterns, a failure case happens more likely when the difference between the total intensity of the input pattern and that of the incident light is larger. This is because we need to apply much modification to the pattern to adjust the total intensity. As a result, some specific region cannot be represented well. For example, in Figure 11(b), the total intensity of the input pattern greatly exceeds that of the incident light. Thus, we need to suppress the intensity variation to reduce the total intensity. As a result, we cannot represent the gradation pattern well. We show another failure case in Figure 11(c). This example is created by increasing the intensity of the hair of the character shown in Figure 10(e). Thus, the intensities of some of the pixels in the resulting pattern need to be reduced for the conservation of light. Although our method tries to make the modification as smooth as possible, since the regions of the hair are separated by the hair accessory, the resulting intensities become

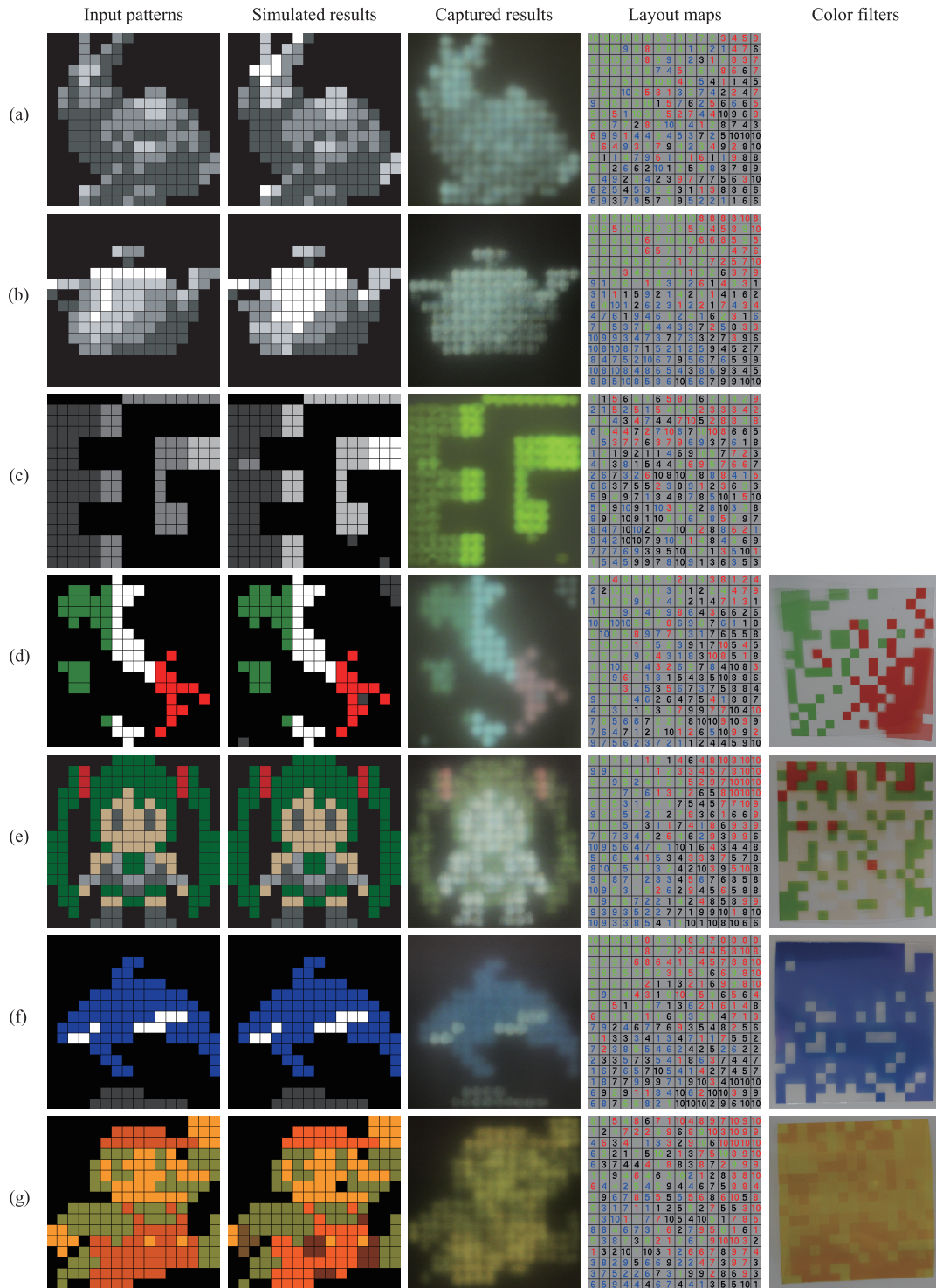


Figure 10: Results of monochromatic patterns ((a) to (c)) and colored patterns ((d) to (g)). (a), (b), (c), (d) and (f): we prepared the input images according to the Stanford bunny, the Utah teapot, the Eurographics logo, the landscape and the national flag of Italy, and an image of a killer whale, respectively. Please see Acknowledgments for the details about the input image (e). The input image (g) is copyrighted by Nintendo Co., Ltd. The captured results show the images captured from the screen.

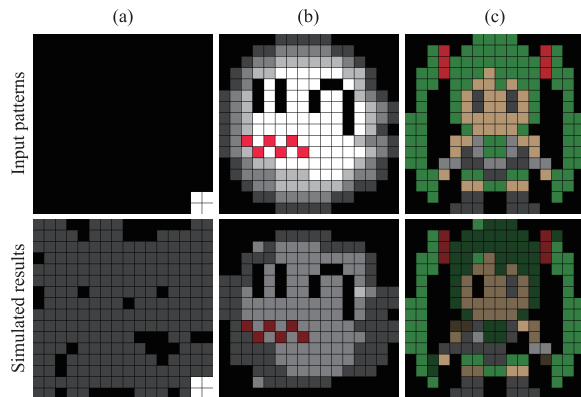


Figure 11: Typical failure cases. The input image (b) is copyrighted by Nintendo Co., Ltd. Please see Acknowledgments for the details about the input image (c).

different among these regions. The user could modify these patterns to obtain better results, though.

5. Conclusions and Future Work

We have presented a new type of creative and enjoyable pixel art. By rearranging a set of sticks made from acrylate resin, and by positioning this set appropriately between a parallel light source (e.g. sun-light or a projector) and a viewing-surface (screen), one can observe the resulting pixel art on the screen. According to the user specified pattern, our system solves an optimization problem to find the layout of the sticks so that the resulting pattern on the screen represents the specified pattern. We have shown that the user can produce various patterns by rearranging the sticks, using only a single set of sticks. Another interesting aspect of our art is that the resulting pattern is only visible when the sticks are appropriately inserted between the light source and the screen. Thus, our art can also be used as a tool for steganography.

For future work, we would like to create sticks with other base shapes, such as triangles or hexagons. Furthermore, we would like to increase the pixel resolution by creating a jigsaw-puzzle like version, where some of the sticks are glued together to form basic blocks. Instead of using the individual sticks, it would then be easier for the user to rearrange these blocks.

Acknowledgments

We gratefully acknowledge helpful comments and thoughtful suggestions from the anonymous reviewers. We also thank Prof. Chandrajit Bajaj for his useful discussions and proofreading. This work was supported by JSPS Grant-in-Aid for Research Activity Start-up (23800011). The input images shown in Figures 1 (d), 10 (e) and 11 (c) are

derivative work of the character "Hatsune Miku", originally copyrighted by Crypton Future Media, INC. A modification of the character is permitted under Piapro Character License (<http://piapro.jp/license/pci>) by Crypton Future Media, INC.

References

- [AM10] ALEXA M., MATUSIK W.: Reliefs as images. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2010)* 29, 4 (2010), 60:1–60:7. [1](#)
- [CPL] IBM ILOG CPLEX Optimizer. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>. [4](#)
- [DWP*10] DONG Y., WANG J., PELLACINI F., TONG X., GUO B.: Fabricating spatially-varying subsurface scattering. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2010)* 29, 4 (2010), 62:1–62:10. [2](#)
- [FDL10] FINCKH M., DAMMERTZ H., LENSCH H. P. A.: Geometry construction from caustic images. In *Proceedings of the 11th European Conference on Computer Vision: Part V* (2010), pp. 464–477. [2](#)
- [FRSL08] FUCHS M., RASKAR R., SEIDEL H.-P., LENSCH H. P. A.: Towards passive 6D reflectance field displays. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2008)* 27, 3 (August 2008), 58:1–58:8. [1](#)
- [HFM*10] HAŠAN M., FUCHS M., MATUSIK W., PFISTER H., RUSINKIEWICZ S.: Physical reproduction of materials with specified subsurface scattering. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2010)* 29, 4 (2010), 61:1–61:10. [2](#)
- [HSHF10] HOSKINSON R., STOEBER B., HEIDRICH W., FELS S.: Light reallocation for high contrast projection using an analog micromirror array. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2010)* 29, 6 (2010), 165:1–165:10. [2](#)
- [Jea10] JEAN P.: Pixels. *SIGGRAPH Asia 2010 Electronic Theater* (2010). [1](#)
- [KL11] KOPF J., LISCHINSKI D.: Depixelizing pixel art. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2011)* 30, 4 (2011), 99:1–99:8. [1](#)
- [Liu09] LIU B.-Y.: DEADLINE. *SIGGRAPH Asia 2009 Electronic Theater* (2009). [1](#)
- [MP09] MITRA N. J., PAULY M.: Shadow art. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2009)* 28, 5 (2009), 156:1–156:7. [1](#)
- [PJJ*11] PAPAS M., JAROSZ W., JAKOB W., RUSINKIEWICZ S., MATUSIK W., WEYRICH T.: Goal-based caustics. *Computer Graphics Forum (Proc. of Eurographics 2011)* 30, 2 (2011), 503–511. [2](#)
- [QBL] Q-BLOCK: Creating 3D pixel art online. <http://kyucon.com/qblock/>. [1](#)
- [RRMG10] REGG C., RUSINKIEWICZ S., MATUSIK W., GROSS M.: Computational highlight holography. *ACM Transaction on Graphics (Proc. of SIGGRAPH Asia 2010)* 29, 6 (2010), 170:1–170:12. [2](#)
- [WPMR09] WEYRICH T., PEERS P., MATUSIK W., RUSINKIEWICZ S.: Fabricating microgeometry for custom surface reflectance. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2009)* 28, 3 (2009), 32:1–32:6. [2](#)