

# Fabricate 2.5D Shadow Art Sculpture

Yunjoo Park

Dept. of Computer Science and Engineering, Ewha Womans University  
Seoul, Republic of Korea  
yunjoopark12@gmail.com

Jyh-Ming Lien

Dept. of Computer Science, George Mason University  
4400 University Dr., Fairfax, VA 22124, USA  
jmlien@cs.gmu.edu

## Abstract

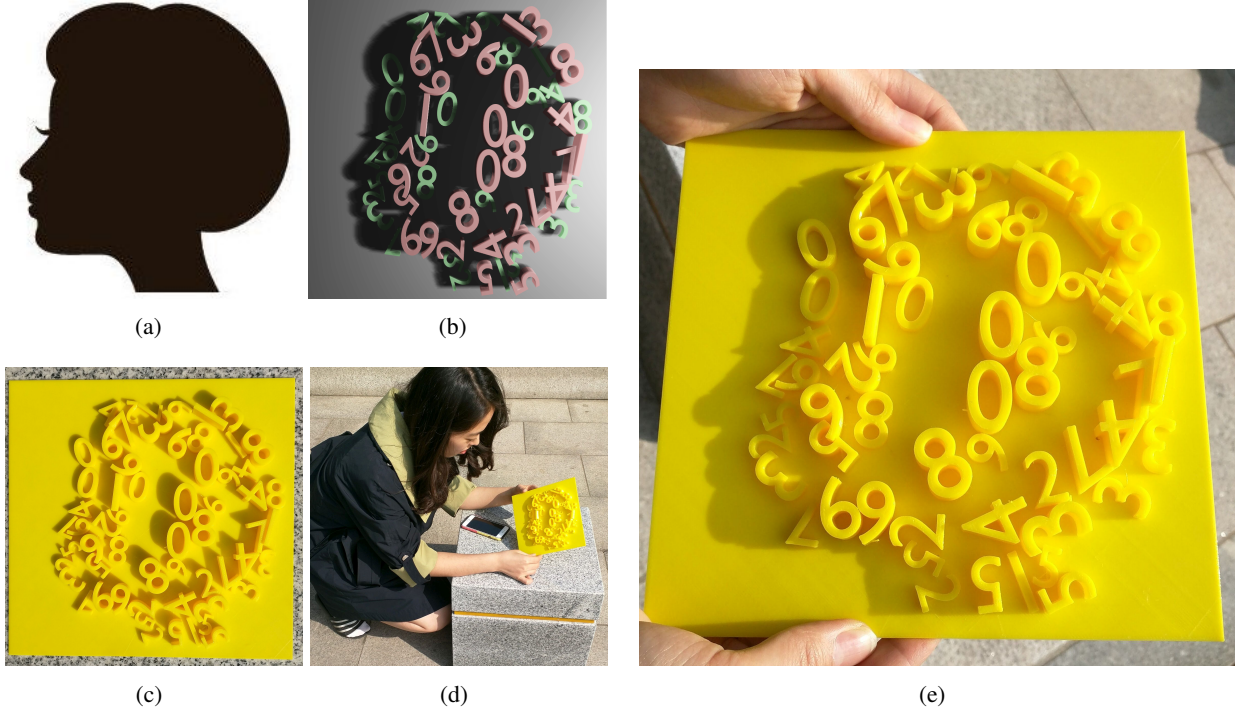
Shadow art is a practice of arranging objects in space to produce surprising shadow. Creating such an art piece has always been labor intensive and time consuming, through trial-and-error methods. Indeed, the time complexity of solving the *shadows art problem* is inevitably NP hard. In this work, we present a computational method that determines a 2D arrangement of 3D objects to produce a desired shadow image. We demonstrate that the proposed tool successfully reproduce the desired shadow through both simulation and physical fabrication.

## 1 Introduction

The interplay between light and shadow has been a study of art in composing intriguing scenes and images for centuries. In this work, we look into the problem of placing objects in 3D space so they collectively cast a desired shadow. Such an installation consisting of carefully arranged objects and shadows is usually known as *shadow art*. Figures 1 and 2 show two of such examples. The relationship between the light blocking elements and their composite shadow is usually counterintuitive due the complicated spatial relationships of projection and occlusion and is sometimes even considered as a type of illusion. Because of this creating such an art piece has always been labor intensive and time consuming through trial-and-error methods.

In this work, we seek an efficient computational method that automatically determines 3D arrangement of light blockers to cast a desired composite shadow; a problem that, from now on, will be referred to as the *shadow art problem*. The shadow art problem is closely related to several optimal placement problems, such as geometry packing and covering problems [6, 2, 16, 4]. Therefore, not surprisingly, finding the optimal solution of the shadow art problem has time complexity in NP-hard. Traditional solutions using metaheuristics, e.g. simulated annealing [5], may be used to attack the shadow art problem by packing the given objects in the visual hull of the target shadow, but they often produce unpleasing shadows with jaggy boundaries.

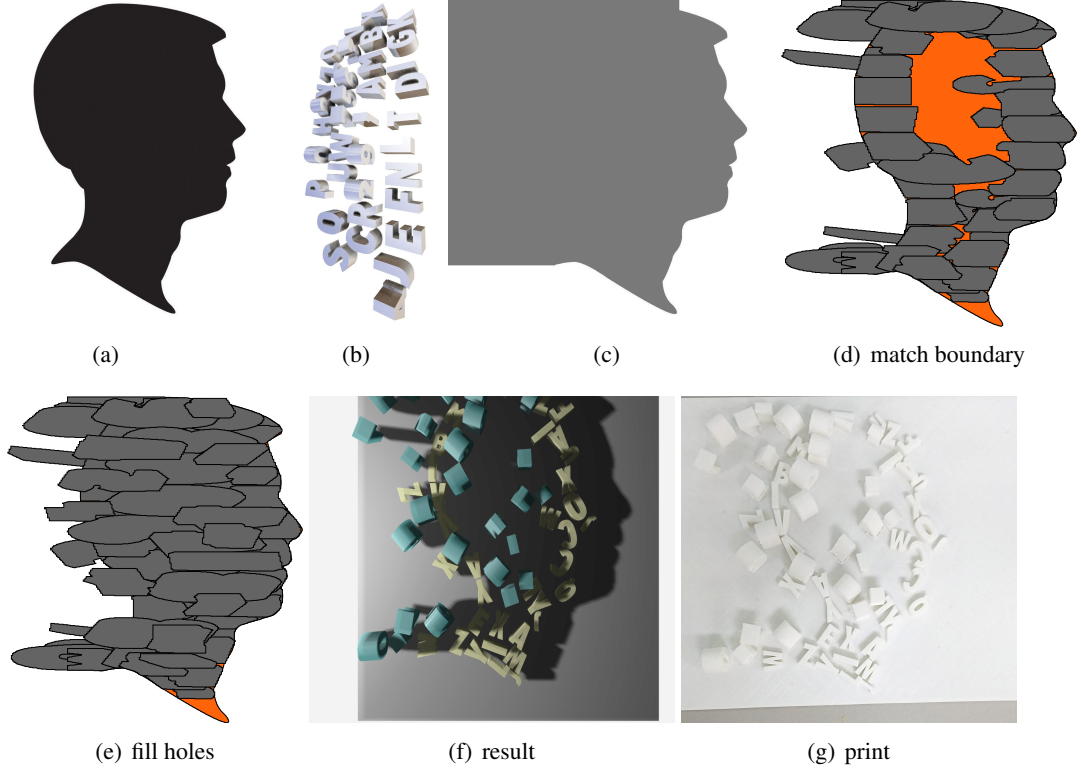
Motivated by the ad-hock nature in practice and its computational challenge, we aim to develop a tool that can assist artists creating shadow art more effectively. For example, we will show that our method can help artists create shadow arts with objects arranged on a 2D plane. We believe that the proposed method can produce complicated shadow art that is extremely time consuming to create manually without such a computational tool. A video showing the shadow arts created for this paper can be viewed at: [https://youtu.be/yVx\\_seXzCvQ](https://youtu.be/yVx_seXzCvQ)



**Figure 1 :** A shadow art (b) of a given profile image (a) created by the proposed method (60 light blockers used). This example is inspired by Kumi Yamashita’s work [7]. A shadow art sculpture shown in (c) is fabricated via additive manufacturing. The scale of the fabrication is indicated by a regular cell phone next it (d). The desired shadow (e) is reproduced by placing the printed sculpture under the sun.

## 2 Related Work

Shadow can be found in almost all aspects of computer graphics, ranging from rendering, to modeling and to animation. However, there are only a few methods that are focused on manipulating shadows. Recent work in modeling considered various *inverse problems* that aim to create 3D models that can produce desired shadow or shade. For example, Lee et al. [11] considered the problem of using one set of 2D billboards and two lights to create two target shadow images, which is known as billboard shadow art. This problem is similar to [13], which focused on creating a 3D model that can cast three shadows on three different orthogonal directions. Mitra and Pauly [13] studied the problem of constructing 3D sculpture that can cast three distinct shadows. Given three desired shadow images, the sculpture is constructed by taking the intersection of the shadow hulls of the images. An iterative optimization is applied to ensure the consistency between the cast shadows and the shadow images. The key idea is to deform slightly the shadow images in each iteration of the proposed optimization so that the consistency (between the cast shadows and the deformed shadow images) is maximized. A brief discussion of the shadow art problem is provided in their paper that considers the idea of iterative closest points and 3D collage [8] by filling the space inside the constructed sculpture mesh with shapes in a small database of four models. This indirect approach has several problems: (1) the models are usually tightly packed and can have inter-model collision, (2) this method provides no control over the arrangement over the parts (this is the most critical defect in their approach), (3) in their example-based shadow art, the models are much smaller in scale compare to the final sculpture, (4) discontinuity at the shadow boundary is quite visible even after subdivision of the mesh is applied, and finally (5) useful for composing the sculpture using lego block, 3D printers but not from existing 3D shapes as their space



**Figure 2 :** *Inputs of the proposed method: (a) Target shadow. (b) Light blockers including English alphabets (horizontal) and numerical digits (vertical). (c) The grey area indicates allowable locations of the shadows cast by the light blockers. Method overview (d)–(g). The shadow created by the printed model can be found in Fig. 6. The orange areas in (d) and (e) indicate uncovered regions of the target shadow.*

filling techniques allow scaling. Since their approach is indirect, their approach either requires a large shape database or operations (such as scaling) that are only easily achievable in virtual world. Finally, shadowpix [3] discussed a local and global algorithm to display several prescribed images (at most four) formed by self shadowing of the surface when lit from certain directions. Researchers also looked into the problem of color shadow. For example, Baran et al. [1] presented a method for creating objects that cast color shadow images when illuminated by prescribed lighting configurations. One of their main concerns is the color consistency.

### 3 Problem Statement and Preliminaries

Given a set of 3D meshes  $\{M_i\}$  as the light-blocking elements, a 2D stencil image  $S_t$  as the desired shadow, and a light source, the shadow art problem is to find an arrangement  $A$  of  $\{M_i\}$  such that their shadows reproduce the stencil  $S_t$  on a given flat surface, where  $A$  is a list of rigid transforms  $\{R_i\}$  for each mesh  $M_i$ . Given an arrangement  $A$ , we call  $R_i(M_i)$  a *footprint* of mesh  $M_i$  that occupies a 3D volume in space, blocks the light and casts a shadow  $S_{R_i(M_i)}$ . Let  $S_M$  be the union of all stencil images  $\{S_{R_i(M_i)}\}$  cast by  $\{R_i(M_i)\}$ . Then the shadow art problem can be formally defined as:

$$\arg \min_A (d(S_t, \bigcup S_{R_i(M_i)})) , \quad (1)$$

subject to the constraint that  $\bigcap R_i(M_i) = \emptyset$ . We consider the difference  $d(S_t, S_M)$  between  $S_t$  and  $S_M$  in terms of both curve and root-mean-square deviation (RMSD) as discussed below.

**Shape similarity and partial shape matching** of stencils are two important components in the proposed algorithm. By breaking down the contours of a stencil into **curve segments** with various arc lengths, we obtain a database of curve segments. This database enables the shape matching at different levels of detail. A curve segment is parameterized as  $(\alpha, \delta)$ , where  $\alpha$  is the arc length of the curve and  $\delta$  is the offset from a specific vertex  $v_0$  on the contour. Given a contour  $\partial S$  of a stencil  $S$ , curve segments are defined as  $\{(i\alpha_0, j\delta_0)\}$ , where  $i$  and  $j$  are positive integers, and  $\alpha_0$  and  $\delta_0$  are the unit arc length and unit offset, respectively. Moreover,  $i\alpha_0$  is upper bounded by the arc length of  $\partial S$ .

The similarity of two curve segments  $c_i$  and  $c_j$  that may have different arc lengths is measured via curvature signature and RMSD of the curve segments. For each curve segment, we re-sample the segment via Gaussian convolution so that all curve segments have the same number of vertices. These vertices are evenly distributed along the segment and the curvature at each vertex is estimated. Given  $c_i$  and  $c_j$ , their difference  $d_{curvature}(c_i, c_j)$  is the  $L_2$  distance between their curvatures along the segment. Given  $c_i$  and  $c_j$ , a rigid transform can also be determined by minimizing their RMSD using the principal component analysis for the scale component and Kabsch algorithm [10] for rotational component.

**Overview.** The proposed shadow art tool is designed to first arrange shadows that outline the target boundary and then to cover the remaining interior holes. See an overview in Fig. 2. Not only does the boundary usually provide more important visual impact, but matching the projection to the target boundary is also more challenging as a small difference is noticeable and unforgiving. To fill the holes interior to the target shadow, there exists higher flexibility to arrange the projected shadows as long as the shadows are contained inside the target boundary and the projected shadows overlap with the holes.

## 4 Our Method

We will now describe our method in detail. The proposed method first generates a set of representative shadows for each given light blockers (Section 4.1). Then these representative shadows are transformed and matched to different sections of the boundaries of the stencil image. A subset of these shadows are selected to maximize the matching scores and simultaneously ensure that no multiple shadows are selected from the same object (Section 4.2). In the next step (Section 4.3), the remaining objects are used to cover the holes (i.e, the regions that are not occupied by the shadows in the previous step). The hole covering step is accomplished by a similar optimization used to match image boundary.

### 4.1 Create Representative Shadow Images

Representative shadows of a given mesh are projected images that can distinctively identify the mesh. Before we can find these representative shadows, a comprehensive set of shadow images are created. We denote this set of shadow as  $S_M$ . From  $S_M$ , we then compute representative images  $S'_M$  that summarize  $S_M$  such that, in a given distance metric, each image in  $S_M$  has a corresponding image in  $S'_M$  that is at most  $\epsilon$  distance away. Note that, even though only representative shadows are used to match the target boundary, the final step of the proposed method will further optimize the matching by searching around the shadows in the neighborhood of the representative shadows to provide better matching.

**Select orientations that create distinctive shadows.** Given a three dimensional mesh, we first create a set of orientations that can create many distinct projected shadows. Instead of selecting random orientations, we pick orientations so that the normal directions of the faces, vertices and edges are perpendicular to the direction of light. This decision is based on the fact that similar orientations usually produce very similar projection except near a critical orientation, which occurs only when the normal directions of some faces, vertices and edges are perpendicular to the direction of light. For a complex model, the number of such normal directions is large and require clustering. In this case, we use  $k$ -mean clustering to reduce the number



of orientations with  $k = 100$ . Then shadows are created using the orientation obtained from the center of these clusters.

**Cluster shadows via shape similarity analysis.** Shadows are further clustered using shape similarity metric. This step allows us to cluster shadows produced by different orientations but with similar outlines due to symmetries and occlusions.

The similarity measure of two stencils  $S_i$  and  $S_j$  is carried out on a bipartite graph formed between their curve segments: the nodes are curve segments and edge weights are  $d_{curvature}(c_i, c_j)$  of the curve segments  $c_i \in S_i$  and  $c_j \in S_j$ . Solving the minimum bipartite graph matching provides a set of good candidates for measuring the similarity of  $S_i$  and  $S_j$ . Next, for each match  $(c_i, c_j)$  in the bipartite graph matching, we determine a rigid transform  $T$  that minimizes the RMSD between  $c_i$  and  $c_j$ . Finally, the minimum overall RMSD between  $S_i$  and  $S_j$  is then reported as their similarity measure.

We then construct a distance matrix to cluster and extract a small number of representative contours for representing the shadows of a given model using spectral clustering method [14]. For each cluster, a representative shadow is defined as a shadow with the smallest total distance to other shadows in its cluster.

## 4.2 Optimal Shadow to Target Boundary Matching

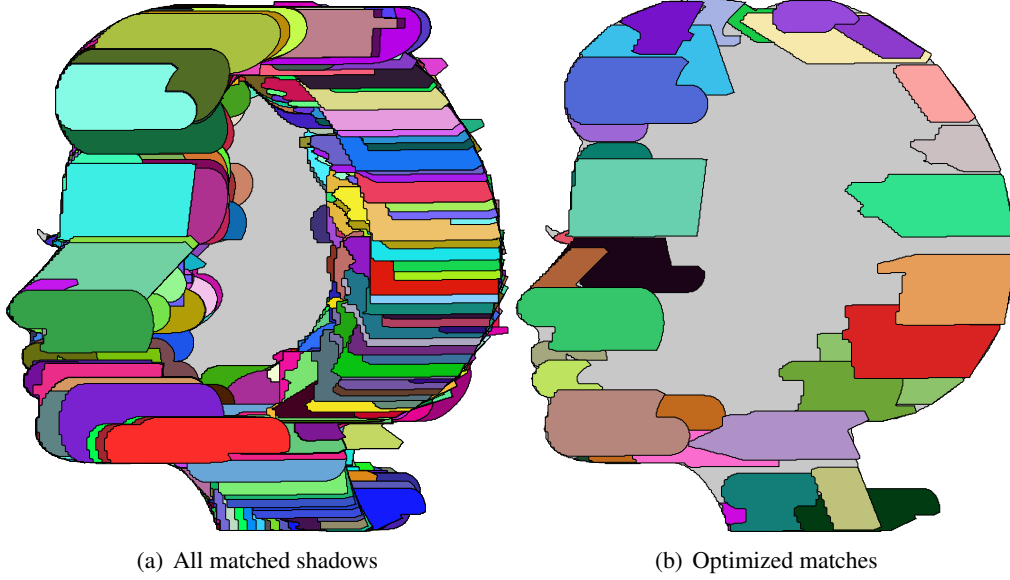
Correspondences between the representative shadows and the boundary of the target shadows are discovered in two steps. In the first step, a curve segment database is built for the target shadow  $T$  as described in Section 3. For a given curve segment of the target (target segment) in the database, we find a best matching transform for every representative shadow stencil. In order to ensure that the match is valid, the transformed shadow stencil also needed be sufficiently contained within the target shadow.

In the second step, each curve segment of the target starts with a list of matching shadows accompanied with their matching error  $\epsilon$ . The optimal matches are selected by solving a 0-1 linear programming problem. Each 0-1 variable  $x_i$  indicates if a given match should be picked toward the final matching. The objective is designed so that the solution will minimize the matching error, and the constraints ensure that each model can only cast a single shadow. We further constrain that every unit curve segment of the target contour is covered at least by one representative shadow boundary.

More specifically, given  $n$  matching, there will be  $n$  the binary variables  $x_1, x_2, x_3, \dots, x_n$ . The optimization is subject to at least two types of constraints. The first type of constraints prevents multiple shadows cast by the same model:  $\sum x_i^M \leq 1$  for all shadows  $x_i$  cast from a model  $M$ . If the model has  $n_m > 1$  instances, then we simply rewrite the constraint as  $\sum x_i^M \leq n_m$ . The second type of constraints ensures that at least one match should be selected to cover a unit-length curve segment. Given a target segment that has unit length and its associated matches  $x_i^t$ , we need to ensure that  $\sum x_i^t > 0$ . Note that it is possible that the second constraints are too tight and no feasible can be found. To address this, we add a binary auxiliary variable  $a_t$  for each constraint:  $a_t$  with value 1 indicates that no associated matches  $x_i^t$  are selected. Therefore, auxiliary variables are added only when we failed to find a solution in the first attempt.

## 4.3 Filling Remaining Holes

To create the complete shadow, the interior of the target shadow should be filled. Filling the holes in some way is easier than matching the boundaries as it has less significant visual impact to the final shadow. However, unlike the classic object packing and cover problems, each object has multiple potential shadows that make the shadow art problem more challenging (due to larger search space). Instead of taking the traditional approaches [8, 13], which greedily pick a candidate shadow that covers the largest hole area, we propose a new iterative optimization method that, in each iteration, identifies multiple shadows that cover as much hole



**Figure 3 :** *Before and after optimization.*

space as possible. Details are discussed next.

For each remaining candidate shadow  $S_i$ , we perform  $K$  covering simulations: for each simulation  $k$ , we randomly pick a point inside the candidate shadow and a point inside holes polygons, and the vector between these two points is the translation vector  $t_k$ . Then, we randomly sample a rotation angle  $\alpha_k$ . Note that the value of  $t_k$  and  $\alpha_k$  can be biased using either the principal axis or Kabsch algorithm [10] that minimizes the RMSD between  $S_i$  and the hole to increase their overlapping area.  $t_k$  and  $\alpha_k$  will form the transformation matrix  $m_k$  for this shadow  $S_i$  in the  $k$ -th simulation. Let  $S_{i,k}$  be the overlapping region between the shadow  $S_i$  and the holes under this transformation  $m_k$ . Our optimization problem is to select a subset of shadows so that the values of  $\sum_i \text{area}(S_{i,k})$  is maximized while ensuring that each model casts only one shadow.

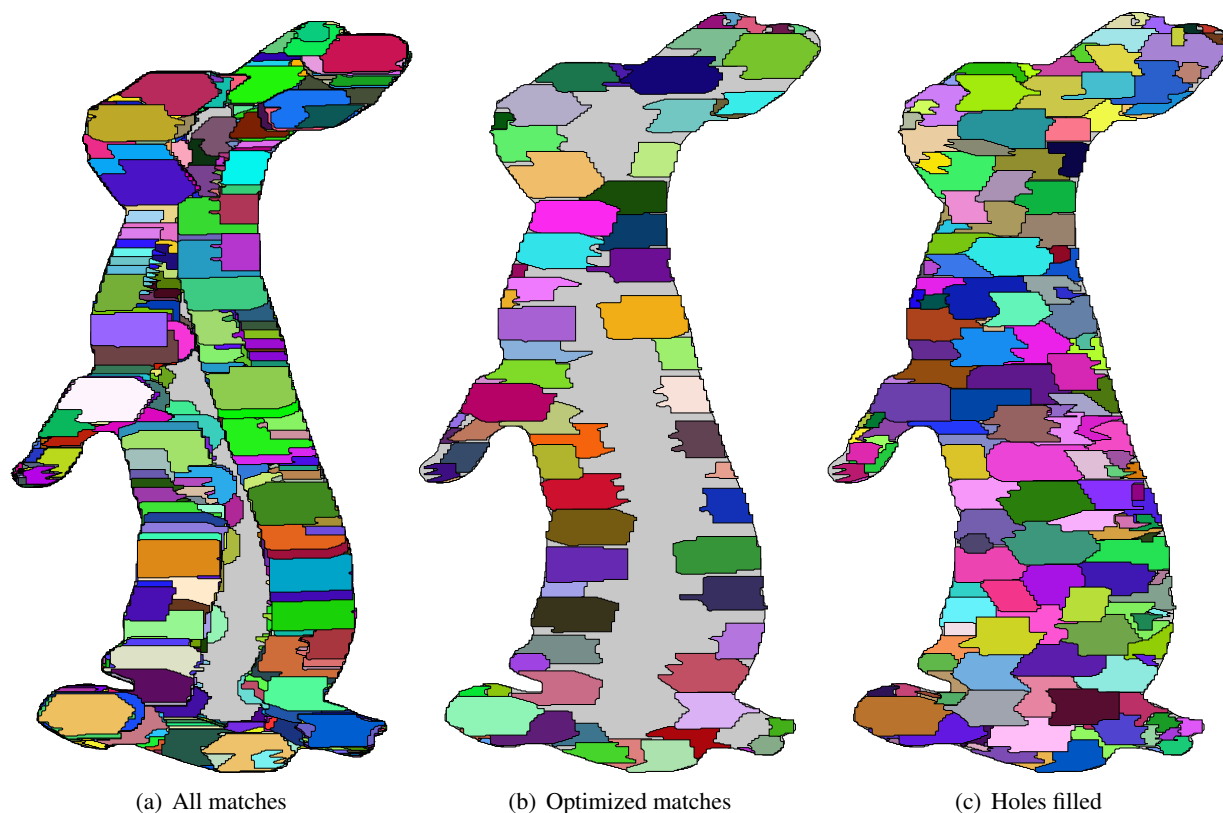
Note that we use  $\sum_i \text{area}(S_{i,k})$  to approximate  $\text{area}(\bigcup_i S_{i,k})$ , therefore, if the overlap between shadows is significant, the difference  $D = \sum_i \text{area}(S_{i,k}) - \text{area}(\bigcup_i S_{i,k})$  will be unacceptable. To control  $D$ , we impose additional constraints that ensure the shadows whose overlapping area is greater than a threshold cannot be selected simultaneously. Given two shadows  $S_i, S_j$  and their hole coverage regions  $S_{i,k}$  and  $S_{j,l}$ , if their overlapping ratios  $\text{area}(S_{i,k} \cap S_{j,l})/\text{area}(S_{i,k})$  and  $\text{area}(S_{i,k} \cap S_{j,l})/\text{area}(S_{j,l})$  is greater than a threshold  $S_\tau$  then the shadows  $S_i, S_j$  at their corresponding transforms  $m_k$  and  $m_l$  are considered to be conflicting shadows.

We repeat the above optimization procedure until all the models are used or the remaining holes become insignificant.

## 5 Additional Results

The proposed method is implemented in C++, and lights and shadows are simulated using OpenGL shader language. The shadow art sculptures are fabricated using a consumer-level 3D printer and take between 6 to 12 hours to print.

In addition to the lady and man profiles shown in Fig. 1 and Fig. 2, we also produced another example shown in Fig. 5. Fig. 4 shows the boundary fitting and hole matching between the shadows and target in creating the rabbit shadow art.



**Figure 4:** *Steps in creating the rabbit shadow produced by the proposed method.*

To study the shadow under different light sources, in Fig. 6, we demonstrated the results of our shadow arts using projector light. The output perfectly matches the expected outcome. In Fig. 7, we compare the shadows of a city skyline produced by simulated light, projector light, and cell phone light. Because cell phone light is a spot light, its shadow deviates from the target shadow more than the other two.

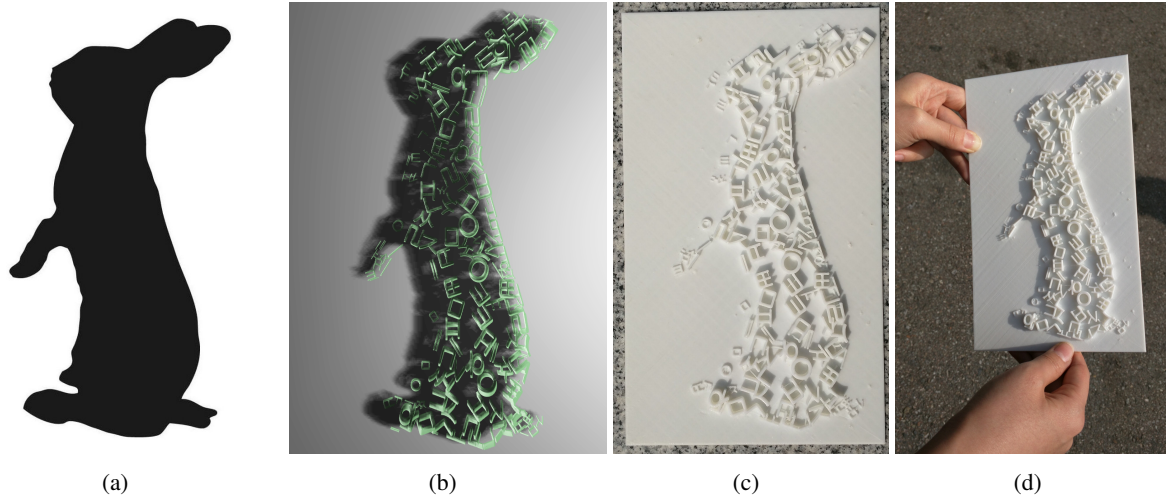
Making the physical copies of the digital shadow art also allows us to enjoy the art in playful ways. For example, we show that, in Fig. 8, different faces can be produced by tilting the sculpture in different angles.

## 6 Conclusion

In this paper, we presented a computational method that determines 2D arrangement of objects from a desired shadow image. The proposed method involves three main steps: light blocker shadow analysis, target boundary matching, and hole filling. We showed that the proposed method produce high quality shadows efficiently. We also physically made the shadow art sculpture using consumer-level 3D printer and reproduce the shadow using sun light, projector light and cell phone light.

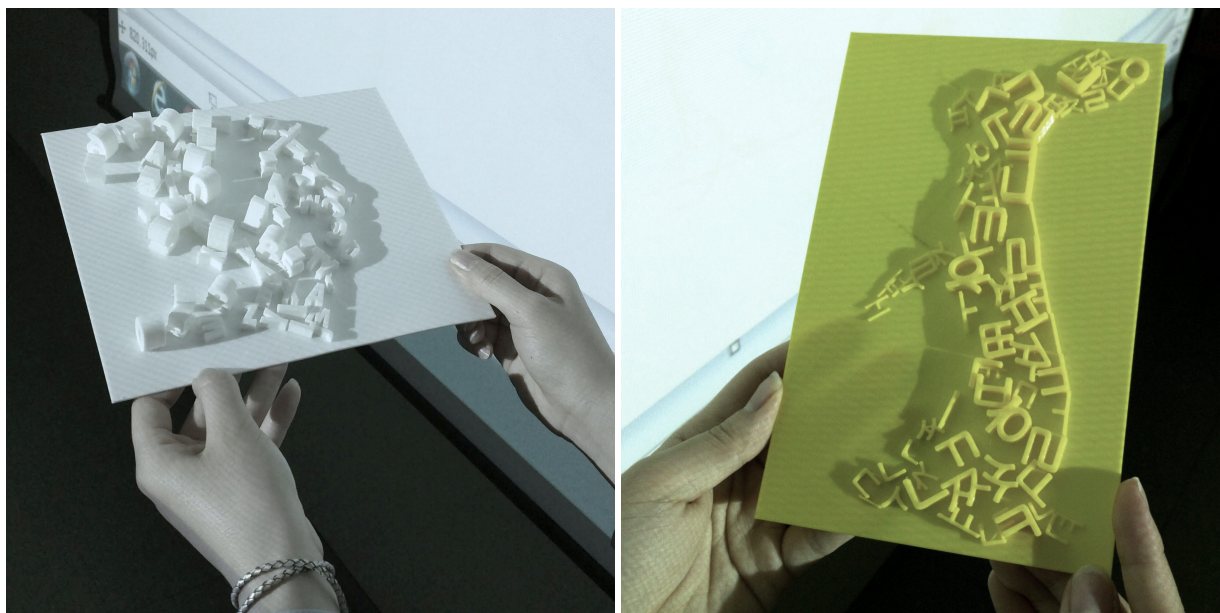
## References

- [1] Ilya Baran, Philipp Keller, Derek Bradley, Stelian Coros, Wojciech Jarosz, Derek Nowrouzezahrai, and Markus Gross. Manufacturing layered attenuators for multiple prescribed shadow images. In *Computer Graphics Forum*, volume 31, pages 603–610. Wiley Online Library, 2012.



**Figure 5:** A shadow art (b) of a rabbit silhouette (a) created by the proposed method. The fabricated shadow art sculpture (c,d) accurately cast the shadow of the rabbit under the sun. The light blockers are made of 46 Hangul alphabets in various scales.

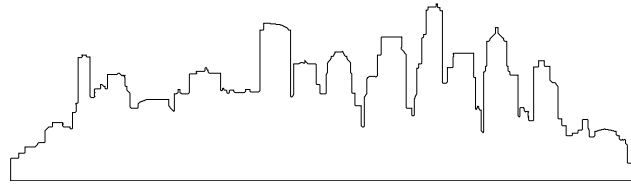
- [2] J Bennell, Guntram Scheithauer, Yuriy Stoyan, and Tatiana Romanova. Tools of mathematical modeling of arbitrary object packing problems. *Annals of Operations Research*, 179(1):343–368, 2010.
- [3] Amit Bermano, Ilya Baran, Marc Alexa, and Wojciech Matusk. Shadowpix: multiple images from self shadowing. In *Computer Graphics Forum*, volume 31, pages 593–602. Wiley Online Library, 2012.
- [4] Xuelin Chen, Hao Zhang, Jinjie Lin, Ruizhen Hu, Lin Lu, Qixing Huang, Bedrich Benes, Daniel Cohen-Or, and Baoquan Chen. Dapper: decompose-and-pack for 3d printing. *ACM Transactions on Graphics (TOG)*, 34(6):213, 2015.
- [5] N Chernov, Yu Stoyan, and Tatiana Romanova. Mathematical model and efficient algorithms for object packing problem. *Computational Geometry*, 43(5):535–553, 2010.
- [6] John Horton Conway and Neil James Alexander Sloane. *Sphere Packings, Lattices and Groups*. Springer Verlag, 1988.
- [7] Designboom. Interview with artist Kumi Yamashita. <http://www.designboom.com/art/kumi-yamashita-interview-03-05-2015/>, March 2015.
- [8] Ran Gal, Olga Sorkine, Tiberiu Popa, Alla Sheffer, and Daniel Cohen-Or. 3d collage: expressive non-realistic modeling. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 7–14. ACM, 2007.
- [9] Shu-Wei Hsu and Tsai-Yen Li. Planning character motions for shadow play animations. *Proc. CASA*, 5:184–190, 2005.
- [10] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.
- [11] Yunjin Lee, Jae-Yun Lee, and Junho Kim. Billbaord shadow art using two lights. In *Computer Graphics International*, 2012.
- [12] Oliver Mattausch, Takeo Igarashi, and Michael Wimmer. Freeform shadow boundary editing. In *Computer Graphics Forum*, volume 32, pages 175–184. Wiley Online Library, 2013.
- [13] Niloy J Mitra and Mark Pauly. Shadow art. In *ACM Transactions on Graphics*, volume 28, pages 156–1, 2009.



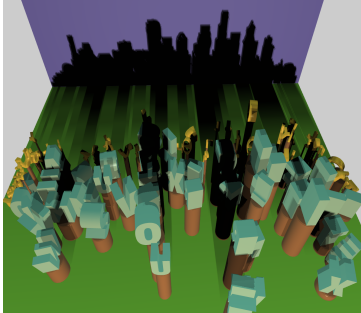
**Figure 6:** *Shadows of the man (Fig. 2) and the rabbit (similar to Fig. 5 but with larger Hangul light blocker) created using projector light.*

- [14] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.
- [15] Fabio Pellacini, Parag Tole, and Donald P Greenberg. A user interface for interactive cinematic shadow design. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 563–566. ACM, 2002.
- [16] J Vanek, JA Galicia, B Benes, R Měch, N Carr, O Stava, and GS Miller. Packmerger: A 3d print volume optimizer. In *Computer Graphics Forum*, volume 33, pages 322–332. Wiley Online Library, 2014.

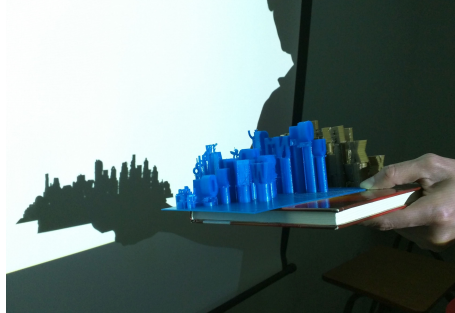




(a) target shadow



(b) simulated light

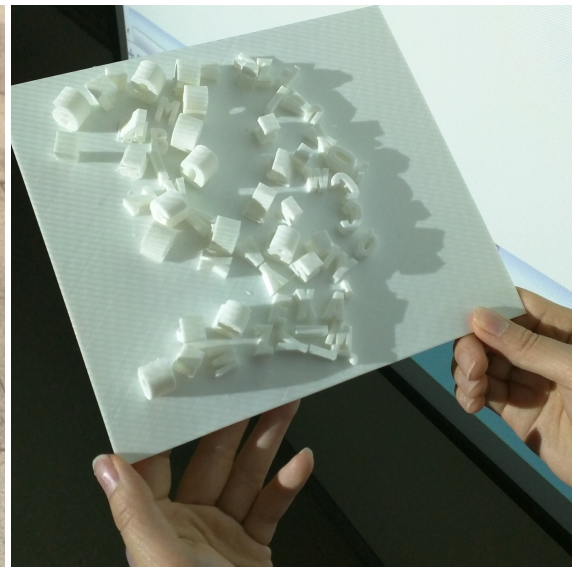
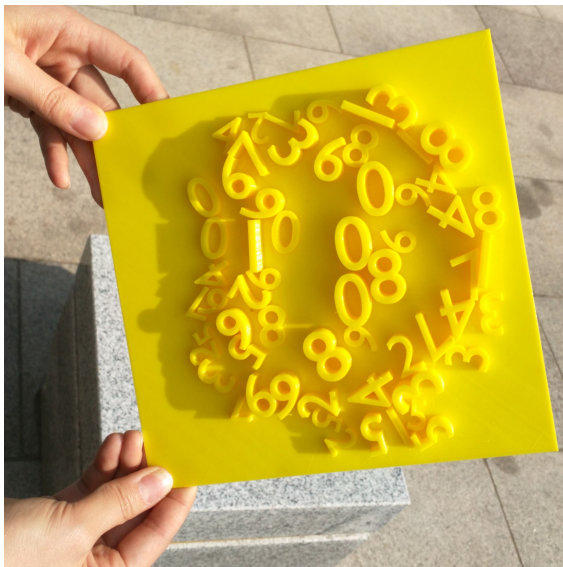


(c) projector light



(d) cell phone light

**Figure 7:** *Shadows of a city skyline created using various light sources. The 3D sculpture is created in two prints (blue and brown) due to the limited print volume.*



**Figure 8:** *Having fun with the shadow art prints. Different lady and man profile shadows created by tiling the shadow art sculptures at a different angle.*