

TrayGen: Arranging Objects for Exhibition and Packaging

Yong-Liang Yang¹ and Qi-Xing Huang²

¹KAUST, KSA

²Stanford University, USA

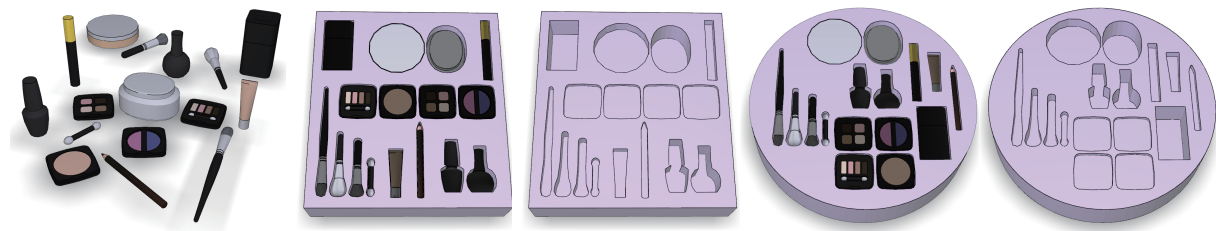


Figure 1: Starting from a collection of unorganized objects, our framework generates tray designs for exhibition and packaging.

Abstract

We present a framework, called TrayGen, to generate tray designs for the exhibition and packaging of a collection of objects. Based on principles from shape perception and visual merchandising, we abstract a number of design guidelines on how to organize the objects on the tray for the exhibition of their individual features and mutual relationships. Our framework realizes these guidelines by analyzing geometric shapes of the objects and optimizing their arrangement. We demonstrate that the resultant tray designs not only save space, but also highlight the characteristic of each object and the inter-relations between objects.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms

1. Introduction

How to effectively pack objects is one of the fundamental problems in product design and operational research. A good packing strategy can dramatically decrease the loading and delivery cost. Many computational algorithms have been proposed to generate good product layout, where the main goal is to save space.

In this paper, we focus on the design of a packaging tray for a collection of solid objects (see Fig. 2). Trays are widely used in daily life to protect and store products. The objects to be packed together are usually from the same category and/or for the same purpose. They can be tools with similar functionality, assemblies that form a real product, etc.

Packaging trays have a special layered structure, which requires additional design considerations other than space saving. On one hand, unlike a standard packaging problem

where the goal is to facilitate transportation, a packaging tray also exhibits objects to the consumer. As a result, it is important to organize and place objects so that their functionalities are revealed. On the other hand, how to design the tray such that a user can easily assemble the objects back on the tray, is also an important factor.

While it is possible to generate a packaging tray by hand, the design is often intuitively achieved without exploring the potential layout space, requiring several rounds of refinement. Thus, it is extremely desirable to have an automatic system that facilitates the design process.

We present a computational framework which can automatically generate tray design solutions with respect to flexible user constraints (see Fig. 1). The core of our framework is a set of design guidelines from shape perception and visual merchandising, which provides cues on how objects should

be placed. For example, objects with similar functionality should be grouped together and placed closer. Moreover, objects in the same group should be ordered in a way that their differences can be easily recognized. This helps the customer to gain a quick impression of the functionalities of the products. Similar to [FCODS08, MYY*10], our approach is based on the observation that the functionality of an object can be inferred from its geometry and its surrounding shapes. We show that these high level guidelines can be effectively implemented using shape analysis and optimization techniques.

We demonstrate our framework on a range of data sets from different categories. The experimental results and user studies show that our framework can deliver good packaging suggestions.

In summary, our work makes two major contributions. First, we identify a set of guidelines to effectively arrange a collection of objects for exhibition and packaging. Second, we develop a computational framework that applies these guidelines to generate plausible tray packaging solutions.

2. Related Work

Arranging objects. Object arrangement is well studied in the area of operational research. Heuristic-based algorithms usually attempt to solve the problem where the main goal is space saving [IC01, LMM02]. Although other constraints such as spatial relations can be added by probabilistic optimization of a multi-modal objective function [CSY02], how the objects are exhibited in the arrangement is not considered. Recently, techniques for arrangement optimization have been applied to different scenarios in graphics. Merrell et al. [MSL*11] and Yu et al. [YYT*11] optimize furniture layout in a room. Yeh et al. [YYW*12] synthesize layouts with varying number of objects. In these approaches, the inter-relations of objects are pre-defined. In contrast, Fisher et al. [FRS*12] learn spatial relations between objects from a set of input scenes and synthesize context-preserving scenes from a database. Our work differs from the above as follows. First, we organize objects purely based on their geometric



Figure 2: Trays are widely used in our daily life.

attributes and no label (e.g., ‘table’, ‘chair’) is required. Second, we focus more on how the layout exhibits object features instead of the functionality of the layout itself.

Shape/Product perception. A comprehensive review on shape perception and the well-known Gestalt laws of grouping is beyond the scope of this paper. Interested readers may refer to [WEK*12] for a survey on this topic and to the references in [NSX*11] for the applications in computer graphics. In the area of marketing, visual merchandising aims at creating a positive image of a business to prospective customers [BSZP91]. Product packaging as a special kind of visual merchandising is also referred to as ‘the salient salesman’ [Jon99], meaning the package layout itself should try to reflect the feature of the product. We show that the rules in shape perception and visual merchandising (e.g., similarity, proximity, regularity) can also be applied to tray packaging.

Viewpoint selection. Viewpoint selection seeks to find the best viewpoint to observe the features of a 3D shape. There exist a variety of objectives for this task such as optimizing viewpoint entropy [VFSH01] and surface saliency [LVJ05]. Please refer to a recent paper [SLF*11] for a survey of this problem. The major difference is that instead of processing a single object, we consider multiple objects together.

3. Overview

In this paper, we consider the setting, where distinct objects are usually nested in separated cells in a tray structure (see Fig. 2). In particular, the shape of the corresponding cell is roughly determined by the contour of the object when placing on the tray. The tray design problem is formulated as computing the 2D projected contour of each object such that (i) each contour reflects the semantic information of the corresponding object, and (ii) the arrangement of the 2D contours reflects their mutual relations.

Specifically, the input to our framework comprises a set of unorganized 3D objects $\mathcal{S} = \{S_1, \dots, S_N\}$ represented by triangle meshes (see Fig. 3). Our system explores the geometric features of each object, inter-object relations based on perception-driven design guidelines, to determine a 2D projected contour for each object and an arrangement over all objects, which will be used to generate the output tray design. The whole pipeline has two major steps:

Contour selection. This step determines a projected contour for each object and reduces the arrangement problem from 3D to 2D. During the selection, we generate groups for geometrically similar objects, optimize both individual group saliency, which favors the preservation of geometric features, and the inter-group saliency which prioritizes the dissimilarity between different groups.

Contour arrangement. The second step initializes the contour arrangement by considering the orientation of the projected contours as well as the regular structure of the objects in the same group. Then we optimize the contour arrangement in 2D to make the layout compact and clear, while the inter-group relations can also be highlighted.

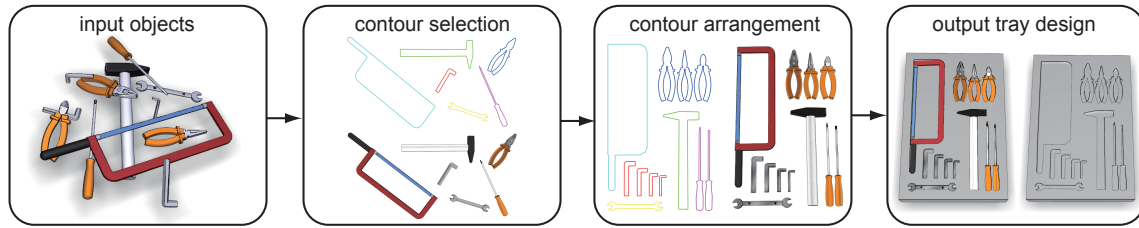


Figure 3: Overview of our framework.

The following two sections elaborate the guidelines and algorithms of the two major steps respectively.

4. Contour Selection

Tray design can be considered as a special *packaging* problem. However, the primary goal of solving a standard packaging problem is to save space. While for tray design, the exhibitions of shapes and inter-shape relations become the dominant factors. By studying the principles on shape perception [WEK*12] and visual merchandising [BSZP91], we summarize several perception-based guidelines for tray design. We classify these guidelines into two categories: *contour* and *arrangement* related design guidelines. The contour guidelines specify rules for choosing a 2D contour for each object, such that the selected contours convey the semantics of the objects. The arrangement guidelines specify how to arrange these 2D contours such that the inter-relations of the objects are highlighted.

4.1. Contour Guidelines

Similarity: The relations among geometrically similar objects should be preserved in the contours. This is particularly important for many tool sets, where several objects have the same functionality but different sizes (e.g., pliers, screwdrivers). (Section 4.2.1)

Saliency: Each object should be well displayed from the exposed part when embedded in the tray. In addition, each contour shall preserve the unique features of the corresponding object, i.e., those features that rarely appear on other objects (Section 4.2.2). This allows a user to quickly identify an object from the corresponding contour and easily put it back.

4.2. Contour Selection Algorithm

4.2.1. Shape Clustering and Alignment.

We begin with performing shape clustering to divide the input shape collection into groups of similar objects $\mathcal{G} = \{G_1, \dots, G_M\}$. We propose to use D2 descriptor [OFCD02] to represent global geometric feature of each object due to its simplicity and robustness. Shape clusters are generated by a standard spectral k -means clustering [NJW01]. The number of clusters k is automatically determined by identifying the largest drop point in the k -means cost curve [KT03].

Note that more sophisticated shape descriptors and clustering methods can be used to further improve the result. The user may also manually re-group objects when the shape descriptor failed to capture high level semantic features. Figure 4 shows a typical classification result of our system.

If a group contains more than one object, we align these objects so that their contours are selected in a consistent manner (see Fig. 5). Specifically, we first compute an approximate minimal bounding box (MBB) of each object using Principal Component Analysis followed by rotating calipers [Tou83]. Then we select the object with the longest MBB diagonal as reference. Other objects in the same group are aligned to it by aligning their MBBs with respect to three principal axes. We scale each object such that the length of its MBB diagonal is the same as that of the reference. Since the direction of a principal axis is ambiguous, we try different combinations and the alignment error is measured by root-mean-square distance from the object to the reference.

4.2.2. Distinctive Contour Selection

The next step is to select a good view direction \vec{d}_i for each group G_i . The overall goal is that when observing objects along the selected directions, the saliency of each group and the dissimilarities between pairs of groups can be highlighted as much as possible. The objects in the same group will later be placed on the tray along the selected direction, while the corresponding contours will be used for object arrangement in Section 5.

The key idea is to formulate view direction selection as

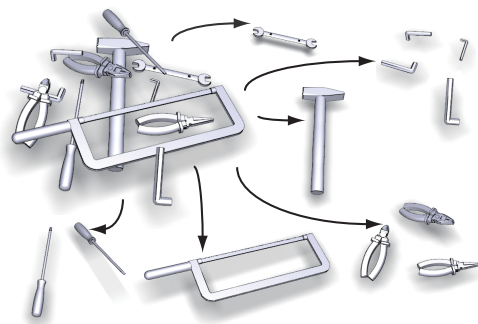


Figure 4: Shape classification leads to groups of geometrically similar objects.

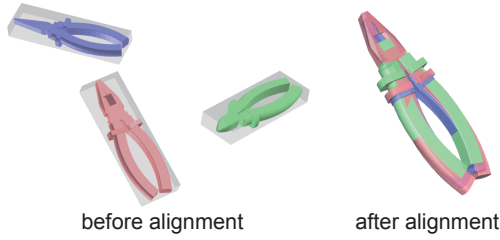


Figure 5: We perform shape alignment for the objects in the same group based on their minimal bounding boxes (MBBs).

a quadratic integer program, whose variables correspond to the candidate directions of all the groups. The first and second order term characterize the saliency of each group and the dissimilarities between pairs of groups, respectively.

Candidate view directions. In our system, we assume that each group of objects is placed on the tray with a ‘canonical’ pose (see Fig. 2). More precisely, we generate 6 candidate view directions, say \vec{d}_{ij} , for each group G_i . Each direction is orthogonal to one MBB face and points to the center. One may also consider additional view directions if necessary.

View-dependent saliency: For each view direction \vec{d}_{ij} , we compute an extreme outer contour curve C_{ij}^k and a visible (partial) mesh M_{ij}^k for each group object S_k ($S_k \in G_i$). The observed saliency score s_{ij}^k of S_k is combined by a contour saliency score and a surface saliency score (in equal weights). We normalize two scores independently by the maximal score among all view directions. The group saliency score s_{ij} is the sum of the saliency scores of all group objects.

In our implementation, we compute the contour saliency score using a center-surrounded mechanism [IKN98]. More specifically, we estimate curvature at each contour point and apply two Gaussian filters with small and large kernel sizes. The contour point saliency is defined as the difference between two filtered curvatures. The contour saliency is the integration of the saliency values over all contour points. The surface saliency score is defined using the *visibility ratio* criterion [PPB*05], which is a product of the visible surface area and the projection area along the view direction.

View-dependent dissimilarity: To emphasize the specialty of each group when placing on the tray, we compute a dissimilarity score $s_{ij,i'j'}$ for two different candidate view directions \vec{d}_{ij} , $\vec{d}_{i'j'}$ of two different groups G_i , $G_{i'}$. We use shape contexts [BMP02] as a contour-based object descriptor. For comparing groups with multiple objects, the dissimilarity score is computed as the sum of the per-pair scores.

Optimization: We associate each candidate view direction \vec{d}_{ij} with a binary indicator $x_{ij} \in \{0, 1\}$, where $x_{ij} = 1$ means the j -th candidate is selected for group G_i and $x_{ij} = 0$ otherwise. Based on the view-dependent group saliency score and inter-group dissimilarity score, we formulate the following

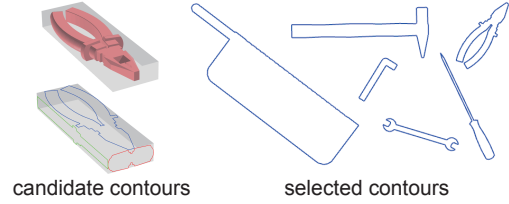


Figure 6: The candidate contours are based on MBBs (left). The selected group contours for the ‘tools’ data set (right).

quadratic problem to jointly select the view direction (projected contour) of each group:

$$\begin{aligned} \max \quad & \lambda \sum_{ij:i'j'} x_{ij}x_{i'j'}s_{ij,i'j'} + \sum_{ij} x_{ij}s_{ij}, \\ \text{s.t.} \quad & \sum_j x_{ij} = 1, \\ & x_{ij} \in \{0, 1\}. \end{aligned} \quad (1)$$

As the quadratic term in the above equation is dense, we use the iterative coordinate ascent method [LH06] to compute the optimal solution. We set $\lambda = 0.1$ for all our experiments. Fig. 6 shows the selected contours (at a group level) for the ‘tools’ data set in Fig. 3.

5. Contour Arrangement

Based on the contours selected in the previous section, we now describe the guidelines and algorithms on how to arrange them (in the xy -plane without loss of generality), leading to a packaging layout of all objects.

5.1. Arrangement Guidelines

Equilibrium: Each object should be placed on the tray with a ‘natural’ orientation, e.g., it is the same orientation as people usually place/hold the object. (Section 5.2.1)

Proportion: Objects in the same group should be placed regularly and ordered by dimension (Section 5.2.2).

Compactness: Many objects with exactly the same shape should not be displayed as multiple instances. It is better to be stacked (Section 5.2.2). In addition, objects should have a compact arrangement to save space (Section 5.3).

Clearance: Objects should not overlap. Moreover, for non-cluttered exhibition, they should admit some distance in between. (Section 5.3)

Communication: For objects with strong spatial interrelations, e.g., components of a rifle (see Figure 9), the arrangement should reflect their inter-relations (Section 5.3).

5.2. Arrangement Initialization

Now we describe how to initialize the arrangement by considering the orientation of the selected contours and the regularity of the in-group layout.

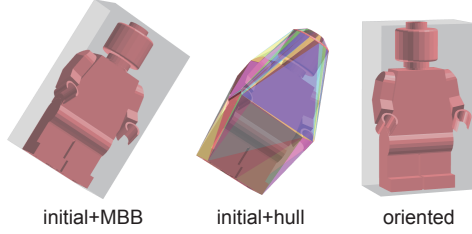


Figure 7: For the upright orientation, we use MBB to guide the selection of supporting base from convex hull.

5.2.1. Contour Orientation

After contour selection, we still have freedom to rotate the selected contour in the xy -plane (the corresponding object is rotated accordingly, with the rotation axis passing through its centroid and parallel to z -axis). To determine the ‘natural’ orientation, we consider four canonical poses for each object determined by its MBB. For each pose, one MBB’s face (orthogonal to y -axis) serves as the base of the object. We apply the convex hull based approach in [FCODS08] to measure a stability score for each pose. More precisely, for each candidate face, we compute the coincident supporting area from the convex hull of the object. The MBB’s face with the largest supporting area will be selected as the supporting base of the object (see Fig. 7). Moreover, for the group with multiple objects, we orient the reference object only and align others accordingly.

For the object without a strongly preferable upright orientation (can be inferred from the supporting area), the user may specify one or allow it to rotate. To keep the pose canonical, we restrict the rotation to be multiples of 90 degrees.

5.2.2. In-group arrangement

For objects in the same group, we prefer a ‘regular’ in-group arrangement. Such regularity allows people to (i) quickly perceive the common characteristics of the objects, and (ii) easily identify the difference among objects.

Since the group objects are aligned already, a good way to highlight the proportion is to sort out their varying sizes (see Fig. 8). For objects with similar sizes, we generate a translational symmetry pattern (1-parameter or 2-parameter) as shown in Fig. 8. When several objects have exactly the same shape, arranging them one after another results in duplicated visual information and requires too much space. Thus we only highlight one instance and stack the others by using minimal principal axis as stacking direction [LAZ*12], the view directions are changed accordingly (see Fig. 8).

5.3. Objectives for Arrangement Optimization

In this section, we describe the objective terms that measure the quality of an arrangement L regarding to the arrangement guidelines of all the groups.



Figure 8: Objects in the same group are arranged regularly.

Compactness. The first objective term f_{cp} measures the compactness of an arrangement:

$$f_{cp} = \frac{A_{bb}}{\sum_i A_i}, \quad (2)$$

where A_{bb} is the area of the axis-aligned bounding box that encloses all contours (we aim at rectangular trays for now, variations can be found in Section 6), and A_i is the area of the 2D contour of object S_i .

Clearance. We define the clearance term f_{cr} as the sum of the overlapping areas between pairs of contours from different groups:

$$f_{cr} = \sum_{S_i, S_j} \frac{A_i \cap A_j}{A_i + A_j}, \quad (3)$$

where S_i and S_j sum over all pairs of objects that do not belong to the same group. We use PQP [GLM96, LGLM99] - a robust and efficient collision detection library to detect and measure contour overlaps. In practice, avoiding overlap turns out to be insufficient. The objects should have certain distance in between. To address this issue, we perform dilation for all the object contours using a circular kernel. The circle radius r controls the clearance scale.

So far the inter-group spatial relation is not addressed. For some data sets, it is very helpful for understanding the connection between groups and the functionality of the whole product. Thus we consider the following two terms.

Relative position. The relative position term f_{rp} considers groups that can be assembled into a single layered functional product (see Fig. 9). In this case, it is much better to arrange them in a way that the overall assembly can be inferred from the layout. Suppose such product is provided with a natural pose and each component (group) is in the right position. We first manually generate an exploded layout [LACS08] as the initial layout. During the optimization, we add an extra term f_{rp} as below to keep the relative positions and we mainly allow groups to translate.

$$f_{rp} = \sum_{i=1}^M \sum_{j=i+1}^M (\vec{r}_{ij} - \vec{r}_{ij}^0)^2, \quad (4)$$

where $\vec{r}_{ij}^0 = \vec{o}_i - \vec{o}_j$ is the initial relative position between the contour barycenters of G_i and G_j , and \vec{r}_{ij} stands for the



Figure 9: The relative position term can help to infer the overall assembly.



Figure 10: The closeness term can keep pairs of groups close to highlight their inter-relation (e.g., the brush group and powder box group).

current layout. Fig. 9 shows the effect of incorporating the relative position term.

Closeness. The closeness term f_{cl} considers groups of objects that tend to be used together (see Fig. 10):

$$f_{cl} = \sum_{(G_i, G_j) \in \mathcal{I}} (\vec{o}_i - \vec{o}_j)^2, \quad (5)$$

where \mathcal{I} contains all the close group pairs. Since it is challenging to infer functional dependency from geometry alone, we expect the user to label group pairs that should stay close.

5.4. Optimization

To search for a good contour arrangement L , we define the following objective function that combines the four objective terms described above:

$$f(L) = f_{cp} + w_{cr}f_{cr} + w_{rp}f_{rp} + w_{cl}f_{cl}, \quad (6)$$

where w_{cr} , w_{rp} and w_{cl} specify the tradeoffs between different terms. For all the examples used in this paper, we set $w_{cr} = 0.02$, $w_{rp} = 0.005$ and $w_{cl} = 0.005$ (w_{rp} and w_{cl} are not included unless otherwise noticed). Fig. 11 shows the effect of different terms.

For optimization, we extend the stochastic pattern search method [YC00] to explore the layout space and generate multiple optimized arrangements. We are aware of other optimization strategies such as integer linear programming [Dyc90] and branch-and-bound [KMP10]. However, the complexity of our problem (i.e., optimizing positions and

Figure 11: The effect of individual terms in the optimization.

orientations of multiple objects with arbitrary shapes) makes it hard to apply these techniques.

Specifically, starting from an initial arrangement L_0 , we first evaluate the objective function $f(L_0)$. We then apply an iterative process to improve the arrangement. At each step, we generate a new layout L_1 by performing one of the following three exploratory moves: (i) translating a random selected group by perturbing all group objects using a Gaussian term; (ii) rotating a random selected group according to its admissible rotations ($0^\circ, 90^\circ, 180^\circ, 270^\circ$); (iii) swapping two randomly selected groups. L_1 is accepted if it decreases the objective function, otherwise we go back to L_0 and try another move. The search continues until the stopping condition is met, which can either be the maximal iteration number (10000) is reached, or the function value does not decrease for a certain number of iterations (100).

To explore the multi-modal layout space, starting from random initial arrangements (by moving all groups to the center and randomly perturbing each group), we run several optimizations in parallel (32 in our experiments). The resultant arrangements are sorted according to objective function values. We always select the best arrangement in the paper, the user can also select lower ranked ones if preferable.

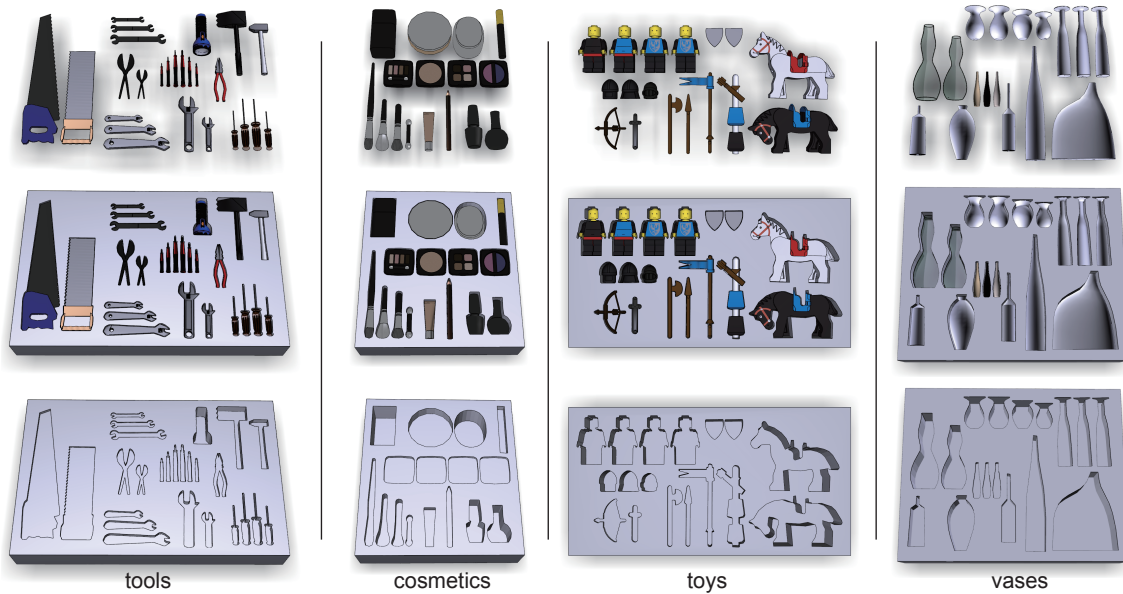


Figure 12: Our framework can generate tray designs for data sets from different categories.

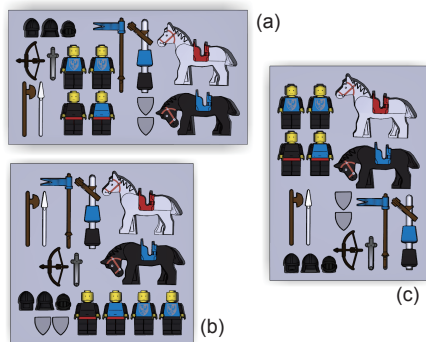


Figure 13: Multiple design suggestions by our framework with objective function value: (a) 1.713 (b) 1.733 (c) 1.740.

6. Results

We have implemented a tray design framework called TrayGen, which combines the presented contour selection and arrangement algorithms. The final tray model is generated by performing a (boolean) difference of a tray box and the contour-based bounding volumes of all transformed objects. The size of the tray box can be adjusted by adding offsets to the bounding box of all the objects (see Fig. 3). Fig. 12 shows the results on a range of data sets from different categories (tools, cosmetics, toy set, personal collection).

By default our system generates rectangular trays with flexible sizes (i.e., no restriction on the width and height). Fig. 13 shows multiple design suggestions generated by running TrayGen multiple times and selecting the best ones. Note that different in-group layouts can be applied as well.

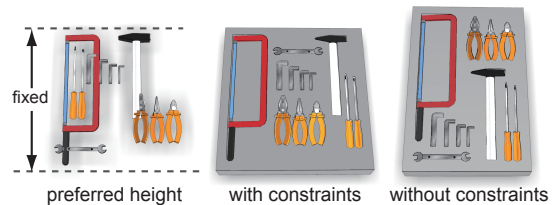


Figure 14: Effects of user specified constraints.

The user may also have some preference on the tray size. For example, the height of the tray is predefined. In such case, we allow the user to add additional hard constraints to the system (see Fig. 14). Starting from an initial layout satisfying the constraints, we only accept the contour arrangement in the feasible layout space during the optimization.

It is also possible to generate non-rectangular trays based on our framework. For the circular case, this can be done by slightly changing the compactness term in Equation 2. Instead of using the axis-aligned bounding box, we compute the minimal bounding circle. The resultant tray design can be found in Fig. 1.

Performance. Our experimental platform is with an Intel Xeon X5550 CPU (2.66 GHz, 4 cores, 8 threads). The timing of our framework is shown in Table 1. In particular, in contour selection, about 80% of the time is for computing D2 shape descriptor.

Evaluation. To evaluate the performance of our tray design framework, we have compared its output (we call it TrayGen result here) with layouts generated by *TexturePacker* [Tex13] a popular software for packing multiple small textures into

data set	# objects	# faces	select	arrange
tools1 (Fig. 3)	12	15633	11.6	10.3
tools2 (Fig. 12)	26	26449	23.3	15.1
cosmetics	16	17476	15.5	10.0
toys	17	45733	17.7	18.6
vases	17	36372	18.5	10.5

Table 1: Performance statistics. The timings are in seconds.



Figure 15: A comparison between the layouts generated by TrayGen (left) and the result of TexturePacker (right).

a single texture image. It applies different heuristics within a prescribed image design domain, and the goal is to save space. To make fair comparisons, we convert each object into a 2D texture image based on the selected contour. The sizes of the texture images and the image design domain are the same as TrayGen. Fig. 15 shows one comparison (see supplemental material for additional results). It is clear that TrayGen better exhibits the inter-relations of objects.

We also asked a designer (whose expertise is product design) to manually generate tray layouts. All the models are loaded into a 3D scene of a modeling software. The designer can translate/rotate each object by the provided user interface. We did not show the designer TrayGen result or any other references. In average, it took the designer 20-25 minutes to generate one layout. Fig. 16 shows one comparison (see supplemental material for additional results).

To quantitatively evaluate the quality of the results generated by TrayGen, we have conducted a user study, where we asked 20 local residents of various majors to compare the TrayGen results with the results from TexturePacker and the design expert. We asked each participant to select the design that he/she favors. The results are shown in Fig. 17.

Compared with TexturePacker, almost all participants se-



Figure 16: A comparison between the results generated by TrayGen (left) and the manual result (right).

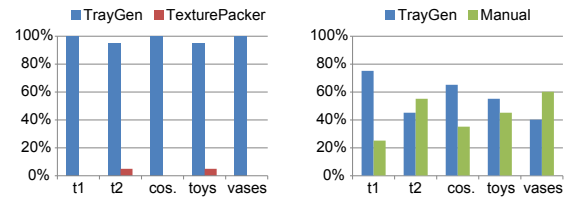


Figure 17: Comparisons of TrayGen result with TexturePacker result (left) and manual design (right). The corresponding histogram shows the statistics of design preference from the user study.

lected TrayGen result without hesitation. Their explanation was that the TrayGen result looked better organized, and it was much easier to identify individual objects. On the other hand, the manual design results share common characteristics (similarity, alignment, etc.) in the object arrangement with those of TrayGen, thus it apparently took the participants more time to select the better one. From the statistics we can see that TrayGen is slightly better in terms of overall performance. In particular, TrayGen highlights more of the object inter-relations (e.g., by setting smaller in-group distance and bigger inter-group distance). It leads to better results for the tools in Fig. 3 and the cosmetics set in Fig. 12. In contrast, for the vase data set, where aesthetics is a dominant factor (e.g., alignment to the boundary), participants tend to pick the manual design.

Limitations. Since only the outer contour is considered to avoid clutter, one limitation of our framework is that it does not allow nested objects. For example, in Fig. 3, small item could be put in the inner empty area of the saw to save more space. This could be solved by grouping different type of objects in a preprocessing stage. Another limitation is that we only process rigid objects. Deformable objects (e.g., cable, cloth) certainly have more freedom for packing and display.

7. Conclusions and Future Work

In this paper, we have presented a tray design framework called TrayGen for the exhibition and packaging of a collection of solid objects. The framework optimizes space occupation, exhibits the visual features of each object and highlights the inter-relations among multiple objects. Experimental results demonstrate that TrayGen can generate pleasing tray designs on a variety of data sets.

There are a number of ways to improve the current framework. First, although similar objects are arranged in a regular way, and the clearance term prevents cluttered layout, the overall aesthetics of the layout such as group-wise alignment and layout balance, can still be improved. Second, so far we have focused on designing trays with a single layer. In the future, we plan to consider how to arrange objects in multiple layers. Third, our analysis is mainly based on geometry. However, other object properties such as weight and texture (if applicable) could be incorporated as well. Finally,

the tray model from boolean operation is only a prototype. In some cases, auxiliary structures are necessary for a real design, e.g., partially dilated cells that facilitate the grasp of objects (see Fig. 2).

In the future, we would like to apply the perception based arrangement guidelines for visualizing 3D model collections from online shape repositories, e.g., generating a thumbnail of a model collection for instant on-line preview.

Acknowledgement. We thank the anonymous reviewers for their comments; Niloy Mitra and Helmut Pottmann for helpful suggestions; Ming-Ming Cheng for shape contexts; and Charlotte Rakhit for carefully proofreading the paper. Qi-Xing Huang is supported by NSF grants 914833 and 1011228, AFOSR grant FA9550-12-1-0372, as well as Google research award.

References

- [BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 4 (Apr. 2002), 509–522. 4
- [BSZP91] BASTOW-SHOOP H. E., ZETOCHA D. F., PASSEWITZ G.: *Visual Merchandising: A Guide for Small Retailers*. North Central Regional Center for Rural Development, Iowa State University of Science and Technology, 1991. 2, 3
- [CSY02] CAGAN J., SHIMADA K., YIN S.: A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design* 34, 8 (2002), 597 – 611. 2
- [Dyc90] DYCKHOFF H.: A typology of cutting and packing problems. *European Journal of Operational Research* 44, 2 (1990), 145–159. 6
- [FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 42:1–42:7. 2, 5
- [FRS*12] FISHER M., RITCHIE D., SAVVA M., FUNKHOUSER T., HANRAHAN P.: Example-based synthesis of 3D object arrangements. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 135:1–135:11. 2
- [GLM96] GOTTSCHALK S., LIN M. C., MANOCHA D.: Obbtree: a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 171–180. 5
- [IC01] IGARASHI T., COSGROVE D.: Adaptive unwrapping for interactive texture painting. In *Proceedings of the 2001 symposium on Interactive 3D graphics* (2001), pp. 209–216. 2
- [IKN98] ITTI L., KOCH C., NIEBUR E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 11 (1998), 1254–1259. 4
- [Jon99] JONES J. P.: *The Advertising Business: Operations, Creativity, Media Planning, Integrated Communications*. SAGE Publications, Inc., 1999. 2
- [KMP10] KORF R., MOFFITT M., POLLACK M.: Optimal rectangle packing. *Annals of Operations Research* 179 (2010), 261–295. 6
- [KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.* 22, 3 (July 2003), 954–961. 3
- [LACS08] LI W., AGRAWALA M., CURLESS B., SALESIN D.: Automated generation of interactive 3D exploded view diagrams. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 101:1–101:7. 5
- [LAZ*12] LI H., ALHASHIM I., ZHANG H., SHAMIR A., COHEN-OR D.: Stackabilization. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 158:1–158:9. 5
- [LGLM99] LARSEN E., GOTTSCHALK S., LIN M. C., MANOCHA D.: Fast proximity queries with swept sphere volumes. In *Proc. IEEE Int. Conf. Robot. Autom.* (1999), pp. 3719–3726. 5
- [LH06] LEORDEANU M., HEBERT M.: Efficient map approximation for dense energy functions. *ICML '06*, pp. 545–552. 4
- [LMM02] LODI A., MARTELLO S., MONACI M.: Two-dimensional packing problems: A survey. *European Journal of Operational Research* 141, 2 (2002), 241 – 252. 2
- [LVJ05] LEE C. H., VARSHNEY A., JACOBS D. W.: Mesh saliency. *ACM Trans. Graph.* 24, 3 (July 2005), 659–666. 2
- [MSL*11] MERRELL P., SCHKUFZA E., LI Z., AGRAWALA M., KOLTUN V.: Interactive furniture layout using interior design guidelines. *ACM Trans. Graph.* 30, 4 (July 2011), 87:1–87:10. 2
- [MYY*10] MITRA N. J., YANG Y.-L., YAN D.-M., LI W., AGRAWALA M.: Illustrating how mechanical assemblies work. *ACM Trans. Graph.* 29 (July 2010), 58:1–58:12. 2
- [NJW01] NG A. Y., JORDAN M. I., WEISS Y.: On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems* (2001), MIT Press, pp. 849–856. 3
- [NSX*11] NAN L., SHARF A., XIE K., WONG T.-T., DEUSSEN O., COHEN-OR D., CHEN B.: Conjoining gestalt rules for abstraction of architectural drawings. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 185:1–185:10. 2
- [OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. *ACM Trans. Graph.* 21, 4 (Oct. 2002), 807–832. 3
- [PPB*05] POLONSKY O., PATANĚ G., BIASOTTI S., GOTSCHMAN C., SPAGNUOLO M.: What is an image? towards the computation of the "best" view of an object. *The Visual Computer* 21, 8-10 (2005), 840–847. 4
- [SLF*11] SECORD A., LU J., FINKELSTEIN A., SINGH M., NEALEN A.: Perceptual models of viewpoint preference. *ACM Trans. Graph.* 30, 5 (Oct. 2011), 109:1–109:12. 2
- [Tex13] <http://www.codeandweb.com/texturepacker.7>
- [Tou83] TOUSSAINT G.: Solving geometric problems with the rotating calipers. In *Proceedings of IEEE Melecon* (1983), pp. A10.02/1–4. 3
- [VFSH01] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference 2001* (2001), VMV '01, Aka GmbH, pp. 273–280. 2
- [WEK*12] WAGEMANS J., ELDER J. H., KUBOVY M., PALMER S. E., PETERSON M. A., SINGH M., VON DER HEYDT R.: A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological Bulletin* 138, 6 (2012), 1172–1217. 2, 3
- [YC00] YIN S., CAGAN J.: An extended pattern search algorithm for three-dimensional component layout. *Journal of Mechanical Design* 122, 1 (2000), 102–108. 6
- [YYT*11] YU L.-F., YEUNG S.-K., TANG C.-K., TERZOPOULOS D., CHAN T. F., OSHER S. J.: Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph.* 30, 4 (July 2011), 86:1–86:12. 2
- [YYW*12] YEH Y.-T., YANG L., WATSON M., GOODMAN N. D., HANRAHAN P.: Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. *ACM Trans. Graph.* 31, 4 (July 2012), 56:1–56:11. 2