

Chapter 8

Shape Compaction

Honghua Li and Hao Zhang

Abstract We cover and discuss techniques that are designed for compaction of shape representations or shape configurations. The goal of compaction is to reduce storage space, a fundamental problem in many application domains. We consider compaction both at the representation level (i.e., digital storage) and in physical domains (i.e., physical storage). Shape representation compaction focuses on reducing the memory space allocated for storing the shape geometry data, whilst shape compaction techniques in the physical domain reduce the physical space occupied by shape configuration. We use the term *shape configuration* to refer to how a shape, real or conceptual, is physically modeled (e.g., design and composition of its parts) and spatially arranged (e.g., shape parts positioning and possibly in relation to other shapes). In this paper we briefly cover the representation compaction techniques whilst placing our focus on the less explored realm of shape compaction approaches on physical configurations.

8.1 Introduction

Memory space is valuable in digital environment. Digital models of 3D shapes are widely used in a vast number of industrial and scientific applications. Typically the same shape admits multiple mathematical representations which may vary significantly in storage cost. Among them the most compact ones in terms of storage cost are usually more preferable since they can reduce the cost of storage, transmission, computation and visualization, as well as facilitate shape understanding and intelligent shape processing.

Physical space is also costly and thus the demand for compact products is strong in practice. Objects that can change the arrangement of their parts or their spatial relation with other shapes (the so-called *shape configuration*) to save space when storing or transporting them, are of great value for survival (e.g., fire fighter equipment, army weapons and tools), camping in the wild (e.g., tent, pocket knife), living (e.g., IKEA furniture) and leisure (e.g., LEGO assembling toys).

H. Li (✉) • H. Zhang
Simon Fraser University, Burnaby, BC, Canada
e-mail: howard.hhli@gmail.com; haoz@cs.sfu.ca

In this paper, we use the term *shape compaction* to refer to techniques that can either assist human beings to reduce the storage size of shapes on both representation level and configuration domain, or automatically accomplish this goal. While numerous algorithms have been proposed for compaction of shape representations in literature, including simplification, abstraction, compression, etc., the compaction of shape configuration is still a realm that remains unexplored.

8.1.1 *Compaction of Shape Representation*

Given shape representation R_0 of a 3D object S , *representation compaction* is to (1) reorganize the data of R_0 to reduce its storage space or (2) find a new representation R of S which occupies less storage space subject to some criteria.

Shape representations are mathematical models conveying the geometry of 3D objects, and their size is measured as the amount of memory required to store such models. There are two key factors that influence the data size of a shape representation: the number of low-level primitives, and the statistical redundancy in geometric data. Shape representation compaction approaches addressing the former factor fall into the category of shape simplification and abstraction, while methods addressing the latter are usually regarded as shape compression techniques.

Shape simplification and abstraction The basic idea is to find a proxy with fewer primitives to represent the original object that consists of many finer primitives. Shape simplification aims to preserve the geometric fidelity within a prescribed error tolerance, while shape compaction has more freedom to modify the topology or geometry as long as the new generated representations are perceptually equivalent to the original shapes.

Compression Data compression techniques either exploit statistical redundancy in the underlying data to represent data more concisely (lossless), or modify the data in a subtle manner such that the statistical redundancy is enhanced (lossy). Mesh compression is the application of data compression on polygonal meshes. Typical mesh compression algorithms encode the connectivity and geometry data separately. Both natural and man-made objects present huge amount of regular and repeated substructures, which are usually captured by symmetries within the shape. Traditional mesh compression approaches do not explicitly utilize this statistical redundancy on the structure level. In a recent trend of research, several hierarchical representation techniques have been proposed to compactly represent complex shapes with rich symmetries in their structures.

8.1.2 *Compaction of Shape Configuration*

Size reduction of physical storage space is significantly different from that of memory space occupied by shape representations. The redundancy in digital models can be efficiently encoded to reduce the total storage space, which however isn't useful for physical storage reduction at all. For example, a shape with reflective symmetry can be compactly represented by half of its geometry and the associated reflection plane. In contrast, the two identical halves (in terms of reflection) both need to physically exist and thus occupy the same amount of space.

An intriguing problem about compact shapes is: what makes some objects more amenable to saving space than others? In an excellent introduction to space-saving designs, [37] discussed twelve collapsible principles. Collapsible objects are able to adjust in size by switching between two opposite configurations: one unfolded and *functional*, the other folded for *storage*. The existence of functional and storage configurations only makes it possible for an object to be collapsible. To be *practically* collapsible, the transformation between these two configurations must be feasible and easy to conduct.

Shape configuration is the arrangement of shape parts and/or the spacial relationships between shapes.

Collapsible objects can save space either individually involving the organization of parts within a shape, which is called *intra-shape configuration*, or cooperatively involving spacial relationships among multiple objects, which is called *inter-shape configuration*. The chairs in Fig. 8.1a, b demonstrate two examples of intra-shape collapsing strategies: folding and decompose-and-pack. The stackable chair in Fig. 8.1c has a set of identical chairs involved while storing them. The outdoor tea table set in Fig. 8.1d consists of one tea table and four seats, which as a group can be packed compactly when not in use.

Given a 3D object S , *configuration compaction* is to find a new 3D object T such that (1) T is close to S , (2) T is able to change configuration to save space.

The problem has a trivial solution if sufficiently large perturbation from the source object S is allowed (e.g., let T be a cube). The “closeness” between two objects needs to be formalized such that it preserves the essence, i.e., structure and functionality of the original shape S . We classify compaction techniques into two categories based on the type of shape configuration they attempt to tackle.

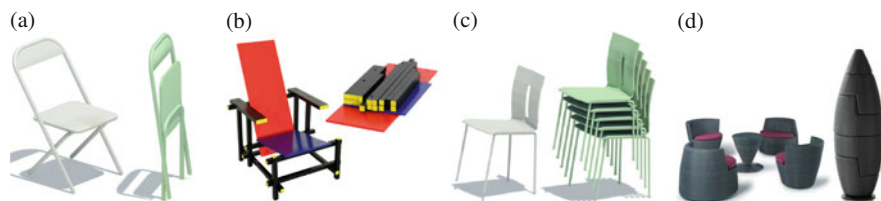


Fig. 8.1 Four collapsible mechanisms with the functional(*left* in each cell) and storage(*right* in each cell) configurations. (a) Folding. (b) Decompose-and-pack. (c) Stacking. (d) Group packing

Intra-shape configuration is the arrangement of shape parts within a shape. Compaction approaches in this category produce new shapes that preserve the essences of the original shapes in terms of either geometric appearance or functionality. Shape parts of the output can be transferred into a storing configuration which takes much less physical storage space than the original shape.

Inter-shape configuration indicates the internal relations between multiple shapes. Given a set of shapes, finding out the optimal configuration itself is a very challenging problem. Moreover there are algorithms that can modify the original shapes subtly such that the final packing result can be more space-saving.

8.2 Simplification and Abstraction

Given a representation, simplification and abstraction approaches output another representation option for the underlying shape which consists of fewer primitives than the original one. Dozens of simplification algorithms have been proposed by researchers in computer graphics. A detailed review of simplification techniques in literature is beyond the scope of this paper. Interested readers should refer to [14, 28, 29] for a broader survey on simplification approaches.

Representing complex objects with low bit budget goes beyond the capability of an error-metric-driven simplification method and the answer often lies in the area of human perception and cognition. Given a shape S , the goal of shape abstraction is to produce a proxy \mathcal{S} such that perceptually S and \mathcal{S} are comparable, but representationally $|\mathcal{S}| \ll |S|$. Note that S and \mathcal{S} are likely to be quite different from a purely geometric point of view. These compact representations are visually more appealing than the detailed original models, which might appear visually cluttered. Therefore they are widely used for prototyping and concept communication.

The boundary between shape simplification and abstraction sometimes is blurry. Simplification with extremely low bit budget can be considered as abstraction, and abstraction at a very fine level may produce comparable results to simplification. The key characteristic of abstraction is that it directly extracts the shape defining features of objects which usually are inspired from human perception and cognition (Fig. 8.2).

Curve networks Sparse characteristic feature curves are typically sufficient for humans to identify a shape. Despite the fact that CG lines (image intensity edges, geometric ridges and valleys, suggestive contours, and apparent ridges) seem likely to succeed in conveying shapes [5, 6], they are usually not well organized and might be view-dependent. De Goes et al. [8] proposed the so-called *exoskeleton* to convey both the perceptual and the geometric structure of a 3D model. They first segment the input shape into parts, and further divide the shape surface into patches. The boundaries of these resulted patches form the exoskeleton.

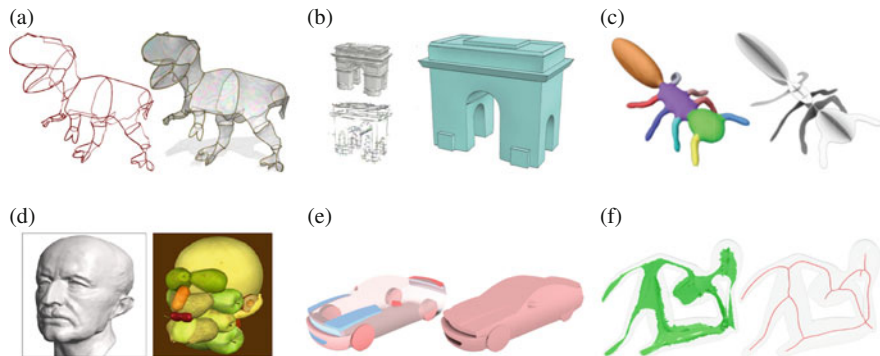


Fig. 8.2 Various shape abstractions. (a) Exoskeleton. (b) Surface model. (c) Slices. (d) Collage. (e) Subvolume. (f) Skeletons

Surface models Unlike simplification approaches which operate at low-level primitives and usually do not preserve shape defining features under extreme simplification, [33] extracts a sparse network of space curves that capture the essential characteristic features of a given man-made object, from which an abstract surface model can be reconstructed. Their method operates in two steps. First, a closed manifold envelope surface that wraps the entire input model is extracted from the voxelization of the input object. Second, they extract a network of curves or vectors from the computed envelope.

Planar sections Inspired by section planes in medical and engineering visualization which illustrate the interior details of complex shapes, [32] proposed an approach for generating shape proxies consisting of planar sections. In their method, planes are progressively selected to maximally capture shape features weighted by their importance, which is learned from the user study trying to discover how humans define planar section representations for various 3D shapes.

Collage Collage is an abstract and expressive visual style that build a new whole by assembling given primitives in a database. In a collage, both the parts and the whole can be easily recognized. Gal et al. [13] created 3D collages that express the target shape using a database of objects as primitive building block. In a parallel thread of work, [47] generated animation collage from mesh animation. In a recent work, [19] developed an algorithm for creating a collage which represents a given image with multiple Internet images. Note that the primitives used for collage are usually more complex than simple geometry primitives, therefore the collage techniques are considered as shape abstraction approaches solely because the little number of primitives.

Subvolumes Yumer and Kara [53] proposed an abstraction method that is built on *subvolumes*. The most abstract form is generated first and more details that are represented by volume chunks can be added or subtracted to the current abstraction. The main contribution of this work is that they can generate a spectrum

of abstractions for each shape, and rely on the co-analysis on the associated shape collections to determine the “right” abstraction.

Skeletons The most well-known skeletal shape representation is probably the medial axis transform (MAT) [3]. In computer graphics, curve skeletons [7] are more broadly utilized due to their compactness and ease of manipulation. We refer the interested readers to recent advances on curve skeleton extraction [2, 18, 45, 46] for more details.

8.3 Compression

Data compression means to encode information using fewer bits than the original representation. Compression can be either lossless or lossy. Lossless compression is conducted by eliminating the statistical redundancy in the data. Some information lost is acceptable in lossy compression. By modifying in a subtle way, the data could be more amenable to coding, thus higher compression rate can be achieved.

The output of shape compression has to be decoded to be used, which is never a free lunch. However, shape compression has the advantage of using a given budget of storage space to represent more detailed shapes. Moreover, compression techniques can be used together with shape simplification and abstraction to obtain more compact shape representations. We refer interested readers to [1, 40] for a deeper and broader review of mesh compression techniques.

With the recent advance on shape structure analysis [36], compression techniques have been proposed to address data redundancy at structure level. Repeating substructures in digital models can be explicitly encoded to reduce its space complexity [39]. Due to the nested nature of symmetries, the simple strategy may encode the same symmetry multiple times. A hierarchical encoding, however, can reflect the nested structure and produce a more compact representation of the entire shape.

As a recent advance, there has been three pieces of work that develop hierarchical representation of single objects or complex scenes to address this type of structural redundancy.

Folding mesh Simari et al. [43] used a *folding tree* data structure to encode the reflective symmetries within a mesh by recursively applying a symmetry detection algorithm. The data structure encodes the non-redundant regions of the original geometry as well as the reflection planes. The folding tree can eventually be unfolded to recover the original shape approximately, see Fig. 8.3 (top right).

Symmetry and instancing Martinet [30] proposed the hierarchical assembly graph (HAG) to represent the structural information in scenes. A HAG is a directed graph, in which each node denotes an *object* and a arc denotes the sub-part relation between two objects. An object is defined as a *closed frequent pattern*, which is a part of the scene that does not have subpart having higher frequency than itself, see Fig. 8.3 (left).

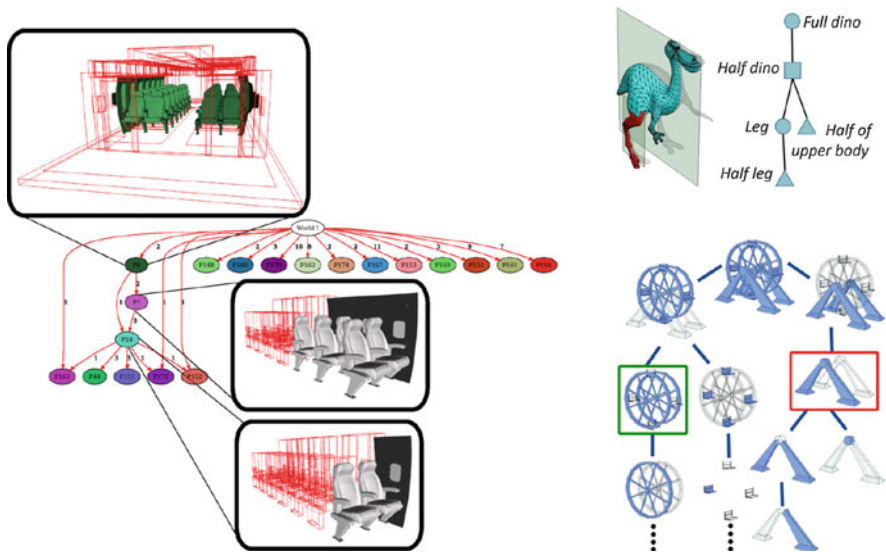


Fig. 8.3 Structural shape compression techniques: hierarchical assembly graph (*left*), folding mesh (*top right*), and symmetry hierarchy (*bottom right*)

Symmetry hierarchy Wang et al. [50] described an analogous method to construct the symmetry hierarchical (SYMH) organization of object parts by using perceptual grouping criteria. The input mesh is initially segmented into parts which are refined by symmetries detected in the original shape. An initial graph is built to encode inter-part symmetry and connectivity relations among the resulting segments, as well as self-symmetry for individual segments. The symmetry hierarchy is then constructed from the initial graph via graph contraction, which either groups parts by symmetry, or assembles connected sets of parts. The order of graph contraction is determined by a set of rules designed to respect human perceptions and the principle of compactness. See Fig. 8.3 (bottom right) for an example.

The HAG proposed by [30] is a directed graph which is different from the tree structures described by the other two. The advantage of a graph structure is that different part of the shape can share the same set of geometry primitives stored in leaf nodes. Primitives geometry represented by leaf nodes are building blocks when establishing hierarchical representations. Although different algorithms have been explored, finding the “best” primitive geometries still remains an open problem.

8.4 Compaction of Intra-shape Configuration

Shapes can reduce size individually by changing their own configuration. Given a 3D object, compaction of intra-shape configurations is to find another object that is close to the input in terms of geometrical appearance, structural form, or functional essence, but also is able to adjust its configuration to meet the requirement of space-saving. This can be achieved by either modifying the original shape or creating a new shape via approximation. In this section we discuss two mechanisms – folding and decomposing – that are frequently utilized for shape compaction.

8.4.1 Folding

Folding via hinges is a popular collapsible principle that impacts many tools in our daily lives. Generally speaking a hinge is a movable joint that connects two objects and typically allows rotation between them. The most popular form of folding is probably paper folding [9, 20], with origami [31] being the best known instance.

Pop-up design Popups are paper arts that can be closed down to a flat surface and opened up again without tearing the paper or introducing new creases other than those in the design. A popup is collapsible since it has both functional and storing configurations, one of which can be easily transformed into the other without extra forces other than holding and turning two support pages.

Origamic architectures, also called paper architectures (PA), are paper buildings created by cutting and folding from a single piece of paper. The simple mechanisms of parallel PA enabled development of automated algorithms to construct a PA from an input 3D model [25, 35] as well as interactive tools [34], see Fig. 8.4a. Li et al.

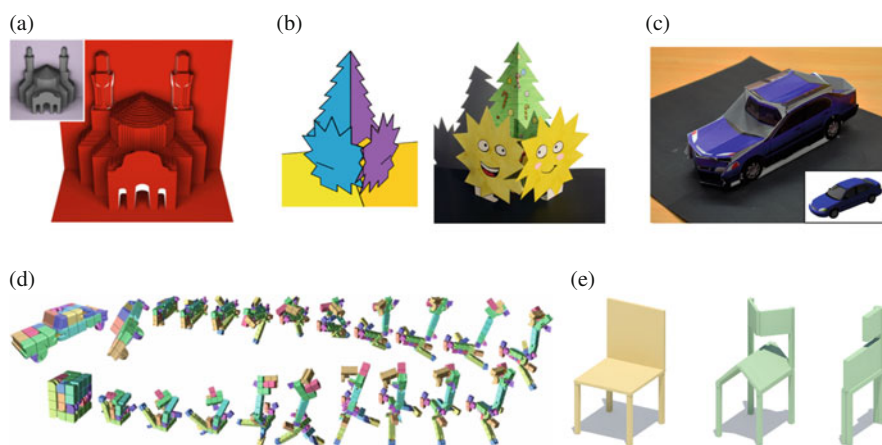


Fig. 8.4 Shape compaction techniques utilizing the folding mechanisms. (a) Pop-up. (b) V-style pop-up. (c) Multi-style pop-up. (d) Boxelization. (e) Foldabilization

[24] studied the general v-style popups, which contain two more parallel groups of planes with multiple pieces of paper, see Fig. 8.4b. Ruiz et al. [41] extend the popup design to multi-style by fitting volumetric primitives and mapping to selected mechanisms, see Fig. 8.4c.

Foldable puzzle design In a recent work, [54] approximates the input shape using a voxel-tree that can fold from the input shape into a cube. The goal of boxelization is to find a physically achievable solution for transforming a shape into a cube. Compactness is one of the objective terms in the optimization procedure and shape compaction is therefore achieved as a by-product. Their algorithm involves three major steps: finding a good voxelization, finding the tree structure that can form the input and target shapes' configurations, and finding a non-intersecting folding sequence.

Foldable furniture design Space-saving furniture designs are ubiquitous in our lives and folding is perhaps the most popular mechanism observed and practiced [37]. However the design process of foldable furniture has to follow the trial-and-error iteration, which is usually both tedious and time consuming. Here we pose an open *foldabilization* problem: given a 3D furniture, how to apply a minimum amount of modification to the input to allow it to be folded? Figure 8.4e provides an example solution: by introducing hinges on the seat and back and shrinking the back, the modified chair is able to fold into a flat configuration.

8.4.2 Decomposing

The functional configuration of an object usually leaves large amount of free space among its parts, which increases the cost for fabrication or storing. Decomposing provides an option to reorganize shape parts to reduce this unused space.

Decompose-and-print Layered printing has been widely used in 3D printers. Usually support structure has to be printed together with the object itself to allow complex shapes to be fabricated, which however causes material cost and takes longer time to print. The amount of support material depends on the free space within the projection volume of an object. Inspired by pyramidal shapes which always have solid projection volume with respect to the given base, [17] proposed an algorithm to decompose the input 3D model into approximately pyramidal parts, see Fig. 8.5a. By printing each pyramidal parts individually and gluing, the original object can be fabricated. The pyramidal composition is more compact than the original object in terms of projection volume.

Decompose-and-pack Decompose-and-pack is a time-honored collapsible principle. A number of separate parts are assembled into a whole to perform functions, and then later are dismantled again into its parts for storage. An excellent example is flat pack furniture which supports almost the entire business of IKEA.

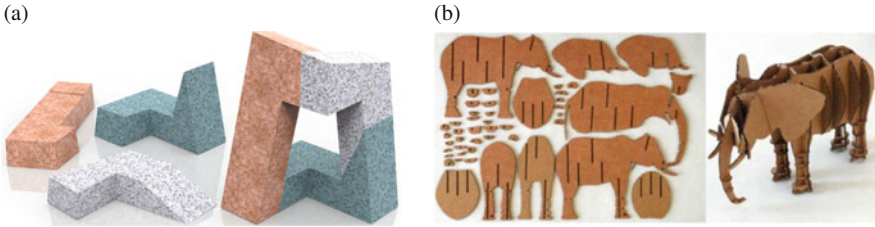


Fig. 8.5 Shape compaction techniques using the decomposing mechanism. (a) Approximate pyramidal decomposition. (b) Cardboard sculpture

The cardboard sculpture is another example where cardboard pieces have prefabricated slits along which they can be slid to assemble the whole shape. Obviously these cardboard pieces can be stored much more compact than as a whole. Hildebrand et al. [15] proposed an algorithm to automatically generate cardboard sculptures with guaranteed constructibility, see Fig. 8.5b.

Given an arbitrary 3D model, searching for the decomposition and packing strategy that leads to the most compact packing remains an open problem.

8.5 Inter-shape Configuration Compaction

Shapes can work cooperatively to save space. This group strategy involves changing the spatial relations with other shapes. A set of objects can be packed more compactly under rigid transformations as long as the unused space within one shape's bonding volume can be used by another shape, see Fig. 8.1c, d.

Without modifying input shapes in any way, the problems we are discussing here degenerate into the classic *nesting problems*. As a specific type of *cutting and packing (C&P) problems*, nesting problems consider packing irregular shapes in order to optimize the packing volume. The problem is NP-hard and as a result solution methodologies usually utilize heuristics. The term “compaction” was also used by [26] to refer to a simultaneous motion of the components that generates a more densely packed layout.

A dense nesting is possible only if the irregular shapes can fit into each other very well. An extreme case is tiling [49], where each tile can exactly fit into its neighbours such that all tiles together can cover the entire plane. However arbitrary shapes usually do not have such nice properties. In many cases the input geometries do not have to keep unchanged but their essences, e.g. main features and functionality. In fact, allowing subtle changes to the input shapes can greatly improve nesting results [21, 23].

In this section, we first briefly overview the challenges and state-of-the-art solutions of nesting problems, then follow up with techniques that modify and optimize the geometry of input shapes for more compact packing results.

8.5.1 *Without Shape Modification*

The topic of cutting and packing covers a variety of problems of a common logical structure which is usually classified under the heading of packing, packaging, layout, configuration, container stuffing, pallet loading or spatial arrangement in the literature. Dyckhoff and Wäscher et al. [10, 51] introduced a useful typology of C&P problems, where C&P problems can be classified into regular packing and irregular packing, the latter is also called *nesting problems*.

The nesting problem is usually abstracted as an optimization problem where an assignment of the positions and orientations of components that minimizes an objective is sought. Comparing to regular packing [27], irregular components increase the complexity of the solution space. The problem is a NP-hard combinatorial problem [38] such that meta-heuristics are typically used to generate acceptable solutions. Hopper and Turton [16] reviewed these meta-heuristic algorithms, in particular genetic algorithms, for both 2D regular and irregular packing problems. As research progressed, new breakthroughs have been achieved in recent years. Timmerman [48] compared different optimization methods using benchmarks and concludes that extended local search [22] is the best method currently available.

3D nesting problem shares most characteristics with its two-dimensional counterpart, but the geometric complexity of 3D irregular components makes it a more challenging problem. Cagan et al. [4] reviewed a spectrum of approaches ranging from deterministic algorithms to stochastic algorithms proposed for solving 3D layout problems. The geometric representation and interference detection approaches of 3D components are also discussed in that survey. Most algorithms are originally designed for 2D nesting problems and have the potential to be extended to 3D [11, 44]. In contrast, the extended pattern search algorithm [52] was particularly designed for 3D nesting problems.

8.5.2 *With Shape Modification*

Modifying the input shapes is not necessary, but when applied it has the potential to improve the nesting density. Shape modification is not always possible or allowed, since traditionally nesting is an independent post procedure after the design of a product has been fixed. If nesting quality is not considered during the product design, a nesting algorithm solely is doomed to fail on finding very dense packing layout. In fact products that are successful in space saving are originally designed to be so. Instead of barely relying on the designer's experience and letting the designers improve their design in a trial-and-error iteration, computational algorithms can be designed to either assist designers to speed up the iterations or automatically modify the design in a subtle manner to achieve more compact packing layout.

Escherization Tiling is a special case of 2D nesting problems because each component (tile) can exactly fit into its neighbors such that the entire plane can be

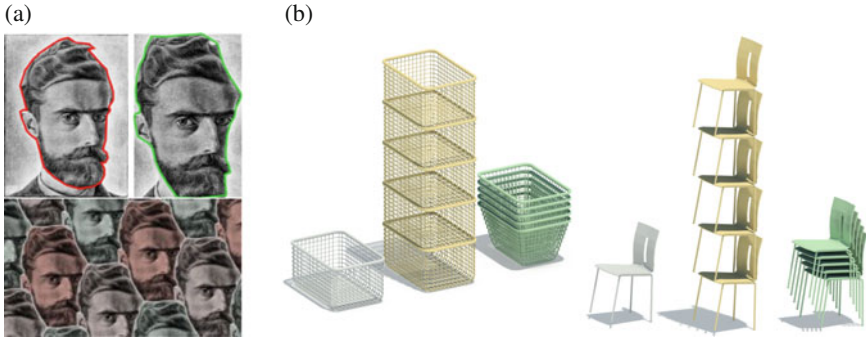


Fig. 8.6 Shape compaction techniques with modification. **(a)** Escherization. **(b)** Stackabilization

covered seamlessly. The Dutch artist M.C. Escher spent his career on producing a notebook with more than a hundred of ingenious and playful designs of tiling [42]. Inspired by Escher’s work, [21] presented a solution to the “Escherization” problem: given a closed figure, find a new closed figure that is similar to the input and tiles the entire plane, see Fig. 8.6a. Their approach utilizes a simulated annealer to optimize over a parameterization of the *isohedral* tilings, which is flexible enough to encompass nearly all of Escher’s own tilings.

Stackabilization Stacking objects on top of each other is a common strategy performed by humans to save space. The nesting layout of a stack along a stacking direction can be achieved by repeated application of a translation and a possible rotation on object copies until two adjacent objects are just touching each other without overlap or gaps. One of the most celebrated examples of stackable objects are chairs [12].

Li et al. [23] first introduced the geometric problem of stackabilization: how to geometrically modify a 3D object so that it is more amenable to stacking? They consider the class of stackings that involves only translation in the stacking configuration. The main challenge in stackabilization lies in the desire to modify the input geometry only subtly so that the intended functionality and aesthetic appearance of the original object are not significantly affected.

8.6 Conclusion

This is the first general introduction on shape compaction techniques, at both the digital representation level and the physical configuration domain. These two compaction categories share the same goal of finding economy solutions for storing and transporting objects, which is beneficial in a large range of applications. They also share the spirit of utilizing shape modification to facilitate the compaction results. In particular, simplification and abstraction of shape representation would

have strong connection to shape compaction of intra-shape configuration, e.g. popup and cardboard sculpture.

Due to the strong practical demands on compact digital and physical objects, more effort from researchers is expected to commit in this realm. To conclude this paper we list a few possible future directions along this thread of research.

Because of the conceptual nature, shape abstraction is worth more creative investigation. Structure analysis has attracted tremendous attention recently, which provides opportunities for finding better structural compression approaches.

By now 3D nesting problems have not drawn comparable amount of attention from researchers as that in 2D. The needs arising in the product layout, rapid prototyping, and efficient use of resources (e.g., 3D printing material) justify the development of efficient nesting approaches for 3D components with complex geometry.

Most intra-shape configuration compaction approaches, e.g. popup designs, and boxelization, can only approximate the appearance of the input in a very rough manner. The reason is that the feasibility of particular collapsibility usually serves as hard constraint, while sacrificing the appearance and even the essential of the given 3D model. An open problem is how to develop generic computational approaches for generating collapsible objects that can preserve the functionality or at least the structure of the input.

Generally speaking, compaction of shape configuration is a relatively unexplored area with numerous open problems waiting to be studied. Solving these problems will benefit a huge amount of practical applications which are sensitive to physical storage space.

References

1. Alliez, P., Gotsman, C.: Recent advances in compression of 3d meshes. In: *Advances in Multiresolution for Geometric Modelling*, pp. 3–26. Springer, Berlin/London (2005)
2. Au, O.K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., Lee, T.-Y.: Skeleton extraction by mesh contraction. *ACM Trans. Graph.* **27**(3), 44 (2008)
3. Blum, H.: A transformation for extracting new descriptors of shape. In: Wathen-Dunn, W. (ed.) *Models for the Perception of Speech and Visual Form*, pp. 362–380. M.I.T. Press, Cambridge (1967)
4. Cagan, J., Shimada, K., Yin, S.: A survey of computational approaches to three-dimensional layout problems. *Comput. Aided Des.* **34**, 597–611 (2002)
5. Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H.S., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S.: Where do people draw lines? *ACM Trans. Graph. (Proc. SIGGRAPH)* **27**(3), 88 (2008)
6. Cole, F., Sanik, K., DeCarlo, D., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S., Singh, M.: How well do line drawings depict shape? *ACM Trans. Graph. (Proc. SIGGRAPH)* **28**(3), 28 (2009). *Proc. SIGGRAPH*
7. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. Comput. Graph.* **13**(3), 530–548 (2007)
8. De Goes, F., Goldenstein, S., Desbrun, M., Velho, L.: Technical section: exoskeleton: curve network abstraction for 3d shapes. *Comput. Graph.* **35**(1), 112–121 (2011)

9. Demaine, E.D., O'Rourke, J.: *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, Cambridge/New York (2007)
10. Dyckhoff, H.: A typology of cutting and packing problems. *Eur. J. Oper. Res.* **44**(2), 145–159 (1990)
11. Egeblad, J., Nielsen, B.K., Odgaard, A.: Fast neighborhood search for two- and three-dimensional nesting problems. *Eur. J. Oper. Res.* **183**(3), 1249–1266 (2007)
12. Fiell, C., Fiell, P.: *1000 Chairs*. Taschen, New York (2000)
13. Gal, R., Sorkine, O., Popa, T., Sheffer, A., Cohen-Or, D.: 3D collage: expressive non-realistic modeling. In: *NPAR: Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, San Diego, pp. 7–14 (2007)
14. Heckbert, P.S., Garland, M.: Survey of polygonal surface simplification algorithms. In: *Multiresolution Surface Modeling Course SIGGRAPH'97*, Los Angeles (1997)
15. Hildebrand, K., Bickel, B., Alexa, M.: crdbrd: shape fabrication by sliding planar slices. *Comput. Graph. Forum* **31**, 1583–592 (2012)
16. Hopper, E., Turton, B.C.H.: A review of the application of meta-heuristic algorithms to 2d strip packing problems. *Artif. Intell. Rev.* **16**(4), 257–300 (2001)
17. Hu, R., Li, H., Zhang, H., Cohen-Or, D.: Approximate pyramidal shape decomposition. In: *Proceedings of SIGGRAPH*, Vancouver (2014)
18. Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., Chen, B.: L1-medial skeleton of point cloud. *ACM Trans. Graph.* **32**, 65:1–65:8 (2013)
19. Huang, H., Zhang, L., Zhang, H.-C.: Arcimboldo-like collage using internet images. *ACM Trans. Graph.* **30**(155), 1–8 (2011)
20. Jackson, P.: *Folding Techniques for Designers: From Sheet to Form*. Laurence King Publishing, London (2011)
21. Kaplan, C.S., Salesin, D.H.: Escherization. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH'00*, New Orleans, pp. 499–510. ACM Press/Addison-Wesley Publishing Co. (2000)
22. Leung, S.C., Lin, Y., Zhang, D.: Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. *Comput. Oper. Res.* **39**(3), 678–686 (2012)
23. Li, H., Alhashim, I., Zhang, H., Shamir, A., Cohen-Or, D.: Stackabilization. *ACM Trans. Graph.* **31**(6), 158:1–158:9 (2012)
24. Li, X.-Y., Ju, T., Gu, Y., Hu, S.-M.: A geometric study of v-style pop-ups: theories and algorithms. *ACM Trans. Graph.* **30**(4), 98:1–10 (2011)
25. Li, X.-Y., Shen, C.-H., Huang, S.-S., Ju, T., Hu, S.-M.: Pop-up: automatic paper architectures from 3d models. *ACM Trans. Graph.* **29**(4), 111:1–9 (2010)
26. Li, Z.: *Compaction algorithms for non-convex polygons and their applications*. Ph.D. thesis, Harvard University (1994)
27. Lodi, A., Martello, S., Monaci, M.: Two-dimensional packing problems: a survey. *Eur. J. Oper. Res.* **141**(2), 241–252 (2002)
28. Luebke, D., Watson, B., Cohen, J.D., Reddy, M., Varshney, A.: *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York (2002)
29. Luebke, D.P.: A developer's survey of polygonal simplification algorithms. *IEEE Comput. Graph. Appl.* **21**(3), 24–35 (2001)
30. Martinet, A.: *Structuring 3D geometry based on symmetry and instancing information*. Ph.D. thesis, INP Grenoble (2007)
31. McArthur, M., Lang, R.J.: *Folding Paper: The Infinite Possibilities of Origami*. Turtle Publishing, Tokyo (2013)
32. McCrae, J., Singh, K., Mitra, N.J.: Slices: a shape-proxy based on planar sections. *ACM Trans. Graph.* **30**(6), 168:1–168:12 (2011)
33. Mehra, R., Zhou, Q., Long, J., Sheffer, A., Gooch, A., Mitra, N.J.: Abstraction of man-made shapes. *ACM Trans. Graph.* **28**(5), 137:1–137:10 (2009)
34. Mitani, J., Suzuki, H.: Computer aided design for origamic architecture models with polygonal representation. In: *Proceedings of Computer Graphics International*, Crete, pp. 93–99 (2004)

35. Mitani, J., Suzuki, H., Uno, H.: Computer aided design for origamic architecture models with voxel data structure. *Trans. Inf. Process. Soc. Jpn.* **44**(5), 1372–1379 (2003)
36. Mitra, N.J., Wand, M., Zhang, H., Cohen-Or, D., Bokeloh, M.: Structure-aware shape processing. In: *EUROGRAPHICS State-of-the-art Report*, Girona (2013)
37. Mollerup, P.: *Collapsible: The Genius of Space-Saving Design*. Chronicle, San Francisco (2001)
38. Nielsen, B.K., Odgaard, A.: Fast neighborhood search for the nesting problem. Technical Report 03/03, Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen Ø (2003)
39. Pauly, M., Mitra, N.J., Wallner, J., Pottmann, H., Guibas, L.: Discovering structural regularity in 3D geometry. *ACM Trans. Graph.* **27**(3), 43:1–11 (2008)
40. Peng, J., Kim, C.-S., Jay Kuo, C.C.: Technologies for 3d mesh compression: a survey. *J. Vis. Commun. Image Represent.* **16**(6), 688–733 (2005)
41. Ruiz Jr., C.R., Le, S.N., Yu, J., Low, K.-L.: Multi-style paper pop-up designs from 3d models. *Comput. Graph. Forum (Special Issue of Eurographics)* **33**(2), 487–496 (2014)
42. Schattschneider, D., Escher, M.C.: *Visions of Symmetry*. W.H. Freeman, New York (1990)
43. Simari, P., Kalogerakis, E., Singh, K.: Folding meshes: hierarchical mesh segmentation based on planar symmetry. In: *Symposium on Geometry Processing*, Cagliari, pp. 111–119 (2006)
44. Stoyan, Y., Romanova, T.: Mathematical models of placement optimisation: two- and three-dimensional problems and applications. In: Fasano, G., Pintér, J.D. (eds.) *Modeling and Optimization in Space Engineering*. Springer, New York (2013)
45. Tagliasacchi, A., Alhashim, I., Olson, M., Zhang, H.: Mean curvature skeletons. *Comput. Graph. Forum* **31**(5), 1735–1744 (2012)
46. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph.* **28**(3): 71, 9 (2009)
47. Theobalt, C., Rössl, C., de Aguiar, E., Seidel, H.-P.: Animation collage. In: *Symposium on Computer Animation*, San Diego, pp. 271–280. *Eurographics* (2007)
48. Timmerman, M.: Optimization methods for nesting problems. Master’s thesis, University West (2013)
49. van Lemmen, H.: *Tiles: 1000 Years of Architectural Decoration*. Harry N. Abrams, Inc., New York (1993)
50. Wang, Y., Xu, K., Li, J., Zhang, H., Shamir, A., Liu, L., Cheng, Z., Xiong, Y.: Symmetry hierarchy of man-made objects. *Comput. Graph. Forum (Special Issue of Eurographics)* **30**(2), 287–296 (2011)
51. Wäscher, G., Haußner, H., Schumann, H.: An improved typology of cutting and packing problems. *Eur. J. Operat. Res.* **183**(3), 1109–1130 (2007)
52. Yin, S., Cagan, J.: An extended pattern search algorithm for three-dimensional component layout. *ASME J. Mech. Des.* **122**(1), 102–108 (2000)
53. Yumer, M.E., Kara, L.B.: Co-abstraction of shape collections. *ACM Trans. Graph.* **31**, 158:1–158:11 (2012). *Proceedings of SIGGRAPH Asia 2012*
54. Zhou, Y., Sueda, S., Matusik, W., Shamir, A.: Boxelization: folding 3d objects into boxes. *ACM Trans. Graph.* **33**(4), 71:1–71:8 (2014)