K-set Tilable Surfaces

Chi-Wing Fu Chi-Fu Lai Ying He Daniel Cohen-Or
Nanyang Technological University Tel Aviv University

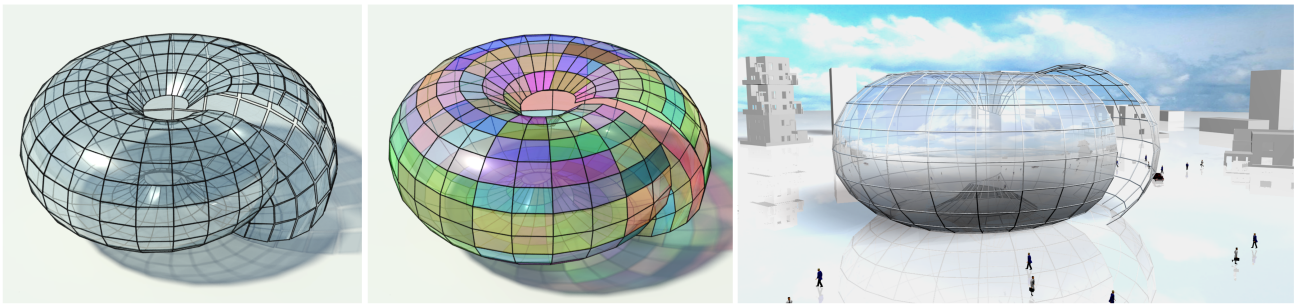


Figure 1: *Left: simply replacing each quad by its cluster mean shape, the* SEASHELL *surface breaks; middle: The K-set tilable surface approximates the input surface with instances of* 60 *distinct quad shapes; we shade different groups with random colors; right: an architectural design of a museum building using a K-set tilable surface on* SEASHELL *with* $K = 60$.

Abstract

This paper introduces a method for optimizing the tiles of a quad-mesh. Given a quad-based surface, the goal is to generate a set of K quads whose instances can produce a tiled surface that approximates the input surface. A solution to the problem is a K-set tilable surface, which can lead to an effective cost reduction in the physical construction of the given surface. Rather than molding lots of different building blocks, a K-set tilable surface requires the construction of K prefabricated components only. To realize the K-set tilable surface, we use a cluster-optimize approach. First, we iteratively cluster and analyze: clusters of similar shapes are merged, while edge connections between the K quads on the target surface are analyzed to learn the induced flexibility of the K-set tilable surface. Then, we apply a non-linear optimization model with constraints that maintain the K quads connections and shapes, and show how quad-based surfaces are optimized into K-set tilable surfaces. Our algorithm is demonstrated on various surfaces, including some that mimic the exteriors of certain renowned building landmarks.

Keywords: Computer-aided-geometric design, architectural geometry, computational differential geometry, freeform surface, tiling.

1 Introduction

Computer-aided-geometric design (CAGD) systems offer powerful solutions with accurate engineering constraints in the design and manufacturing of wide ranges of 3D geometric models. One emerging area is modern architecture, where the exteriors of a fast growing number of specially-designed building landmarks were using CAGD methods [Pottmann et al. 2007a-b]. These buildings typically come with noticeable aesthetic-looking curved surfaces, such as those in Figure 2. However, compared to the immense research effort that has been devoted to the design of professional CAD systems, until recently relatively little attention has been paid to geometric techniques to improve the quality and efficiency in architectural surface modeling.

In modern architecture, prefabricated components, or semi-finished parts, are used to assemble walls, panels, and ceilings [Blanc et al. 1993]. These prefabricated components are pre-built offsite, transported to the construction site, and further assembled onsite. The use of prefabricated components has a number of clear advantages. Its applicability to model curved building exteriors is, however, limited by the distinct panel shapes that curved surfaces require. Undoubtedly, up to basic symmetries, typically each prefabricated panel has a unique shape on the curved surface.

Given a quad-based surface as an input, the goal of our work is to generate a K-set tilable surface (or KT-surface) with a prescribed K while maintaining the KT-surface close to the input surface. Beyond being close to the surface, we expect the KT-surface to have similar geometric characteristics, such as the surface curvature, as in the input surface, and thus taking the square faces of a voxelization of surface is an invalid solution. Note, however, that we do not require the quads to be planar. The main contribution of this work is two-fold. First, the introduction of the notion of K-set tilable surfaces. Then, we present a solution to the problem based on non-linear optimization. We show the construction of various KT-surfaces that are quantitatively close to the input surfaces.

Overview of our approach. To generate a K-set of tiles, we take an optimization approach that consists of iterative clustering and

Figure 2: *Buildings with curved exteriors: Swiss Re Tower in London, De Beers Ginza in Japan, and Turning Torso in Sweden.*

ACM Reference Format
Fu, C., Lai, C., He, Y., Cohen-Or, D. 2010. K-set Tilable Surfaces. *ACM Trans. Graph. 29*, 4, Article 44 (July 2010), 6 pages. DOI = 10.1145/1778765.1778781 http://doi.acm.org/10.1145/1778765.1778781.

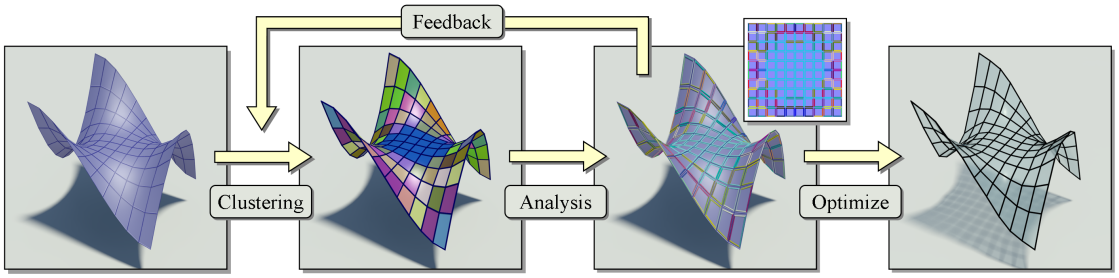


Figure 3: *Overview of our approach over a* MONKEY SADDLE *surface: We iteratively cluster and analyze: For this example we (i) cluster the quads into* 24 *clusters, and (ii) analyze the edge connections to detect* 33 *distinct edge lengths in the specific clustering configuration. Then (iii) we optimize the given surface into a KT-surface.*

analysis (see Figure 3): after clustering the quads, we generate K representatives and learn the flexibility from the clustering by analyzing the relations between the clusters and the quad-mesh. The surface vertices are then optimized so that all quads agree with the representatives while remaining close to the given surface. We assume that the given surface is tiled with quads only, possibly initially of a unique shape. To reduce the number of distinct quad shapes to K, one can first cluster together the surface quads by their shape similarity, using some rotation-invariant similarity measures between quads, and replace each quad with a representative quad from the associated cluster. In the following, the representative quads are denoted by *S-quads*.

Simply taking the mean shape of quads in a cluster as the shape of an S-quad and replacing the surface quads by their respective S-quads will break the surface, because adjacent quads will no longer agree on their common edges, see Figure 4. Clearly, the geometry of S-quads cannot be determined merely from the clusters alone. The S-quad shapes must respect also the spatial constraints among their instances over the tiled surface to guarantee their proper connectivity. In addition, one also has to ensure the shape consistency of the instances so that they have compatible lengths in diagonals and edges, as well as consistent signed volume occupied by the quad instances. Since we consider non-planar quads, the vertex arrangement of quads can affect the shape uniqueness in addition to the lengths. Lastly, one also needs to ensure that the surface tiled with the quads instances approximates the input surface.

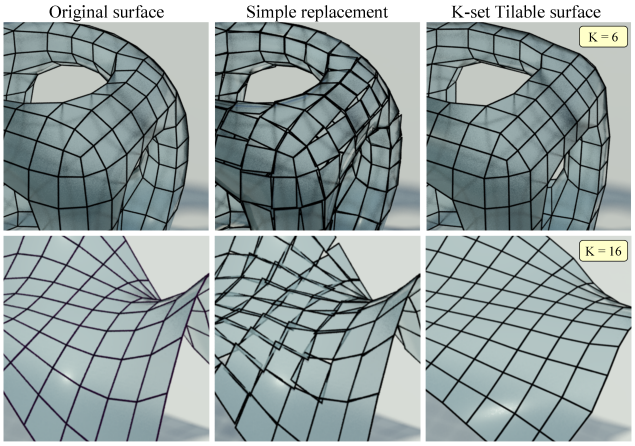


Figure 4: *Left: original surface; middle: simply replacing surface quads with S-quads; right: optimized S-quads tiled on KT-surface.*

Our optimization model solves for the vertex coordinates with various constraints that respect the edge connectivity and shape consistency among the S-quads instances, as well as the surface approximation. Consequently, we can iteratively optimize the input surfaces into K-set tilable surfaces.

2 Related works

Discrete differential geometry (DDG) [Bobenko and Suris 2008] is an emerging research area where differential geometry interacts with discrete geometry. With the discrete equivalences of the geometric notions and methods in classical differential geometry, recently researchers have explored the potential of DDG for computer graphics modeling and applications. DDG allows formulating and optimizing various geometric properties in a discrete manner. Some recent examples are the construction of various models [Mitani and Suzuki 2004; Massarwi et al. 2007; Shatz et al. 2006; Mori and Igarashi 2007; Rose et al. 2007]. Another example where a model is analyzed and then optimized is the work of Gal et al. [2009]. They introduce an editing mechanism that analyzes and edits man-made models by learning the inter-relation among the model parts. Our optimization tends to capitalize on the inherent symmetry of the given shape [Xu et al. 2009]. In that sense, a related work is the model symmetrization of Mitra et al. [2007].

The optimization method that we present here is applied on quad-meshes. Quad-meshes draw more attention recently since they are attractive for modeling surfaces: the quad elements can be nicely aligned with the principal directions of the surface. The work of Daniels et al. [2008] simplifies quad-meshes. Our work also simplifies quad-meshes, but not in terms of the surface quad count.

One recent CAGD area emerged at the borders between discrete differential geometry and architectural engineering is architectural geometry. This area is primarily driven by the increasing demands in designing and modeling freeform surfaces. Motivated by practical architectural need, Liu et al. [2006] introduced conical meshes, which have planar faces and yet possess offset meshes at constant face-to-face distance from the base mesh. They developed an optimization model to convert quad-meshes into canonical meshes and combined Catmull-Clark subdivision with their approach. Yan et al. [2006] proposed a variational approach to extract general quadric surfaces from mesh surfaces; quadric proxies are progressively inserted (or merged) against an error threshold to improve the surface approximation. Building upon the concept of parallel meshes, Pottmann et al. [2007b] developed methods to optimize meshes with offset properties relevant to architectural modeling. Later, Pottmann et al. [2008] systematically investigated semi-discrete surfaces and presented algorithms to optimize freeform surfaces into developable strips, leading to elegant solutions for surface panelization. Kilian et al. [2008] analyzed developable surfaces with curved creases and applied them to assorted architecture and industrial designs. More recently, Schiftner et al. [2009] introduced circle-packing meshes, where the incircles on neighboring triangle faces touch one another and the induced spheres centered at each vertex also form a packing.

Concurrent to our work, there are two similar pieces of independent research work that share the same general motivation with us. All

aim at optimizing the number of tile shapes required to construct a surface. The approach of Singh and Scott Schaefer [2010] is similar to ours, but their tiles are triangles. They search for a clustering of triangles that can be optimized to approximate the given model. Starting from a single cluster, they keep adding clusters until the approximation is sufficient. The work of Eigensatz et al. [2010] considers molds and panels rather than congruent tiles. Their focus is on the reusability of molds in fabricating panels for forming globally coherent surfaces.

3 Clustering

The purpose of the clustering phase is to group quads with similar shapes and define an initial set of representative S-quads. Then the shapes of these S-quads are optimized, so instances of them can then tile the KT-surface. First, we define the metric for computing shape dissimilarity between two quads p and q, whose vertices are p_i and q_i, respectively:

$$s(p,q) \;=\; \min_j \; \Sigma_i ||q_i - T_j(p_i)|| \;\; \text{for} \;\; i \in [1,4]\,,$$

where T_j is a rigid body transformation involving only arbitrary translation and rotation. This definition searches for the best possible registration between the two quads among the eight different ways to correspond vertices from the two quads. Subsequently, we can construct an affinity matrix of quads, see Figure 5.

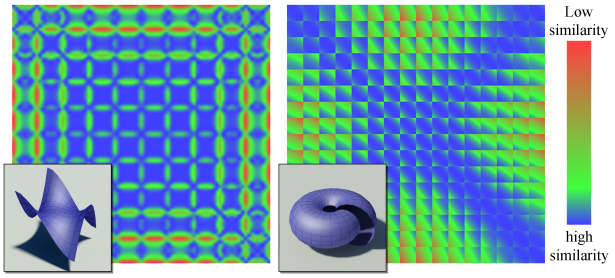


Figure 5: *Affinity matrices (linearly color-mapped) for* MONKEY SADDLE *(left) and* SEASHELL *(right) surfaces.*

Initial Clustering. The clusters are formed by a series of cluster merge, where at each step two clusters are merged until we are left with exactly K clusters. We start off with a large number of small clusters created simply by grouping together all quads with strong similarity defined by a prescribed threshold. The result of this step yields an initial set of N clusters. Once a set of clusters, or say a *clustering configuration*, is formed, the mean shape of each cluster defines an initial S-quad. The S-quad instances are now placed over the surface to replace the original quads. Since the quads have eight possible orientations, each S-quad instance is tagged with the orientation that maximizes its similarity with the quad tile it replaces.

To continue the discussion, the *weight* of a cluster C_i is defined as the total sum of all pairwise dissimilarity among its quads, and is denoted as $S(C_i)$. The cost of merging two clusters C_i and C_j is $S(C_i \cup C_j) - S(C_i) - S(C_j)$ and the weight of a clustering configuration is the sum of weight of the individual cluster inside.

However, the clustering configuration with the lowest weight is not necessarily the one that will incur the least constraints on the consequent KT-surface. The *quality* of a configuration is defined also by the degree of *flexibility* it imposes on the KT-surface. The "flexibility" is learnt by analyzing the relations between the clusters and the quad-mesh, and the degree of freedom the S-quads definition imposes on the mesh edges. For that we next analyze the edge connections and flexibility.

Analyzing the edge connections. Recall that the S-quads are templates of quads and their instances over the resultant KT-surface must have matched length along their shared edges. We thus examine the edge connections between S-quads over all edges on the input surface and formulate constraints for the optimization of S-quad shapes. To this end, constraints of individual edges are aggregated to form the *edge connection constraints* with the assistance of an *edge sharing graph*. Basically, it is an undirected graph, say $G = (V, E)$, where each vertex in V corresponds to one of the edges of the k S-quads, i.e., $|V| = 4k$. Two vertices in G are connected if their associated S-quad edges (for some S-quad instances) share an edge on the given surface.

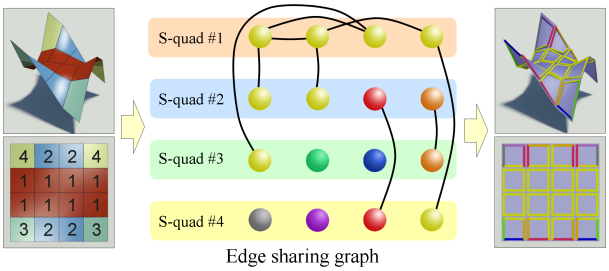


Figure 6: *Given four clusters of quads on* MONKEY SADDLE *(left), we analyze the edge sharing conditions for S-quad instances by an edge sharing graph (middle). Edges marked with the same color should have matched edge length on the KT-surface (right).*

Taking the 4-by-4 MONKEY SADDLE surface in Figure 6 as an example. The connections (edge sharing) between instances of S-quads on the surface define the connections between related vertices in the edge sharing graph. Since the same S-quad may have more than one instance over the surface, instances of the same S-quad may sometimes be placed next to one another, and hence, some vertices in G may connect to vertices of the same S-quad. However, we can ignore edges that loop back to the same vertex in G since loops do not pose any constraint.

Based on the edge sharing graph, we define the edge connection constraint by searching over connected components in the edge sharing graph. Each disjoint subgraph, i.e., a connected component, in the graph is a group of S-quad edges that should have compatible edge length on the resultant KT-surface.

Following the example shown in Figure 6, the edge sharing graph consists of seven disjoint subgraphs, and hence there are seven distinct edge lengths in the KT-surface. In Figure 6, the graph vertices belonging to the same subgraph are labeled with the same color and these colors are applied to visualize the edge connection constraints over the surface. During the optimization, edges of the same color are constrained to have compatible length.

The larger the number of disjoint subgraphs, the higher the number of distinct edge lengths exist on the KT-surface. This number defines the degree of freedom in modeling a KT-surface. In other words, the number indicates the *flexibility* of a KT-surface. Now, the degree of flexibility can be directly determined by the clustering configuration; it serves as a useful indicator on the quality of the clustering, and is used in searching for the best configuration. In practice, we found that for some clustering configurations with a relatively large number of clusters, their degrees of flexibility may still be low due to poor arrangement of S-quads.

Cluster Merge. Recall that the clusters are formed by a series of merge operations. Each series of such binary merges yields a clustering configuration that is associated not only with cost, but also

with a degree of flexibility learnt from its edge sharing graph. Ideally, we would like to exhaustively test all possible series, and learn which clustering configuration yields the best KT-surface, but since our problem is intrinsically similar to the NP-hard optimization problem *Minimum K-clustering Sum* [Ausiello et al. 1999], this, however, is not computationally feasible and we employ the following heuristics: We sort all the initial N clusters by their weights $S(C_1) \leq S(C_2) \leq S(C_3) \leq ... \leq S(C_N)$. Then, we subsequently pick two clusters from the front of the list and compute the merging cost as well as the change in degree of flexibility. Meanwhile, we keep track of 10 candidate configurations[1] and the lowest cost among them serves as an upper bound for the merging cost. Since the weight of the merged cluster is always larger than the sum of weights of the two individual clusters being merged, this upper bound is used for an early termination of an otherwise exhaustive search. In addition, we penalize the merging cost (with a user-specified parameter) if the merge results in severe reduction in the degree of flexibility. Notice also that upon each binary merge, we can amend the edge sharing graph rather than re-building it in order to learn the degree of flexibility. In the spirit of genetic algorithms for searching a very large space, the resultant ten candidate configurations are later used as the seed configurations to generate the next ten candidates when we further reduce K. The different series of binary merges yield candidate clustering configurations, or in fact, sets of S-quads.

4 Optimization

The KT-surface geometry is generated by optimizing the $\mathbf{v}_{i,j}$ vertices positions, under the constraints defined by the edge sharing graph. The vertices positions aim to remain closer to the original surface, expressed by the F_d term, which measures the distances between the corresponding vertices of KT-surface and input surface:

$$F_d := \sum_{i,j} \|\mathbf{v}_{i,j} - \mathbf{v}^0_{i,j}\|^2,$$

where $\mathbf{v}^0_{i,j}$ is the original position of vertex $\mathbf{v}_{i,j}$.

To preserve the surface smoothness, we employ the fairness term F_f and apply it to vertices that are not on sharp features, such as corners and crest lines. Based on [Liu et al. 2006], we use the following fairness term on interior vertices:

$$F_f := \sum_{i,j} \text{fair}(\mathbf{v}_{i,j})$$

$$\text{fair}(\mathbf{v}_{i,j}) := \begin{cases} \|\mathbf{v}_{i,j} - \frac{1}{m}\sum^m (\text{neighbors of } \mathbf{v}_{i,j})\|^2 & \text{if } m \neq 4 \\ \|\mathbf{v}_{i+1,j} - 2\mathbf{v}_{i,j} + \mathbf{v}_{i-1,j}\|^2 & \\ \quad + \|\mathbf{v}_{i,j+1} - 2\mathbf{v}_{i,j} + \mathbf{v}_{i,j-1}\|^2 & \text{if } m = 4, \end{cases}$$

where m is the valence of $\mathbf{v}_{i,j}$. For vertices along the surface boundaries (excluding corner points on boundary), we use:

$$\text{fair}(\mathbf{v}_{i,j}) := \sum_{i,j} \|\mathbf{v}_{i+1,j} - 2\mathbf{v}_{i,j} + \mathbf{v}_{i-1,j}\|^2 ,$$

where $\mathbf{v}_{i+1,j}$ and $\mathbf{v}_{i-1,j}$ are neighbors of $\mathbf{v}_{i,j}$ along the boundary. Note that the fairness term in Liu's method is a discrete bending energy. Minimizing this energy is equivalent to minimizing the sum of squared principal curvatures, i.e., $k_1^2 + k_2^2$, while minimizing the squared Laplacian is equivalent to minimizing the square of mean curvature, i.e., $(k_1 + k_2)^2$. These terms are close, but we found the bending energy more effective for vertices of valence four, which predominate the quad-meshes that were experimented with.

[1]The number of candidate configurations is controllable as a parameter, and a larger number, such as 50, is needed for surfaces with more quads.

For each subgraph, say $G_s \in G$, we extract all its associated edges on the surface: e_0, e_1, ..., e_{n-1}, and apply the following edge connection term to enforce compatible edge lengths, so that we can connect instances of S-quads on the KT-surface:

$$F_e := \sum_{G_s} \sum_{i=0}^{n-1} \Big(||e_i||^2 - ||e_{i'}||^2 \Big)^2, \text{ where } i' = (i+1) \text{ modulus } n .$$

In addition to edge lengths, we further need to constrain the diagonal lengths to enforce compatible shape for instances of S-quads. Given q_0, q_1, ..., q_{n-1} as the instances of a given S-quad, we define $ac(q_i)$ and $bd(q_i)$ as the two diagonals of q_i, corresponding to the diagonals of the S-quad. The following diagonal term enforces compatible diagonal lengths among the instances:

$$F_a := \sum_{G_s} \sum_{i=0}^{n-1} \Big[\big(||ac(q_i)||^2 - ||ac(q_{i'})||^2 \big)^2 + \big(||bd(q_i)||^2 - ||bd(q_{i'})||^2 \big)^2 \Big] .$$

Using the diagonal term alone is insufficient to guarantee compatible shape among the instances of a given S-quad. Since we consider non-planar quads, which act like tetrahedrons in 3-space, instances in the same cluster could be optimized to be mirror of each other, but yet still having compatible edge and diagonal lengths. Since mirror reflection is not rigid body transform, we cannot tile them with the same S-quad. Hence, the following orientation term is introduced to enforce compatible orientation for instances in the same cluster. This is expressed by the signed volume (determinant) of 4-vertices in 3-space:

$$F_o := \sum_{G_s} \sum_{i=0}^{n-1} \big(V(q_i) - V(q_{i'}) \big)^2 ,$$

where $V(q_i)$ denotes the signed volume of q_i.

The following objective function summarizes all the above terms:

$$\min f := \mu_1 F_d + \mu_2 F_f + \mu_3 F_e + \mu_4 F_a + \mu_5 F_o ,$$

where μ_i's are user-specified weighs. Since we regard F_e, F_a, and F_o as hard constraints whereas F_d and F_f as soft constraints, we set in our current implementation $\mu_3 = \mu_4 = \mu_5 = 1$, and the typical values of μ_1 and μ_2 to be 0.0001. Compared to the edge/diagonal/orientation constraints, the fairness and distance contribute very little to the objective function, and therefore, these terms can hardly converge to zero. We use the Conjugate Gradient method (which implements the Polak-Ribiere minimization) from the book *Numerical Recipe* to solve the above unconstrained nonlinear programming.

5 Implementation and Results

Clustering Enhancement. Recall that *Minimum K-clustering Sum* is an NP-hard optimization problem [Ausiello et al. 1999], we use a number of heuristics to further improve and control the clustering quality: First, we allow interactively merging and breaking clusters. After each update, our system can amend the edge sharing graph and instantly feedback the degree of flexibility and edge connection constraints. Two examples, which were edited, are TOWER ($K = 9$) and SEASHELL ($K = 21$): the approximation errors of their resultant KT-surfaces are 1.43 and 3.55, respectively; after re-arranging the clusters, the approximation errors are reduced to 1.37 and 3.34, respectively, see Table 1. Another way is to enlarge the size of candidate set in hierarchical clustering. The larger the size is, the closer to an exhaustive search the process will be. In addition, we also initiate an auto-merge that examines the cluster pattern, i.e., the four subgraphs that the associated S-quad edges belong to. If two clusters have exactly the same pattern, they can be

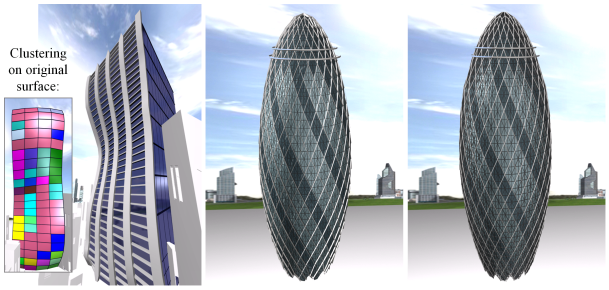


Figure 7: *Left: K-set* DE BEERS GINZA *($K = 27$); middle: original surface of* SWISS RE TOWER*; right: K-set* SWISS RE TOWER *($K = 21$). There is only a little visual degradation for smaller K.*

merged without reducing the degree of flexibility. This can quickly reduce K, particularly in symmetric surface regions. Lastly, we find it useful to add also a clustering configuration defined by a standard K-means clustering to enrich the candidate pool because hierarchical clustering may be locked in a local minimum.

Results. From Figure 8 and Table 1, the general observation is that the larger the K is, the better the KT-surface approximates the original input is. When K decreases, we gradually lose flexibility to retain geometric details in the KT-surface. Table 1 summarizes the KT-surface properties; the columns (from left to right) are: K, degree of flexibility, the terms in the objective function, and the approximation error. For fair comparison, we uniformly scale the surfaces into a unit cube and the terms in the objective function are normalized by the number of vertices, edges, or faces, accordingly. Two data models are particularly interesting. When K decreases, the opening in the SEASHELL gradually shrinks, appearing like a symmetrization process. For the MONKEY SADDLE, we gradually lose the flexibility to represent the frontal curly part as it has mirrored left and right halves and its quads are highly curved, thus limiting the clustering choices due to the orientation constraint, see below for summarized symmetry conditions:

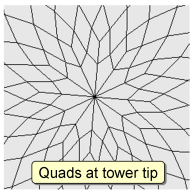
- TUNNEL: multiple rotation symmetry about its medial axe and (one global and two local) reflection symmetry;
- TOWER: multiple rotation symmetry about its medial axe and approximate reflection symmetry about its middle;
- DECOCUBE: multiple mirror-reflection / rotational symmetry;
- MONKEY SADDLE: mirror-reflection (left and right) and rotational symmetry (front and back);
- SEASHELL: one mirror-reflection symmetry;
- DE BEERS GINZA: no symmetry.

Surface	K	D.O.F.	$\widetilde{F}_d$	$\widetilde{F}_f$	$\widetilde{F}_e$	$\widetilde{F}_a$	$\widetilde{F}_o$	Approx. Error
Tower	24	40	5.63E-04	2.43E-02	3.44E-08	1.37E-10	2.23E-07	0.05
	18	29	4.33E-03	3.94E-02	6.41E-07	3.39E-08	3.95E-05	0.36
	9	9	1.67E-02	5.61E-02	6.28E-06	1.97E-07	2.17E-04	1.37
Tunnel	12	35	1.78E-01	5.48E+00	2.63E-04	1.30E-06	1.88E-05	1.10
	10	29	9.35E-01	7.90E+00	1.37E-03	2.66E-06	4.35E-05	5.75
	8	22	2.11E+00	1.17E+01	1.84E-03	5.70E-06	5.56E-05	12.97
Sea Shell	96	156	3.91E-03	9.78E-03	2.84E-09	4.46E-09	1.99E-08	0.89
	60	71	7.03E-03	5.95E-03	2.30E-08	1.93E-08	8.96E-08	1.35
	44	52	8.16E-03	1.12E-02	3.15E-08	2.01E-08	9.81E-08	1.86
	21	27	1.47E-02	1.81E-02	5.66E-09	5.43E-09	1.79E-08	3.34
Decocube	15	27	5.52E-04	2.31E-02	6.73E-09	8.37E-12	6.74E-08	0.26
	9	13	1.22E-03	2.51E-02	8.98E-09	4.24E-11	1.75E-07	0.58
	3	5	7.13E-03	3.42E-02	7.13E-08	3.34E-10	8.17E-07	3.38
	1	3	1.85E-02	4.87E-02	3.27E-15	9.34E-14	3.54E-13	8.77
Monkey Saddle	34	67	1.99E-04	5.55E-03	9.62E-09	3.92E-08	1.09E-06	0.02
	28	47	4.47E-04	5.50E-03	1.75E-08	9.97E-08	1.04E-06	0.05
	22	27	1.05E-03	6.83E-03	9.10E-08	3.22E-07	2.91E-06	0.13
	16	19	3.58E-02	1.15E-03	3.39E-08	1.32E-07	9.06E-07	4.33
DeBeers Ginza	48	73	5.35E-04	3.56E-02	1.82E-06	4.23E-10	8.41E-08	0.00
	44	65	8.15E-04	3.56E-02	2.16E-06	8.83E-10	1.65E-07	0.01
	39	60	2.76E-03	4.20E-02	1.34E-07	8.85E-11	3.20E-09	0.03
	34	51	4.52E-03	4.94E-02	1.99E-07	1.88E-10	3.66E-09	0.04
	30	44	1.03E-02	5.79E-02	4.13E-07	1.77E-10	9.50E-08	0.09
	27	40	7.08E-02	4.47E-03	9.65E-07	1.25E-09	2.52E-06	0.65

Table 1: *Optimization results on surfaces shown in Figure 8.*

For TOWER, we found from our experiment that both hierarchical and k-mean clustering can exploit approximate symmetry about its horizontal plane in the middle, see the clustering pattern shown in Figure 8. See also Figure 7 for the demonstration of KT-surfaces as the building exteriors of DE BEERS GINZA and TOWER (that mimicks Swiss Re Tower).

Limitations. (i) The input surface is assumed to be noiseless so that the similarity metric can provide good initial clustering. (ii) Our optimization does not scale well. Currently we are limited to surfaces of less than 1500 quads. A possible solution is to use quad simplifications [Daniels et al. 2008], and another is to first segment the surface and optimize it in parts. (iii) Constraining the quads to be planar is a different and harder problem, and it was not considered in this current work. (iv) For many models, we may not be successful in reducing K to small number. Like in Figure 8, TOWER gets severe distortions already with $K = 9$ tiles. If we push it further down to $K = 2$, the TOWER degenerates and collapses due to the non-uniform arrangement of quads around the tower tip. Also when the quads have irregular shapes, like in the TOWER, it becomes too ambiguous to cluster them. However, for some models we succeeded to go down to $K = 1$, e.g., the DECOCUBE becomes cube-like while the MONKEY SADDLE is completely flattened.



6 Conclusion

This paper presents a challenging problem in freeform surface modeling, namely the K-set tilable surface. It allows us to closely approximate an input surface with instances of a small set of K quads. This solution may contribute to an effective cost reduction in the physical assembly of the given surface. In detail, we introduce the edge sharing graph structure to analyze the edge connection constraints and also to learn the degree of flexibility from the clustering. We further formulate constraints according to the S-quad tilability and the surface approximation, and devise a non-linear optimization model to progressively iterate the vertices positions to achieve the construction of KT-surfaces. In the paper, we push K to be particularly small. However, we believe that the advantage of the technique is for designing surfaces with rather moderate K. We also consider extending this technique to home-made assembly of toys for recreational purposes.

Acknowledgements. We thank anonymous reviewers for the constructive comments and Lim Tze Yuen of Game Lab. for helping to create the architectural scenes. This work was supported in part by the Singapore A*Star PSF SERC grant (SERC grant no. 092 101 0063), NTU Startup Grant (M58020007.500000), Israel Science Foundation founded by the Israel Academy of Sciences and Humanities, and NRF IDM grant NRF2008IDM-IDM-004-006.

References

AUSIELLO, G., CRESCENZI, P., GAMBOSI, G., KANN, V., MARCHETTI-SPACCAMELA, A., AND PROTASI, M. 1999. *Complexity and Approximation*. Springer. ISBN 3-540-65431-3.

BLANC, A., MCEVOY, M., AND PLANK, R., Eds. 1993. *Architecture and construction in steel*. Taylor & Francis. 640 pages, ISBN 978-0419176602.

BOBENKO, A. I., AND SURIS, Y. B. 2008. *Discrete Differential Geometry*. American Math. Society. ISBN 978-0821847008.

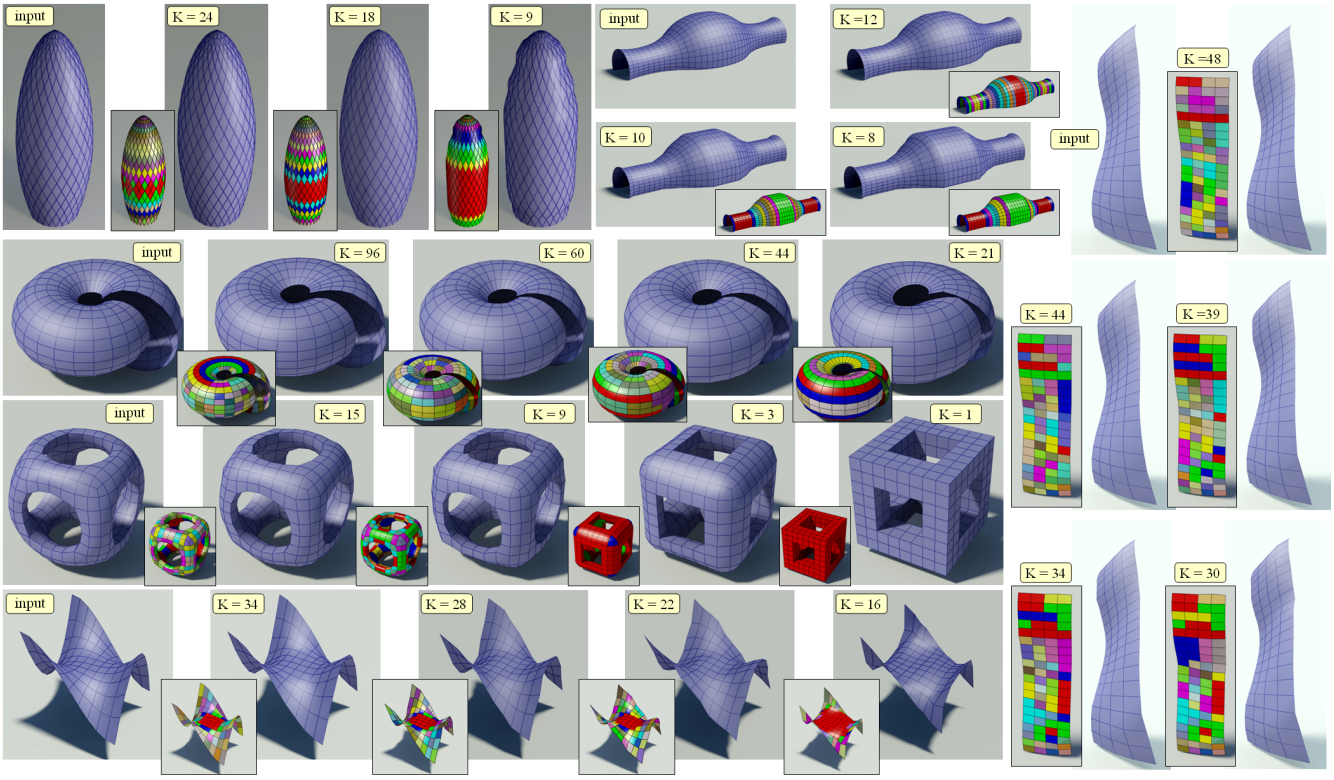


Figure 8: *Resultant K-set tilable surfaces (from top to bottom):* SWISS RE TOWER *(528 quads);* TUNNEL *(1152 quads);* SEASHELL *(400 quads);* DECOCUBE *(480 quads);* MONKEY SADDLE *(100 quads); and* DE BEERS GINZA *(80 quads).*

DANIELS, J., SILVA, C. T., SHEPHERD, J., AND COHEN, E. 2008. Quadrilateral mesh simplification. *ACM Trans. Graph. (SIGGRAPH ASIA 2008) 27*, 5.

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N., POTTMANN, H., AND PAULY, M. 2010. Paneling architectural freeform surfaces. *ACM Tran. Graphics (Proc. of SIGGRAPH) 29*, 3.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Trans. Graph. (SIGGRAPH 2009) 28*, 3.

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N. J., SHEFFER, A., AND POTTMANN, H. 2008. Developable surfaces with curved creases. In *Advances in Architectural Geometry*, 33–36.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. (SIGGRAPH 2006) 25*, 3, 681–689.

MASSARWI, F., GOTSMAN, C., AND ELBER, G. 2007. Papercraft models using generalized cylinders. In *5th Pacific Conf. on Comp. Graphics and App.*, 148–157.

MITANI, J., AND SUZUKI, H. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Tran. Graphics (Proc. of SIGGRAPH) 23*, 3, 259–263.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2007. Symmetrization. *ACM Tran. on Graphics (SIGGRAPH 2007) 26*, 3.

MORI, Y., AND IGARASHI, T. 2007. Plushie: an interactive design system for plush toys. *ACM Tran. Graphics (Proc. SIGGRAPH) 26*, 3.

POTTMANN, H., ASPERL, A., HOFER, M., AND KILIAN, A. 2007. *Architectural Geometry*. Bentley Institute Press. 724 pages, ISBN 978-1-934493-04-5.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Tran. Graphics (SIGGRAPH 2007) 26*, 3.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM Tran. Graphics (SIGGRAPH 2008) 27*, 3.

ROSE, K., SHEFFER, A., WITHER, J., CANI, M.-P., AND THIBERT, B. 2007. Developable surfaces from arbitrary sketched boundaries. In *Symposium on Geometry Processing (SGP)*, 163–172.

SCHIFTNER, A., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2009. Packing circles and spheres on surfaces. *ACM Trans. Graph. (SIGGRAPH ASIA 2009) 28*, 5.

SHATZ, I., TAL, A., AND LEIFMAN, G. 2006. Paper craft models from meshes. *Visual Computer 22*, 9, 825–834.

SINGH, M., AND SCHAEFER, S. 2010. Triangle surfaces with discrete equivalence classes. *ACM Tran. Graphics (Proc. of SIGGRAPH) 29*, 3.

XU, K., ZHANG, H., TAGLIASACCHI, A., LIU, L., LI, G., MENG, M., AND XIONG, Y. 2009. Partial intrinsic reflectional symmetry of 3D shapes. *ACM Trans. Graph. (SIGGRAPH ASIA 2009) 28*, 5.

YAN, D.-M., LIU, Y., AND WANG, W. 2006. Quadric surface extraction by variational shape approximation. In *Geometric Modeling and Processing*, Springer Berlin / Heidelberg, 73–86.