Journal for Geometry and Graphics
Volume 19 (2015), No. 1, 107–118.

Mesh Approximation for Generating Development with Creases and Slits

Kuniko Takahashi[1], **Tomohiro Tachi**[2], **Yasushi Yamaguchi**[3]

[1] *The University of Tokyo*
3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
email: kunikot@me.com

[2] *The University of Tokyo / JST PRESTO*
3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
email: tachi@idea.c.u-tokyo.ac.jp

[3] *The University of Tokyo / JST CREST*
3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
email: yama@graco.c.u-tokyo.ac.jp

Abstract. The creation of papercrafts from 3D mesh models is useful not only for entertainment but also in engineering applications. Many studies have been conducted on approximating a mesh model using sheets of paper based on the developability of surfaces. In these studies, one of the objectives was to make the shape of each part simple in order to maintain its manufacturability. However, this approach may increase the number of pieces necessary for building an accurate model. In this paper, we propose a novel method to approximate a triangular mesh model with a small number of parts. The method achieves high accuracy with few parts by introducing creases and slits within each part.

Key Words: papercraft, development, slits and creases.

MSC 2010: 68U05, 65D17, 68W25, 52B10

1. Introduction

Digital processing and visualization of 3D shapes using 3D computer graphics techniques have become popular. The 3D mesh model is one of the most popular representations used to describe 3D shapes. A 3D mesh uses a very considerable amount of small triangles to represent a smooth surface.

Papercraft, a hobby enjoyed by many people, is a good method to easily obtain a physical model of a digitally modeled 3D shape. It is useful not only for evaluating the shape using a physical model but also for realizing the final product, e.g., an architectural structure or

ISSN 1433-8157/$ 2.50

a machine part, using metal sheets instead of paper. Therefore, a method for reproducing a mesh model using papercraft will provide general and practical solutions in different fields. For this purpose, it is necessary to develop the shape onto a plane and to obtain its development. Since the shapes in a papercraft model are limited to those that are developable because of the non-stretching nature of paper, studies to approximate a mesh model under the developability constraint have been conducted. Most of these studies adopt the approach of separating the mesh model into small regions, called charts, each of which is then approximated by a developable surface. The target shape needs to be approximated well by the charts and realized in an easy-to-glue manner. In previous methods, the ease of gluing was defined by the number of charts and the simplicity of the chart boundaries, i.e., their closeness to a circle and their smoothness.

In this paper, we propose an approximation method, the objective of which is to reduce the number of charts by introducing slits and creases into them. To achieve the method, we use the angle constraint around vertices so that the total angle around a developable vertex must be 2π, instead of approximating each chart by using a developable surface, as in previous studies.

2. Developable shapes

In the fabrication of a 3D figure out of paper, the critical condition is its *developability*. The developability is the property of a shape that it can be developed onto a plane without stretching. In this research, we call such a shape a *developable shape*. Developable shapes include, but are not necessarily limited to, developable surfaces, such as a cylinder, cone, tangent surface, and plane. For example, a developable shape can consist of multiple developable surface patches joined by creases.

The general approach for making the development of a given figure is first to segment the figure into charts and then to develop each chart onto a plane. In order to fabricate the figure using paper, each chart must constitute a developable shape, which may be a single developable surface or consist of various kinds of developable surfaces. Any shapes can be separated into various kinds of surfaces, but it is quite rare that they consist only of developable surfaces. Therefore, it is normally necessary to approximate a chart by using a developable shape. Although any triangular mesh model consists of developable surfaces, i.e, triangles, it is not feasible that a mesh model is developed as a collection of original triangles, because the number of triangles tends to be huge.

Hoppe [1] and Garland et al. [2] introduced methods of surface simplification. In these methods, the triangles that compose a mesh model are combined, while the shape features of the original model are preserved. These methods are able to reduce the number of triangles and render the surface development feasible. However, the resulting shape appears no smoother than the original and does not approximate it precisely. Other methods for approximating a mesh with developable surfaces such as cones or cylinders were proposed by Julius et al. [3], Shatz et al. [4], and Massarwi et al. [5].

D-Charts, introduced by Julius et al. [3], is a method that segments a mesh into quasi-developable charts. However, these charts are not absolute developable surfaces, and therefore, it is impossible to develop them onto a plane without stretching. Shatz et al. [4] introduced segmentation of a mesh model, as well as an approximation by developable surfaces, so that each chart can be unfolded onto a plane without stretching. However, this method requires that the charts' boundaries be refined, because approximation by developable surfaces may result in non-intersection between neighboring charts. Therefore, after the mesh model approximation, chart boundaries are refined such that they intersect each other. This method may generate non-uniformly small charts, which makes it difficult to glue charts into a papercraft model.

Mitani et al. [6] introduced a method that produces strip-based approximation segmentation of a mesh model. This method succeeds in generating a relatively smooth papercraft model. However, strip-based segmentations are long and usually need more charts to achieve better approximation, and therefore, may be difficult to cut and glue.

3. Proposed method

In general, a papercraft model consists of charts, each of which can be unfolded onto a plane. We require that the papercraft model accurately approximate the original mesh model, while being easy to assemble by gluing the charts together. Over the last decade, the objectives of most studies were both to reduce the number of the charts and to simplify their shapes, because the authors claimed that the ease of cutting and gluing depends on both the number and simplicity of the charts. However, our method focuses only on reducing the number of charts for two reasons. The first is that recently available cutting plotters can automatically cut complicated charts. The second reason is that the ease of gluing is expected to depend more on the number of charts and less on the total length of the boundary edges. This is because the correspondence of edges to be glued within the same chart is trivial; a pair of edges can be glued together one by one along the slit.

Now, we consider origami as an example of papercraft modeling. In origami, a complicated shape can be generated by folding a sheet of paper. In other words, origami obtains flexibility of expression by introducing creases into the paper. We propose a segmentation and approximation method that introduces slits and creases into charts. By using creases, different developable surfaces can be combined into one chart. In addition, by using slits, it is expected that the accuracy of the approximation improves, and that the process of gluing becomes easier, because edge correspondence along a slit is trivial, as stated above. By introducing creases and slits, we expect to reduce the number of charts and make the process of gluing charts easier as compared to that in previous methods.

However, it is difficult to introduce creases and slits into a method based on the developable surfaces, such as cones and cylinders, used in previous research. Instead, we use the angle constraint around vertices [7]. This constraint states that the total angle around a

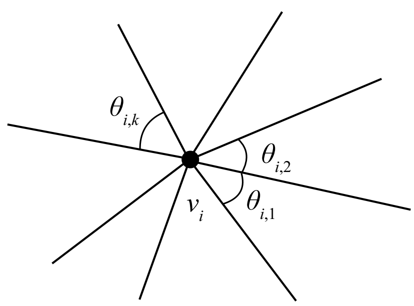


Figure 1: Developability around a vertex.

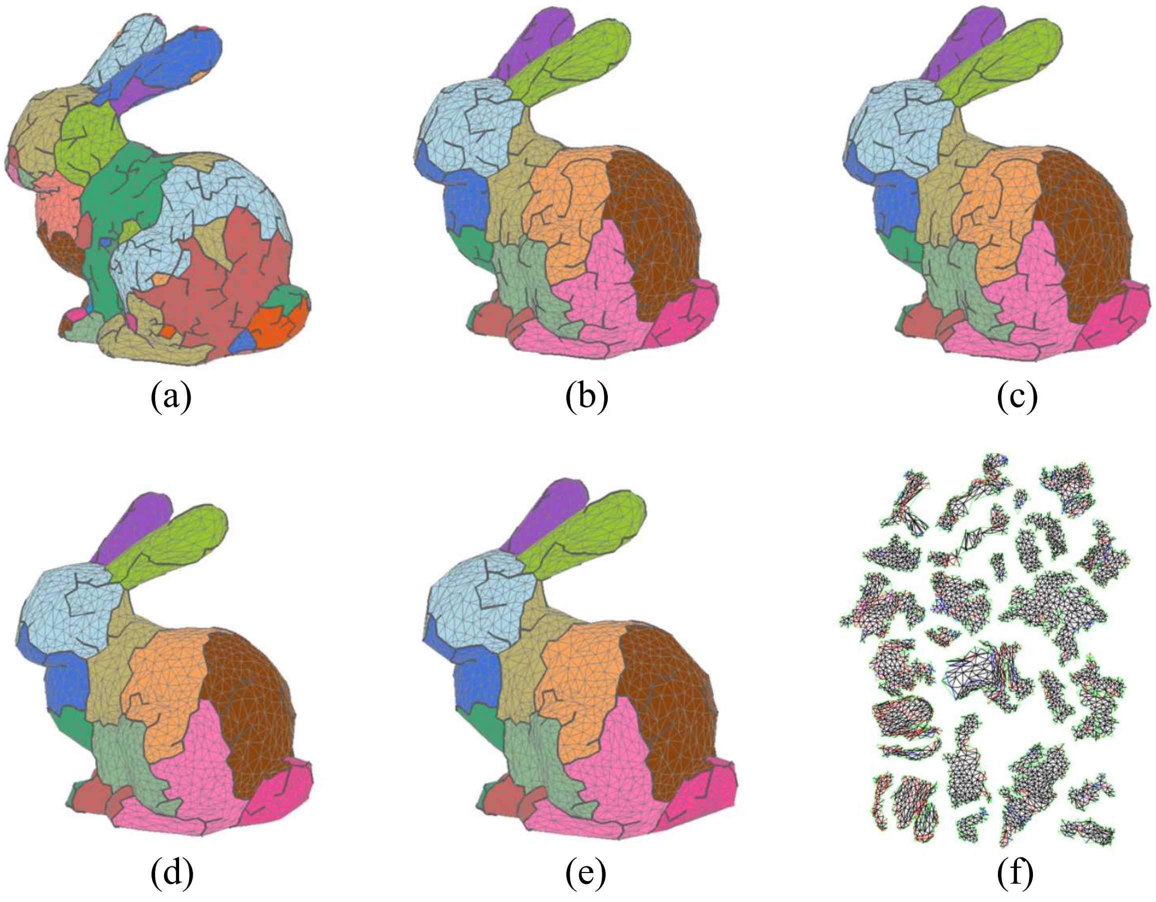


Figure 2: Steps of the proposed method.

developable vertex must be 2π (Figure 1):

$$2\pi - \sum_{k=1}^{m_i} \theta_{i,k} = 0, \tag{1}$$

where $\theta_{i,k}$ is the k-th incident sector angle of vertex v_i with m_i incident facets. This angle constraint allows creases in charts. The constraint is used for chart segmentation, as well as for developable deformation, which will be explained later in this paper. The developability based on the angle constraint can be realized by changing the coordinates of the vertices regardless of the mesh topology. Thus, the connectivity of charts can be achieved easily, which was the challenging issue in the method proposed in [4].

3.1. Feature of the method

Our method achieves the following properties.

1. Each chart may contain creases within itself.
2. Each chart may have slits, but their total length should be as short as possible.
3. The entire mesh model preserves its topology.
4. The user can control the number of charts.

The first property was mentioned previously. The second property is realized by leaving "less developable vertices" on the boundary of charts when segmenting a mesh model into charts. After mesh segmentation into charts, a certain part of a slit is closed like a zipper if the closing does not critically affect the developability of the entire chart. Edges of a slit are judged to be closed by taking into account the slits' length, as well as the angle defects. The third property is achieved by adjusting only the positions of vertices in order that the entire model is developable. Thus, the boundary reconstruction required in [4] is no longer necessary. The last property is achieved by reducing the number of charts. It is accomplished by repeatedly culling the smallest chart so that a given mesh is segmented into evenly-sized charts. Properties 1 through 3 in our approach are novel; their realization has not been attempted in previous research. They may lead to a significant reduction in the number of charts, as well as an improvement in approximation accuracy. The last property, that the user can control the number of charts, is also novel.

3.2. Algorithm

Our method consists of two phases: mesh segmentation (Steps 1–2) and deformation and development (Steps 3–5). In the former phase, a mesh is segmented into charts, each of which approximately satisfies the angle constraint. In the latter phase, all the charts are deformed so that they become absolutely developable by solving the angle constraint. Figure 2 illustrates the entire algorithm of our method.

In the first step of initial segmentation, a mesh model is segmented into tentative charts using the angle constraint around vertices (Figure 2a). This step is explained in Section 3.2.1. In the second step, charts are selected by repeatedly culling the smallest chart so that every chart has almost the same number of triangles (Figure 2b). In Section 3.2.2, this step is described in detail. Third, all the charts are deformed so that they become completely developable by moving vertices (Figure 2c). In Section 3.2.3, this step is shown in detail. Fourth, some parts of slits are closed like zippers if the closing does not critically affect the developability of the entire charts (Figure 2d, e). This step is described in Section 3.2.4. Finally, each chart is developed onto a plane and separated if a self-intersection occurs (Figure 2f). Section 3.2.5 illustrates the details of this step.

3.2.1. Initial segmentation (Step 1)

In this step, first we choose a primary vertex, called a seed. Each chart grows from the seed that has the minimum cost of developability among vertices not yet belonging to any existing chart. The cost is defined by

$$E(v_i) = \left| 2\pi - \sum_{k=1}^{m_i} \theta_{i,k} \right|, \tag{2}$$

which is based on the angle constraint specified in Equation (1). The initial chart consists of all the triangles around the seed (Figure 3a).

Two kinds of vertices belong to a chart: a boundary and an inner vertex. A boundary vertex lies on the boundary of the chart. An inner vertex is not located on the boundary so that it is completely surrounded by the triangles that comprise the chart. Therefore, the seed is the only inner vertex of the initial chart. A chart C is grown by changing one of the boundary vertices into an inner vertex, as shown in Figure 3. The boundary vertex is selected

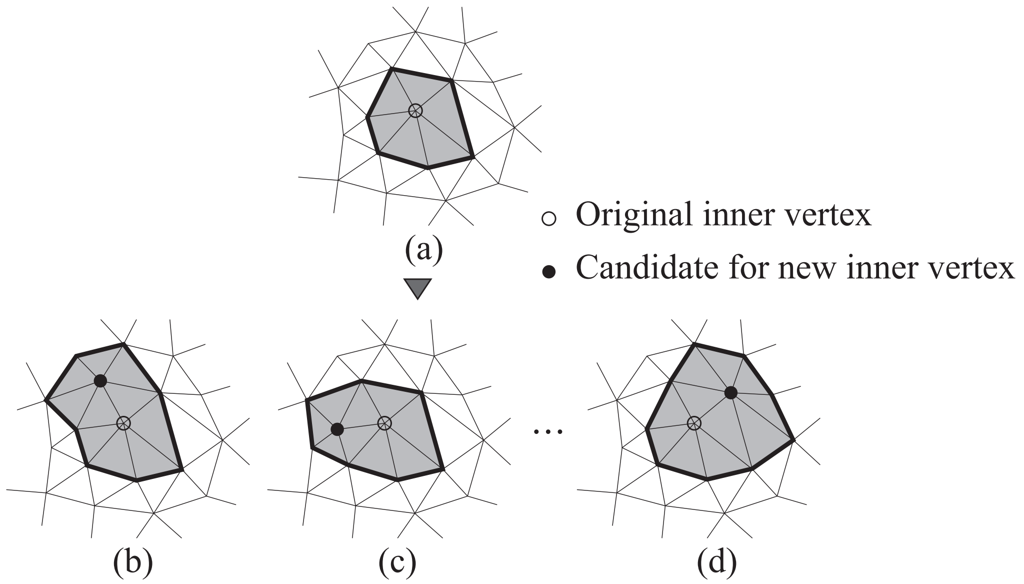


Figure 3: Chart growth by changing one of the boundary vertices into an inner vertex.

by evaluating the cost function (C) of chart C, defined as

$$Cost(C) = \sum_{v_i \in V_I(C)} E(v_i). \tag{3}$$

Here, $V_I(C)$ is a set of the inner vertices of chart C. Figures 3b–d show the candidates for the next charts grown from the initial chart shown in Figure 3a. The chart having the minimum cost among all the candidate charts is selected next. The chart grows until $Cost(C)$ reaches a predefined threshold.

This step naturally produces slits in a chart. Figure 4 illustrates this process of slit emergence and growth. A slit occurs when one of the boundary vertices is surrounded by the triangles of the chart without being an inner vertex, as shown in Figure 4b. This type of boundary vertex surrounded by chart triangles is excluded from the candidates of chart growing. Consequently, a slit is never eliminated, and therefore, naturally grows as in Figure 4c.

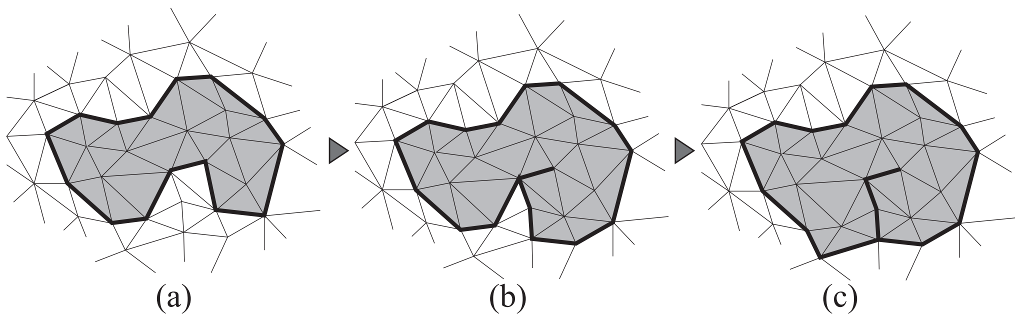


Figure 4: Emergence and growth of a slit.

Seed selection and chart growth are repeated until no more seed candidates exist, i.e., vertices not belonging to any charts. After this step, every vertex of the mesh model belongs to one of the charts. However, there may exist some triangles not belonging to any chart near the chart boundaries. These triangles are incorporated into one of the neighboring charts so that the resulting boundary becomes the smoothest among possible choices. Figure 2a shows an example of initial segmentation.

3.2.2. Chart selection (Step 2)

The charts produced by the initial segmentation step explained in Section 3.2.1 differ widely in size. Charts produced earlier tend to be big, while those produced later tend to be small, because the earlier charts usually grow by taking highly developable vertices. The number of charts in the previous step is unpredictable, as it is determined by the shape of the given mesh model and the threshold value of chart growing. In this step, we reduce the charts to a user-specified number by selecting the seeds produced in the initial segmentation and re-segmenting the mesh model with the selected seeds.

First, initial charts C_i $(i = 1, 2, \ldots, n)$ are generated around the seeds obtained in the initial segmentation. Then, the chart with the lowest cost, as evaluated by Equation (3), i.e., the chart having the highest developability, is selected. The selected chart grows by one vertex, i.e., changes one of its boundary vertices into its inner vertex. By repeating this process, that is, growing a chart of the highest developability, the growing speed of each chart keeps pace with that of the other charts. This process is completed when all vertices on the mesh model belong to n charts.

When the previous process terminates, the seed of the chart with the least number of triangles is removed from the seed set. This means the number of seeds is decremented by one. Based on the new seed set, the mesh model is segmented again, which results in $n-1$ charts. By repeating this process, the number of charts reaches the user-specified number. Figure 2b shows the results of chart selection. As compared with the results of the initial segmentation (Figure 2a), the chart number is significantly decreased and the sizes of the charts are almost uniform.

3.2.3. Deformation for developability (Step 3)

The charts obtained by the chart selection step explained in Section 3.2.2 are nearly, but not perfectly, developable. This step calculates the coordinates of all vertices so that all the charts become developable. This means that all inner vertices exactly satisfy Equation (1).

The coordinates of all vertices are denoted by

$$\mathbf{V} = \begin{bmatrix} x_1 & y_1 & z_1 & x_2 & y_2 & z_2 & \ldots & x_n & y_n & z_n \end{bmatrix}^{\mathrm{T}},$$

where n is the number of vertices of the mesh model. A set of functions corresponding to the inner vertices is defined as

$$\mathbf{F}(\mathbf{V}) = \begin{bmatrix} 2\pi - \sum_{k=1}^{m_1} \theta_{1,k} \\ 2\pi - \sum_{k=1}^{m_2} \theta_{2,k} \\ \vdots \\ 2\pi - \sum_{k=1}^{m_{n_I}} \theta_{n_I,k} \end{bmatrix}, \tag{4}$$

where n_I is the number of the inner vertices of all the charts. Each function represents the difference between 2π and the total angle around the inner vertex, which represents the developability of the vertex. Thus, the charts become developable if $\mathbf{V}$ satisfies the constraint

$$\mathbf{F}(\mathbf{V}) = \mathbf{0}. \tag{5}$$

All the charts of the mesh become developable by solving the nonlinear equation system (5). This system is solved by using the numerical method proposed by TACHI [7], which gives a feasible solution if the given charts are almost developable. It should be noted that this constraint is provided only for inner vertices, and the boundary vertices are not constrained at all. The boundary vertices can adaptively move in order to satisfy the constraint on the inner vertices. The neighboring charts sharing the boundary vertices preserve their connectivity, and thus, the re-evaluation of chart boundaries required in [4] is no longer necessary. Figure 2c depicts a result of the deformation. The curvature of the mesh model is concentrated at the slits or the boundaries.

3.2.4. Slit closing (Step 4)

The chart at this point may have many slits that were produced during the growth of charts in the chart selection step. These slits allow the charts to be highly developable. On the other hand, charts with fewer slits can be glued together more easily. This step closes certain parts of slits like zippers, unless the closing critically affects the chart's developability. The effect of closing a part of a slit is influenced by not only the angle around a vertex but also the slit's length. The cost function representing the developability of a slit is defined as

$$E_s(v_l) = \sum_{i=1}^{l} \left(E(v_i) \sum_{j=i}^{l} \|e_j\| \right), \tag{6}$$

where $E(v_i)$ is the developability cost of vertex v_i defined by Equation (2) and $\|e_j\|$ is the length of edge e_j (Figure 5). Therefore, Equation (6) considers both the residual error of vertex developability and the length of the closing slit. After the slit closing step, the mesh model is no longer perfectly developable. Thus, the deformation explained in Section 3.2.3 is applied to the mesh model once again.

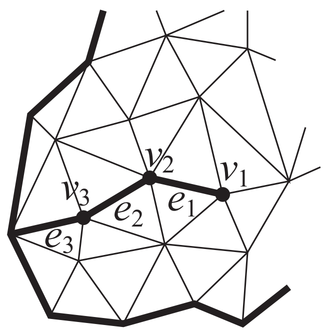


Figure 5: Evaluation of developability along a slit.

3.2.5. Chart development (Step 5)

In the final step of our method, each chart is developed onto a plane. Figure 6 illustrates an example of a developed chart. Dark solid and dotted lines indicate mountain and valley creases, respectively. Edges are detected as creases if their folding angles are sufficiently large. The threshold angle is $\pi/8$ for Figure 6. The development of Figure 6 is a part of *Stanford Bunny*, corresponding to a chart of the *Bunny*'s front leg, shown in dark orange in Figure 2e. This step may result in self-intersection in a development, as indicated by the shaded circle

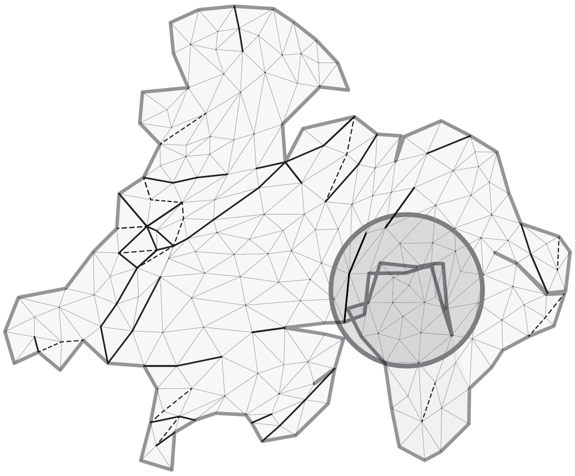


Figure 6: A self-intersecting development.

Figure 7: Separated developments.

in Figure 6. Such developments are separated into several pieces to resolve self-intersection. Figure 7 shows the developments that result when the development is divided, as shown in Figure 6. This separation was done manually.

4. Results

Figure 8 shows sample developments created by the proposed method. They are the developments of *Stanford Bunny* and consist of 27 parts, which is fewer than the 35 parts obtained in SHATZ's method [4]. Figure 9 depicts the resulting papercraft model fabricated with the developments of Figure 8.

Figure 10 shows the effect of chart selection. Figure 10a shows the results when the number of charts is reduced to five. Figures 10b and c show the results with 10 and 20 charts, respectively. When reducing the number of charts, the accuracy of approximation is preserved by increasing the number of slits instead. Thus, the mesh model consisting of a smaller number of charts may lose approximation accuracy after the slit closing. As the number of charts becomes smaller, the sizes of charts become bigger. This may result in self-intersection among charts. As a result, chart selection cannot control the final number of charts.

Figure 11 presents another sample of results generated by the proposed method. Machinery components, such as the *Fan Disk* shown in Figure 11, usually consist of planes and cylinders. It is expected that the number of charts can be reduced by connecting them with creases. The proposed method succeeded in reducing the number of charts of *Fan Disk* to only five.

5. Conclusions and future work

This paper introduced a novel method for achieving developable mesh segmentation and approximation. Our method realizes the following properties.

1. Each chart may contain creases within itself.

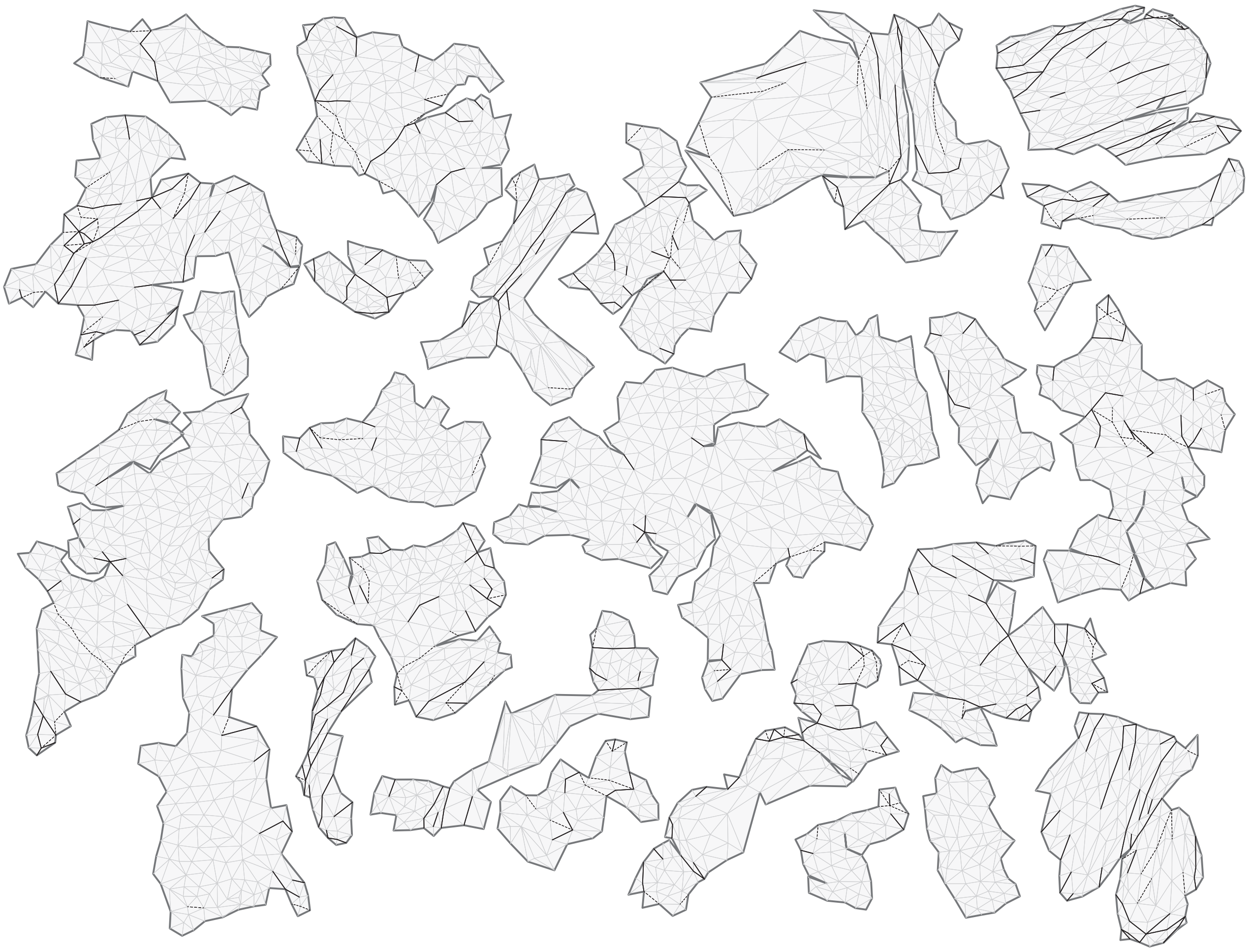
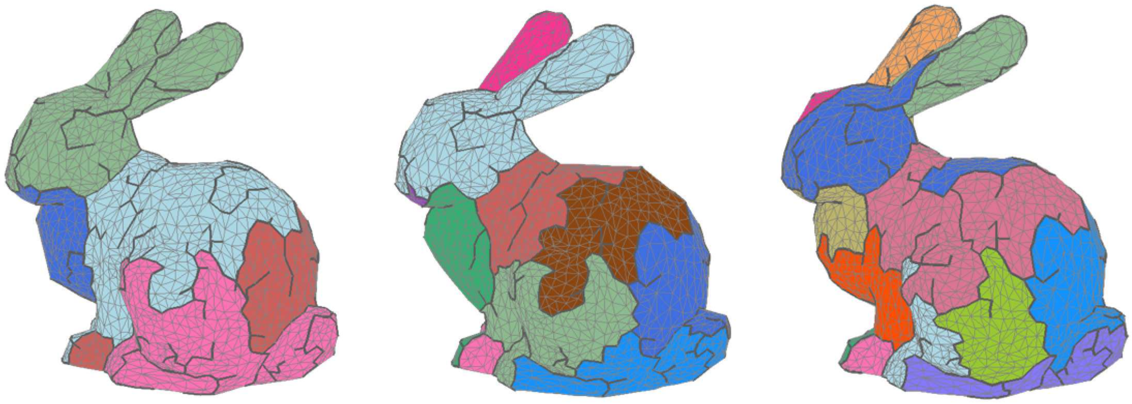


Figure 8: *Stanford Bunny*'s developments generated by the proposed method.

Figure 9: Papercraft model resulting from the developments in Figure 8.



(a) (b) (c)

Figure 10: Effects of the number of charts. (a) Results with 5 charts, (b) results with 10 charts, (c) results with 20 charts.

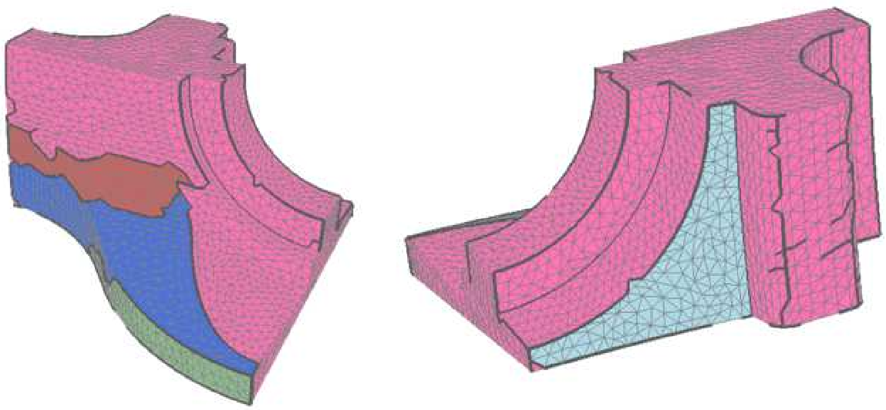


Figure 11: Example charts of *Fan Disk* (5 charts).

2. Each chart may have slits, but their total length should be as short as possible.

3. The entire mesh model preserves its topology.

4. The user can control the number of charts.

Properties 1 and 2 are advantageous for making a papercraft model for the following reasons. First, the original model can be well approximated. Second, the number of charts becomes small. The charts with creases and slits achieve a better approximation with a smaller number of charts than do those with developable surfaces having simple boundaries. Property 3 frees the algorithm from reconstruction of chart boundaries. Property 4 allows the user to control the ease of assembly. However, there exists a trade-off between approximation accuracy and the number of charts. An additional issue is that, as the number of charts decreases, the size of each chart increases, and therefore, finally more self-intersections appear in each chart. In the future, we would like to improve the segmentation step of the proposed method. Currently, after development, some charts may have self-intersections. By improving the segmentation step, we would like to resolve this issue.

Acknowledgements

This work was partially supported by JSPS Grant-in-Aid for Scientific Research (B) (22300030).

References

[1] H. Hoppe: *Progressive meshes.* ACM SIGGRAPH 96 Conference Proceedings, 1996, pp. 99–108.

[2] M. Garland, P.S. Heckbert: *Surface simplification using quadric error metrics.* ACM SIGGRAPH 97 Conference Proceedings, 1997, pp. 209–216.

[3] D. Julius, V. Kraevoy, A. Sheffer: *D-Charts: Quasi-developable mesh segmentation.* Computer Graphics Forum **24**(3), 581–590 (2005).

[4] I. Shatz, A. Tal, G. Leifman: *Papercraft models from meshes.* Visual Computer **22**(9), 825–834 (2006).

[5] F. Massarwi, C. Gotsman, G. Elber: *Papercraft models using generalized cylinders.* Proc. 15th Pacific Conference on Computer Graphics and Applications 2007, pp. 148–157.

[6] J. Mitani, H. Suzuki: *Making papercraft toys from meshes using strip-based approximate unfolding.* ACM Transactions on Graphics **23**(3), 259–263 (2004).

[7] T. Tachi: *Freeform variations of origami.* J. Geometry Graphics **14**(2), 203–215 (2010).

Received August 5, 2014; final form April 20, 2015