布偶外觀縫線導向之網格分割方法

饒 亢* 黃群凱* 陳奕麟† 陳炳宇†
國立臺灣大學
*{mbrsl, chinkyell}@cmlab.csie.ntu.edu.tw †{yilingchenntu, robin}@ntu.edu.tw

ABSTRACT

絨毛布偶是很多人的童年回憶，布偶因為獨特的觸感和可愛的外觀而深受小朋友們的喜愛。然而，布偶的設計並不容易，設計師必須在布料上剪成特定的平面形狀才能將平面的布料縫成想要的立體形狀，而這些平面形狀我們稱之為「版型」。傳統上版型的設計必須經過多次的錯誤嘗試才能完成，整個過程既繁瑣又費時。目前雖然有自動的演算法可以從立體形狀逆推出版型，但這些版型卻沒有考慮到最後成品的外觀，使布偶表面充滿不規則、疤痕似的縫線，進而影響布偶的形象。 本研究取出影響布偶外觀的因素作為權重，使用最短路徑最佳化縫線的路徑，再利用簡單的貪婪法消去不必要的縫線。最後的結果除了確保產生的版型可以縫成想要的立體形狀外，還減少了縫線的不自然感。文末將附上使用此方法縫製的實際成品。

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling;

General Terms

Algorithms

1. INTRODUCTION

In industrial fabrication, approximating a given shape of triangular mesh by planar material is an important topic. Such process can be found from aircraft, furniture, garments, stuffed toys design, etc. To form the target shape, one must cut the material into a number of patterns of specific planar shapes and then assembled them. However, such planar shape does not always exist if the target shape is not developable. To solve the problem, one possible solution is to split the target shape into smaller parts which can be approximated by developable surfaces. However, these applications sometimes require complex shapes that are challenging for human to design the patterns.

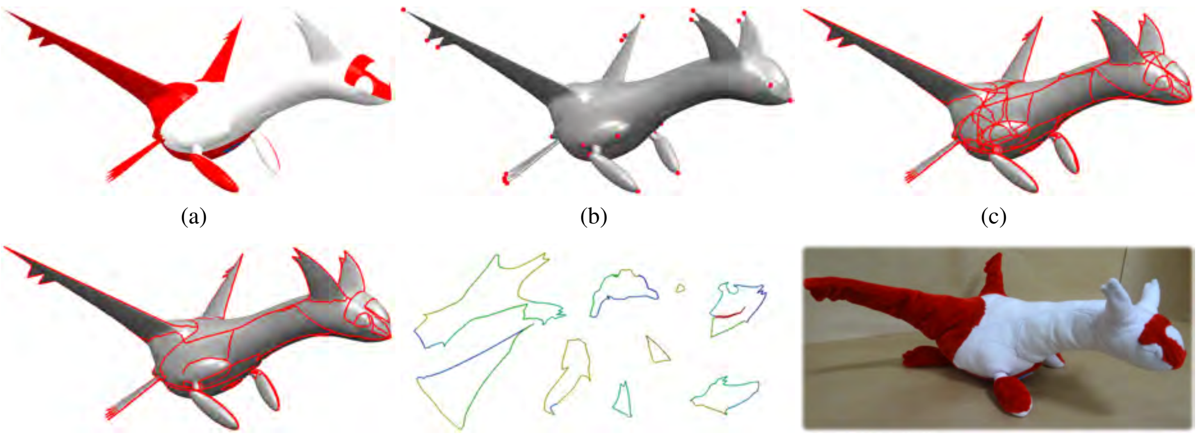
Other than developability, designers should also pay attention to the boundary of the patterns, *i.e., seams*. During fabrication, complex and curvy seams are undesired since they greatly increase the difficulty of assembling. After assembling, the area around seams becomes trenches which is visually noticeable on the surface as shown in Figure 2(a). Depending on the layout, seams capturing the prominent structure of the target shape would be automatically neglected by human while non-structural seams would cause scar-like impressions, which degrades the appearance of fabrication result. Previous works like D-Charts [3] only focused on developability without appearance consideration. The seams they produce does not fit the nature structure of the given shape, resulting in non-negligible impressions on seams.


(a) (b)

Figure 2: (a) Trenches formed by seams on a stuffed toy. (b) Segmentation seams are lines in red. Structural seams are lines in green and darts are lines in blue.

The salient structure, however, can not be easily formulated by simple heuristics. In this work, we take the strategy of *over-segmenting* the target shape to obtain a candidate set of the final seams. These candidates are then split into smaller seam segments to be examined whether they satisfy the desired properties. Finally, unnecessary seam segments are removed by a greedy method similar with [9].

By observing a variety of fabricated objects (we particularly focus on plush toys in this work) designed by experienced designer, we define three types of seams as shown in Figure 2(b). The first type is called *segmentation seams* used to efficiently reduce non-developable areas. They go through the high curvature area (*e.g.,* areas between limbs and torso of creatures) which is usually non-developable. Segmentation seams can be successfully captured by traditional segmentation algorithm. The second type is called *structural seams*. For large smooth non-developable surface without apparent curvature guidance, it's relatively free to put seams to anywhere since there are no obvious answer. Based on the interview of experienced designers, we learned that seams resembling geodesic curves and parallel to the skeleton direction of target shape is likely to be treated as "natural" and being ignored by human eyes. Based on this assumption we introduce a method for evaluating whether a seam candidate is suitable or not. The last type is called *dart* used in pattern design. Darts are used to further flatten a small area after the overall seam layout is


(a) (b) (c)

(d) (e) (f)

Figure 1: The system workflow: (a) input mesh, (b) extrema points selected by conformal factor, (c) initial seams, (d) result of seam removal, (e) flattened patterns (partial) and (f) fabricated plush toy.

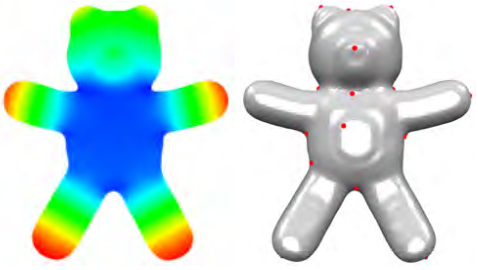

(a) (b)

Figure 3: (a) The visualization of conformal factor. (b) Local extrema points $\mathbf{V}$.

decided. They can be distinguished by the ending point vanished in the middle of the shape. Unlike [3], darts will be generated naturally during our algorithm without post-process.

2. OVERVIEW

The basic idea of the proposed method is to *over-segment* the input mesh and then filter out the unwanted segmentation boundaries. For an input mesh $\mathcal{M}$ with vertices $\mathcal{V}$ and edges $\mathcal{E}$, we first compute its conformal factor [1] to find the local extrema point set $\mathbf{V}$ (Figure 1(b) and Figure 3) and then compute the pairwise shortest paths with custom weight $\mathbf{W}$ among $\mathbf{V}$ as candidate seams $\mathbf{S}$ (Figure 1(c)). Such method does not restrict search area in local space so it can find long seams across multiple regions. After shortest path search, structural seams are contained in $\mathbf{S}$ as sub-path of the original shortest path. To ensure developablility, user can give a threshold $\mathbf{h}$ measuring area changes after flattening. We test all the sub-meshes $\mathcal{M}_i$ partitioned by $\mathbf{S}$ to see if they are developable. If not, we further segment them by adding local seams into $\mathbf{S}$ until all $\mathcal{M}_i$ satisfiy the given threshold $\mathbf{h}$. Since $\mathbf{S}$ contains too much seams that we don't need, they need to be removed. We compute the intersection points $\mathbf{V}'$ (Figure 4(a)) from $\mathbf{S}$ to extract seam segments $\mathbf{S}'$ (Figure 4(b)). For each seam segment in $\mathbf{S}'$, we evaluate its priority using weight $\mathbf{W}$ defined previously. Finally, we apply a greedy method like [9] with this priority to eliminate unwanted seams (Figure 1(d)). For the first type of seams, we use traditional segmentation method like [5] to find them. While for the third type, they will be found during our method so we don't need to worried about them.

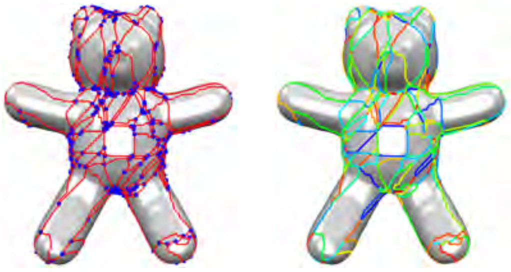

(a) (b)

Figure 4: (a) The intersection points $\mathbf{V}'$ are points in blue. (b) Seam segments $\mathbf{S}'$. Different segments are marked in different colors.

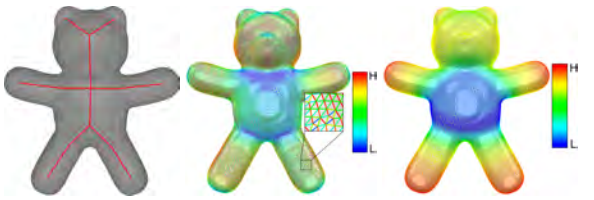
3. PROPOSED METHOD

The proposed algorithm consists of three main stages, including structural seam generation, local seam generation and seam removal. Structural seams determine the overall layout of seam pattern and split mesh into an initial set of patches at the same time. Local seam generation minimizes the non-developable areas by further splitting the sub-meshes into smaller patches. Seam removal eliminates unnecessary seams and reduce the number of patches to alleviate the efforts to assemble patches.

3.1 Structural Seam Generation

The overall layout of seams are defined by structural seams. Structural seams should fit the nature feel of shape when people look at it. By observing physical plush toys designed by experienced designers, we propose several geometric criteria to measure the "nature feel" of the given seams and then generate seams by minimize these criterion functions.

Inspired by [6], we formulate this problem as a shortest path problem with our criterion as edge weight. Instead of the Steiner tree used in [6], we calculate pairwise shortest paths $\mathbf{S}$ among a vertex subset $\mathbf{V}$ of $\mathcal{V}$. To select $\mathbf{V}$, one should start from vertices in non-developable areas so that the final seams split these area into small developable parts. These vertices can be distinguished by



(a) (b) (c)

Figure 5: (a) The extracted skeleton. (b) Visualization of $\cos\theta$. (c) Visualization of $\alpha(e)$.

their curvature which is higher than average. However, curvature information is not always reliable for complex meshes, so we use conformal factor [1] which produces more stable results. In general case, only a small number of vertices is needed for $\mathbf{V}$. We select at most $\sqrt{|\mathcal{V}|}$ vertices from $\mathcal{V}$ whose conformal factors form local extrema among their 2-rings neighbors as $\mathbf{V}$.

To get the desired result of seams, we introduce the weights used in shortest path as follow. **Geodesic distance:** A simple yet elegant line along mesh surface without abrupt turning is preferred for seam. Geodesic paths usually satisfy such properties. Edge weight of this property is defined as its length divided by the maximum edge length.

$$W_g(e) = \frac{\|\vec{e}\|}{l_{max}}, \tag{1}$$

$$l_{max} = \max_{e\in\mathcal{E}} \|\vec{e}\|\,. \tag{2}$$

Skeleton direction: If skeleton information is available, for example, the skeleton of animals and creatures shape showed in Figure 5(a) can be extract by algorithms such as [8], we can use such information to help guiding the direction of seam. Laying seams parallel to the axial direction not only gives a nature look but also reduces the seams required for flattening. We formulate this weight as the directional deviation between a given edge and the averaged direction.

$$W_s(e) = 1 - \alpha(e)\cdot|\cos\theta|\,, \tag{3}$$

where θ is the angle between $\vec{e}$ and $\vec{b_{avg}}$ is described below. $\cos\theta$ measures the consistency between $\vec{e}$ and $\vec{b_{avg}}$. As showed in Figure 5(b), the edges with direction from north-west to south-east is marked in red, which indicates a high consistency to the skeleton nearby.

Given n bones, b_{avg} is the averaged bone direction.

$$\vec{b_{avg}}(e) = \sum_{i=1}^{n} \vec{b_i}\cdot w_{i,e}, \tag{4}$$

$$w_{i,e} = \frac{w_{i,a} + w_{i,b}}{2}. \tag{5}$$

For i-th bone, $\vec{b_i}$ is the direction and $w_{i,a}$, $w_{i,b}$ are the influence of this bone to the two vertices connected by e respectively. There are many ways of determining the influence factor, for example, the vertex weight used for skinning. As long as it represents the spatial relationship from bone to vertex, it's suitable in our application. Here we assume the weights of all bones to sum up to 1. In our implementation, we use the weights produced by [7].

In some cases like chest where many bones gathered together, the averaged direction is not reliable since there are no dominant directions. This is visualized in Figure 5(c) where chest, crotch, and forehead are marked in a relative low color. To prevent from getting incorrect result, we introduce an influential factor α to weaken the effects in such case. This factor measures the direction deviation of effective bones.

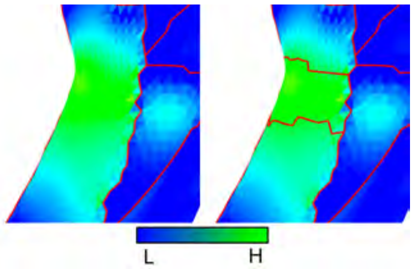


Figure 6: Left: Before local seam generation. Right: After local seam generation. The color represents area changes

$$\alpha(e) = \sum_{i=1}^{n} |\cos\phi|\cdot w_{i,e} \tag{6}$$

ϕ is the angle between $\vec{b_i}$ and $\vec{b_{avg}}$.

Curvature: Curvature plays an essential role in estimating developability. By placing seams on high curvature area, flattening distortion can be alleviated.

$$W_c(e) = \frac{1}{1 + |c_a| + |c_b|} \tag{7}$$

c_a and c_b are the mean curvature of the two vertice connected by e respectively.

Finally the weight can be written as:

$$W = W_g^{\alpha}\cdot W_s^{\beta}\cdot W_c^{\gamma}. \tag{8}$$

α, β, γ are the parameters to control the effectiveness of each term. In our implementation, we use $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.8$.

There are some cases that users may want to mix multiple materials in fabrication. Apparently these materials should be separate by seams as hard constraints. After the calculation of shortest path is completed, these seams are added to $\mathbf{S}$.

3.2 Local Seam Generation

Though we generate appearance-aware seams during the last step, the sub-meshes segmented by these seams are not guaranteed to be developable. To fix this problem, we employ surface flattening to test if a sub-mesh is developable. For a non-developable sub-mesh, we flatten it by calculating its length-preserving free boundary [9] and measure the area change after flattening. We split this sub-mesh into two new sub-meshes by the isoline of area changes until the area change is less than the threshold h given by user. The area change of a sub-mesh $\mathcal{M}_i$ is defined by:

$$\delta_{area} = \max_{t\in\mathcal{T}_i}(area'_t - area_t)/area_t \tag{9}$$

$\mathcal{T}_i$ is the set of faces of $\mathcal{M}_i$ and t is one of its face.

3.3 Seam Removal

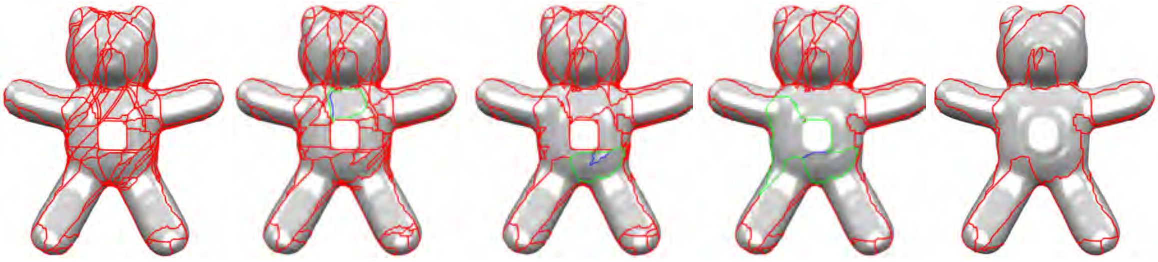

(a) (b) (c) (d) (e)

Figure 7: (a) Initial seams before seam removal; (b)(c)(d) Intermediate results of seam removal. The blue line indicates the seam to be removed and the green lines are the neighboring boundary used to estimate developability. (e) Final result.

It is important to keep the number of generated planar shapes low to simplify fabrication process. The final layout should contain only the essential seams so that it will not generate too much planar shapes. A greedy method similar to [9] is adopted with different priority to remove unnecessary seams.

To remove seams, it's essential to determine which seam is importance or not. For one seam, some part of it may be essential and some other parts may not because there are already other better seams nearby. Thus, for every seam in **S**, we split it into smaller seam segments by the intersection points with $valance \neq 2$ formed by **S** and re-evaluate the importance of each seam segment using the weight defined above. To prevent from making shortest segment being always least important, the importance of a segment is divided by the number of edges it contains. These seam segments are then insert into a minimum heap with the importance associated with it.

Starting from the least important seam segment, we calculate a new sub-mesh by merging two sub-meshes beside this seam and test if this new sub-mesh is developable using the test described in the last section. This process is equivalent to testing whether this seam segment can be removed. If some seam segment is removed, the neighboring seam segment should be updated due to the possible change of intersection points. This process loops until no seam segments can be removed. Figure 7 illustrates the process of iterative seam removal.

4. RESULTS AND DISCUSSIONS

We test the proposed method using different types of models, showing the intermediate results and final results. We also fabricate a real plush using the result generated by the proposed method. There are few obvious overlapping errors in the generated patches caused by instability of LPFB [9]. In such case, we have to manually correct them during fabrication.

The weight function we proposed is composed with several terms which can be combined by summation or multiplication. In our experiments, we found multiplication yield a better result. This can be explained by the influence of each term. We design each term to be in range of $[0, 1]$. For a given edge, if one of these terms give a weight close to zero, it means this term is a dominant strategy than others. By multiplication, other terms will not affect the output weight even if they are close to 1. If summation is used, though the dominant one still exist, the output weight will become larger due to other terms.

Performance. We implemented the proposed method as a plugin of OpenFlipper [4] and runs on a MacBook Pro with a 2.7GHz dual-core Intel CPU. The test models and the corresponding execution time is listed in Table 1. Most of the processing time is spent on computing LPFB to test whether a sub-mesh is developable.

Limitation. We rely on LPFB to test whether a sub-mesh is developable. However, LPFB becomes unstable when the given boundary is not smooth. Such case is inevitable since the boundary is generated by union of seams. Different types of seams will form abrupt turns or spikes at intersection point.

The greedy algorithm we used for seam removal does not give an optimal result. The order of seams removal plays an essential role in the final result. If one seam is removed, the neighboring two patches are merged and become harder to merge with other patches. This is the reason why our result contains many small patches because they are surrounded by larger patches that can not be merged anymore. We plan to tackle this problem by Exact Vertex Cover as suggested in [2].

5. CONCLUSION

In this paper, we proposed a segmentation method minimizing visual defects of plush seams by considering the natural structures that will affect the feeling of human perception. Due to the lack of canonical answer, the whole idea is to find all possible answers and then prune less important candidates. We formulated the whole problem into a shortest path problem and find seams that comply with the natural structure we defined. By union of different seams, we obtain a set of all possible seams containing the sub-seams we need. The unwanted seams is pruned by a greedy method starting from least important candidates. We maintain developability by testing developability of each new sub-mesh formed by removing a seam segment during pruning. If the new sub-mesh is non-developable, this seam segment will not be removed. Finally, we got the a set of seams that fit the human perception while still maintain developability required for plush fabrication.

6. 致謝

本論文感謝科技部經費補助，計畫編號：MOST103-2221-E-002-158-MY3、MOST103-2811-E-002-040 與 MOST103-2218-E-002-030。

7. REFERENCES

[1] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *Proc. 3DOR '08*, pages 1–8, 2008.

Figure 8: Fabricated plush from different views.

Models	#verts	#faces	Time of seam generation (sec)	Time of seam removal(sec)
Latias	13573	27142	47.77	272.2
Teddy	12561	25118	17.50	132.74
Bird	2497	4990	6.71	19.17
Triceratops	2832	5660	3.81	53.26
Horse	8078	16152	87.64	16.59

Table 1: The number of models and corresponding execution time.

[2] R. Hu, H. Li, H. Zhang, and D. Cohen-Or. Approximate pyramidal shape decomposition. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2014)*, 33(6):213:1–213:12, 2014.

[3] D. Julius, V. Kraevoy, and A. Sheffer. D-Charts: Quasi-developable mesh segmentation. *Comput. Graph. Forum (Proc. EG '05)*, 24(3):581–590, 2005.

[4] J. Möbius and L. Kobbelt. Openflipper: An open source geometry processing and rendering framework. In *Proceedings of the 7th International Conference on Curves and Surfaces*, pages 488–500, 2012.

[5] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.*, 24(4):249–259, 2008.

[6] A. Sheffer and J. C. Hart. Seamster: Inconspicuous low-distortion texture seam layout. In *Proc. IEEE Vis '02*, pages 291–298, 2002.

[7] L.-T. Shen, S.-J. Luo, C.-K. Huang, and B.-Y. Chen. SD models: Super-deformed character models. *Comput. Graph. Forum (Proc. PG '12)*, 31(7):2067–2075, 2012.

[8] A. Tagliasacchi, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph. (Proc. SIGGRAPH '09)*, 28(3):71:1–71:9, 2009.

[9] C. Wang. Computing length-preserved free boundary for quasi-developable mesh segmentation. *IEEE Trans. Vis. Comput. Graph.*, 14(1):25–36, 2008.

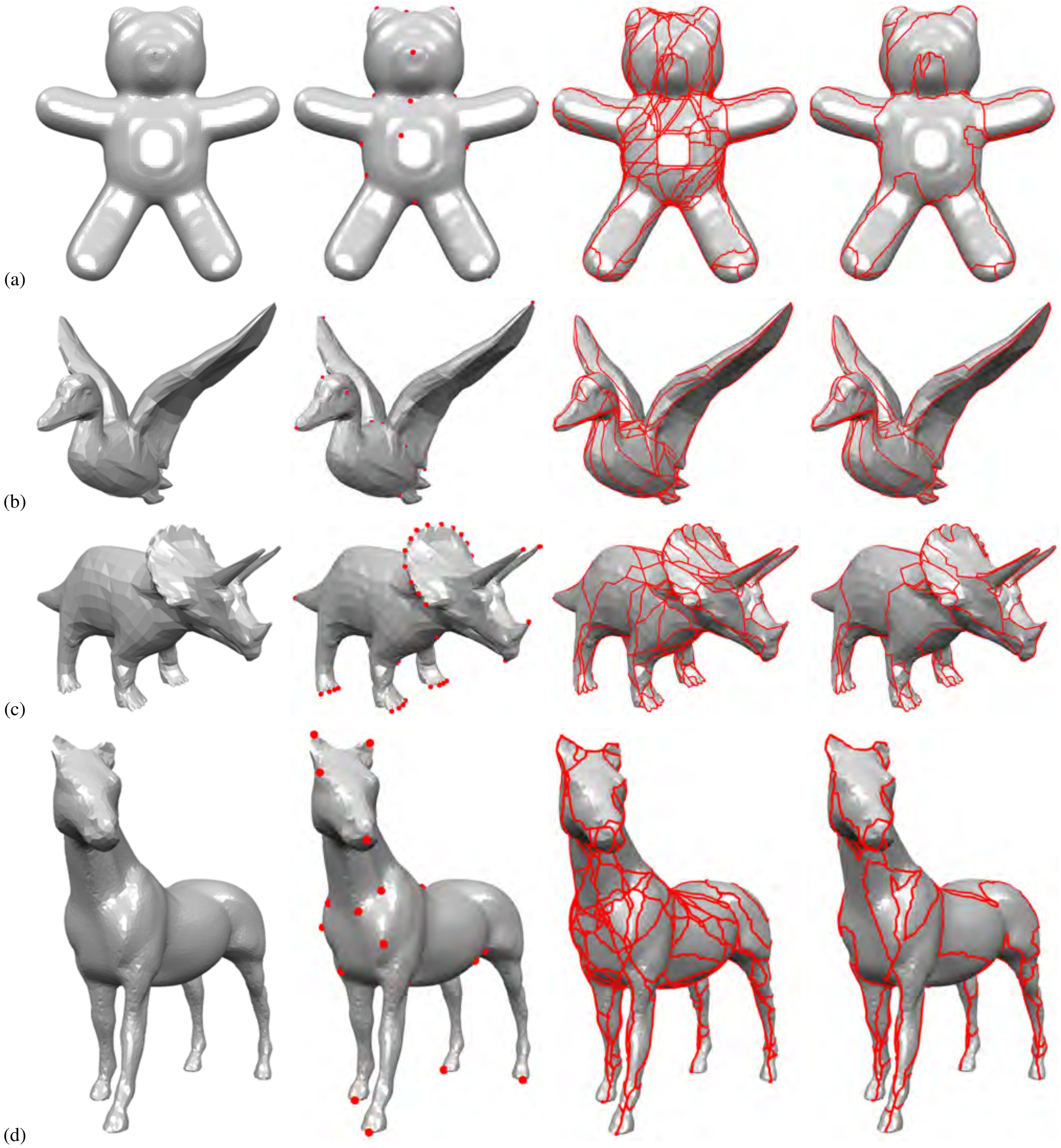


Figure 9: From left to right: input model, extrema points, initial seams, after seam removal. From upper to lower: (a)Teddy (b)bird (c)Triceratops (d) Horse.