Sketch-based Mesh Cutting: A Comparative Study

Lubin Fan   Min Meng   Ligang Liu*

Department of Mathematics, Zhejiang University, Hangzhou 310027, China

Abstract

In this paper we present the first comprehensive study and analysis on different sketch-based mesh cutting approaches. To compare a representative number of state-of-the-art sketch-based mesh cutting methods, we conduct a large scale user study which was carried out via extensive user experiments. To address the objective assessment of the performances of different algorithms, a complete framework with various intuitive sketching interfaces was developed to enable interactive mesh cutting by simply drawing sketches on mesh surface. To address the subjective assessment of user's experience, we presented the analysis of the user's responses, where the analytic hierarchy process was employed to quantify the performance of algorithms in terms of multiple criteria. Our results suggest that human in general agree on the evaluation of the performance of algorithms, and some sketch-based mesh cutting methods are consistently more favorable than others. The importance of our work lies in studying users' experience on operating various sketch-based mesh cutting tools, to motivate more practical interactive systems in the future.

Keywords: sketching interface, mesh cutting, evaluation, analytic hierarchy process

1. Introduction

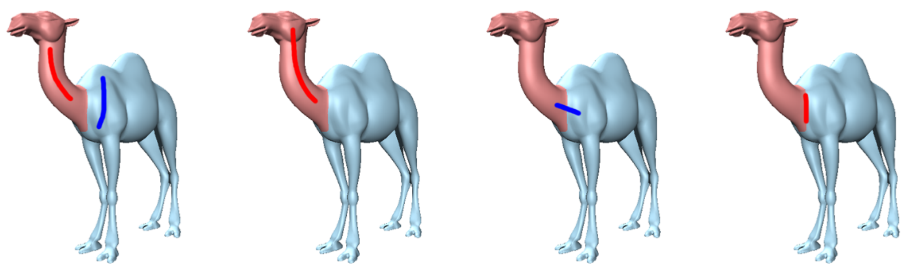
Decomposing 3D shapes into semantic parts is a critical component in shape understanding and many computer graphics applications [1]. Due to the complicated human perception it remains a challenge to develop fully automatic algorithms to define semantic parts. Therefore, interactive mesh

*Corresponding: ligangliu@zju.edu.cn

cutting, which guides the mesh segmentation process by user interaction, has received much attention in recent years [2, 3, 4, 5].

As users prefer using pen and paper to communicate, *sketch-based user interfaces*, which were first introduced in cutting out 3D shapes in [3], have become a great success in interactive mesh cutting as only a few freehand strokes suffice to help users cut out semantic parts meeting users' intention and expectation.

Since then, various sketch-based mesh cutting algorithms have been proposed [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Generally they can be categorized into two types: boundary based user interfaces, which require the user to draw strokes across the desired cut [4] or along the desired cut [12], and region based user interfaces, which allow the user to draw strokes to specify the foreground/background areas [3] or only the foreground area [5]. Fig. 1 shows the different user interfaces for mesh cutting.



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

Figure 1: Different user interfaces of various sketch-based mesh cutting algorithms: (a) foreground/background sketching interface[3]; (b) foreground sketching interface[5]; (c) cross-boundary sketching interface[4]; (d) along-boundary sketching interface[12]. We present a comprehensively comparative study on evaluating the performances of these algorithms and the user experiences of these user interfaces.

However, it is not easy to judge which sketch-based mesh cutting algorithm is superior to the others. Moreover, different users might have different experience on applying the various user interfaces for these algorithms. Therefore, the user can hardly choose the most appropriate algorithm in their own applications.

In this paper, we make an intensive study on evaluating the different sketch-based mesh cutting algorithms with various user interfaces and help better understand their mechanisms and characteristics. Research on evaluating the quality of automatic mesh segmentation algorithms has recently been recognized as important [13, 14]. A more recent work made a comparative evaluation of various foreground/background sketch-based interactive mesh cutting algorithms [15]. Different from these works which evaluate the performances of various segmentation algorithms, our work tries to study the user experience on operating various sketch-based mesh cutting tools. Apparently, the latter is rather nontrivial as the user experience is highly *subjective*.

To our knowledge, this is the first time that an intensive and comparative study has been provided for evaluating both the performance and the user experience of the different sketch-based mesh cutting algorithms and interfaces. The contributions of this work are summarized as follows:

- a comprehensive and perceptual study on sketch-based mesh cutting algorithms was conducted, analyzing the strengths and weaknesses of state-of-the-art interactive algorithms with different interfaces;
- a systematic approach based on the analytic hierarchy process (AHP) was developed for evaluating the user experience on different sketch-based mesh cutting tools in a quantitative and qualitative manner;
- extensive analysis and comparisons of the experimental results on user studies were substantially demonstrated, and valuable insights for the interactive algorithms were provided.

2. Related work

Interactive mesh cutting algorithms. A number of interactive mesh segmentation algorithms have been developed in the literature. Simple methods require the user to specify points on the cuts and use geodesic lines connecting the points as cuts [2, 13]. As sketch-based interfaces provide a more natural and flexible interactions between computers and users, various sketch-based mesh cutting algorithms have been proposed during the last few years. Some methods require users to specify sketches near the cutting boundaries [16, 4, 12] while the other methods allow users to specify sketches in the foreground and/or background regions [3, 6, 8, 9, 10, 11, 5]. In this paper we make a comparative study on evaluating four different sketch-based mesh cutting algorithms (see Fig. 1) and provide valuable insights on user preferences and experiences.

Table 1: The evaluated algorithms and their abbreviation

User interface	Algorithm	Abbreviation
Foreground/background sketching	Easy mesh cutting [3]	EMC
Foreground sketching	Paint mesh cutting [5]	PMC
Cross-boundary sketching	Mesh decomposition with cross-boundary brushes [4]	CBB
Along-boundary sketching	iCutter: A direct cut out tool for 3D shapes [12]	ICC

Comparative evaluation on mesh cutting algorithms. Recently, research on evaluating the quality of mesh segmentation algorithms has become a hot topic. Attene et al. [17] proposed the comparative evaluation of 3D mesh segmentation. Evaluation and comparison was performed by showing side-by-side images of 3D segmented meshes produced by five algorithms [18, 19, 20, 21, 22]. Chen et al.[13] and Benhabiles et al.[23] have proposed the quantitative evaluation of automatic mesh segmentations. Both works developed systems and benchmarks with respect to a ground-truth corpus respectively. Benhabiles et al.[14] presented an experimental comparison of existing metrics for the quantitative assessment of mesh segmentation. The recent work of Meng et al.[12] proposed an extensive analysis and comparison of 5 mesh segmentation algorithms which are based on the foreground/background sketch-based interfaces. Different from all the above evaluation works, we present an intensive and comparative study on evaluating the performances of different sketch-based mesh cutting algorithms and the user experiences of these user interfaces.

3. Sketch-based mesh cutting algorithms

The set of sketch-based mesh cutting methods used in our study covers most of the recent major publications in the field, and equally samples from these approaches according to their interfaces, as shown in Table 1.
Foreground/background sketch-based mesh cutting. The foreground/background sketch-based mesh cutting methods [3, 6, 8, 9, 10, 11] allow the user to draw two types of strokes to specify the foreground or background regions respectively as shown in Fig. 1(a). We chose the easy mesh cutting (EMC for

short) [3] as the representative method of this type as it performs relatively better than the others across most of the performance factors [15].

Foreground sketch-based mesh cutting. The paint mesh cutting (PMC for short) [5] provides a foreground sketching user interface for cutting out meshes, in which the user is allowed to only draw strokes on the foreground region as shown in Fig. 1(b).

Cross-boundary sketch-based mesh cutting. The cross-boundary brushes tool (CBB for short) [11] allows the user to draw strokes across the desired cutting boundary, as shown in Fig. 1(c).

Along-boundary sketch-based mesh cutting. Several along-boundary sketch-based mesh cutting methods [2, 16, 12] have been proposed. We chose the iCutter tool (ICC for short) [12] as the representative method in this type as it provides an easy-to-use interface for cutting out meshes by allowing the user to draw *rough* strokes along the cutting boundary and performs in a more robust and stable manner than the other methods.

4. Evaluation system and task assignment

We describe our evaluation system in this section.

4.1. Ground-truth corpus

Considering the characteristics of sketch-based mesh cutting, we selected 16 categories from the Princeton segmentation database [13] to construct our ground-truth corpus, with five models in different poses for each category. 3 categories are discarded, such as tables owing to their strong symmetry, glasses owing to their simplicity, busts owing to their patch-type segmentations.

Within our corpus, each model has an average of 11 segmentations. So one part for each model is chosen that it can be unambiguously described to the user for extraction. For the clarity of the task description, we associated each part task with 5 images, describing the part segmentation from the model it belongs to via different views. Fig. 2 shows the models selected from the corpus, one model from each category, with one segmentation per model. For further details for all the models and their selected parts, please see the supplementary file.

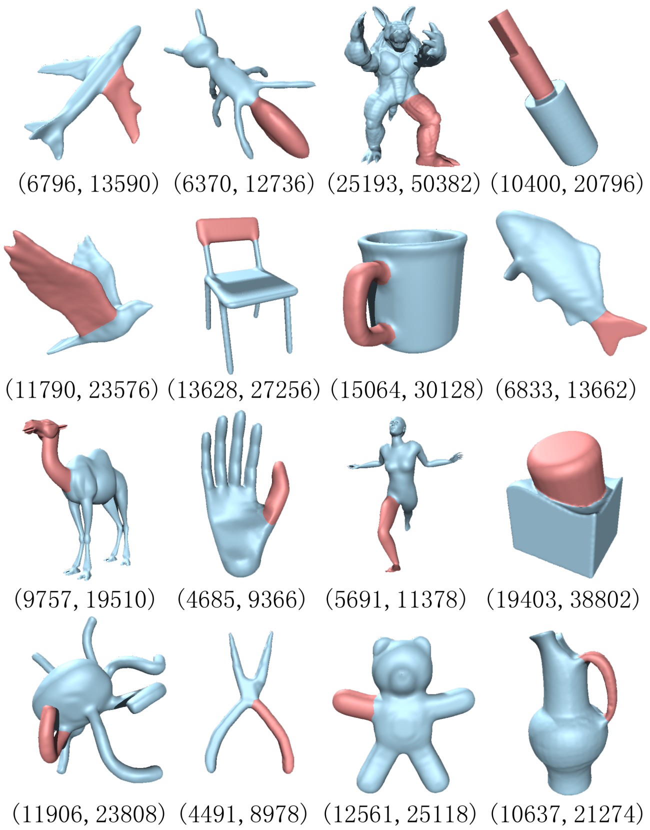
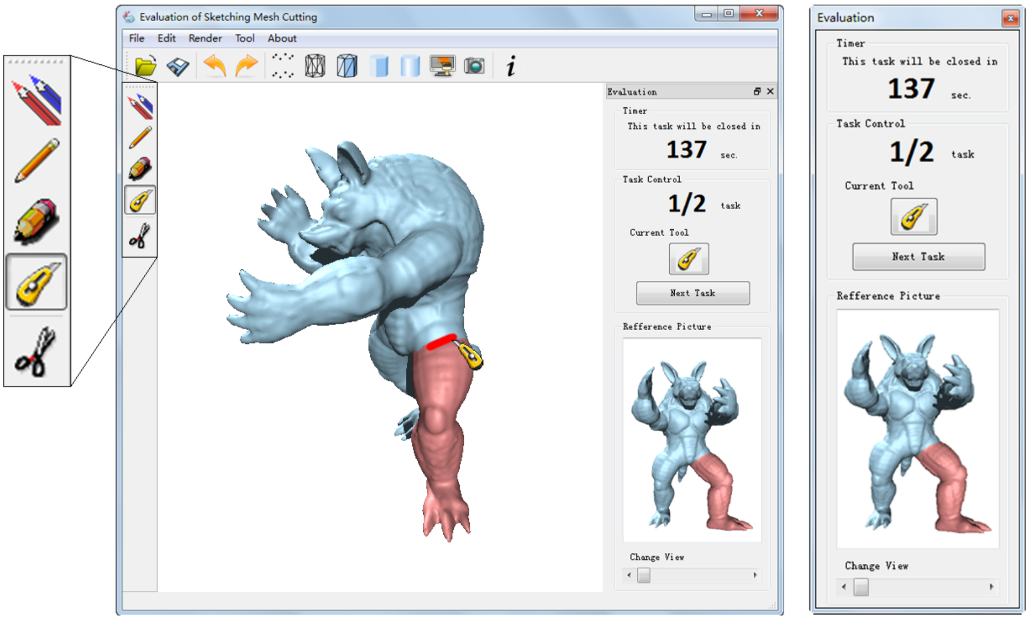


Figure 2: Models in the ground-truth corpus, one part per model of each category. Numbers in brackets denote vertices and triangles of models, respectively.

4.2. Evaluation system

To facilitate the comparison, we have implemented a complete system, allowing participants to segment the semantic parts from models using the evaluated algorithms with corresponding sketching user interfaces. All the evaluated algorithms have been implemented and integrated in the system.

Fig. 3 shows a screenshot of our evaluation system. In the beginning, the participants were shown a video, then given a user guide and sufficient time to help them get familiar with the system. Also some sample models were provided for training. After that, the users were required to load a model into the system and familiarize themselves with the segmentation task by viewing the images associated with the model. The users can drag the scroll



(a) (b)

Figure 3: Screenshot of our evaluation system: (a) the evaluation system for different sketch-based mesh cutting algorithms; (b) the task panel for evaluation.

bar at "Change view" in the "Evaluation panel" located on the right of the system (shown in Fig. 3(b)), to browse the images to see which part needs to be extracted from the model. fig:AHPFor each algorithm, the users can select the corresponding brush tool located on the left of the system (shown in Fig. 3(a)), roughly draw the strokes to specify their requirements, and obtain the segmentation results, then decide whether more strokes are needed to refine the segmentation. We restricted participants to a maximum of 3 minutes per model. They are allowed to proceed to the next task earlier if satisfied with their current segmentation.

4.3. Questionnaire

Once the participants had finished their segmentation tasks, they were asked to fill out a questionnaire which aims at comparing the evaluated algorithms based on their experiences on using the various cutting tools. When examining the sketch-based interactive mesh segmentation methods from the user's perspective, we need to consider four main criteria:

- ease of use: how easily the user specified the strokes to get the desired segmentation;

- user intention: to which degree the algorithm meets the user's intention and expectation;

- stability: how stably the users perceived the algorithm performs;

- efficiency: how efficiently the users perceived the algorithm performs;

These criteria are difficult to interpret and evaluate because of their subjectivity. More importantly, the performance of the evaluated algorithms cannot be represented in a linear scale[24]. Thus we utilized the psychophysical technique of paired comparisons[25] into our evaluation problem, where the participants were asked to compare the relative priority of two elements in pairs. Firstly, the pairwise comparisons of four criteria for the sketch-based mesh cutting in the questionnaire are as follows:

1. Is ease of use more important than user intention?
2. Is ease of use more important than stability?
3. Is ease of use more important than efficiency?
4. Is user intention more important than stability?
5. Is user intention more important than efficiency?
6. Is stability more important than efficiency?

A ratio scale of 1 to 9 is added to the ordinal ranking provided by the responses to these questions to provide the relative importance of one criterion over another for interactive mesh cutting.

After the pairwise comparisons of four criteria were established, the participants were asked to compare the evaluated algorithms in pairs on the basis of each of the four criterion respectively. Similar as above, the corresponding pairwise comparisons of the evaluated algorithms for each criterion are also denoted by the responses on a ratio scale of 1 to 9, to provide the relative performance of one evaluated algorithm over another for each criterion.

4.4. Task assignment

121 individuals participated in our experiment, of which 68 participants had experience in geometry processing, and the rest needed to be trained for the task. There were 87 males and 34 females in all the participants, whose ages are ranged from 20 to 29 years with an average of 24. Most of the participants were computer science graduates.

4.4.1. Corpus division

Since there would lead to learning effects when participants performed all four tests on the same model, we randomly divided the corpus into 80 sets, ensuring that each set contains 6 models from different categories and different models correspond to different pairwise comparisons which covers all 6 paired comparisons for the evaluated algorithms.

4.4.2. Task distribution

Each participant was assigned to run the experiment on the models of one set. After loading a mesh model, the participants were asked to extract the required part from the model using two evaluated algorithms for the specific paired comparison respectively, with corresponding sketching interface.

So at each run, the participant performed the segmentation task on the model using two algorithms in random order, and then is asked the questions about the comparison in the questionnaire, to provide the relative performance of the two algorithms for each criterion on a ratio scale of 1 to 9. Once completed all the segmentation tasks, each participant was also asked to fill out the questions in the questionnaire, to provide the relative importance of the four criteria.

4.5. Experiments collection

All 121 participants completed the experiments. 1452 segmentations were collected for objective evaluation, of which 1327 were accepted, and 125 were discarded as the segmentation conflicted with the requirement of the task. By distributing the model sets to participants equally, each model obtained an average of four segmentations for each algorithm. Additionally, 121 survey responses for the questionnaire were collected for subjective evaluation. Thus all the experimental results can be used to evaluate the performance of the interactive algorithms.

5. Objective evaluation

To objectively evaluate the performance of sketch-based mesh cutting algorithms, three criteria need to be considered [12]:

- Accuracy: the degree to which the extract part corresponds to the ground-truth;

- Efficiency: the amount of time or effort required to perform the desired segmentation for users;

- Stability: the extent to which the same result would be produced over different segmentation sessions when the user has the same intention.

5.1. Accuracy

According to recent works [13, 15], existing measures used to evaluate mesh segmentation can be classified into two categories: boundary measures and region measures. Following the work [15], we employed five normalized metrics, such as NCD, NHD, RI, NLCE and NGCE, to quantify the similarity between segmentations.

To study the accuracy of each evaluated algorithm, we computed:

- the average initial accuracy: the boundary and region accuracy measured when the participant had completed the initial interaction;

- the average final accuracy: compared to the initial accuracy, the boundary and region accuracy measured till the participant had finished the task or the task timing was up, averaged across all the models for each algorithm;

using the accuracy measures mentioned above.

5.1.1. Initial accuracy

Fig. 4 shows the initial boundary and region accuracy statistics for segmentation tasks using the evaluated algorithms. According to Fig. 4, the best performing algorithms, in terms of the measured initial boundary and region accuracy, are CBB and ICC, which perform equally well, followed by PMC, and EMC has the worst performance. Perhaps the reason is that, both the sketching interfaces of CBB and ICC provide good user control over the initial segmentation, by allowing the user to draw freehand strokes to roughly specify where cuts should be made. Comparatively, EMC and PMC allow the user to draw strokes to roughly specify the foreground/background regions. Hence the user loses control over the cutting boundary, and obtains relatively poor initial segmentation.

Additionally, this figure also shows the standard error of measured initial accuracy, computed across all the models for each algorithm. Compared with the other three algorithms, EMC gives the highest standard error of

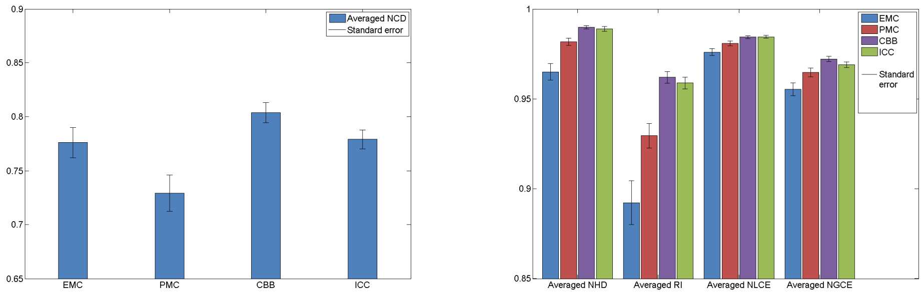


Figure 4: Comparison of the initial boundary and region accuracy for the evaluated algorithms. (Left) averaged initial boundary accuracy (shown as vertical bars) and its standard error (shown as vertical lines on top of the bars); (Right) averaged initial region accuracy (shown as vertical bars) and its standard error (shown as vertical lines on top of the bars).

initial accuracy across all the models in the corpus. This result is reasonable, the lower the accuracy, the higher the standard error for the evaluated algorithms.

5.1.2. Final accuracy

Fig. 5 shows the final boundary and region accuracy statistics for segmentation tasks using the evaluated algorithms. According to Fig. 5, in terms of the measured incremental boundary and region accuracy, EMC and PMC surpass the performance of CBB and ICC, obtaining a wide margin of improved accuracy during the update process. EMC performs poorest initially, but also obtains the best accuracy in the final segmentation. Perhaps because the region growing scheme in EMC allows highly efficient refinement when additional interactions are specified. Both CBB and ICC give superior initial segmentation, but tend to inhibit iterative improvement during the update process. Probably the reason is, both CBB and ICC depend on the global solutions of poisson equation, which give relatively small changes when new interactions are added.

Furthermore, according to this figure, the evaluated algorithms have comparable standard error of measured final accuracy, computed across all the models. Compared with the initial accuracy, the standard error of final accuracy for EMC has been significantly reduced. This result is not surprising. With the increase of accuracy, the standard error will be decreased.

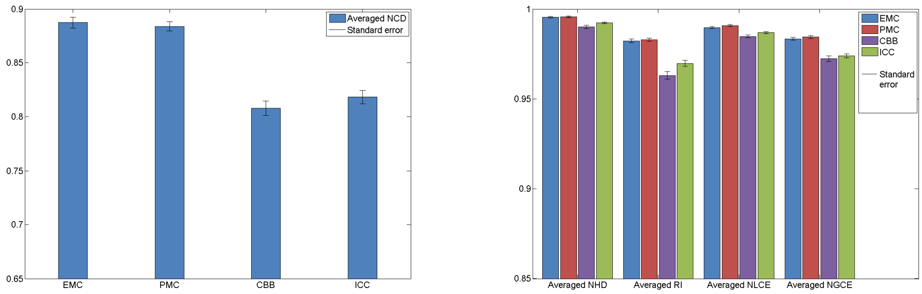


Figure 5: Comparison of the final boundary and region accuracy for the evaluated algorithms. (Left) averaged final boundary accuracy (shown as vertical bars) and its standard error (shown as vertical lines on top of the bars); (Right) averaged final region accuracy (shown as vertical bars) and its standard error (shown as vertical lines on top of the bars).

5.2. Efficiency

Fig. 6 shows the statistics of the time required for mesh segmentation and user interaction for each algorithm. Overall, the averaged time required for the evaluated algorithms are relatively close. All the evaluated algorithms can provide realtime feedback for the models in our corpus, therefore the time is mainly related to the number of user interaction. Specifically, users spent the least amount of time on CBB and ICC, followed by PMC, and the most with EMC. This result is very reasonable. Both CBB and ICC provide good control over the cutting boundary, thus require the minimal amount of user interaction during the whole segmentation process. However, EMC and PMC allow the user to specify in-segment strokes, thus lose control over the cutting boundary. Compared with PMC, EMC needs the user specify foreground and background regions, and requires relatively more amount of user interaction.

According to this figure, the standard error of time required for the evaluated algorithms are comparable with each other. We can see that, the time required for each algorithm have significant consistency when the users cut the mesh models with sketching interfaces.

5.3. Stability

According to the stability test in [7, 12], we computed:

- the averaged normalized coverage: the percentage of triangles with the same labels (foreground or background) found when using different user

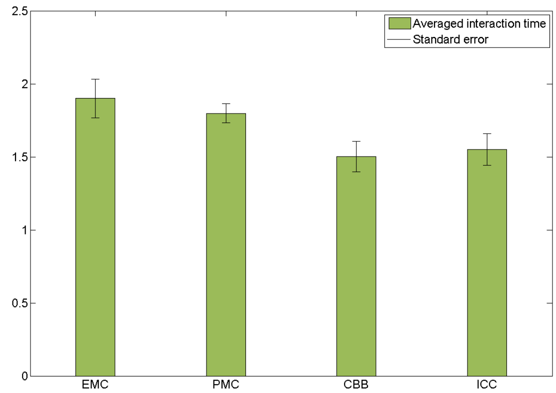


Figure 6: Comparison of the averaged time (shown as vertical bars) required for mesh segmentation and user interaction and its standard error (shown as vertical lines on top of the bars) for the evaluated algorithms, averaged over all the models.

inputs per model, averaged across all models,

for evaluating the stability of each algorithm with respect to different user inputs.

Fig. 7 shows the comparison of the stability test of the evaluated algorithms for the initial segmentation and final segmentation respectively. For the initial segmentation, we see that CBB and ICC are more stable, that is to say, they are less sensitive to different user inputs probably due to the global property of the solution for the poisson equation they depend on. Comparatively, EMC is more sensitive to different user inputs owing to the local property of the greedy region growing technique it employs, PMC is also unstable because of the progressive local optimization it proceeds. In comparison, the levels of stability of the evaluated algorithms for the final segmentation are very close to each other. This result is not surprising. Despite EMC and PMC are more sensitive to the different strokes drawn by the user, both of them allow efficient refinement during the update process, thus can obtain more accurate final segmentation and higher final stability. Comparatively, CBB and ICC are more stable to different user inputs in the initial segmentation, but tend to inhibit iterative improvement in the update process.

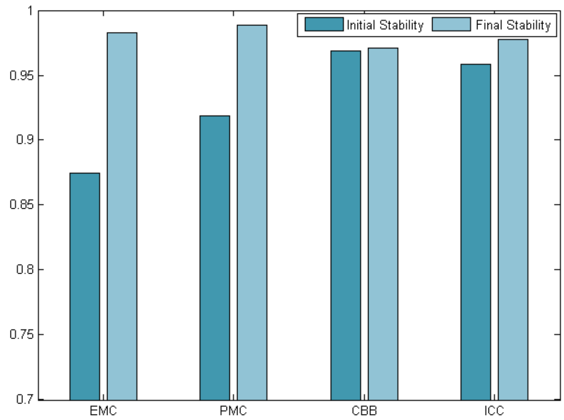


Figure 7: Comparison of the the evaluated algorithms by the stability tested for the initial and final segmentations.

6. Subjective evaluation

In this section, we present an approach for performing an subjective evaluation on the various sketch-based mesh cutting algorithms.

6.1. Analytic hierarchy process (AHP)

The analytic hierarchy process (AHP) developed by Saaty [26, 27, 28] aims at quantifying relative priorities for a given set of alternatives on a ratio scale, based on the judgement of participants. As a participation-oriented methodology, AHP can aid coordination and synthesis of multiple measures [29]. It is ideally suited to help resolve problems that arise when multiple criteria are concerned in performance evaluation. The most important contribution of AHP for performance evaluation is that it provides a *systematic* approach for weighting performance to provide a comprehensive performance measure, which can be used to assess the overall performance of the evaluated algorithms.

The AHP approach consists of four steps:

Step 1. decide upon the criteria for evaluation;

Step 2. rate the relative importance of these criteria using pair-wise comparisons;

Step 3. rate each potential choice relative to each other choice on the basis of each criterion;

Step 4. combine the ratings derived in Step 2 and Step 3 to obtain an overall relative rating for each potential choice. Following the AHP process, the hierarchy of the performance evaluation in our work was developed as shown in Fig. 8.

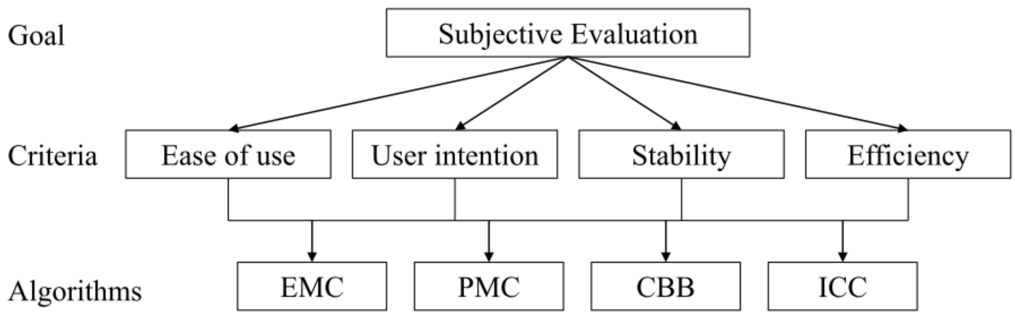


Figure 8: Hierarchy of the AHP in our performance evaluation.

Based on the pairwise comparison results in the questionnaire, the AHP approach can be applied to stress the importance of the intuitive preference of a participant as well as the consistency of the comparisons of the evaluated algorithms in the subjective evaluation.

6.2. Criteria evaluation

According to the AHP hierarchy, the relative priorities for the criteria can be quantified on a ratio scale, which were indicated based on the preferences of the participants in the questionnaire. To determine the relative rating of the criteria, the following need be computed (shown in Table 2):

1. synthesize the pairwise comparison matrix;
2. calculate the largest eigenvalue λ_{max} and the priority vector;
3. calculate the consistency index CI and consistency ratio CR;
4. check the consistency of the pairwise comparison matrix to check whether the participant's comparisons were consistent.

Based on the priority vector in Table 2, the four criteria are ranked as follows: user intention, stability, ease of use, and efficiency. According to the rating of the criteria, it is clear that the criteria of stability and user intention are more important than the other two. This is not surprising. All the evaluated algorithms provide the simple and intuitive sketching interfaces, which can help the user to easily extract the required part from the mesh.

Moreover, all the evaluated algorithms can give realtime feedback for the models in our corpus. Comparatively, the participants preferred to concern the problems of stability and user intention, for example, how stable the evaluated algorithm performs, to which degree the evaluated algorithm meets user's intention and expectation. We are much inspired by this observation, which suggests that the researchers should pay more attention to the stability and user intention issues for the interactive mesh cutting methods.

Table 2: Synthesized matrix for the criteria. λ_{max} denotes the largest eigenvalue of the synthesized matrix, and CI, RI, CR denote the consistency index, random consistency ratio, and the consistency ratio respectively.

Criteria evaluation	Ease of use	User intention	Stability	Efficiency	Priority vector
Ease of use	0.161	0.200	0.121	0.193	0.1688
User intention	0.323	0.401	0.444	0.396	0.3910
Stability	0.423	0.286	0.317	0.300	0.3315
Efficiency	0.093	0.113	0.118	0.111	0.1088
$\lambda_{max} = 4.0407, CI = 0.0136, RI = 0.9, CR = 0.0151 < 0.1$					

6.3. Algorithm evaluation with respect to each criterion

In addition to the comparison of the criteria, the relative priorities for the evaluated algorithms can also be quantified based on the preferences of the participants in the questionnaire. The participants were asked to compare the relative importance of the evaluated algorithms in pair on a ratio scale, in terms of how it contributes to each criterion. Similar as above, the pairwise comparison matrices and priority vectors of the evaluated algorithms on the basis of each criteria can be computed respectively. For further details, please see the supplementary file. According to the priority vectors, the rating of the evaluated algorithms with respect to each criterion is shown in the Fig. 9.

Ease of use. According to the results shown in Fig. 9(a), in terms of ease of use, ICC received the best rank, followed by PMC, EMC got the worst rank. Compared with the other three algorithms, EMC needs the user to draw in-segment strokes for specifying the foreground and background regions. Hence the user loses control over the cutting boundary, and usually needs more additional strokes to refine the segmentation. In comparison, ICC provides the most intuitive interface, which allows freehand strokes along the

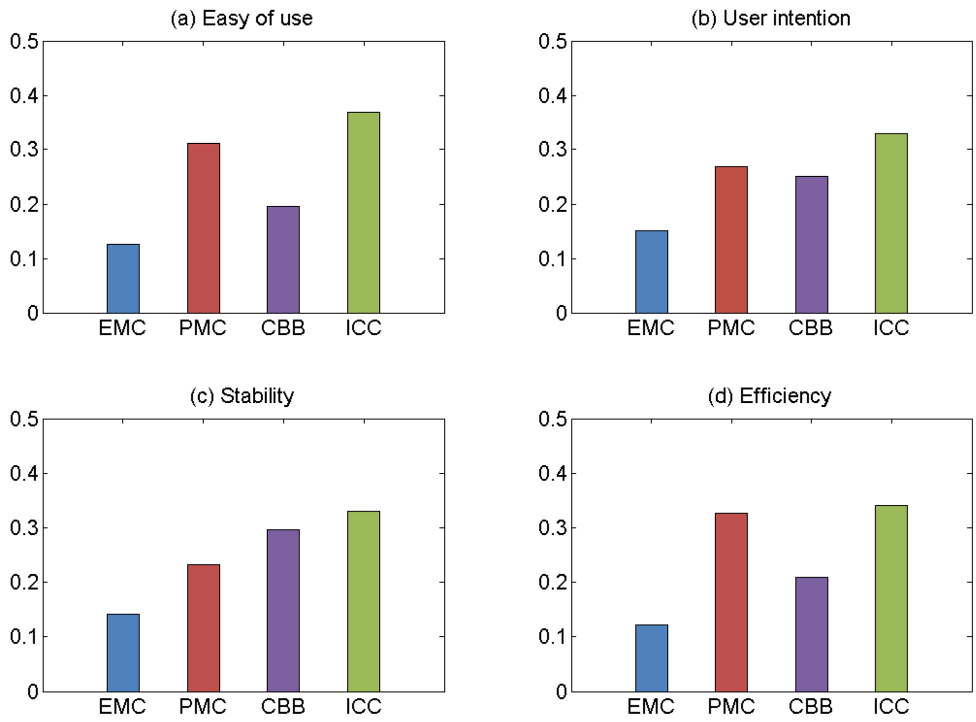


Figure 9: The rank of the evaluated algorithms with respect to each criterion: (a) ease of use; (b) user intention; (c) stability; (d) efficiency.

boundary. Therefore, from the user's perspective, the criterion of ease of use is related to the user interface, the more intuitive the interface, the easier the algorithm.

User intention. According to the results shown in Fig. 9(b), the best ranking algorithm, in terms of user intention, is ICC, PMC and CBB received comparable ranks to each other, and EMC got the worst rank. This result is closely related to the sketching interface. ICC provides the most intuitive interface among all four algorithms, embodying the motif "what you draw is what you get". Similar as ICC, the interface provided by CBB allows the user to draw strokes across the desired cutting boundary, and is also intuitive to users from their perspectives. Additionally, PMC provides more intuitive interface than the other two algorithms, embodying the motif "what you paint is what you get". Comparatively, EMC provides the least intuitive interface, which requires the maximal amount of in-segment strokes and gives the weakest control over the cutting boundary. Hence, similar as the criterion of ease of use, from the user's perspective the criterion of user intention is related to the user interface, the more intuitive the interface, the more intuitive the algorithm.

Stability. According to the results shown in Fig. 9(c), the most stable algorithm perceived by the participant is ICC, followed by CBB and PMC, EMC received the worst rank. This result can be explained by the initial segmentation. As shown in Fig. 4, both ICC and CBB provide superior initial segmentation, thus do not require the user to refine the segmentation using additional strokes. Furthermore, as shown in Fig. 7, these two algorithms perform more stable, that is to say, they are less sensitive to different user inputs across all the models in the corpus. Comparatively, EMC gives the poorest initial guess and perform the most unstable for the initial segmentation. This implies that, the perceived stability is closely related to the initial segmentation, the more accurate the initial segmentation, the more stable the user perceived.

Efficiency. According to the results shown in Fig. 9(d), the most efficient algorithm perceived by the participant is ICC, followed by PMC, and EMC received the worst rank. This result is very reasonable. We know that all the evaluated algorithms can provide realtime feedback for the models in our corpus, therefore the performing time is mainly related to the number of user interaction. Usually. ICC provides the best initial segmentation which meets the user's expectation, and does not require additional refinement. Both PMC and CBB provide the simple interfaces, comparatively, CBB requires more interactions to refine the segmentation owing to its poorer accuracy. EMC requires the user to draw two in-segment strokes for specifying foreground and background regions. Moreover, EMC provides the poorest initial segmentation, always need more additional interactions to refine the segmentation. This implies that both the interface and initial accuracy of the algorithm can affect the perceived efficiency. The simpler the user interface, the more accurate the initial segmentation, the higher efficiency the user perceived.

6.4. Overall performance evaluation

By combining the criterion priorities and the priorities of each evaluated algorithm relative to each criterion, we can develop an overall priority ranking of the evaluated algorithms, which is termed as the priority matrix (shown in Table 3). For illustration purpose, the calculations for finding the overall priority of algorithms are given in the supplementary file.

Table 3: Priority matrix for the overall performance evaluation of the algorithms

	Ease of use (0.1688)	User intention (0.3910)	Stability (0.3315)	Efficiency (0.1088)	Overall priority vector
EMC	0.1258	0.1507	0.1410	0.1225	0.1402
PMC	0.3109	0.2679	0.2323	0.3270	0.2698
CBB	0.1952	0.2510	0.2962	0.2091	0.2520
ICC	0.3682	0.3303	0.3304	0.3415	0.3380

According to the overall priority vector in Table 3, the algorithms in our study are ranked as follows: ICC, PMC, CBB, EMC. This indicates that the most preferred algorithm by the participants is ICC, followed by PMC and CBB, and EMC is the least preferred algorithm. This result is very interesting.

- ICC and CBB provide the simple and intuitive interfaces which strength the control over the cutting boundary, and both of them depend on the harmonic fields defined on the mesh surface. However ICC is more favorable than CBB, perhaps the reasons are: (1)compared with CBB, the user interface provided by ICC is more intuitive to allow users to draw freehand strokes along the boundary, embodying the motif "what you draw is what you get"; (2)compared with CBB, ICC can provide superior initial segmentation, usually does not require additional strokes, thus obtains high efficiency from the user's perspective.

- In spite of its poor rank for the criteria of stability, PMC received comparable rank for the overall performance evaluation. Probably the reasons are: (1) it provides the intuitive user interface to allow users to progressively paint the part of interest, embodying the motif "what you paint is what you get"; (2) it also allows highly efficient improvement during the update process, thus results in high accuracy for the final segmentation.

- Compared with the other three algorithms, EMC provides the least intuitive interface and gives poorest initial segmentation, which increases the number of user interaction and decreases the performing efficiency. Hence it is the least preferred algorithm from user's perspective.

7. Summary

We summarize the analysis on both the algorithms and user experience in this section. They are useful not only to study the properties of the current sketch-based mesh cutting algorithms but also to inspire new interactive segmentation techniques.

7.1. Analysis on experimental results

According to the objective experimental results, several observations on the characteristics of the evaluated algorithms are described as follows:

- both CBB and ICC provide simple and intuitive sketching interfaces and can give good initial segmentations, but tend to inhibit further improvement;
- compared with CBB, ICC is more stable and efficient as ICC employs an averaged harmonic field which can efficiently filter out those incorrect foreground/background point pairs. Furthermore, the user interface of ICC is more natural than that of CBB as it imitates the physical notion of a cutter in real sculpting;
- PMC provides poor initial segmentation and requires the user to progressively paint the part of interest, but provides accurate final segmentation by allowing highly efficient improvement during the update process;
- despite EMC provides satisfied updates during the whole process, it gives the poorest initial guess and usually requires the user to add new strokes to refine the segmentation in many cases.

7.2. Analysis on user experience

When examining the user's preference in the experiments, several insights, which are believed to be helpful and beneficial, are as follows:

- according to the relative priorities of the criteria, researchers should concern more about the user intention and stability issues of the algorithm when they design new interactive mesh segmentation approaches in the future;

- according to the overall ranking of the evaluated algorithms, the algorithms the user preferred either provide a superior initial segmentation, or allow efficient refinement during the update process;

- both the experimental results and user responses indicate that, accuracy and stability are highly related to each other. This implies that researchers not only concern with the improvement in the accuracy but also the stability with respect to different user inputs when designing new interactive systems in the future.

8. Conclusion

We have presented a thorough study on sketch-based mesh cutting algorithms with various sketching interfaces. To facilitate the study, we developed a complete framework integrating four sketch-based mesh cutting tools, which allow users to segment models in the unified system. We constructed the ground-truth corpus as a benchmark based on the Princeton segmentation database, and conducted a large scale user experiments comparing four state-of-the-art sketch-based mesh cutting methods. By studying the experimental results, we further analyzed the performance of the evaluated algorithms both in the objective and subjective manners. We believe that our evaluation methodology will lead to improved sketch-based mesh cutting algorithms, as well as a better understanding of user experiences.

9. Acknowledgements

This work is supported by the National Natural Science Foundation of China (61070071) and the 973 National Key Basic Research Foundation of China (2009CB320801).

References

[1] A. Shamir, A survey on mesh segmentation techniques, Computer Graphics Forum 27 (6) (2008) 1539–1556.

[2] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, D. Dobkin, Modeling by example, ACM Transactions on Graphics (Proc. SIGGRAPH) 23 (2004) 652–663.

[3] Z. Ji, L. Liu, Z. Chen, G. Wang, Easy mesh cutting, Computer Graphic Forum (Proc. Eurographics) 25 (3) (2006) 283–291.

[4] Y. Zheng, C.-L. Tai, Mech decomposition with cross-boundary brushes, Computer Graphics Forum (Proc. Eurographics) 29 (2) (2010) 527–535.

[5] L. Fan, L. Liu, K. Liu, Paint mesh cutting, Computer Graphic Forum (Proc. Eurographics) 30 (2) (2011) 603–611.

[6] H.-Y. Wu, C. Pan, J. Pan, Q. Yang, S. Ma, A sketch-based interactive framework for real-time mesh segmentation, in: Proc. Computer Graphics International, 2007.

[7] Y.-K. Lai, S.-M. Hu, R. R. Martin, P. L. Rosin, Fast mesh segmentation using random walks, in: Proc. ACM Symposium on Solid and Physical Modeling, 2008, pp. 183–191.

[8] M. Meng, Z. Ji, L. Liu, Sketching mesh segmentation based on feature preserving harmonic field, Journal of Computer-Aided Design and Computer Graphics (in Chinese) 20 (9) (2008) 1146–1152.

[9] C. Xiao, H. Fu, C.-L. Tai, Hierarchical aggregation for efficient shape extraction, The Visual Computer 25 (3) (2009) 267–278.

[10] S. Brown, B. Morse, W. Barrett, Interactive part selection for mesh and point models using hierarchical graph-cut partitioning, in: Proc. Graphics Interface, 2009, pp. 23–30.

[11] J. Zhang, C. Wu, J. Cai, J. Zheng, X. cheng Tai, Mesh snapping: Robust interactivemesh cutting using fast geodesic curvature flow, Computer Graphics Forum (Proc. Eurographics) 29 (2) (2010) 517–526.

[12] M. Meng, L. Fan, L. Liu, iCutter: A direct cut out tool for 3d shapes, Journal of Computer Animation and Virtual World (2011) to appear.

[13] X. Chen, A. Golovinskiy, T. Funkhouser, A benchmark for 3D mesh segmentation, ACM Transactions on Graphics (Proc. SIGGRAPH) 28 (3) (2009) 1–12.

[14] H. Benhabiles, J.-P. Vandeborre, G. Lavoué, M. Daoudi, A comparative study of existing metrics for 3d-mesh segmentation evaluation, The Visual Computer 26 (12) (2010) 1451–1466.

[15] M. Meng, L. Fan, L. Liu, A comparative evaluation of foreground/background sketch-based mesh segmentation algorithms, Computer & Graphics (Proc. Shape Modeling International) 35 (3) (2011) 650–660.

[16] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, H.-P. Seidel, Mesh scissoring with minima rule and part salience, Computer Aided Geometric Design 22 (2005) 444–465.

[17] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, A. Tal, Mesh segmentation - a comparative study, in: Proc. Shape Modeling and Applications, 2006, pp. 14–25.

[18] M. Attene, B. Falcidieno, M. Spagnuolo, Hierarchical mesh segmentation based on fitting primitives, The Visual Computer 22 (2006) 181–193.

[19] S. Katz, A. Tal, Hierarchical mesh decomposition using fuzzy clustering and cuts, ACM Transactions on Graphics (Proc. SIGGRAPH) 22 (3) (2003) 954–961.

[20] S. Katz, G. Leifman, A. Tal, Mesh segmentation using feature point and core extraction, The Visual Computer 21 (2005) 649–658.

[21] M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, J. Rossignac, Blowing bubbles for multi-scale analysis and decomposition of triangle meshes, Algorithmica 38 (2003) 227–248.

[22] M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, J. Rossignac, Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies, in: Proc. Shape modeling and Applications, 2004, pp. 339–344.

[23] H. Benhabiles, J.-P. Vandeborre, G. Lavoué, M. Daoudi, A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models, in: Proc. Shape Modeling and Applications, Beijing, China, 2009, short paper.

[24] M. G. Kendall, B. Babington-Smith, On the method of paired comparisons, Biometrica 31 (1940) 324–345.

[25] H. A. David, The Method of Paired Comparisons, Oxford University Press, 1988.

[26] T. Saaty, The analytic hierarchy process, McGrawHill, 1980.

[27] T. Saaty, How to make a decision: the analytic hierarchy process, European Journal of Operational Research 48 (1990) 9–26.

[28] T. Saaty, K. Kearns, Analytical planning: the organization of systems, The analytic hierarchy process series, 1991.

[29] Y. Chan, B. Lynn, Performance evaluation and the analytic hierarchy process, Journal of Management Accounting Research 3 (2002) 57–87.