First of all, we would like to thank all anonymous reviewers for their constructive and valuable comments, which will certainly help us improve our manuscript. We have revised the paper (changes colored in yellow). Here we provide response to the individual review comments.

Summary Review

**[1]** A quantitative evaluation of the accuracy and a comparison with the alternative slower k nearest neighbor graph methods.

Response: We have made comparison with the alternative slower k nearest neighbor graph methods; See the last paragraph of the RC-kNNG Retrieval Performance part (Sec. 8) of the revision. The qualitative improvement in part extraction is reported in Table 1. The retrieved parts generated with the four different methods is shown in Figure 17.

**[2]** A comparison to simply segmenting the database.

Response: We have made comparison to the pre-segmentation approach; See the comparison to pre-segmentation approach part (Sec. 8) of the revision.

Review 1

**[1]** The main question I have is if this method has any advantages over the existing kNNG methods other than query runtime performance. Table 1 clearly shows the queries are faster, at a cost of slower precompute. But I find it a bit vexing that there is no similar comparison of matching error, such as in Table 3. Reading between the lines, I am left with the impression that the contribution is only faster queries? This could still be important, but for most of the proposed applications I am skeptical.

Response: We have compared the matching error among different retrieval methods (reported in Table 1); See the last paragraph of RC-kNNG Retrieval Performance part (Sec. 8) of the revision.

**[2]** Why the descriptor as described in sec 5.1 is invariant to the index ordering of the contour polylines. IE, for two contours, the descriptor only makes sense if the (i,j) indices of the matrix refer to points in comparable relative spatial locations. In other words, where does 'i' start on each polyline?

Response: The contours are sampled in the anticlockwise direction. In each section, ‘i’ is the first sampling point in the sense of anticlockwise direction.

**[3]** Fig 12 shows a global symmetry example. Would local symmetry/similarity matching also be possible?

Response: If the contour of the local symmetric part is ambiguous, our method could fail. Incorporating cues from surrounding context could help disambiguate contours. We leave it for future; See the future work part (Sec. 9) of the revision.

Review 2

**[1]** Explanation of Recommendation: I would argue that the main novelty comes from putting together the complete system. The system contains multiple components that have been designed / adopted by the authors. The technical contribution of the individual parts is reasonable engineering, but I am not convinced that they can be claimed as major contributions.

Response: We would like to highlight that we have proposed two novel algorithms: 1) the fast, sketch-based, partial 3D shape matching method based on multi-view projections; 2) the novel customized segmentation method based on a super-face graph. We demonstrate several applications of our method to shape design and exploration.

**[2]** One problem is the quality of the results. The system is demonstrated by some figures. However, in these figures, the retrieved results are not very similar to the sketches. This can be good or bad. In the good case, no great matches are available in the system and the system does a good job in retrieving the best available matches. In this case that indicates robustness. In the bad case, there would be many similar or better retrieval candidates that the system fails to identify. How can I tell? Therefore, the traditional evaluation of retrieval problems uses quantitative metrics, e.g. precision, recall, F-score, precision-recall curves, average precision, ... and a comparison of these metrics to competing algorithms. This submission does not provide quantitative results of the quality of the retrieval. Instead of analyzing the quality, the paper analyses the speed of the retrieval. The speed however is not relevant, without a clear demonstration of quality.

Response: We have quantitatively evaluated the retrieval quality for different retrieval methods; See the last paragraph of the RC-kNNG Retrieval Performance part (Sec. 8) of the revision.

**[3]** The partial shape matching problem is not well specified. It is difficult to define a metric that computes the quality of partial matches and it is already difficult to get humans to agree on what partial matches are good or not. This is a general problem, also for other methods in this area, but I feel it could have been tackled a bit better.

Response: I agree with you that it is a difficult problem to propose a recognized definition for the partial shape matching problem. We leave this for future; See the future work part (Sec. 9) of the revision.

**[4]** More retrieval results could be presented. Instead of showing more retrieval results, the paper contains very rough sketches of quite adventerous applications. I understand that other papers follow the same template, but the underlaying strategy is to replace quality with quantity.

Response: We have added more retrieval results; See the last sentence of the first paragraph of Sec. 8.

Review 3

**[1]** I would not say that sketch-based modeling "typically" interprets sketches in a data-driven manner by searching a repository for a matching high-quality part. That is sketch-based shape retrieval, and only one, atypical approach to sketch-based modeling.

Response: We have reworded the sentence; See the first paragraph (Sec. 1) of the revision.

**[2]** Are contours sampled uniformly from directions around the unit sphere (Section 4, third paragraph)?

Response: Yes. The contours are sampled uniformly in the anticlockwise direction.

**[3]** When sampling uncovered points on the contour ("Construction of the RC-kNNG" point 1), what happens when a point is sampled that does not have 10 consecutive points on either side? (Relatedly, 20 is an even number; how can a point be at the center of 20 consecutive points.)

**Response:** 1) Since the contour is end-to-end and the number (50, 150, and 250) of the sampled points in each contour is greater than 21, there will always be enough consecutive points on either side.

2) We have corrected the mistake of the number of the sampled points; See the Construction of the RC-kNNG part (Sec. 4) and paragraph 3 (Sec. 5.1) of the revision.

**[4]** Why must every one of the k-nearest neighbors be within a distance threshold for a section to be considered valid (last paragraph of Section 4)? Why not simply disconnect edges above the threshold? Does clustering the list mean clustering all valid sections? Is this clustering based on the same distance defined in Section 5.1?

**Response:** 1) If every one of the kNNs of the section be within a distance threshold, the section has the potential to be as the seed section to represent other sections.

2) We keep the edges above the threshold to establish more connectivity among nodes in the graph.

3) Yes. Clustering the list means clustering all the valid sections. We have reworded the third sentence in the last paragraph (Sec. 4) of the revision.

4) Yes. The clustering is based on the same distance defined in Section 5.1. We have added one sentence in the last paragraph (Sec.4) of the revision.

**[5]** How are super faces selected when the retrieved shape has multiple contours matching the query? Is the process described in the first paragraph of Section 6.2 repeated for each contour?

**Response:** The process described in the first paragraph of Section 6.2 is repeated for each of the contour matching the query.

**[6]** How do you "filter out invalid parts and collapse multiple instances into single part" (last sentence of 6.2)?

**Response:** The invalid parts (generated with depth-discontinuous contour section Figure 8(a)) are identified by a group of isolated super-faces. This kind of parts are discarded. The multiple instances of the same part (Figure 8(c)) are identified by matching the sets of the super-faces. The repetitive instances are removed. We have added some sentences in the last paragraph (Sec 6.2) of the revision.

**[7]** What is the sub-SFG (Section 7, "Multi-scale part suggestion")? What makes the suggested segments multi-scale? If the normalized cuts produces sets of faces, these faces may not have silhouettes, so how can the contour descriptor and distance be computed? More explanation is needed.

**Response:** 1) The sub-SFG is the SFG corresponding to the part retrieved. We have reword the second sentence of the Multi-scale part suggestion (Sec. 7) of the revision.

2) The retrieved part is future partitioned into several segments. We suggest segments according to the distance between the segments and the corresponding part: given the scale S, the segments, whose distance to the corresponding part is less than or equal to S are suggested. We have reworded the latter part of the Multi-scale part suggestion part (Sec. 7) of the revision.

3) The contour descriptor of the segments is not used here. The distance is defined as the Euclidean distance between the OBB of the segment and the corresponding part. We have reworded the latter part of the Multi-scale part suggestion part (Sec. 7) of the revision.

**[8]** Are the whole-object contours represented with m=20 samples when computing distances (Section 8, "For the two kNN graph methods...")?

**Response:** No. Each of the whole-object contours is sampled under 3 different scales. Our levels place 50, 150, and 250 sampled points along the whole-object contour respectively. The contour descriptor is then computed for the sampled points. We have added 3 sentences for the second paragraph of the RC-kNNG retrieval performance part (Sec. 8) of the revision.

**[9]** I don't understand where the scale parameter S fits into the multi-scale part suggestion (Section 7, "Multi-scale part suggestion" and Figure 14). S does not appear in the expression for Tn. The figure shows different scale levels, but all shapes appear to be segmented into 2 or 3 parts uncorrelated with the scale level.

**Response:** Tn is the number of the segments, which the retrieved part could be partitioned into. The scale S is given by the user. The segments, whose distance to the corresponding part are less or equal to S, are suggested. We have reorganized the Multi-scale part suggestion part (Sec. 7) of the revision.

**[10]** Why is it a problem that the contour descriptor is not scale invariant (Section 9, Limitations) if the database stores contours at various scales?

**Response:** Although we have stores contours at various scales, there are some scales are missing.

**[11]** Typos: Section 2: "develop a statistical models"; Section 3: "a online phase"; Section 6.3.2: "an other edge" should be "another edge"

**Response:** We have corrected the typos.

**[12]** Why does increasing the superface count shrink the matched part (Figure 15)? There may be a normalization term missing somewhere.

**Response:** With the increasing of the superface count, the superfaces locating in the boundary of the extracted part are separated into smaller ones. Therefore, the extracted part fits the user’s sketch better and better. With the increasing of the superface count, some extracted parts shrink (Figure 19 (a), (c), (d) of the revision), while others expand (Figure 19 (b) of the revision).

**[13]** The primary weakness is a missing evaluation of the fundamental premise. The premise of the approach is that the example database becomes larger by implicitly containing all possible segmented parts. This is interesting and a worthwhile idea, but it is unevaluated. Why not simply run segmentation on the database during preprocessing? Then any of the approaches mentioned in the related work (sketch-based or 3D shape-based) which retrieve similar entire shapes from a database could be compared to. Can users no longer find the sub-parts they are looking for?

**Response:** 1) I agree with you that the premise of the sketch-based shape retrieval methods is that the example database containing all possible shapes. We leave the evaluation of the fundamental premise for future; See the future work part (Sec. 9) of the revision.

2) We have made a comparison to the pre-segmentation approach; See the Comparison to pre-segmentation approach part (Sec. 8) of the revision.

**[14]** The "RC-kNNG Retrieval Performance" doesn't explain whether the alternative indexing structures retrieve the same parts.

**Response:** We have added the comparison among the alternative indexing structures and visualize the retrieval result generated with the four retrieval methods; See the last paragraph of the RC-kNNG Retrieval Performance part (Sec. 8) of the revision.

**[15]** Does the proposed approach work on a different database than the one it was developed with?

**Response:** I develop the approach on a database including 73 shapes from different categories: humanoid, lamps, quadrupeds, birds, insects, and candlesticks. The database is then extended to 513 shapes for evaluation and application.

Review 4

**[1]** Clarity of Exposition: The exposition is fairly clear, but the number of camera views and how they are determined is not given.

**Response:** We extract contours from 21 camera views: 3 canonical side views, 4 corner views and 14 uniformly sampled views; See the Camera view settings part (Sec. 8) of the revision.

**[2]** The super-faces are analogous to super pixel representations in images — perhaps citation of a super pixel paper would be appropriate.

**Response:** We have cited a super pixel paper; See the third paragraph of the Offline Phase part (Sec. 3) of the revision.

**[3]** — missing discussion of choice number and location of camera views needed. This seems to be a key issue, and would seem to depend on the types of objects in the data base.

**Response:** We have compared different camera view settings: 7 camera views, 21 camera views, 49 camera views, and 112 camera views; See the Camera view settings part (Sec. 8) of the revision.

**[4]** — the extreme simplicity of the partial shapes used. In a significant database an extraordinarily large number of partial geometries could match the very simple sketches used here.

**Response:** I agree that there could be an extraordinarily large number of partial geometries matching the very simple sketches. The ranking algorithm (introduced in Sec. 5.2) could rank the results according to the matching error. The contextual information could be adopted to assist our partial matching algorithm. We have noted this point in the future work part (Sec. 9) of the revision.

**[5]** — the very small size of the data based searched. There are obviously larger data sets available, making it seem suspicious that this doesn’t scale well.

**Response:** We have demonstrated our method on the database with 513 models. The performance is shown in table 1. I agree that it is better to test our algorithm on a larger database. We leave it for future; See the future work part (Sec. 9) of the revision.

**[6]** In the example applications, the ideas are interesting, but the pose of the partial shape used for computing the boundary contours appear carefully selected.

**Response:** The pose of the partial shape used for computing the boundary contours is important for our method. If a suboptimal view is chosen, the method can return inappropriate candidate parts. We have noted this point in the future work part (Sec. 9) of the revision.

**[7]** Similar to FKS04, a compelling test would be to give the user something specific to design from pieces of objects in the data base.

**Response:** Good suggestion! We have added the new application; See the photo-driven part-based modeling part (Sec.7) of the revision.