Intrinsic Reflective Symmetry Axis Curve Generation for Meshes

Batuhan Tosyalı[1 [0009−0006−0243−2028]] and Yusuf Sahillioğlu[1 [0000−0002−7997−4232]]

Middle East Technical University, Çankaya, Ankara, Türkiye
batuhan.tosyali@metu.edu.tr ys@ceng.metu.edu.tr

Abstract. Symmetry is a key structural cue in computer graphics, guiding many shape analysis tasks. We propose a novel method to extract intrinsic reflective symmetry axis curves—curves that split a mesh into two intrinsically symmetric, near-equal regions. Our approach begins with geodesic-based point sampling, followed by point matching via spectral histograms computed along geodesic paths to candidate bisector regions. These histograms are compared using normalized optimal transport. To improve robustness, we aggregate votes from multiple paths and midpoints, selecting the best bisector region and refining the axis iteratively. Our method outperforms state-of-the-art techniques on SCAPE and TOSCA, and produces strong results on the diverse Princeton dataset.

Keywords: Intrinsic Symmetry Axis Curve Generation · Bisector Region · Symmetric Sampling · Symmetric Matching

1 Introduction

Extracting intrinsic symmetry axes aids pose-invariant matching [12], remeshing [20], segmentation, and modeling. It supports mirrored designs, skeletons, UVs, and speeds up rendering and compression via structural reuse. We address this challenging task amid irregular geometry and shape variation.

We propose a multistage method for extracting intrinsic reflective symmetry axis curves on polygonal meshes, focusing on 3D. The pipeline begins with robust geodesic-based sampling, followed by candidate matching using spectral signatures. We construct bisector regions and refine the symmetry axis through optimal transport-based histogram comparisons. This layered approach handles geometric noise, structural variability, and mesh irregularities effectively.

While designed for complete meshes, our algorithm tolerates minor topological variations and limited partiality (e.g., missing limbs). It performs best on complete, well-structured meshes. Code and executables of our research can be found at [30].

The contributions of this paper are as follows:

– **Accuracy:** For each matched pair, we construct a bisector region and identify the curve that minimizes the geodesic distance to the pair's midpoint, selecting it as the candidate symmetry axis. We then iteratively refine both the

correspondence set and the axis curve itself, resulting in a robust and precise extraction of the intrinsic symmetry axis. Our method achieves state-of-the-art performance in partitioning meshes into symmetric regions, as evidenced by superior area ratio results.
– **Efficiency**: We achieve point matching via Optimal Transport with minimal overhead.

2 Related Work

Symmetry detection has shifted from extrinsic planes to intrinsic spectral and functional-map methods for greater robustness. We extend this progress with a compact intrinsic approach for reflective axis extraction that addresses prior limitations.

2.1 Symmetry Detection and Shape Correspondence

Symmetry detection is a mature field in image processing and digital geometry processing, featuring diverse methodologies; surveys such as [27] and [15] offer comprehensive comparative evaluations on this topic.

Recent methods extract symmetry planes for extrinsically symmetric shapes. He et al. [8] use PCA-based vertex distances with iterative refinement, while [13] fits planes via curvature-guided sampling. Nagar et al. [16] adapt [29] for point clouds, and [18] refines a randomly initialized plane descriptor-free. In contrast, our method targets intrinsic symmetry using geodesic distances .

Hierarchical methods decompose 3D shapes via structural symmetries [31], while intrinsic ones use Möbius transforms for symmetry-aware sampling [11], [9]. Eigenfunction-based techniques match regions via Laplace–Beltrami spectra [24], and coarse-to-fine pipelines combine isometric alignment with flip correction [26]. Inspired by [19], we replace sign sequences with spectral histograms and Optimal Transport in Algorithm 1. Embedding-based methods [34] reduce flips to rotations, and [23] improves robustness via orientation-penalizing regularization.

2.2 Symmetry Axis Curves

Intrinsic symmetry-axis extraction has been studied in [22], which surveys detection methods. [28] addresses noisy meshes via statistical voting. [33] uses bisector-based selection with scalar field refinement, extended by [32] through multi-scale clustering of scale-aware geodesic profiles.

Deep learning has advanced symmetry detection, with early works [7],[10] targeting extrinsic symmetries. [21] recasts intrinsic symmetry as a functional map task for real-time, robust results, while [5] learns orientation-aware features to resolve flips and improve consistency.

Our method builds on shape analysis advances, using the Heat Kernel Signature (HKS) [29] for robust, multiscale intrinsic descriptors and geodesic-based sampling via average and minimum distances [9].

3 Method for Symmetry Axis Curve Generation

3.1 Overview

We present our method, highlighting key innovations and improvements over prior work. An overview of the full pipeline is shown in Figure 1.

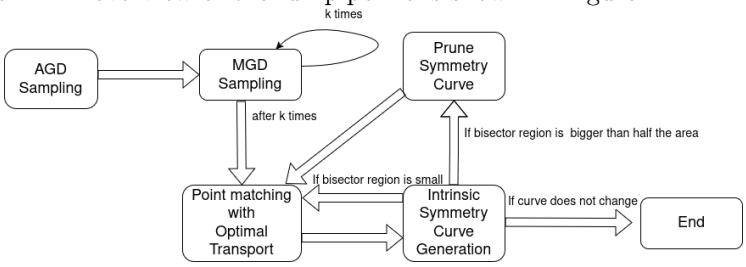


Fig. 1: Overview of the algorithm.

We sample symmetry-aware points using AGD and MGD [9], this strategy increases the chance that for every sampled point, we also sample its symmetrical counterpart, yielding reliable initialization pairs. Correspondences rely on HKS [29], normalized by geodesic distance from midpoints. From bisector regions, we trace paths, bin HKS values, and compare histograms via log-scaled Optimal Transport [14]. A multi-pass scheme selects regions minimizing inverse-weighted midpoint-to-boundary distance (Fig. 2). Overextended bisectors are pruned, same-side pairs removed, and the refined axis reused to update or seed new pairs.

3.2 Symmetry-Aware Point Sampling

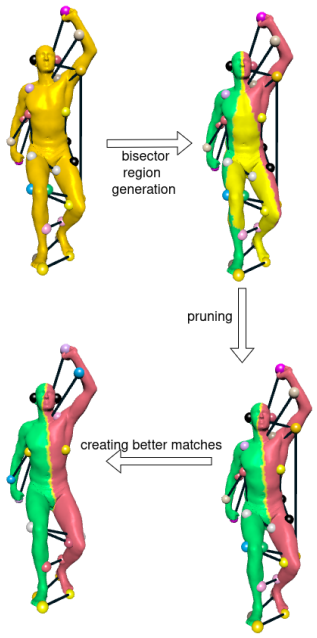


Fig. 2: Sequence of symmetry-curve generation and pruning; colored spheres mark corresponding pairs, linked by black lines.

We sample using Average Geodesic Distance (AGD), summing distances from each vertex to all others. After 1-ring smoothing, local extrema are selected (Fig. 3-a). This extends to any k-ring [9].

We compute Minimum Geodesic Distance (MGD) to find closest points to each AGD sample—treated as bisector centers—and apply a distance threshold to prevent clustering. Fig. 3-b–d shows the iterative MGD process.

3.3 Candidate Point Matching

Comparing Intrinsic Properties Candidate pairs from AGD/MGD sampling are filtered by HKS similarity and midpoint distance (Algorithm 1) to retain only well-placed, similar pairs. Nearby pairs are discarded to avoid oversized bisector regions and reduce computation.

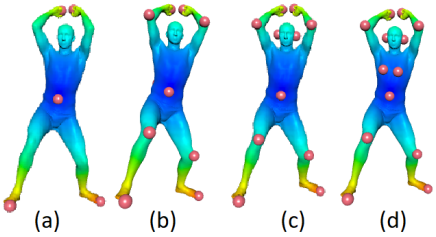


Fig. 3: (a) AGD values from low (blue) to high (red) and samples (spheres) based on AGD local extrema. (b, c, d) Iterative sampling that inserts new MGD samples as we go from (b) to (d).

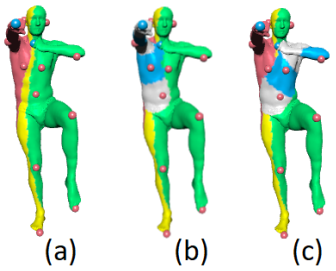


Fig. 4: Red spheres show sampled points. Blue ones mark the bisector pair defining the yellow bisector (red/green sides). Alternating bands indicate histogram bins.

Algorithm 1 Early Candidate Matching Test

```
procedure CheckPointsFeasible(mesh, p1, p2)
    isHks ← CheckHKS(mesh, p1, p2)
    isDistance ← CheckDistanceRatio(mesh, p1, p2)
    return isHks ∧ isDistance
end procedure
```

Pairing with Histogram For valid pairs, we compute bisector regions, sample normalized HKS histograms along geodesics to the boundary (Fig. 4), and compare them via optimal transport—lower costs imply stronger matches.

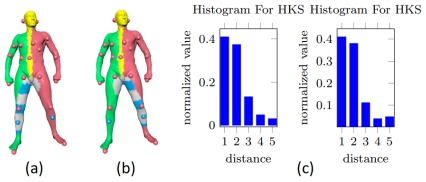


Fig. 5: (a) Area influenced by the right blue sphere on the shinbone; (b) same for the left blue sphere. Alternating blue and gray indicate histogram bins. (c) HKS histograms from (a–b) show strong similarity, supporting their correspondence.

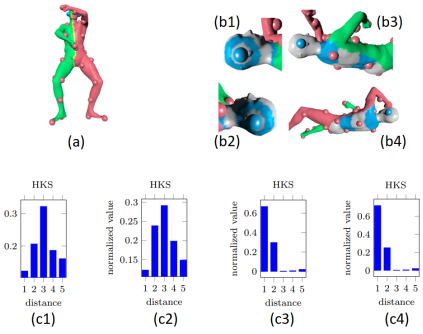


Fig. 6: (a) Yellow bisector with red (samples) and blue (selected pairs) spheres. (b1–b2) Histograms from blue points to the closest bisector boundary; (b3–b4) to the farthest. (c1–c4) Corresponding HKS histograms. X-axis: binned geodesic distances, with alternating colors marking successive bins.

To ensure uniform sampling, we normalize geodesics by regional bounds and trace N paths per pair (Figs. 5, 6). Higher N improves accuracy at cost. For

each path γ, we gather points q within threshold F, i.e., $\min_{x\in\gamma} \text{dist}_g(q,x) \leq F$, then bin their normalized HKS values by endpoint distance to form the final histogram.

Comparing histograms We evaluated L2 norm, KL divergence, Chi-Square, EMD, and KS metrics for histogram comparison, but found Optimal Transport (OT) [4] to yield the most accurate and robust results. We compare normalized HKS histograms along bisector-boundary geodesics using entropy-regularized Optimal Transport (OT), solved via the Sinkhorn algorithm [4], for fast and stable correspondence scoring.

We use OT cost to score similarity—lower values reflect stronger intrinsic matches. Unlike L_2 or KL, OT captures subtle shifts; its log variant [14] improves robustness. Top-ranked pairs seed symmetry extraction; same-side and residual pairs are pruned.

3.4 Generating Symmetry Axis Curve

We extract the symmetry axis by inversely weighting midpoint-to-bisector geodesics, then prune outliers for a smooth, meaningful curve.

Selecting Best Region We select the bisector with the lowest inverse-weighted midpoint-to-boundary sum (Fig. 7-a–d) and prune outliers. Algorithm 2 prioritizes short, central matches with normalization. Area-weighting showed no benefit.

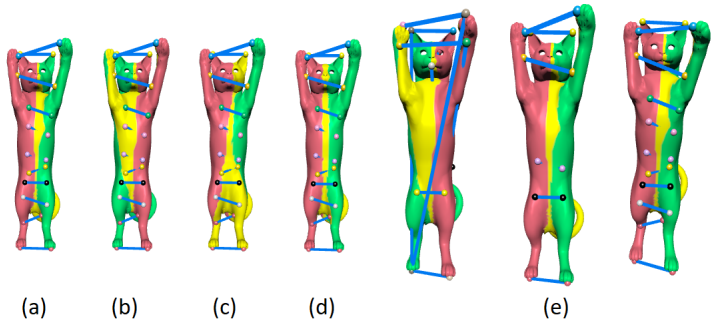


Fig. 7: (a–d) Example bisector regions (yellow) on one model: (a) and (d) are valid; others are flawed. Yellow indicates the bisector, red and green mark opposing sides. (e) Initial bisector (left), after pruning invalid matches (center), and final refined symmetry axis (right).

Pruning and Iterations Bisectors from overly close pairs often overflow (Fig. 9-a). We address this using our pruning routine (Fig. 9-b).

The algorithm computes each pair's geodesic midpoint, splits the mesh along its normal, selects the farthest bisector point as the front, and ray-casts backward to find the rear midpoint.

Algorithm 2 Best bisector selection: Voting algorithm for selecting symmetry axis curve.

```
procedure (mesh, pairs)
    for i ← 1 to pairs.size do
        bisector ← GenerateBisector(mesh, pairs[i].p1, pairs[i].p2)
        result ← 0
        for j ← 1 to pairs.size do
            distance ← MidpointToBisector(bisector, pairs[j])
            weight ← GeodesicPathLength(pairs[j])
            result ← result + 1/weight × distance
        end for
        results.append(result)
    end for
    return min(results)
end procedure
```

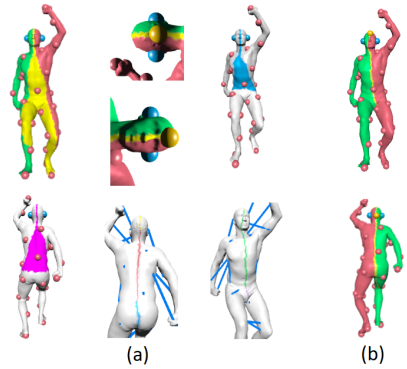


Fig. 8: (a) Pruning steps: (left) initial bisector in yellow with blue samples, (middle) matched midpoints (orange) and their ray-cast counterparts, (right) mesh split into front (blue) and back (magenta) via midpoint normals. Geodesic paths from each midpoint to its inverse and farthest matches form the symmetry axis curve. (b) Final result after pruning (front: top, back: bottom). Compare with top-left for improvement.

After refining the axis (Fig. 9-c), we discard same-side pairs and remove others with low Dijkstra-based distance ratios (Fig. 9-d). We re-run Algorithm 2 to update matches and, if needed, the bisector. Figures 7-e and 8 show improved accuracy with refined yellow axes and green/red mesh partitions.

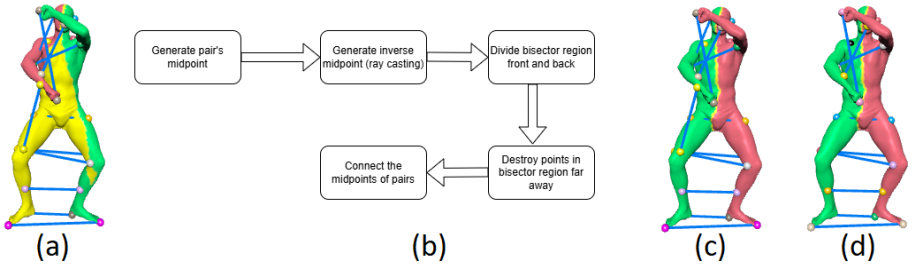
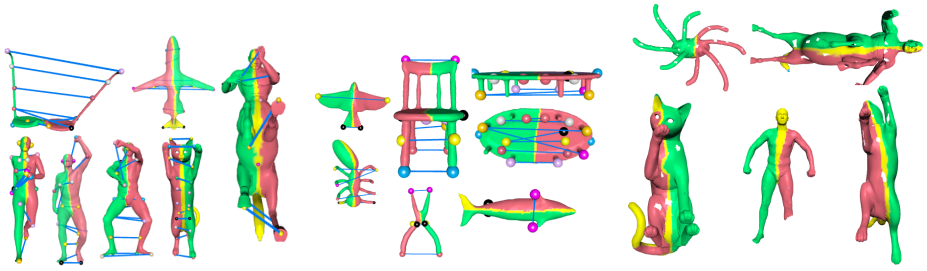


Fig. 9: Pruning steps: (a) Initial mesh with overextended yellow bisector and matched pairs. (b) Intermediate pruning. (c) Refined bisector aligned to the symmetry axis. (d) Final result after removing incorrect matches.

4 Results

In this section, we evaluate our algorithm on different datasets, SCAPE [1], TOSCA [2] and Princeton [3], focusing on symmetry accuracy, area balance, and runtime. Then, we discuss our ablation study which confirms Optimal Transport's benefit. Figure 10 shows robust results on diverse, imperfect meshes.



(a) SCAPE and TOSCA results (b) Princeton results (c) Modified results (missing parts)

Fig. 10: Results across SCAPE, TOSCA, and Princeton datasets. Our algorithm consistently extracts coherent symmetry-axis curves.

Table 1: Performance change with parameters and comparative performance summary.

(a) Using Different Parameters

(a) Area-based weighting

Dataset	No Area	Area
SCAPE	92.52	94.38
TOSCA	89.21	91.45

(b) Histogram count per vertex

Count	1	3	5	7	10	20
SCAPE	71.1	93.3	94.4	93.1	92.6	91.3
TOSCA	70.1	85.6	91.5	95.2	93.3	92.1

(c) Iteration count

Iter #	1	2	3	4	5	6
SCAPE	82	85	88	88	92	93
TOSCA	77	83	85	86	86	89

(b) Comparison with State-of-the-Art

(d) Overall dataset accuracy

Dataset	LIRSD [21]	FAISD [17]	Ours
SCAPE	97.5	97.5	94.4
TOSCA	98	97.8	92.1

(e) Mesh-level accuracy

Dataset	LIRSD [21]	FAISD [17]	Ours
SCAPE	100	100	98.6
TOSCA	99.3	100	95.4

(f) TOSCA per-class rates

Class	LIRSD [21]	FAISD [17]	Ours
Cat	95.5	95.6	**98.3**
Centaur	100	100	100
David	**97.2**	96.2	85.4
Dog	100	98.8	100
Horse	**96.4**	97.3	87.1
Michael	**98.7**	96.5	92.2
Victoria	**97.8**	96.2	84.1
Wolf	**100**	100	98.6
Gorilla	**100**	100	95.3

Our final results can be seen in Figure 10-a and Figure 10-b. Sampling AGD maxima followed by six MGD extrema gave the best accuracy at higher cost.

Area-weighting improved correspondence but reduced robustness on partial data. Increasing histogram comparisons consistently boosted accuracy (Table 1c). Following [9] [17], a match is correct if $\text{error}(k, m') < \frac{M}{N}$, with M as surface area and $N = 20$; mesh-level accuracy reflects meshes with over 75% correct matches.

Accuracy improves with denser sampling, as well-distributed midpoints better capture the true axis. Filtering limb matches, applying area-based weighting, and pruning over-segmented curves further enhance results (Table 1a). Increasing histogram comparisons boosts accuracy (Table 1b), though gains taper and may fluctuate. Final results (Tables 1d, 1e, 1f) show competitive performance against both the learning-based [21] and axiomatic FAISD [17] methods.

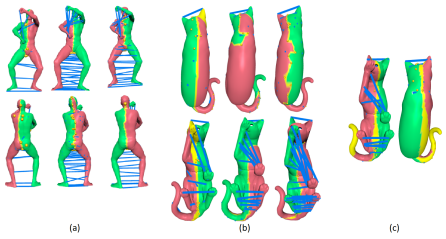


Fig. 11: (a) Symmetry axis curves on SCAPE: ours (left), [21] (middle), [17] (right). Orange spheres show midpoints; blue lines indicate correspondences. (b) Same for TOSCA. (c) Reconstructed axis for [21] using our midpoint-joining protocol.

Figure 11 shows that while some methods slightly outperform in correspondence rate, our sampling creates more stable and accurate symmetry axes. Using local extrema avoids the upper-body clustering seen in [21] [17] (Fig. 11-a,b). As these methods lack explicit axes, we reconstruct them via midpoint chains. If their area ratio drops below 60% and trails ours by over 10%, we regenerate it using our method (Fig. 11-c). Tables 4–3 confirm better balance and 96% SCAPE accuracy.

Table 2: Area ratios separated by the symmetry axis curve: Princeton dataset (left) and aggregated/TOSCA data (right).

Table 3: Princeton dataset.

Mesh/Method	LIRSD [21]	FAISD [17]	Ours
Human	88.12	91.52	**96.27**
Cup	73.12	**78.19**	69.82
Glasses	92.53	93.14	**93.78**
Ant	91.20	92.80	**94.41**
Chair	91.45	82.56	**94.92**
Octopus	**94.91**	88.43	93.81
Table	93.87	84.12	**94.80**
Teddy	90.30	91.25	**92.49**
Hand	89.90	88.92	**91.26**
Plier	92.86	93.46	**95.17**
Fish	92.34	88.90	**96.49**
Bird	91.57	92.13	**93.82**
Bust	92.70	84.43	**94.32**
Armadillo	93.25	94.17	**95.32**
Mech	89.91	91.75	**94.27**
Bearing	92.18	**93.15**	92.32
Vase	**91.57**	88.14	89.23
Fourleg	93.70	92.41	**94.82**

Table 4: Overall dataset-level ratios.

Dataset	LIRSD [21]	FAISD [17]	Ours
SCAPE	88.32	91.9	**96.58**
TOSCA	92.89	93.75	**94.61**
Princeton	91.71	90.69	**95.31**

Table 5: TOSCA class-level ratios.

Name	LIRSD [21]	FAISD [17]	Ours
Cat	85.89	90.44	**94.32**
Centaur	93.45	91.41	**95.21**
David	89.13	92.80	**93.24**
Dog	93.57	91.16	**94.71**
Horse	92.91	89.71	**93.41**
Michael	91.23	90.11	**92.91**
Victoria	92.80	**96.81**	93.81
Wolf	94.21	93.70	**98.60**
Gorilla	91.80	90.11	**95.32**

4.1 Parameters

We use fixed parameters across all experiments: HKS threshold 0.1, point-to-midpoint 0.8, point-to-point over min-midpoint 0.2, and 5 histogram bins. HKS is computed at t=200 using the 20 smallest eigenpairs.

4.2 Complexity

Table 6 outlines stage-wise costs. AGD and MGD involve Dijkstra per vertex, yielding naive $O(V^3)$ time. We reduce this using a Fibonacci heap to $O(V \log V + E)$, and further optimize by running Dijkstra from $J \ll V$ farthest points [6]. Setting $J = 100$ ensures coverage, maintaining overall complexity at $O(V \log V + E)$.

Table 6: Asymptotic time complexities of stages.

Sampling	Histogram path	Pruning
$O(V \log V + E)$	$O(K^2 P(V \log V + E))$	$O(1)$

Given K sampled points, we form $\binom{K}{2}$ candidate pairs. Each pair's bisector region is defined via two Dijkstra runs from its midpoint. Histogram generation for Optimal Transport involves P Dijkstra calls per pair (one per path point), resulting in a dominant complexity of $\mathcal{O}(K^2 P(V \log V + E))$. To reduce cost, we optionally sample $R \ll P$ points uniformly along the path, following the path augmentation strategy in [25], with minor accuracy trade-offs.

Pruning is lightweight, requiring only two Dijkstra runs per sample pair. With precomputed geodesic distances, time remains constant and space complexity is $O(K \times V)$.

Time evaluation The algorithm scales with mesh size and sample count. A single AGD pass and three MGD iterations yield accurate curves efficiently. Table 7-a shows runtimes on an MSI GV62 8RE (i7-8750H) for sample meshes.

Table 7: Timing vs. MGD iterations on sample meshes.

Mesh / Iteration	3	5	9
SCAPE Human	8.3	34.8	162.6
TOSCA Cat	33.9	212.3	1722.8
Princeton Airplane	2.5	6.8	36.7

4.3 Ablation Study

We ablate histogram comparison metrics to assess their effect on geodesic-path–based descriptors and the accuracy of point matching, detailed below.

Motivation and Experimental Setup We use the log-variant of Optimal Transport (OT) to compare histograms, as it offers minimal morphing cost and outperformed L2, KL divergence, and Chi-Square in both robustness and efficiency.

We evaluate on SCAPE, TOSCA, and Princeton using a fixed setup. Candidate pairs are generated by our pipeline, and histogram descriptors are computed along geodesic paths. Each pair is compared using L_2, Chi-Square, and Earth Mover's Distance. Results appear in Tables 8a (area ratios) and 8b (correspondence rates).

Table 8: Comparison of different histogram distance metrics.

(a) Area Ratios

Dataset / Method	OT	L2	Chi-Square	Earth Mover's
SCAPE	**96.58**	92.78	94.46	89.12
TOSCA	**94.61**	90.15	93.81	91.51
Princeton	**95.31**	93.41	91.90	92.73

(b) Correspondence Rates

Dataset / Method	OT	L2	Chi-Square	Earth Mover's
SCAPE	**94.38**	91.84	92.96	89.41
TOSCA	**92.14**	90.17	89.48	91.73

Considering these Tables 8a and 8b, it is seen that the the method for both correspondence rates and area ratios based on OT is the best solution among other histogram comparison methods.

5 Discussion and Conclusion

5.1 Limitations and Possible Improvements

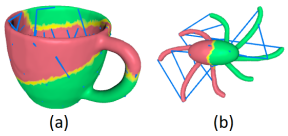


Fig. 12: Failure cases on Princeton meshes: (a) Incorrect axis due to curvature ambiguity on a cup. (b) Reasonable axis on an octopus, but incorrect correspondences.

Our method struggles on uniformly curved meshes where HKS lacks discriminative power (Fig.12-a), and on multi-symmetric shapes (Fig.12-b), where ambiguous correspondences arise despite a correct axis—complicating rotational symmetry detection.

Our single-axis design hinders multi-axial cases, and repeated geodesic queries pose challenges for real-time or high-res use.

While our method tolerates moderate noise and partial cuts (Fig. 10-c), severe point removal disrupts geodesic-based computations, limiting robustness for partial symmetry extraction.

5.2 Future Work

Future work could extend our symmetry-axis extraction to automate mirroring in graphics and CAD, enhance mesh segmentation, UV mapping, and material placement, and support medical imaging and biomechanics. While our method

targets a single dominant axis, hierarchical or multi-scale clustering could enable multi-axis detection. Learning-based approaches may replace hand-crafted HKS with CNN/GNN descriptors, while reinforcement learning could adapt parameters for better generalization.

5.3 Conclusion

We present a method for extracting intrinsic symmetry axis curves on 3D meshes. Starting with AGD and MGD sampling, we filter candidate pairs by geodesic midpoints and HKS differences. For each pair, we construct bisector regions and compare HKS histograms using logarithmic Optimal Transport, offering superior discrimination to area- or diameter-based descriptors. Matched curves are refined via inverse ray casting and iterative pruning. Experiments demonstrate competitive accuracy, robustness, and efficiency. Our approach offers a strong foundation for future research in mesh symmetry and the official implementation can be found at [30].

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. ACM Transactions on Graphics **24**(3), 408–416 (Jul 2005)
2. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer New York (2009)
3. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3D mesh segmentation. ACM Transactions on Graphics (Proc. SIGGRAPH) **28**(3) (Aug 2009)
4. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Journal of Machine Learning Research **15**, 3221–3245 (2013)
5. Donati, N., Corman, E., Ovsjanikov, M.: Deep orientation-aware functional maps: Tackling symmetry issues in shape matching (2022)
6. Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.: The farthest point strategy for progressive image sampling. IEEE Trans. Image Processing **6**, 1305–1315 (1997)
7. Gao, L., Zhang, L.X., Meng, H.Y., Ren, Y.H., Lai, Y.K., Kobbelt, L.: Prs-net: Planar reflective symmetry detection net for 3d models (2019)
8. He, C., Wang, L., Zhang, Y., Wang, C.: Dominant symmetry plane detection for point-based 3d models. Advances in Multimedia **2020**, 1–8 (Oct 2020)
9. Kim, V.G., Lipman, Y., Chen, X., Funkhouser, T.: Möbius transformations for global intrinsic symmetry analysis. Computer Graphics Forum **29**(5), 1689–1700 (Jul 2010)
10. Kobsik, G., Lim, I., Kobbelt, L.: Partial symmetry detection for 3d geometry using contrastive learning with geodesic point cloud patches (2023)
11. Lipman, Y., Funkhouser, T.: Möbius voting for surface correspondence. ACM Transactions on Graphics **28**(3), 1–12 (Jul 2009)
12. Liu, T., Kim, V.G., Funkhouser, T.: Finding surface correspondences using symmetry axis curves. In: Computer Graphics Forum. vol. 31, pp. 1607–1616. Wiley Online Library (2012)
13. Lukas Hruda, J.D.: Estimating approximate plane of symmetry of 3d triangle meshes. Proceedings of CESCG 2017 (2017)

14. Mandad, M., Cohen-Steiner, D., Kobbelt, L., Alliez, P., Desbrun, M.: Variance-minimizing transport plans for inter-surface mapping. ACM Transactions on Graphics **36**(4), 1–14 (Jul 2017)
15. Mitra, N.J., Pauly, M., Wand, M., Ceylan, D.: Symmetry in 3d geometry: Extraction and applications. In: Computer graphics forum. vol. 32, pp. 1–23. Wiley Online Library (2013)
16. Nagar, R.: Robust extrinsic symmetry estimation in 3d point clouds. The Visual Computer (Mar 2024)
17. Nagar, R., Raman, S.: Fast and accurate intrinsic symmetry detection (2018)
18. Nagar, R., Raman, S.: 3dsymm: Robust and accurate 3d reflection symmetry detection. Pattern Recognition **107**, 107483 (Nov 2020)
19. Ovsjanikov, M., Sun, J., Guibas, L.: Global intrinsic symmetries of shapes. Computer Graphics Forum **27**(5), 1341–1348 (Jul 2008)
20. Podolak, J., Golovinskiy, A., Rusinkiewicz, S.: Symmetry-enhanced remeshing of surfaces. In: Proceedings of the fifth Eurographics symposium on Geometry processing. pp. 235–242 (2007)
21. Qiao, Y.L., Gao, L., Liu, S.Z., Liu, L., Lai, Y.K., Chen, X.: Learning-based intrinsic reflectional symmetry detection. IEEE Transactions on Visualization and Computer Graphics **29**(9), 3799–3808 (Sep 2023)
22. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Full and partial symmetries of non-rigid shapes. International Journal of Computer Vision **89**(1), 18–39 (Feb 2010)
23. Ren, J., Poulenard, A., Wonka, P., Ovsjanikov, M.: Continuous and orientation-preserving correspondences via functional maps (2018)
24. Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D.: Partial functional correspondence (2015)
25. Sahillioğlu, Y., Horsman, D.: Augmented paths and reodesics for topologically-stable matching. ACM Transactions on Graphics **42**(2), 1–15 (2022)
26. Sahillioğlu, Y., Yemez, Y.: Coarse-to-fine isometric shape correspondence by tracking symmetric flips. Computer Graphics Forum **32**(1), 177–189 (Jan 2013)
27. Sahillioğlu, Y.: Recent advances in shape correspondence. The Visual Computer **36**(8), 1705–1721 (Sep 2019)
28. Sipiran, I., Gregor, R., Schreck, T.: Approximate symmetry detection in partial 3d meshes. Computer Graphics Forum **33**(7), 131–140 (Oct 2014)
29. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. Computer Graphics Forfum **28**(5), 1383–1392 (Jul 2009)
30. Tosyalı, B.: Intrinsic symmetry curve generation. https://github.com/Batucan2601/Intrinsic-Symmetry-Curve-Generation- (2025), accessed: 2025-06-11
31. Wang, Y., Xu, K., Li, J., Zhang, H., Shamir, A., Liu, L., Cheng, Z., Xiong, Y.: Symmetry hierarchy of man-made objects. Computer Graphics Forum **30**(2), 287–296 (Apr 2011)
32. Xu, K., Zhang, H., Jiang, W., Dyer, R., Cheng, Z., Liu, L., Chen, B.: Multi-scale partial intrinsic symmetry detection. ACM Transactions on Graphics **31**(6), 1–11 (Nov 2012)
33. Xu, K., Zhang, H., Tagliasacchi, A., Liu, L., Li, G., Meng, M., Xiong, Y.: Partial intrinsic reflectional symmetry of 3d shapes. ACM Transactions on Graphics **28**(5), 1–10 (Dec 2009)
34. Yoshiyasu, Y., Yoshida, E., Guibas, L.: Symmetry aware embedding for shape correspondence. Computers amp; Graphics **60**, 9–22 (Nov 2016)