MidSurfer: Efficient Mid-surface Abstraction from Variable Thin-walled Models Supplementary Material

Anonymous

1. Theorems and Lemmas

Theorem 1. *Let S_{left} and S_{right} be two surfaces in FG_{left} and FG_{right}, respectively, and let L be a line intersecting both surfaces at points $A \in S_{left}$ and $B \in S_{right}$. There exists at least one point $Q_0 \in L$ between A and B such that Q_0 lies on the mid-surface between S_{left} and S_{right}.*

Proof. For any point $Q \in L$, define the shortest distance functions:

$$\mathrm{Dist}(Q, S_{left}) = \inf_{P \in S_{left}} \|Q - P\|, \quad \mathrm{Dist}(Q, S_{right}) = \inf_{P \in S_{right}} \|Q - P\|. \tag{1}$$

S_{left} and S_{right} are closed sets, so their distance functions $\mathrm{Dist}(Q, S_{left})$ and $\mathrm{Dist}(Q, S_{right})$ are continuous over $\mathbb{R}^3$, thus both distance functions are continuous on L. Construct the function $f(Q) = \mathrm{Dist}(Q, S_{left}) - \mathrm{Dist}(Q, S_{right})$. At endpoints:

$$f(A) = \mathrm{Dist}(A, S_{left}) - \mathrm{Dist}(A, S_{right}) = 0 - \mathrm{Dist}(A, S_{right}) < 0, \tag{2}$$

$$f(B) = \mathrm{Dist}(B, S_{left}) - \mathrm{Dist}(B, S_{right}) = \mathrm{Dist}(B, S_{right}) - 0 > 0, \tag{3}$$

The function $f(Q)$ is continuous on the line segment $[A, B] \subset L$. Since $f(A) < 0$ and $f(B) > 0$, by the Intermediate Value Theorem, there exists at least one point $Q_0 \in (A, B)$ such that:

$$f(Q_0) = 0 \implies \mathrm{Dist}(Q_0, S_{left}) = \mathrm{Dist}(Q_0, S_{right}). \tag{4}$$

Thus, Q_0 lies on the mid-surface. □

Lemma 1 (Convex Projection Invariance)**.** *Let T be a convex triangle in M, and let $A, B \in L$ be two points on a line segment such that their closest points on T lie strictly inside T. Then, for any $Q \in AB$, the closest point $P_Q \in T$ to Q also lies strictly inside T.*

Proof. By convexity of T and projective continuity, the line segment $P_A P_B \subset T$. Parametrize $Q = (1 - \lambda)A + \lambda B$ and define $P_Q = (1 - \lambda)P_A + \lambda P_B$, by convexity, $P_Q \in T$, and the distance of Q to P_Q follows:

$$\mathrm{Dist}_{\min}(\mathbf{Q}, \mathbf{P_Q}) = Q - P_Q = (1 - \lambda)(\mathbf{A} - \mathbf{P}_A) + \lambda(\mathbf{B} - \mathbf{P}_B). \tag{5}$$

Since

$$\mathbf{A} - \mathbf{P}_A \perp T, \quad \mathbf{B} - \mathbf{P}_B \perp T, \tag{6}$$

and the vector $Q - P_Q$ is a linear combination, it remains **orthogonal** to the T at P_Q, thereby ensuring $\mathbf{P}_Q$ is the closest point to Q on T. □

Theorem 2. *Let M_{left} and M_{right} be triangular meshes generated by two face groups (FG), and let L be a line segment intersecting both meshes. Suppose two points $A, B \in L$ (between the intersections of L with M_1 and M_2) are such that:*

1. *A and B share the same closest triangle indices $T_a \in M_{left}$ and $T_b \in M_{right}$.*
2. *T_a and T_b are convex.*

Then, for any point $Q \in AB$, the closest triangle indices of Q to M_{left} and M_{right} remain T_a and T_b, respectively.

Table 1: The statistical results of efficiency at different sampling points and parallelism for different benchmarks (s).

		M2	M3	M4	M5	M6
N=8	parallel	0.031	0.355	**0.068**	0.384	0.644
	single-threaded	0.122	0.973	**0.220**	1.593	2.011
N=16	parallel	**0.029**	**0.252**	0.085	**0.342**	**0.589**
	single-threaded	**0.108**	**0.963**	0.251	**1.363**	**1.801**
N=32	parallel	0.047	0.321	0.105	0.509	0.798
	single-threaded	0.204	1.779	0.368	2.624	3.313

Note: The data in the table represents the results of the Mid-point Extraction stage. Model 1, being a constant wall thickness model, is not listed in the table.

Proof. For M_{left}, by Lemma 1, if the closest points of A and B to T_a lie inside of T_a, then the closest points of all points on the segment AB also lie inside of T_a. If the closest points of A and B lie on the edge or vertex of T_a, then the closest points of points on AB may move along the edge or vertex but still belong to T_a. Therefore, for all points on AB, the index of the closest face to M_{left} is T_a. Same proof for M_{right}. □

2. Performance of Different Sampling Points

We evaluated the computational efficiency in the mid-point extraction based on the number of sampling points and the use of parallel computing, as detailed in Table 1. For all models, the best efficiency was achieved with 16 sampling points (i.e., $N = 16$) and parallel acceleration (the default setting used in the algorithm). Notably, we pre-tested the impact of different thread counts on the algorithm, revealing only a ±2% variation in efficiency. Thus, we used the default thread count (i.e., the CPU's core count). When $N = 8$, the parallel strategy achieved a speedup of $2.7X - 4.2X$, but the overall time was generally longer than with $N = 16$. This is primarily due to insufficient sampling points leading to increased recursion and more iterations in the binary search. For example, in Model 3, with $N = 8$, the average recursion count was 1.32, and the average binary search count was 6.75, compared to 1.12 and 6.03, respectively, with $N = 16$. Similarly, with $N = 32$, the average speedup ranged from $3.5X$ to $5.5X$, demonstrating enhanced parallelism with more sampling points. However, despite the improved parallel efficiency and favorable recursion (1.03) and binary search counts (4.94), this configuration was slower than the $N = 16$ case across all models. This is mainly because excessive sampling points reduce computational efficiency, as more points with no impact on the final result are calculated. An exception in Table 1 is Model 4, which

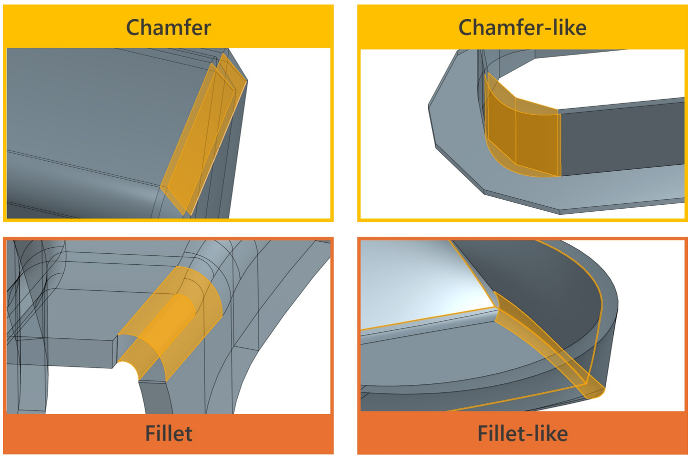

Figure 1: Examples of different transition scenarios, where the highlighted areas are transition areas.

performed better with $N = 8$ than $N = 16$. This is attributed to the gentle transitions in Model 4's variable wall thickness (low curvature variation), allowing fewer sampling points to quickly identify triangular meshes with the same shortest distance. Additionally, the model's low discretization level, resulting in larger average triangular mesh areas, made it easier to hit the same meshes. However, this configuration significantly reduces the geometric accuracy of the generated mid-surface. Based upon this, $N = 16$ was finalized as the default value.

3. Construction of Thin-walled Model Dataset

All models were selected from various datasets using keywords such as "thin-walled parts," "sheet metal parts," and "plastic shells." We searched databases including the Onshape public dataset[1], GrabCAD Library [2], and the ABC dataset [1], covering categories such as aerospace, automotive, industrial design, and machine design. After the initial search, we manually curated the collected models, resulting in a dataset of 213 representative models comprising 7894 faces. All models are thin-walled and include various wall thickness types, transition scenarios, and face pair combinations.

Further, the models were manually classified and annotated. To reflect the diversity of the database, the models were divided into four categories based on their corresponding domains: aerospace, automotive, industrial/mechanical design, and others, with the quantity and proportion of each category presented in Table 2. Moreover, To illustrate the dataset's composition, the count of constant and variable wall thickness models were recorded, and categorized into 1-1 and 1-n/n-n FGP types, along with various transition scenarios, such as chamfers(-like), fillets(-like) structures (see Fig. 1). These statistics highlight the dataset's complexity, as shown in Table 3. Moreover, 12 representative models and their extracted mid-surfaces were visually displayed with corresponding model data in Figs. 2-3, showcasing the diversity and complexity of the model dataset across different domains, wall thicknesses, and FGP types, ensuring the dataset's validity for evaluation.

Table 2: The statistical data for each domain of the model in the dataset.

	Count	Percentage
Aerospace	45	21.12%
Automotive	52	24.41%
Industry/Machine Design	75	35.21%
Others	41	19.25%
Total	213	-

Table 3: The statistical data of each face group pair (FGP) type and transition scenarios in the dataset.

	Constant Wall-thickness		Variable Wall-thickness	
	Count	Percentage	Count	Percentage
1-1 *FGPs*	38	17.84%	30	14.08%
1-n/n-n *FGPs*	27	12.68%	118	55.40%
Chamfer/Fillet (-like)	22	10.33%	121	56.81%
Total	65	30.52%	148	69.48%

References

[1] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, D. Panozzo, ABC: A big CAD model dataset for geometric deep learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9601–9611.

[1] https://cad.onshape.com/documents?nodeId=3&resourceType=filter

[2] https://grabcad.com/library

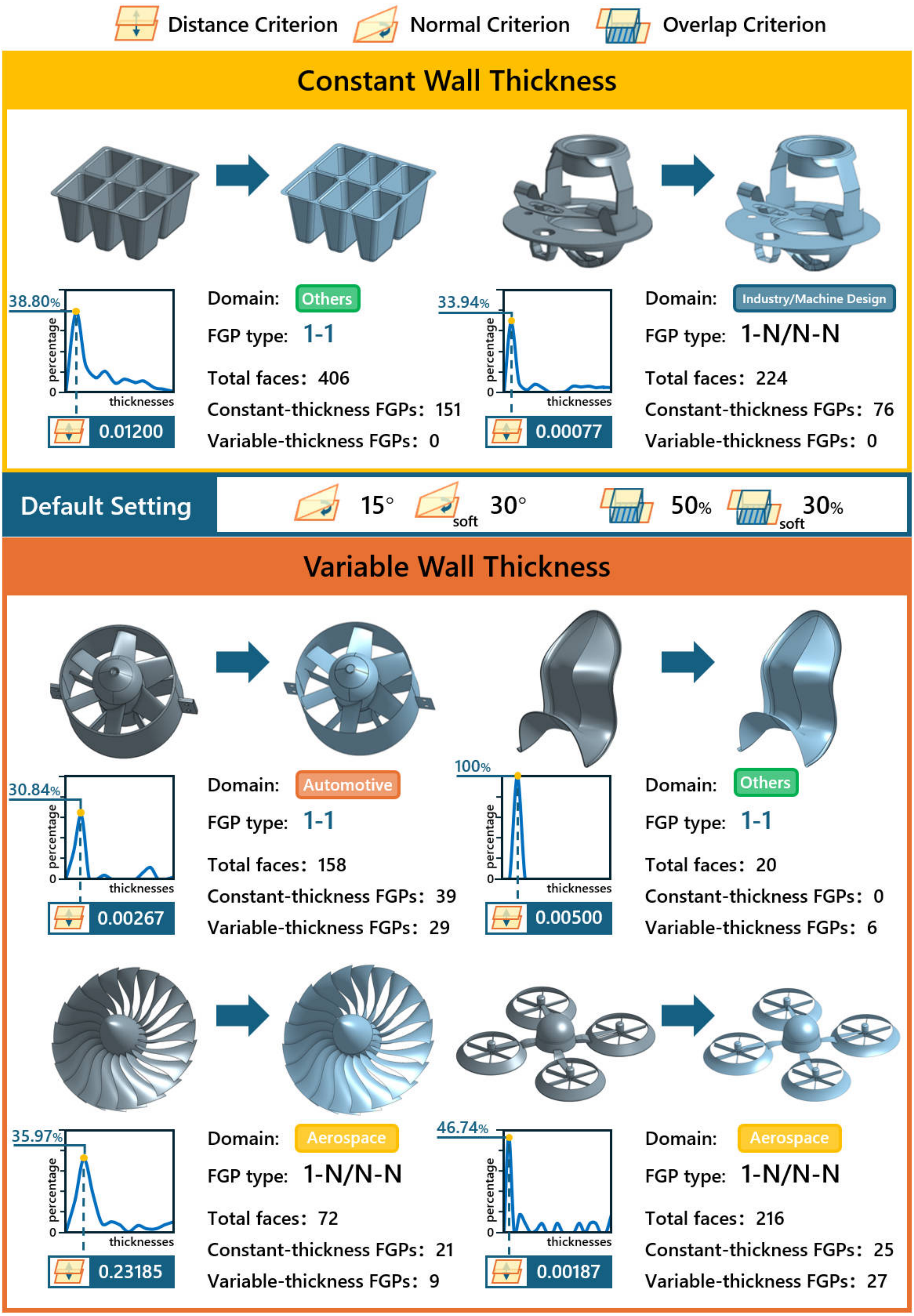

Figure 2: Visual examples of 12 representative models in the dataset. The yellow boxes indicate constant wall-thickness models, and the orange boxes indicate variable wall-thickness models. The domain, FGP type, and parameter settings for each model are listed. The blue boxes show the default parameter settings of our method.

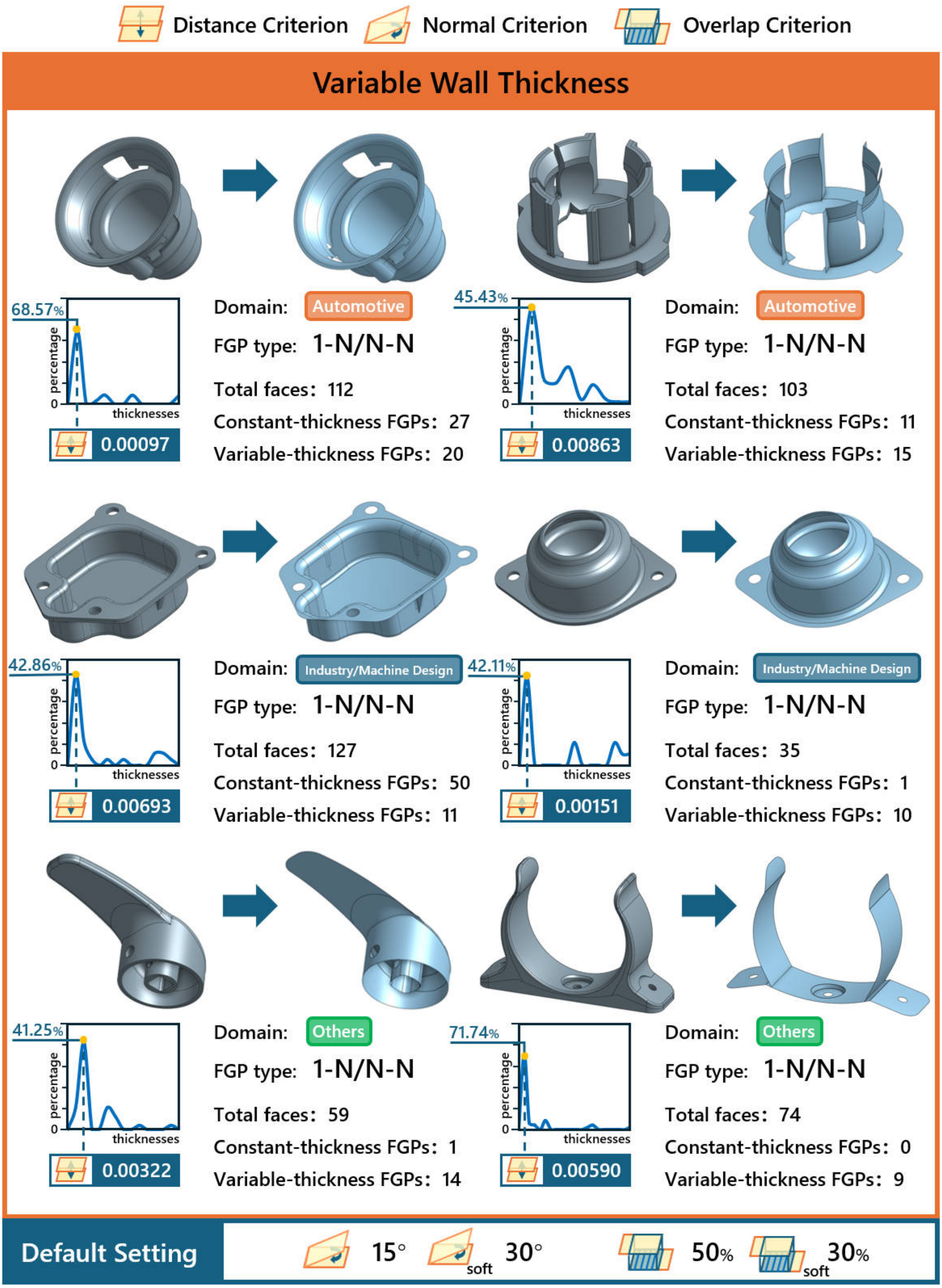

Figure 3: Visual examples of 12 representative models in the dataset. The orange boxes indicate variable wall-thickness models. The domain, FGP type, and parameter settings for each model are listed. The blue boxes show the default parameter settings of our method.