

## Article

# A Data-Driven Parametric Framework for Size-Adaptive Shoe Insole Outline Generation

Ga Eun Lee <sup>1</sup>, Minjun Kim <sup>1</sup>, Jeong Hyeon Lee <sup>1</sup>, Jiwon Kim <sup>2</sup>, Sukwon Lee <sup>3,\*</sup> and Changgu Kang <sup>1,\*</sup>

<sup>1</sup> Department of Computer Science and Engineering, Gyeongsang National University, Jinju 52828, Republic of Korea; rkdsms0623@gnu.ac.kr (G.E.L.); dunta@gnu.ac.kr (M.K.); ljhyun0821@gnu.ac.kr (J.H.L.)

<sup>2</sup> Information Strategy Division, MESSE ESANG, 9, World Cup buk-ro 58-gil, Mapo-gu, Seoul 04516, Republic of Korea; newkimjiwon@gmail.com

<sup>3</sup> Department of AI Content Engineering, Kangwon National University, Wonju 26403, Republic of Korea

\* Correspondence: sukwonlee@kangwon.ac.kr (S.L.); cgk@gnu.ac.kr (C.K.)

## Abstract

With the rapid expansion of AR/VR-based digital platforms, there is an increasing demand for the automated generation of size-varied 3D shoe assets. However, conventional CAD-based linear scaling and PCA-based global statistical shape models are limited in their ability to capture the non-uniform and locally non-linear deformations observed in insole contours. This study proposes a size-adaptive insole contour generation framework that integrates image-based contour extraction, B-spline parametric representation, and type-specific SVR-based local displacement regression. By decomposing control-point displacements into tangent–normal components, the proposed method directly models non-linear curvature variations associated with size progression without relying on dimensionality reduction. Quantitative evaluations under an eight-fold leave-one-insole-out (LOIO) protocol show that the proposed method achieves a mean Hausdorff distance of 4.73 mm, a Chamfer distance of 1.76 mm, and an IoU of 0.929. It significantly outperforms the no-clustering ablation in both the Hausdorff distance (6.13 mm;  $p = 0.032$ ) and IoU ( $p = 0.033$ ), as well as the PCA-based kernel ridge regression baseline across all three metrics ( $p < 0.05$ ). No statistically significant differences were observed between the proposed method and the ratio scaling, Gaussian process, or thin plate spline baselines ( $p > 0.05$ ). PCA-Linear showed a small numerical advantage in the Hausdorff distance (4.16 mm), but the difference was not statistically significant ( $p = 0.187$ ). A sensitivity analysis further reveals the existence of a practical control-point density region that balances geometric fidelity and model complexity. This work provides a data-driven parametric approach for standard size-based digital grading automation and establishes a technical foundation for future extension to full 3D shoe mesh generation.



Academic Editor: Christos Bouras

Received: 8 August 2026

Revised: 7 September 2026

Accepted: 8 September 2026

Published: 11 September 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

**Keywords:** B-spline modeling; digital grading; insole contour generation; support vector regression

## 1. Introduction

The recent digital transformation in the manufacturing and design industries has expanded toward internet-based information systems, including augmented reality (AR), virtual reality (VR), and metaverse platforms [1–7]. In digital platforms for footwear and apparel, it is essential to systematically generate multiple size variations of 3D digital assets to accommodate diverse avatars and user profiles [8–11].

The automated generation of such digital assets extends beyond the mere virtualization of physical products; it constitutes a core information system technology that enables data-driven design decision-making and production automation. Accordingly, there is an increasing demand for automated techniques capable of accurately reproducing real-world grading processes in digital environments.

In traditional apparel grading, uniform scaling is insufficient to preserve the original silhouette and fit because growth rates vary across different body regions [12,13]. Similar non-linear characteristics are observed in footwear, where geometric attributes such as the foot length, forefoot width, and instep height deform at different rates as the size changes [14]. In footwear manufacturing, the last—a three-dimensional mold representing the foot—plays a critical role in determining the overall shape and fit of a shoe. The bottom last contour is functionally equivalent to the two-dimensional insole outline and defines the boundary condition of the three-dimensional shoe geometry [15]. Therefore, the accurate transformation of the insole outline is a fundamental requirement for constructing high-quality 3D digital footwear assets.

However, most industrial CAD-based digital design systems still rely on linear scaling formulas or simple rule-based processing with parallel translations. Such approaches fail to adequately capture the non-linear geometric variations inherent in footwear, often resulting in contour distortions across sizes and requiring manual correction by designers. This not only introduces quality variations depending on individual expertise but also reduces data compatibility and reusability within CAD/CAM information processing pipelines, as curve parameterization is not standardized [8]. Consequently, manually intensive workflows fundamentally limit the scalability and automation of digital manufacturing systems.

Meanwhile, recent studies leveraging 3D scanning technologies have primarily focused on custom-made designs tailored to individual foot geometries [16]. Although this approach is effective for achieving a personalized fit, it presents limitations from an information systems perspective, as it requires acquiring and processing 3D scan data for each user. Furthermore, compatibility with standardized size systems is limited, and the approach is not well suited for mass production environments, where multiple size variations must be generated rapidly from a single base design. In other words, prior research has largely emphasized personalized design automation, whereas data-driven approaches for standardized size-based digital grading remain underexplored.

In parallel, recent advances in computer vision and machine learning have actively explored image-based shape processing and data-driven geometric deformation prediction. Deep learning-based methods, in particular, have demonstrated strong capabilities in modeling complex non-linear deformations [17–20]. However, acquiring large-scale shoe insole contour data remains challenging in practice due to difficulties in non-destructive data acquisition and limited access to proprietary manufacturer design data.

To address these constraints, this study adopts a data-driven parametric shape modeling approach combined with kernel-based regression, which is capable of operating effectively under limited sample conditions. The proposed framework serves as a lightweight information processing module that supports the automated generation of standardized size variations within digital footwear design systems.

This study proposes a data-driven parametric shape modeling framework for size-adaptive insole outline generation in digital footwear design environments. The proposed framework integrates image-based contour extraction, B-spline-based parametric representation, and type-specific local displacement regression. By representing all shapes in a unified control-point space and directly modeling tangent-normal displacements without dimensionality reduction, the method effectively captures non-uniform and locally non-linear geometric variations across size changes.

Unlike conventional global scaling or PCA-based latent-space approaches, the proposed framework preserves local curvature characteristics and provides more stable shape prediction under limited data conditions. In addition, the parametric formulation enables straightforward extension to full 3D shoe mesh grading, supporting practical CAD/CAM applications.

The remainder of this paper is organized as follows. Section 2 reviews related studies, Section 3 describes the proposed size-adaptive insole outline generation framework in detail, Section 4 presents the experimental evaluations, and Section 5 concludes the paper and discusses future research directions.

## 2. Related Work

### 2.1. Digital Fashion Asset Modeling

With the rapid expansion of digital platforms such as AR, VR, XR, and the metaverse, the industrial and technological demand for personalized digital fashion production has steadily increased. The widespread adoption of virtual fitting, real-time style visualization, and interactive product customization in online environments has accelerated the transition from traditional manual apparel and footwear production processes toward digital workflows. A primary driver of this shift is the need to reduce the cost and production cycle associated with physical prototyping. Consequently, 3D-based virtual prototyping and automated digital asset generation have emerged as key enabling technologies. Recent studies have proposed various approaches to rapidly generating high-quality digital fashion assets using scan-based 3D modeling, data-driven shape deformation, and neural rendering-based reconstruction techniques [21–25].

ClothCap [21] introduced a method for capturing the motion of garments worn by real users as 4D sequences and retargeting them onto parametric human body models, thereby establishing a foundation for realistic dynamic garment reconstruction. DeepWrinkles [22] combined a PCA-based garment deformation space with a deep learning-based wrinkle refinement model to reconstruct high-frequency clothing details, demonstrating the feasibility of generating high-quality 3D apparel assets. TailorNet [23] provided a data-driven garment deformation model that generates diverse clothing geometries in real time based on the user body shape, pose, and style conditions, supporting large-scale personalized virtual try-on environments.

In the footwear domain, Seul et al. [24] presented an automated shoe last generation method based on 3D foot scans and anthropometric measurements, facilitating the digitalization of traditional manufacturing workflows and simplifying size grading. Furthermore, Chen et al. [25] proposed a neural volumetric rendering approach to reconstruct 3D clothing models from a single image, demonstrating a viable solution for high-quality asset generation under limited data conditions. Collectively, these studies indicate a paradigm shift from purely geometry-based modeling toward data-driven shape modeling in digital fashion and footwear production, highlighting the need for efficient and accurate geometric transformation techniques capable of handling size variations.

However, most existing approaches primarily focus on the reconstruction or deformation of three-dimensional surface-based assets such as garments or shoe lasts. They do not explicitly address fine-grained geometric transformations of curve-based two-dimensional structures such as insole outlines. Moreover, many methods rely on complex mesh or volumetric representations or on measurement-based uniform scaling strategies, which make it difficult to precisely control curvature variations and localized geometric features across different sizes.

In contrast, the proposed framework represents insole outlines using continuous B-spline-based parametric curves and directly predicts local control-point displacements

through kernel-based regression without dimensionality reduction such as PCA. This formulation enables the accurate modeling of size-dependent geometric transformations, including subtle curvature variations along the tangent and normal directions, rather than simple global scaling.

## 2.2. Statistical Shape Modeling

Statistical shape models (SSMs) have become a standard data-driven approach for analyzing and deforming geometric objects across various domains, including facial modeling, human body analysis, and medical imaging. SSMs typically align a set of shapes into a common reference frame and represent their variability using low-dimensional statistical models such as principal component analysis (PCA). By summarizing high-dimensional shape variations into a small number of latent coefficients, SSMs enable the efficient generation of new shapes under different conditions and provide stable modeling performance even in limited data scenarios. Representative examples include the Active Shape Model (ASM) [26] and Active Appearance Model (AAM) [27].

Blanz and Vetter's 3D Morphable Model [28] introduced a PCA-based shape space for synthesizing three-dimensional facial geometries, forming the foundation for subsequent parametric modeling approaches. In human body modeling, PCA-based statistical representations have also been widely adopted. Allen et al. [29] constructed a human body shape space by aligning range scan data to a high-resolution template mesh through optimization-based registration. The SCAPE model [30] and the SMPL model [31] further demonstrated that human shape and pose variations can be efficiently represented within a linear latent space.

More recent studies have sought to extend statistical models to generate more realistic and physically plausible deformations. For instance, Kim et al. [32] proposed a dual-layer human body model that integrates a volumetric statistical body model (VSMPL) with a finite element method (FEM)-based physical layer, addressing the limitations of purely linear PCA-based models and enabling more realistic soft-tissue deformation.

Although SSM-based approaches effectively capture global modes of shape variation, they exhibit limitations in directly controlling local geometric characteristics such as curvature, contour sharpness, and fine-grained edge structures [30,31]. In scenarios requiring complex silhouette transformations, linear PCA models may introduce local distortions or unnatural deformations. To address these limitations, recent research has explored the combination of PCA-based shape spaces with geometric constraints or the incorporation of non-linear regression models such as Gaussian process regression (GPR), kernel ridge regression (KRR), and neural deformation networks to generate more flexible shape transformations [33–35].

However, most non-linear extensions focus primarily on complex three-dimensional surface modeling, while optimization strategies for two-dimensional contour deformation remain relatively underexplored. To address the information loss introduced by PCA-based dimensionality reduction, this study eliminates the dimensionality reduction stage and directly predicts local control-point displacements through kernel-based regression. Unlike global statistical models that emphasize compact latent representations, the proposed approach prioritizes the preservation of local geometric fidelity, making it well suited for modeling subtle curvature variations in insole outlines.

## 3. Proposed Framework

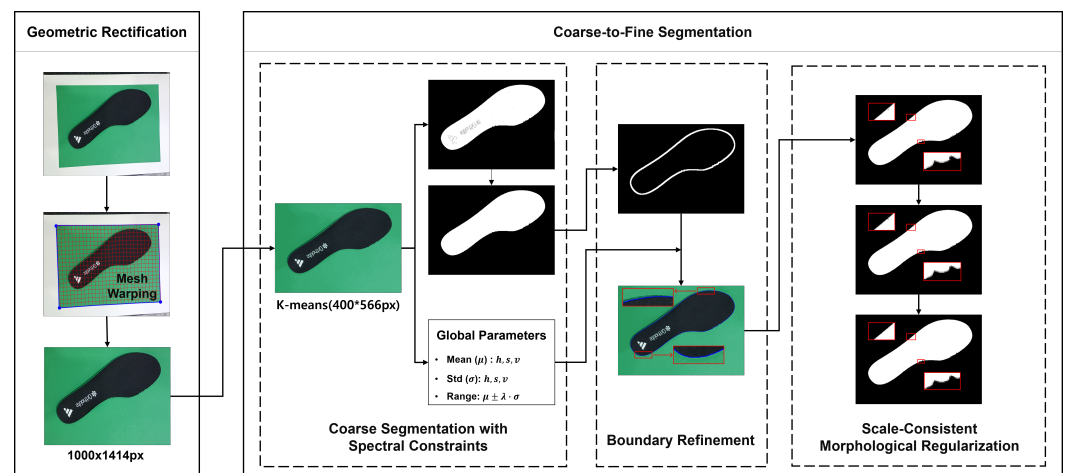
This section presents the proposed size-adaptive insole outline generation framework, which consists of three stages. Section 3.1 describes the data preparation stage, in which geometrically aligned insole contours are extracted from captured images through a coarse-

to-fine segmentation and rectification pipeline. Section 3.2 describes the size-adaptive outline generation stage, in which the rectified contours are parameterized using B-splines to enable controlled interpolation across insole sizes. Section 3.3 describes the type-specific displacement regression stage, in which insole designs are grouped into geometrically consistent types using hierarchical clustering and type-specific support vector regression models are used to predict tangent–normal displacements while enforcing monotonic size progression.

### 3.1. Data Preparation

The objective of the data preparation stage is to reliably extract geometrically aligned insole contours from captured images. To ensure consistent shape processing in the subsequent shape generation stage, geometric distortions caused by the camera perspective, background deformation, and environmental conditions are corrected, and all input images are aligned to a standardized reference space.

To this end, we design a contour acquisition pipeline consisting of (1) geometric rectification for spatial normalization and (2) a coarse-to-fine segmentation strategy that preserves topological consistency and boundary precision. Figure 1 illustrates the overall workflow of the data preparation stage.



**Figure 1.** Overview of the data preparation pipeline, consisting of geometric rectification for spatial alignment and a coarse-to-fine segmentation strategy for robust insole contour extraction. Red boxes and lines indicate the regions selected for enlarged views of the extracted contour.

#### 3.1.1. Geometric Rectification

To correct geometric distortions caused by background deformation (e.g., green screen bending) and perspective effects during image acquisition, we construct a unified preprocessing pipeline that integrates geometric rectification and coarse-to-fine segmentation within a standardized coordinate framework.

First, a Coons patch-based mesh warping algorithm is applied to compensate for non-linear geometric distortions in the input image. Background regions are identified using HSV color space masking, from which the largest contour corresponding to the background boundary is extracted. The HSV space is adopted for this purpose because it separates chromatic information (hue) from the illumination intensity (value), making background identification more robust to the non-uniform lighting and shadow conditions inherent in the chroma-key acquisition setup than RGB, in which color and brightness are entangled across all channels. Instead of directly approximating the contour with a simple polygon, contour points are classified into four directional groups, and robust line fitting with the Huber loss is performed to estimate noise-resistant virtual corner points (robust

corners). Based on the detected four corners, the contour is partitioned into four boundary curves (top, bottom, left, and right), each of which is smoothed using one-dimensional Gaussian filtering ( $\sigma = 10$ ) to suppress local edge noise.

The choice of  $\sigma = 10$  is justified via a sensitivity analysis in Section 4.1.

Since the acquisition protocol places the insole on an A4-sized rectangular backdrop, the background contour is assumed to be approximately rectangular; the “irregularities” addressed here are local deviations from this rectangular premise (e.g., slight bending of the backdrop, shadows, or edge detection noise) rather than arbitrary background shapes. Under this premise, the four boundaries are identified through a robust estimation procedure: a rough minimum-area bounding rectangle is first computed from the detected contour, and every contour point is assigned to its nearest side of this rectangle. For each of the four resulting point groups, a boundary line is then fitted using a Huber loss function (`cv2.fitLine` with `DIST_HUBER`), which down-weights outlier points relative to ordinary least-squares fitting. The four robustly fitted lines are intersected pairwise to obtain the final corner points, with the initial bounding rectangle corners used as a fallback in degenerate cases where line intersection fails.

The sensitivity of the segmentation accuracy to the injected corner localization error is characterized in Section 4.1.

Next, a target grid with a fixed resolution ( $1000 \times 1414$  pixels), following the aspect ratio of A4 paper (1:1.414), is generated. The four boundary curves are combined using linear and bilinear interpolation within the Coons patch formulation to remap the source image pixels onto the target grid. This boundary-driven interpolation scheme propagates boundary curvature information to interior pixels, effectively transforming non-linear distortions into a planar canonical space. Simultaneously, the region of interest is maximized to ensure that the object occupies the available canvas area.

The target aspect ratio (1:1.414) was chosen to match the physical A4-sized backdrop used during image acquisition ( $210 \text{ mm} \times 297 \text{ mm}$ ), preserving the true proportions of the capture setup without introducing artificial distortion; the choice of the absolute resolution (1000 px short side) is justified via a quantitative sensitivity analysis in Section 4.1.

Unlike a simple homography, which relies on four point correspondences and produces only a linear (projective) mapping, the Coons patch formulation interpolates directly from the four detected boundary curves—which may themselves be locally curved due to non-planar background deformation (e.g., bending of the chroma-key backdrop during acquisition)—allowing the rectification to compensate for such non-linear distortions, which a point-based projective transform cannot capture. A quantitative comparison confirming this benefit against a homography baseline is provided in Section 4.1.

As a result, all input images are standardized to share a consistent pixel density and coordinate scale, independent of the camera distance or zoom level. This normalization guarantees geometric equivalence across samples and provides a stable input representation for subsequent regression modeling.

### 3.1.2. Coarse-to-Fine Segmentation

After geometric rectification in the standardized reference space, directly performing pixel-level segmentation on high-resolution images ( $W_{orig} > 3000 \text{ px}$ ) would incur excessive computational and memory costs. To address this issue, we adopt a coarse-to-fine segmentation strategy that simultaneously preserves boundary precision and topological consistency. The proposed segmentation process consists of the following stages.

**Coarse Segmentation with Spectral Constraints:** The rectified input image is first downsampled to a low-resolution proxy ( $W_{proc} = 400 \text{ px}$ ) to reduce the computational complexity. A sensitivity analysis validating this choice is provided in Section 4.1. Consid-

ering the chroma-key acquisition setup, the green spectral range ( $h \in [35, 95]$ ) is restricted as background candidates. This range was empirically determined from the measured hue distribution of the chroma-key backdrop across the captured dataset, with a margin added to accommodate lighting variations. To mitigate shadow-induced misclassification, we introduce an asymmetric safety lower bound defined as

$$\tau_{low} = \max(\mu_{bg} - \lambda \cdot \sigma_{bg}, \delta_{hard}), \quad (1)$$

where  $\mu_{bg}$  and  $\sigma_{bg}$  denote the mean and standard deviation of the background intensities, respectively.

The proxy image is then transformed into the  $L^*a^*b^*$  color space, followed by K-means clustering. Clustering is performed using the K-means++ initialization scheme (cv2.KMEANS\_PP\_CENTERS) with 10 independent attempts (retaining the result with the lowest compactness), a convergence criterion of 20 iterations or an epsilon of 1.0, and a fixed random seed to ensure reproducibility; the cluster count  $K$  is a tuned hyperparameter, reported in Table 1. Clusters violating the spectral constraints are classified as foreground to generate an initial segmentation mask  $M_{init}$ . Formally, this classification (ClusterFilter in Algorithm 1) compares each cluster's mean hue, saturation, and value against per-channel bounds derived from the background statistics—the value-channel lower bound following Equation (1), with analogous bounds for hue and saturation—retaining a cluster as foreground ( $S_{fg}$ ) if its mean falls outside the bounds on any channel. To remove small noise artifacts and fill small holes, morphological opening is applied first, followed by closing, using structuring elements of size  $k_{open}$  and  $k_{close}$  (both tuned hyperparameters, reported in Table 1):

$$M_{open} = (M_{init} \ominus B_{k_{open}}) \oplus B_{k_{open}}, \quad (2)$$

$$M_{close} = (M_{open} \oplus B_{k_{close}}) \ominus B_{k_{close}}. \quad (3)$$

Only the largest connected component (LCC) of  $M_{close}$  is retained, yielding the coarse segmentation mask  $M$ . This assumes that the insole forms a single, spatially contiguous foreground region after segmentation; an empirical assessment of this assumption is provided in Section 4.1.

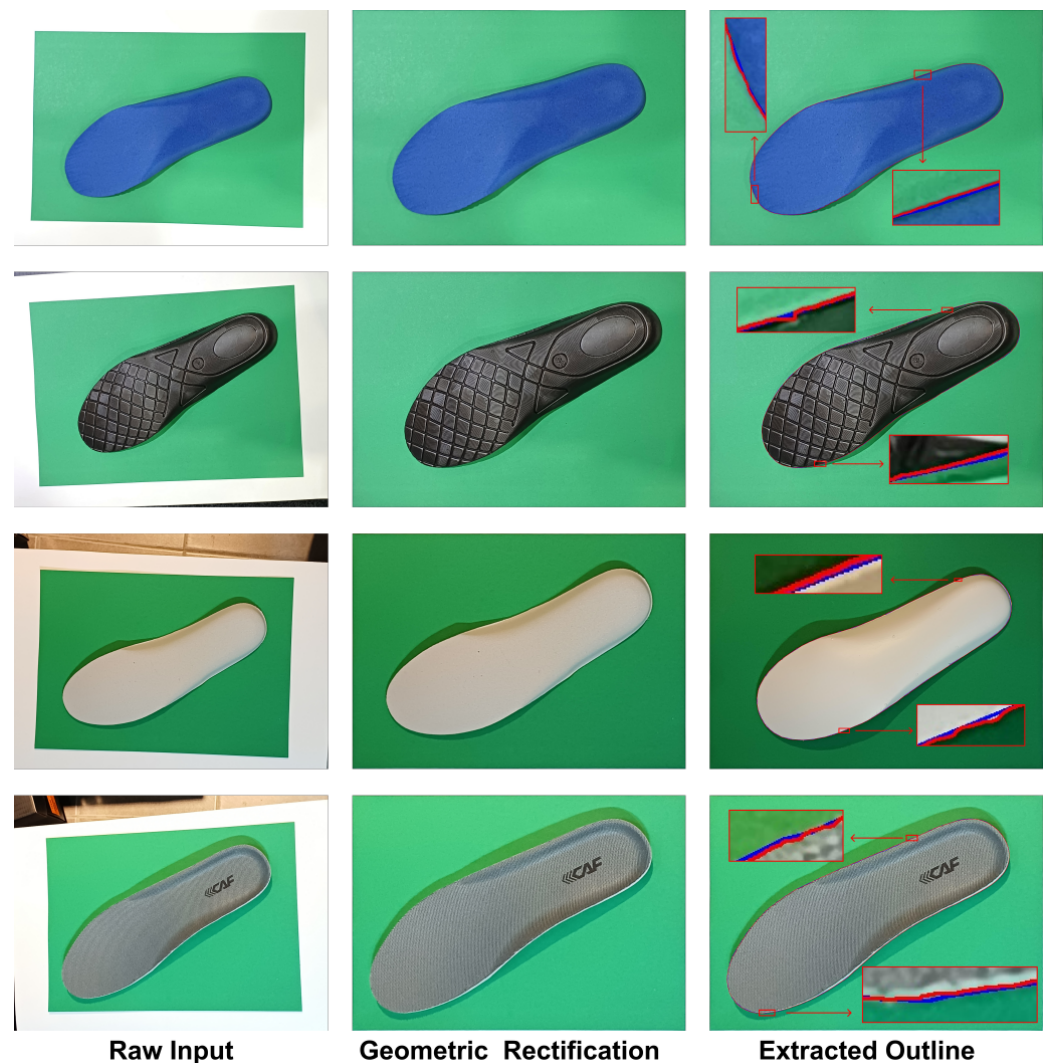
**Boundary Refinement:** To alleviate staircase artifacts introduced by low-resolution segmentation, we perform localized boundary reclassification. Using a circular structuring element  $B_{circ}$ , an uncertainty band is defined as

$$\Omega_b = (M \oplus B_{circ}) \setminus (M \ominus B_{circ}), \quad (4)$$

where  $M$  denotes the coarse segmentation mask. Only pixels within  $\Omega_b$  are reclassified using full-resolution  $L^*a^*b^*$  values. The cluster centroids obtained during the coarse stage are reused to prevent overfitting in boundary regions.

**Scale-Consistent Morphological Regularization:** Since geometric rectification aligns all input images to a standardized resolution, fixed-size morphological kernels can be consistently applied across samples. Kernel sizes determined via Bayesian optimization are used to suppress structural noise, followed by topological filtering to produce a stable and clean insole contour. This topological filtering step (TopologicalFilter in Algorithm 1) reapplies the same morphological opening–closing procedure used in the coarse segmentation stage (using the same kernel sizes  $k_{open}$ ,  $k_{close}$ ) to the boundary-refined mask, immediately prior to final contour extraction.

The complete data preparation procedure is summarized in Algorithm 1, and qualitative results are presented in Figure 2.



**Figure 2.** Qualitative results of the shape acquisition pipeline. For each insole sample, the columns show (left) the raw input image, (middle) the geometrically rectified image via Coons patch mesh warping, and (right) the extracted insole outline (red) overlaid with the ground truth contour (blue).

**Table 1.** Optimized hyperparameters.

| Stage        | Parameter              | Symbol      | Optimized Value |
|--------------|------------------------|-------------|-----------------|
| Segmentation | Cluster Count          | $K$         | 10              |
|              | Std. Multiplier        | $\lambda$   | 4.6             |
|              | Saturation Lower Sigma | $\sigma_S$  | 4.3             |
|              | Value Lower Sigma      | $\sigma_V$  | 7.6             |
|              | Saturation Min Floor   | $\tau_S$    | 66.0            |
|              | Value Min Ratio        | $\tau_V$    | 0.43            |
|              | Hue Margin             | $\delta_H$  | 15.0            |
| Refinement   | Morphological Open     | $k_{open}$  | 3               |
|              | Morphological Close    | $k_{close}$ | 9               |
|              | Boundary Kernel Size   | $k_{bound}$ | 11              |

**Algorithm 1** Data Preparation**Input:** Raw image  $\mathbf{I}_{raw}$ **Output:** Rectified image  $\mathbf{I}_{rect}$ , contour  $\mathcal{C}_{insole}$ **Phase A: Geometric Rectification**

- 1:  $M_{bg} \leftarrow \text{HSVMask}(\mathbf{I}_{raw})$
- 2:  $\mathcal{C}_{mat} \leftarrow \text{LargestContour}(M_{bg})$
- 3:  $\{\mathcal{L}_k\}_{k=1}^4 \leftarrow \text{RobustLineFit}(\mathcal{C}_{mat})$  ▷ Huber Loss
- 4:  $\{\mathbf{v}_k\} \leftarrow \text{Intersect}(\{\mathcal{L}_k\})$
- 5:  $\mathcal{B}_{side} \leftarrow \text{SplitAndSmooth}(\mathcal{C}_{mat}, \{\mathbf{v}_k\})$
- 6:  $\mathcal{M}_{map} \leftarrow \text{CoonsPatch}(\mathcal{B}_{side})$
- 7:  $\mathbf{I}_{rect} \leftarrow \text{Remap}(\mathbf{I}_{raw}, \mathcal{M}_{map})$

**Phase B: Coarse-to-Fine Segmentation**

- 8:  $\mathbf{I}_{small} \leftarrow \text{Resize}(\mathbf{I}_{rect}, W_{proc})$
- 9:  $\mathcal{C}, \mathcal{L} \leftarrow \text{KMeans}(\text{Lab}(\mathbf{I}_{small}), K)$
- 10:  $\mathcal{S}_{fg} \leftarrow \text{ClusterFilter}(\mathcal{C}, \tau_{low})$
- 11:  $\mathbf{M}_{coarse} \leftarrow \text{MaskFromClusters}(\mathcal{L}, \mathcal{S}_{fg})$
- 12:  $\mathbf{M}_{coarse} \leftarrow \text{Morphology}(\mathbf{M}_{coarse})$
- 13:  $\mathbf{M}_{coarse} \leftarrow \text{LargestComponent}(\mathbf{M}_{coarse})$  ▷ preliminary LCC pass; computational optimization, subsumed by the final LARGESTCONTOUR step below
- 14: **for**  $p \in \Omega_b$  **do**
- 15:    $\ell_p \leftarrow \text{NearestClusterLabel}(p, \mathcal{C})$  ▷ nearest center in Lab space
- 16:   **if**  $\ell_p \in \mathcal{S}_{fg}$  **then**
- 17:      $\mathbf{M}_{coarse}[p] \leftarrow 1$  ▷ explicit in-place update
- 18:   **else**
- 19:      $\mathbf{M}_{coarse}[p] \leftarrow 0$
- 20:   **end if**
- 21: **end for**
- 22:  $\mathbf{M}_{final} \leftarrow \text{TopologicalFilter}(\mathbf{M}_{coarse})$
- 23:  $\mathcal{C}_{insole} \leftarrow \text{LargestContour}(\mathbf{M}_{final})$
- 24: **return**  $\mathbf{I}_{rect}, \mathcal{C}_{insole}$

## 3.1.3. Parameter Optimization

To ensure geometric stability and the reproducibility of the contour extraction process, we conducted automated hyperparameter optimization over ten key parameters. The optimization was performed using Optuna, and the objective function was designed as a weighted hybrid metric to mitigate sensitivity to outliers that may arise when relying solely on the average performance.

$$\mathcal{J} = 0.7 \cdot \mu_{\text{BF1}} + 0.3 \cdot \min(\text{BF1}) \quad (5)$$

Here,  $\mu_{\text{BF1}}$  denotes the mean boundary F1 score across the dataset, while  $\min(\text{BF1})$  reflects the worst-case performance. This hybrid formulation balances overall segmentation accuracy with robustness against failure cases, encouraging consistent performance across samples rather than optimizing for the average accuracy alone.

The weighting coefficients (0.7, 0.3) in Equation (5) were chosen to balance average-case performance with worst-case robustness; a post hoc sensitivity analysis characterizing this weighting choice is reported in Section 4.1.

The hyperparameter optimization was performed using the Optuna framework with a Tree-Structured Parzen Estimator (TPE) sampler (200 startup trials, random seed fixed at 1) over 2000 trials. The search space spanned ten segmentation parameters:  $K \in [7, 16]$ ,  $\lambda \in [2.0, 6.0]$ ,  $\sigma_S \in [1.0, 5.0]$ ,  $\sigma_V \in [1.0, 8.0]$ ,  $\tau_S \in [0, 90]$ ,  $\tau_V \in [0.2, 0.7]$ ,  $\delta_H \in [0, 15]$ ,  $k_{open} \in \{1, 3, \dots, 15\}$ ,  $k_{close} \in \{1, 3, \dots, 17\}$ ,  $k_{bound} \in \{5, 7, \dots, 25\}$ . A median pruning strategy (20 startup trials, 5 warm-up steps, evaluated at every step) was applied to terminate clearly unpromising trials early.

This procedure was applied to the training set (22 images), whose ground truth contours were manually annotated via polygon labeling; the resulting configuration is summarized in Table 1. A held-out evaluation of this configuration is reported in Section 4.1.

Hyperparameter importance analysis of the optimized configuration is reported in Section 4.1.

### 3.2. Size-Adaptive Shoe Insole Outline Generation

This section describes the shape generation module of the proposed data-driven parametric framework. The module generates a target-size insole outline from a reference insole shape of a single size. Unlike simple global scaling, the insole geometry exhibits non-uniform local deformation as the size varies. Because PCA-based dimensionality reduction primarily captures the dominant global modes of variation, it may not sufficiently represent fine-grained local shape changes.

To address this limitation, the proposed method introduces type-specific kernel-based local displacement regression. Insole types are first defined via shape similarity-based clustering, and, for each type, local control-point displacements are learned as a function of size. This formulation enables the stable prediction of non-uniform and non-linear geometric transformations across sizes.

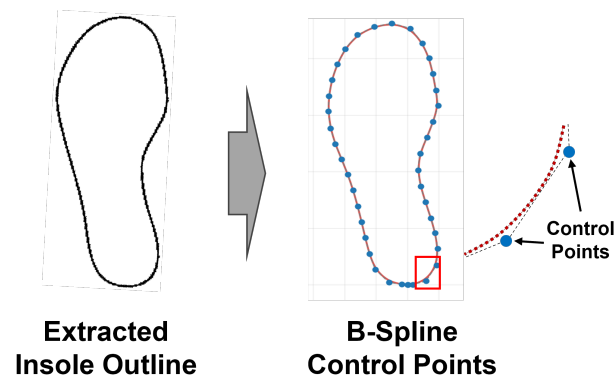
The proposed method consists of two stages: a training stage and a shape generation stage. In the training stage, type-specific regression models are learned. In the shape generation stage, the trained models are used to synthesize the target-size outline. To ensure consistent shape comparison and regression learning, all insole contours are parameterized using a B-spline model with an identical degree and number of control points. Consequently, each shape is represented within a fixed-dimensional control-point space, enabling stable regression modeling and geometric comparison.

Specifically, we employ a cubic B-spline (degree  $d = 3$ ) defined over an open uniform knot vector, with control points obtained by regularized least-squares fitting to chord length-parameterized boundary pixels. To establish point-wise correspondence across shapes with different orientations, we apply a cyclic alignment process that evaluates all circular permutations as well as the reversed ordering of the control-point indexing and selects the permutation that minimizes the Euclidean distance after centering both shapes at their geometric centroids. This alignment step is applied uniformly to all shapes used in both training and inference and is an integral part of the proposed method.

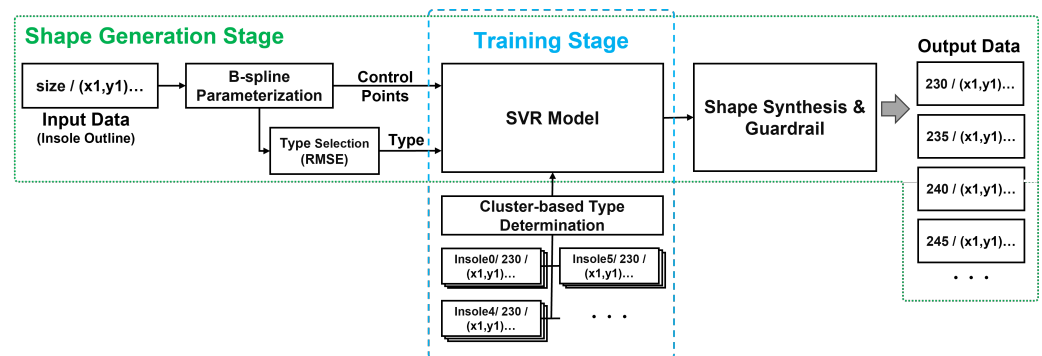
Figure 3 illustrates the B-spline parameterization process. The discrete contour extracted from the input image is converted into a continuous parametric curve defined by uniformly structured control points. The enlarged region demonstrates how the B-spline representation smoothly approximates the local boundary geometry.

Figure 4 presents the overall pipeline of the proposed method. During the training stage, insole outlines represented in the control-point space are clustered based on geometric similarity to define insole types. For each type, a support vector regression (SVR) model is constructed to learn the relationship between size and local control-point displacements.

In the shape generation stage, the same B-spline parameterization is applied to a new input contour. The insole type is determined via RMSE-based type selection in the control-point space. The trained SVR model corresponding to the selected type predicts local control-point displacements for the target size. These predicted displacements are applied to the reference control points to synthesize the final outline. A guardrail validation step is subsequently performed to enforce geometric constraints and prevent physically implausible distortions.



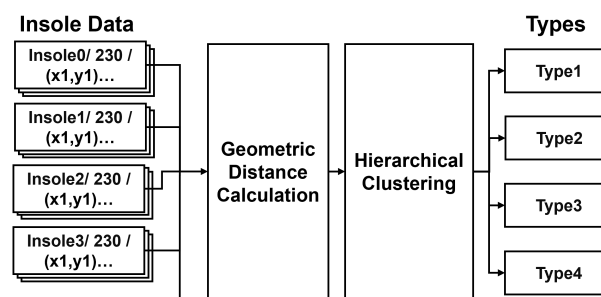
**Figure 3.** B-spline parameterization of the insole outline. The extracted discrete contour (left) is converted into a continuous parametric representation defined by uniformly structured control points (right). Blue points denote the control points, and the red curve represents the B-spline approximation. The enlarged region illustrates the smooth approximation of the local boundary geometry, with the red box and connector line indicating the selected region.



**Figure 4.** Overall pipeline of the proposed method. The training stage constructs type-specific SVR models from control-point representations, and the shape generation stage performs type selection, displacement prediction, and final outline synthesis with guardrail validation.

### 3.3. Type-Specific Displacement Regression and Synthesis

During the training stage, the geometric similarity among insole shapes is quantitatively analyzed to establish a data-driven typology of insole designs. For all pairs of insole samples, control-point-based geometric distances are computed to construct a pairwise distance matrix. Hierarchical clustering with Ward linkage is then applied to group shapes according to geometric similarity. Figure 5 illustrates the cluster-based type determination process.



**Figure 5.** Cluster-based type determination process. Minimum-size insole shapes are compared using RMSE-based geometric distance with cyclic alignment, followed by hierarchical clustering (Ward linkage) to partition the dataset into four geometrically homogeneous types.

The geometric distance used for clustering is defined consistently with the RMSE-based metric employed later in the type selection stage, ensuring methodological coherence between training and inference. To establish  $K = 4$  as a fixed structural hyperparameter, the clustering behavior was first examined on the full 8-design dataset (Table 2). The agglomeration schedule alone does not provide unique justification for  $K = 4$ , as the largest increase in Ward’s dissimilarity cost occurs when merging from five to four clusters (+1.16: 3.36  $\rightarrow$  4.52), exceeding the subsequent four-to-three merge (+0.74: 4.52  $\rightarrow$  5.26). Accordingly,  $K = 4$  is not selected solely based on the agglomeration cost; its selection is further supported by the cluster heterogeneity and downstream ablation analyses presented below. During the actual 8-fold LOIO evaluation, clustering is independently reconstructed within each fold using only the seven-design training pool, ensuring that the held-out test design does not influence its type assignment.

**Table 2.** Clustering validity metrics across different numbers of types ( $K$ ), computed on the full 8-design dataset.  $K = 4$  does not uniquely optimize either index, and its elevated Davies–Bouldin value is associated with the two singleton clusters at this partition level (see discussion below).

| Step (Merge) | Types ( $K$ ) | Ward’s Distance | Silhouette Score | Davies–Bouldin Index |
|--------------|---------------|-----------------|------------------|----------------------|
| 2            | 6             | –               | 0.163            | 0.474                |
| 3            | 5             | 3.36            | 0.159            | 0.597                |
| 4            | 4             | 4.52            | 0.146            | 0.936                |
| 5            | 3             | 5.26            | 0.145            | 0.487                |
| 6            | 2             | 5.87            | 0.291            | 0.446                |

As shown in Table 2, neither the silhouette score nor the Davies–Bouldin index identifies  $K = 4$  as the preferred solution. In particular,  $K = 4$  yields the highest Davies–Bouldin value among the tested configurations (0.936, compared with 0.487 at  $K = 3$  and 0.597 at  $K = 5$ ), while it does not attain the highest silhouette score. This value is associated with the presence of two singleton clusters (Insole02 and Insole07) at  $K = 4$ , highlighting the sensitivity of internal validity indices to cluster size imbalance. Across the tested range ( $K = 2$ – $6$ ), the internal validity indices therefore do not identify a single clustering solution that simultaneously accounts for the geometric and downstream modeling considerations of the proposed framework. Accordingly,  $K = 4$  is not selected solely on the basis of internal clustering validity indices. Instead, it is selected because it preserves Insole02 and Insole07 as geometrically distinct types, consistent with the cluster heterogeneity analysis in Table 3, and because the ablation study in Section 4.6 shows that removing the type-based partitioning (Ablation1, no clustering) significantly degrades the prediction accuracy ( $p = 0.017$ ). We note that this benefit is realized indirectly rather than through Insole02 and Insole07 themselves: under the LOIO protocol, both designs are routed to the ratio scaling fallback when held out, since no other same-type sample remains in the seven-design training pool to support a type-specific SVR model for them. The practical value of retaining them as separate types therefore lies in helping to preserve the geometric homogeneity of the clusters comprising the remaining five designs (Insole00, Insole01, Insole03, Insole04, Insole06), for which the type-specific SVR pathway is used.

For each identified type, a centroid shape  $\bar{\mathbf{P}}^{(k)}$  is computed as the mean of the geometrically aligned control-point representations within the cluster. This centroid is used exclusively during the type selection stage to identify the nearest cluster for a query shape (Algorithm 2, Phase 2, Strategy A) and does not serve as the training reference. Once a candidate set  $\mathcal{T}^*$  is selected, the reference shape  $\mathbf{P}_{\text{ref}}$  used for tangent–normal displacement training (Phase 3) is defined as the individual sample with the minimum size within  $\mathcal{T}^*$ . This ensures that the local displacement basis is anchored to an observed geometry rather

than a synthetically averaged shape. Local control-point displacement data from all samples belonging to each type are then collected to train type-specific SVR models, enabling the models to learn deformation characteristics specific to each geometric group.

---

**Algorithm 2** Overall Pipeline for Size-Adaptive Shoe Insole Outline Generation
 

---

**Input:** Full dataset  $\mathcal{D}$ ; query design  $D_{\text{query}}$  with base shape  $\mathbf{P}_{\text{query}}$  at size  $s_{\text{query}}$ ; target sizes  $\{s_1, \dots, s_{K_s}\}$

**Output:** Generated outlines  $\mathbf{C}_{\text{final}}$  for all target sizes

```

1: Phase 1: Leave-One-Insole-Out and Type Determination
2:  $\mathcal{D}' \leftarrow \mathcal{D} \setminus \{\text{all samples belonging to } D_{\text{query}}\}$  ▷ Enforce LOIO constraint
3:  $M_{\text{dist}} \leftarrow \text{GEOMETRICDISTANCEMATRIX}(\mathcal{D}')$ 
4:  $\{C_k\}_{k=1}^4 \leftarrow \text{HIERARCHICALCLUSTERING}(M_{\text{dist}}, K = 4)$ 
5: for  $k = 1$  to 4 do
6:    $\tilde{\mathbf{P}}^{(k)} \leftarrow \text{COMPUTECENTROID}(C_k)$ 
7: end for
8: Phase 2: Adaptive Reference Selection
9:  $\text{rmse}_A, C_A \leftarrow \arg \min_k \text{SHAPEDIST}(\mathbf{P}_{\text{query}}, \tilde{\mathbf{P}}^{(k)})$  ▷ Nearest cluster centroid
10:  $\text{rmse}_B, C_B \leftarrow \arg \min_{\mathbf{P} \in \mathcal{D}'} \text{SHAPEDIST}(\mathbf{P}_{\text{query}}, \mathbf{P})$  ▷ Nearest individual shape
11: if  $\min(\text{rmse}_A, \text{rmse}_B) > 3.5$  then
12:   return  $\text{RATIOSCALING}(\mathbf{P}_{\text{query}}, \{s_1, \dots, s_{K_s}\})$  ▷ Fallback for outliers
13: else
14:    $\mathcal{T}^* \leftarrow \text{SELECTMEMBERS}(\text{Candidate set with lower RMSE})$ 
15: end if
16: Phase 3: Type-Specific SVR Training
17:  $\mathbf{P}_{\text{ref}} \leftarrow \text{shape with minimum size in } \mathcal{T}^*$ 
18:  $\mathcal{T}_{\text{train}} \leftarrow \emptyset$ 
19: for each sample  $(s_j, \mathbf{P}_j) \in \mathcal{T}^*$  do
20:    $\tilde{\mathbf{P}}_j \leftarrow \text{CYCLICALIGN}(\mathbf{P}_j, \mathbf{P}_{\text{ref}})$ 
21:    $(\hat{\mathbf{d}}_{t,j}, \hat{\mathbf{d}}_{n,j}) \leftarrow \text{COMPUTELOCALDISPLACEMENT}(\tilde{\mathbf{P}}_j, \mathbf{P}_{\text{ref}}, \mathbf{T}_{\text{ref}}, \mathbf{N}_{\text{ref}})$ 
22:    $\mathcal{T}_{\text{train}} \leftarrow \mathcal{T}_{\text{train}} \cup \{(s_j, \hat{\mathbf{d}}_{t,j}, \hat{\mathbf{d}}_{n,j})\}$ 
23: end for
24: Train SVR model  $f_{\text{SVR}} : s \mapsto (\hat{\mathbf{d}}_t, \hat{\mathbf{d}}_n)$  on  $\mathcal{T}_{\text{train}}$ 
25: Phase 4: Relative Deformation Transfer
26:  $\tilde{\mathbf{P}}_{\text{query}} \leftarrow \text{CYCLICALIGN}(\mathbf{P}_{\text{query}}, \mathbf{P}_{\text{ref}})$ 
27: Compute local basis  $\tilde{\mathbf{T}}_{\text{query}}, \tilde{\mathbf{N}}_{\text{query}}$  for  $\tilde{\mathbf{P}}_{\text{query}}$ 
28: for each target size  $s_i \in \{s_1, \dots, s_{K_s}\}$  do
29:    $(\hat{\mathbf{d}}_t(s_i), \hat{\mathbf{d}}_n(s_i)) \leftarrow f_{\text{SVR}}(s_i)$ 
30:   Compute relative displacements  $\Delta \mathbf{d}_t(s_i), \Delta \mathbf{d}_n(s_i)$ 
31:    $\hat{\mathbf{P}}_{\text{pred}}(s_i) \leftarrow \tilde{\mathbf{P}}_{\text{query}} + \Delta \mathbf{d}_t(s_i) \odot \tilde{\mathbf{T}}_{\text{query}} + \Delta \mathbf{d}_n(s_i) \odot \tilde{\mathbf{N}}_{\text{query}}$ 
32: end for
33: Phase 5: Monotonicity Constraint (Guardrail)
34:  $L_{\text{adj}}(s_{K_s}) \leftarrow \text{PRINCIPALAXISLENGTH}(\hat{\mathbf{P}}_{\text{pred}}(s_{K_s}))$ 
35: for  $i = K_s - 1$  downto 1 do
36:    $L_{\text{pred}} \leftarrow \text{PRINCIPALAXISLENGTH}(\hat{\mathbf{P}}_{\text{pred}}(s_i))$ 
37:    $L_{\text{adj}}(s_i) \leftarrow \min(L_{\text{pred}}, L_{\text{adj}}(s_{i+1}))$ 
38:   if  $L_{\text{pred}} > L_{\text{adj}}(s_i)$  then
39:      $\hat{\mathbf{P}}_{\text{final}}(s_i) \leftarrow \text{PRINCIPALAXISSCALING}(\hat{\mathbf{P}}_{\text{pred}}(s_i), L_{\text{adj}}(s_i))$ 
40:   else
41:      $\hat{\mathbf{P}}_{\text{final}}(s_i) \leftarrow \hat{\mathbf{P}}_{\text{pred}}(s_i)$ 
42:   end if
43: end for
44: return  $\{\text{BSPLINERECONSTRUCTION}(\hat{\mathbf{P}}_{\text{final}}(s_i))\}_{i=1}^{K_s}$ 

```

---

**Table 3.** Cluster heterogeneity analysis based on geometric RMSE. The weighted intra-cluster RMSE (3.14 mm) is lower than the global pairwise RMSE (4.65 mm), indicating improved geometric homogeneity after type partitioning.

| Type             | Included Insoles             | N | Intra RMSE (mm) | Inter RMSE (mm) |
|------------------|------------------------------|---|-----------------|-----------------|
| 1                | Insole01, Insole03, Insole06 | 3 | $2.36 \pm 0.85$ | $4.91 \pm 1.48$ |
| 2                | Insole00, Insole04, Insole05 | 3 | $3.92 \pm 0.49$ | $4.41 \pm 0.89$ |
| 3                | Insole02                     | 1 | -               | $5.42 \pm 1.29$ |
| 4                | Insole07                     | 1 | -               | $6.46 \pm 1.35$ |
| Weighted Average |                              | 8 | 3.14            | 4.65            |

The threshold  $\theta = 3.5$  mm is a fixed global constant applied identically across all eight LOIO folds—that is, it is not re-tuned using data from any held-out design. Its value was selected based on the separation observed in the per-fold matched RMSE distribution, where five designs have  $\text{RMSE} \leq 3.29$  mm, indicating a reliable geometric match, while the remaining three exceed 3.68 mm, indicating no reliable match. This leaves a gap of approximately 0.4 mm within which  $\theta$  can be placed without changing the pathway assignment of any fold. We additionally evaluated the robustness of this choice through a sensitivity sweep over  $\theta \in \{3.0, 3.3, 3.5, 3.8, 4.0, 4.5\}$  mm (Table 4). The performance remains stable for  $\theta \geq 3.3$  mm, with Hausdorff distances ranging from 4.64 to 4.76 mm, and degrades only when  $\theta$  is lowered to 3.0 mm, which forces two well-matched designs (matched RMSEs of 3.14 and 3.29 mm) into the ratio scaling fallback.

**Table 4.** Sensitivity of Proposed\_Full to the RMSE fallback threshold  $\theta$  ( $N_{CP} = 40$ , 8-fold LOIO, mean  $\pm$  std; query size excluded from evaluation as described in Section 4.2). “Fallback” counts the held-out designs routed to ratio scaling under each  $\theta$ .

| $\theta$ (mm) | Fallback (n/8) | Hausdorff (mm)    | Chamfer (mm)      | IoU               |
|---------------|----------------|-------------------|-------------------|-------------------|
| 3.0           | 5              | $5.172 \pm 1.754$ | $1.925 \pm 0.761$ | $0.928 \pm 0.024$ |
| 3.3           | 3              | $4.726 \pm 1.892$ | $1.763 \pm 0.831$ | $0.929 \pm 0.026$ |
| 3.5 (used)    | 3              | $4.726 \pm 1.892$ | $1.763 \pm 0.831$ | $0.929 \pm 0.026$ |
| 3.8           | 1              | $4.636 \pm 1.869$ | $1.709 \pm 0.808$ | $0.930 \pm 0.026$ |
| 4.0           | 1              | $4.636 \pm 1.869$ | $1.709 \pm 0.808$ | $0.930 \pm 0.026$ |
| 4.5           | 0              | $4.758 \pm 2.097$ | $1.572 \pm 0.613$ | $0.932 \pm 0.022$ |

The validity of the type partitioning is supported by the shape distance analysis summarized in Table 3. When the four-type structure is applied, the intra-cluster RMSE decreases to 3.14 mm, representing a 32.5% reduction compared to the non-clustered configuration. This reduction indicates improved geometric homogeneity within each type.

During the shape generation stage (also referred to below as the inference stage), a new input control-point set  $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$  is compared against the predefined reference shapes  $\mathbf{C}^{(k)} = \{\mathbf{c}_i^{(k)}\}_{i=1}^N$  for  $K = 4$  types.

For each type  $k$ , the geometric distance is defined as the root mean square error (RMSE) between cyclically aligned control points:

$$D_k = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{c}_i^{(k)}\|^2} \quad (6)$$

Here, cyclic alignment is performed prior to distance computation to account for rotational ordering ambiguity in the closed contour representation.

The optimal type is determined as

$$k^* = \arg \min_{k \in \{1, \dots, K\}} D_k, \quad (7)$$

where  $\|\cdot\|$  denotes the Euclidean norm.

### Synthesis and Guardrail

For the selected type, the trained SVR model predicts the local displacement magnitudes for each control point in the tangent and normal directions. While the SVR models are trained within the canonical coordinate frame of the reference shape ( $\mathbf{P}_{\text{ref}}$ ) to ensure consistent learning, applying these absolute displacements directly during inference would overwrite the unique fine-grained geometric details of the input query shape.

To address this, we introduce a **relative deformation transfer** formulation. Rather than applying the absolute predicted displacements directly, we compute the relative displacement between the target size  $s$  and the query size  $s_{\text{query}}$ . These relative magnitudes are then projected onto the local tangent–normal basis of the cyclically aligned query shape ( $\tilde{\mathbf{P}}_{\text{query}}$ ).

Let  $\Delta d_{t,i}(s)$  and  $\Delta d_{n,i}(s)$  denote the relative displacement magnitudes:

$$\begin{aligned} \Delta d_{t,i}(s) &= \hat{d}_{t,i}(s) - \hat{d}_{t,i}(s_{\text{query}}) \\ \Delta d_{n,i}(s) &= \hat{d}_{n,i}(s) - \hat{d}_{n,i}(s_{\text{query}}) \end{aligned} \quad (8)$$

Given the aligned query shape  $\tilde{\mathbf{P}}_{\text{query}} = \{\tilde{\mathbf{p}}_{\text{query},i}\}_{i=1}^N$  and its corresponding unit tangent and normal vectors  $\tilde{\mathbf{t}}_{\text{query},i}$  and  $\tilde{\mathbf{n}}_{\text{query},i}$ , the synthesized control point  $\hat{\mathbf{p}}_i(s)$  is computed as

$$\hat{\mathbf{p}}_i(s) = \tilde{\mathbf{p}}_{\text{query},i} + \Delta d_{t,i}(s) \tilde{\mathbf{t}}_{\text{query},i} + \Delta d_{n,i}(s) \tilde{\mathbf{n}}_{\text{query},i}, \quad i = 1, \dots, N \quad (9)$$

In vector form, the synthesis process is expressed as

$$\hat{\mathbf{P}}_{\text{new}}(s) = \tilde{\mathbf{P}}_{\text{query}} + \Delta \mathbf{d}_t(s) \odot \tilde{\mathbf{T}}_{\text{query}} + \Delta \mathbf{d}_n(s) \odot \tilde{\mathbf{N}}_{\text{query}} \quad (10)$$

where  $\odot$  denotes element-wise multiplication. This formulation preserves query-specific geometric features while adapting the shape to the target size.

To ensure geometric and physical consistency, the synthesized outline undergoes a guardrail validation step. In particular, a monotonicity constraint is imposed on the principal-axis length of the insole shape to reflect the expected size progression. Let  $L(s)$  denote the principal-axis length at size  $s$ . The following condition is enforced:

$$L(s_{k+1}) \geq L(s_k), \quad \forall k \quad (11)$$

If size inversion occurs (i.e., violation of the monotonicity constraint), a principal-axis scaling correction is applied to restore geometric consistency. This post-processing step prevents non-physical deformation artifacts and ensures stable size progression across the generated outlines.

The detailed procedure of the synthesis and guardrail process is summarized in Algorithm 2.

## 4. Experiments

This section presents the experimental evaluation of the proposed framework. We first evaluate the geometric rectification and segmentation pipeline used for contour extraction (Section 4.1). We then describe the dataset composition and the leave-one-insole-out cross-validation protocol used to prevent information leakage across insole designs (Section 4.2).

Next, we present a qualitative comparison with the ratio scaling baseline (Section 4.3), a sensitivity analysis with respect to the control-point density (Section 4.4), a quantitative comparison with the baseline methods (Section 4.5), and an ablation study evaluating the contributions of the individual algorithmic components (Section 4.6).

#### 4.1. Evaluation of the Geometric Rectification and Segmentation Pipeline

This subsection reports the held-out evaluation and sensitivity analyses supporting the design choices of the geometric rectification and hyperparameter optimization stages (Section 3): sensitivity to the corner localization error, the selection of the boundary smoothing parameter  $\sigma$ , the selection of the target resolution, a quantitative comparison against a homography baseline, the held-out evaluation of the optimized configuration, hyperparameter importance, and the sensitivity of the selected configuration to the hybrid-objective weighting coefficients. These analyses were conducted on the 21-image held-out image test set used to evaluate the contour extraction and geometric rectification pipeline (Section 3.1), which is distinct from the 47-sample insole dataset described in Section 4.2 below.

**Corner Localization Robustness.** To assess the sensitivity of the overall pipeline to corner detection errors, we injected controlled Gaussian noise (standard deviation  $\sigma$  ranging from 0 to 50 px) directly into the four detected corner points prior to mesh warping, with all other settings (resolution, hyperparameters) held fixed, and measured the segmentation accuracy—quantified via the boundary F1 score (BF1; formally defined in Equation (5))—on the 21-image held-out test set (averaged over three repetitions per noise level; Table 5). The accuracy degrades gracefully for small corner errors but deteriorates sharply beyond  $\sigma = 10$  px, particularly for the worst-case performance. This indicates that the proposed corner detection procedure is expected to remain accurate to within approximately 5–10 px under the acquisition protocol used in this study, and we report this as a practical operating boundary rather than claiming unconditional robustness.

**Table 5.** Sensitivity of segmentation accuracy to injected corner localization error on the 21-image held-out test set.

| Corner Error $\sigma$ (px) | Mean BF1 | Min BF1 |
|----------------------------|----------|---------|
| 0 (none)                   | 0.929    | 0.825   |
| 5                          | 0.898    | 0.692   |
| 10                         | 0.832    | 0.469   |
| 20                         | 0.697    | 0.160   |
| 30                         | 0.618    | 0.113   |
| 50                         | 0.461    | 0.055   |

**Boundary Smoothing Sigma Selection.** We investigated the effect of the Gaussian smoothing parameter  $\sigma$  used during boundary curve smoothing by evaluating the segmentation accuracy on the 21-image held-out test set for  $\sigma \in \{0, 2, 5, 10, 15, 20, 30\}$ , with the resolution and hyperparameters held fixed (Table 6). Without smoothing ( $\sigma = 0$ ), boundary noise severely degrades the performance (mean BF1 = 0.699, min BF1 = 0.129). The accuracy improves sharply as  $\sigma$  increases and peaks at  $\sigma = 10$  (mean BF1 = 0.929, min BF1 = 0.825), and it then degrades for larger  $\sigma$  (min BF1 falling to 0.639 at  $\sigma = 30$ ) as excessive smoothing removes genuine boundary detail. We therefore retain  $\sigma = 10$  as the empirically optimal operating point within the tested range.

**Target Resolution Selection.** The target aspect ratio (1:1.414) was chosen to match the physical A4-sized backdrop used during image acquisition (210 mm  $\times$  297 mm), preserving the true proportions of the capture setup without introducing artificial distortion. The absolute resolution (1000 px short side) was selected via a quantitative sensitivity analysis in which the full pipeline was re-run at five target resolutions on the 21-image held-out

test set (Table 7). The accuracy improves sharply up to 1000 px and then plateaus, while the per-image processing time continues to increase, so 1000 px is retained as a practical operating point rather than a uniquely optimal value.

**Table 6.** Sensitivity of segmentation accuracy to the boundary smoothing parameter  $\sigma$  on the 21-image held-out test set.

| $\sigma$         | Mean BF1 | Min BF1 |
|------------------|----------|---------|
| 0 (no smoothing) | 0.699    | 0.129   |
| 2                | 0.902    | 0.721   |
| 5                | 0.927    | 0.824   |
| 10 (used)        | 0.929    | 0.825   |
| 15               | 0.916    | 0.798   |
| 20               | 0.897    | 0.731   |
| 30               | 0.856    | 0.639   |

**Table 7.** Resolution sensitivity analysis on the 21-image held-out test set (short-side resolution; hyperparameters fixed at their 1000 px-tuned values across all resolutions).

| Resolution         | Mean BF1 | Min BF1 | Hybrid Score ( $\mathcal{J}$ ) | Rectification Time |
|--------------------|----------|---------|--------------------------------|--------------------|
| 500 × 707          | 0.423    | 0.255   | 0.373                          | 0.16 s             |
| 750 × 1060         | 0.876    | 0.715   | 0.828                          | 0.29 s             |
| 1000 × 1414 (used) | 0.929    | 0.825   | 0.898                          | 0.45 s             |
| 1250 × 1768        | 0.939    | 0.822   | 0.904                          | 0.73 s             |
| 1500 × 2121        | 0.944    | 0.822   | 0.908                          | 0.98 s             |

**Segmentation Downsampling Width Selection.** The coarse segmentation downsampling width  $W_{proc}$  (Section 3.1.2) was selected via a quantitative sensitivity analysis in which the segmentation accuracy and processing time were measured on the 21-image held-out test set for  $W_{proc} \in \{200, 400, 600, 800\}$  px, with the resolution and hyperparameters held fixed (Table 8).  $W_{proc} = 400$  achieves a mean and worst-case boundary F1 within 0.01 of the highest values observed in the tested range (attained at  $W_{proc} = 600$ ), while requiring roughly half the processing time; we therefore retain  $W_{proc} = 400$  as a practical operating point offering the near-best accuracy at a substantially lower computational cost, rather than the uniquely optimal value within the tested range.

**Table 8.** Sensitivity of segmentation accuracy and processing time to the downsampling width  $W_{proc}$  on the 21-image held-out test set.

| $W_{proc}$ (px) | Mean BF1 | Min BF1 | Segmentation Time |
|-----------------|----------|---------|-------------------|
| 200             | 0.911    | 0.664   | 0.097 s           |
| 400 (used)      | 0.929    | 0.825   | 0.258 s           |
| 600             | 0.934    | 0.832   | 0.507 s           |
| 800             | 0.920    | 0.597   | 0.860 s           |

We note that, in the four sweep experiments above (corner localization error, boundary smoothing  $\sigma$ , target resolution, and segmentation downsampling width  $W_{proc}$ ), the segmentation hyperparameters were held fixed at their originally tuned values rather than being independently re-optimized for each tested condition. Independently re-tuning the hyperparameters for every condition would be a more rigorous but substantially more costly analysis (a separate full Optuna search per condition) and is left to future work.

**Largest Connected Component Retention.** Retaining only the largest connected component (LCC) after morphological filtering assumes that the insole forms a single,

spatially contiguous foreground region after segmentation. To assess how often this assumption is violated, we measured the number of significant connected components (area greater than 0.1% of the mask) immediately before the LCC step, across all 43 images used in this study (22 training and 21 held-out test). Multiple significant components occurred in only 2 of 43 images (4.7%); in both cases, the discarded (non-largest) region did not overlap with the ground truth insole mask, indicating background noise rather than a fragment of the true object. No case in the dataset resulted in the LCC step discarding the genuine insole area; we report this as an empirical observation on the present dataset rather than a general guarantee.

**Comparison against a Homography Baseline.** Unlike a simple homography, which relies on four-point correspondences and produces only a linear (projective) mapping, the Coons patch formulation interpolates directly from the four detected boundary curves—which may themselves be locally curved due to non-planar background deformation (e.g., bending of the chroma-key backdrop during acquisition)—allowing the rectification to compensate for such non-linear distortions, which a point-based projective transform cannot capture. A quantitative comparison against a homography baseline using the same detected corner points on the 21-image held-out test set, summarized in Table 9 (evaluated via the mean IoU and Hausdorff distance, formally defined in Section 4.4), confirms that this curved-boundary interpolation yields substantially improved boundary alignment. Because these ground truth annotations were themselves generated on Coons patch-rectified images—and no independent physical ground truth (e.g., a manufacturer template) of the insole outline was available—this comparison may be biased toward the Coons patch in regions of backdrop curvature, and the reported margin should be interpreted as indicating the direction rather than the precise magnitude of the improvement.

**Table 9.** Quantitative comparison of the proposed Coons patch rectification against a homography baseline on the 21-image held-out test set.

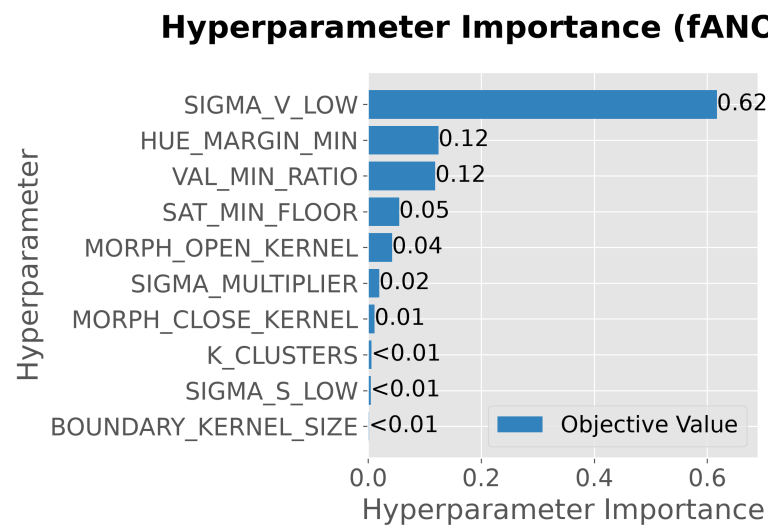
| Method                 | Mean IoU     | Mean Hausdorff (px) | Worst-Case Hausdorff (px) |
|------------------------|--------------|---------------------|---------------------------|
| Coons patch (proposed) | <b>0.995</b> | <b>4.21</b>         | <b>9.00</b>               |
| Homography baseline    | 0.947        | 14.85               | 25.18                     |

**Held-Out Evaluation.** The hyperparameter optimization procedure (Section 3) was applied to the training set (22 images); the resulting configuration is summarized in Table 1. Evaluated on an independent held-out test set of 21 images, whose ground truth contours were likewise manually annotated via polygon labeling, the pipeline achieved a mean boundary F1 of 0.9294, a minimum boundary F1 of 0.8252, and a hybrid score ( $\mathcal{J}$ ) of 0.8981.

**Hyperparameter Importance.** A hyperparameter importance analysis (Figure 6) revealed that the lower-bound coefficient of the value channel,  $\sigma_V$  (Value Lower Sigma), exhibited by far the highest relative importance (0.62), followed by a secondary but non-negligible contribution from the hue margin ( $\delta_H$ , 0.12) and the value ratio floor ( $\tau_V$ , 0.12). This finding indicates that suppressing background shadows and illumination variations in the chroma-key setup plays a critical role in the segmentation accuracy, while morphological and boundary refinement kernel sizes contributed comparatively little.

**Sensitivity to Hybrid-Objective Weighting.** The weighting coefficients (0.7, 0.3) in Equation (5) were chosen to balance average-case performance with worst-case robustness. We conducted a post hoc sensitivity analysis to characterize the effect of this weighting choice, recomputing the optimal configuration under weights ranging from 0.5 to 1.0 (mean coefficient) using the completed set of 2000 Optuna trials described in Section 3 (Table 10). The configuration selected under weights 0.6–1.0 remained identical, whereas equal weighting (0.5/0.5) selected a different configuration exhibiting a marked drop in

held-out worst-case performance—which is consistent with overfitting to the training set’s worst-case sample when excessive weight is placed on the minimum term. This confirms that the reported results are not sensitive to the specific choice of 0.7/0.3 within a reasonably wide range.



**Figure 6.** Hyperparameter importance analysis via fANOVA for the training set-optimized configuration (Table 1).

**Table 10.** Sensitivity of the selected hyperparameter configuration to the weighting coefficients in Equation (5). The configuration selected under weights 0.6–1.0 is identical (Trial #1444); only equal weighting (0.5/0.5) selects a different configuration (Trial #1814), which exhibits markedly lower held-out worst-case performance.

| Weight (Mean/Min) | Selected Trial | Held-Out Mean BF1 | Held-Out Min BF1 |
|-------------------|----------------|-------------------|------------------|
| 0.5/0.5           | #1814          | 0.891             | 0.591            |
| 0.6/0.4           | #1444          | 0.929             | 0.825            |
| 0.7/0.3 (used)    | #1444          | 0.929             | 0.825            |
| 0.8/0.2           | #1444          | 0.929             | 0.825            |
| 0.9/0.1           | #1444          | 0.929             | 0.825            |
| 1.0/0.0           | #1444          | 0.929             | 0.825            |

#### 4.2. Dataset and Evaluation Protocol

This subsection describes the composition of the dataset used in the size-adaptive shape grading experiments and introduces the leave-one-insole-out (LOIO) cross-validation protocol adopted to prevent information leakage across insole designs during training and evaluation.

##### 4.2.1. Dataset Composition

Our dataset comprises 47 insole samples collected from eight distinct insole designs (Insole00–Insole07), with each design represented by multiple physical sizes spanning 230–290 mm. Thus, the 47 samples correspond to the individual size-specific instances collected across the eight designs. Each sample is represented by a set of B-spline control points describing its insole geometry. The sampling density varies across designs because the insoles were collected from commercially available footwear, and only physically available sizes could be obtained during data collection. Accordingly, some designs have relatively dense size coverage (e.g., Insole03 with 10 sizes), whereas others have more limited coverage (e.g., Insole07 with 3 sizes).

The 47 insole samples were obtained from an acquisition batch independent of the 43-image dataset (22 training + 21 held-out test images) used in Section 4.1 to tune and evaluate the rectification/segmentation pipeline. The two datasets share no images and serve different purposes: the 43-image dataset is used to evaluate the image-based preprocessing pipeline itself, whereas the 47 insole samples are used to train and evaluate the size-adaptive shape grading pipeline described in this section. Each of the 47 samples corresponds to a single physical insole at a specific size, photographed under the acquisition protocol described in Section 3. Each photograph was then geometrically rectified and segmented to extract its boundary contour, and the contour was fit with a chord length-parameterized cubic B-spline (degree 3) followed by cyclic alignment, as described in Section 3.2, to yield the fixed-dimension control-point representation used throughout this paper. The same image-to-control-point procedure was applied to every insole shape reported in this work, including the query and target shapes used during inference.

Target predictions are generated on a common prediction grid of 230–280 mm in 5 mm increments; the isolated 290 mm samples present only in Insole01 and Insole05 (Table 11) are excluded from this common grid, as evaluating at a size supported by only two designs would not provide a statistically meaningful per-size comparison. For quantitative evaluation, each design is assessed only at the sizes for which ground truth control points are actually available within this grid (Table 11), rather than a single fixed range common to all types, ensuring that reported metrics are computed exclusively against real, non-interpolated reference geometries.

**Table 11.** Dataset composition across the eight insole designs (Insole00–Insole07). The available size range and sample count are summarized for each design.

| Insole ID | Size Range (mm) | Included Sizes                                     | N  |
|-----------|-----------------|----------------------------------------------------|----|
| Insole00  | 240–275         | {240, 245, 255, 265, 270, 275}                     | 6  |
| Insole01  | 230–290         | {230, 240, 250, 260, 280, 290}                     | 6  |
| Insole02  | 230–270         | {230, 250, 260, 270}                               | 4  |
| Insole03  | 230–280         | {230, 235, 245, 250, 255, 260, 265, 270, 275, 280} | 10 |
| Insole04  | 230–280         | {230, 235, 240, 245, 250, 260, 265, 270, 280}      | 9  |
| Insole05  | 240–290         | {240, 260, 280, 290}                               | 4  |
| Insole06  | 230–280         | {230, 240, 250, 270, 280}                          | 5  |
| Insole07  | 250–280         | {250, 270, 280}                                    | 3  |
| Total     | 230–290         | —                                                  | 47 |

For each held-out design, the shape at its minimum available size serves as the query shape  $\mathbf{P}_{\text{query}}$ , reflecting the practical setting in which grading starts from a single reference size. This query size is excluded from the evaluation grid because the relative deformation transfer (Equation (9)) yields  $\Delta d(s_{\text{query}}) = 0$  by construction, so the prediction at  $s = s_{\text{query}}$  trivially reproduces the input shape. Accordingly, all reported metrics are computed only over target sizes distinct from  $s_{\text{query}}$ .

#### 4.2.2. Leave-One-Insole-Out Cross-Validation

To rigorously evaluate the generalization capabilities of the proposed method to unseen insole geometries, we employ an eight-fold leave-one-insole-out (LOIO) cross-validation protocol. Standard  $k$ -fold cross-validation, which partitions the data purely at the sample level, risks information leakage when differently sized samples of the same underlying insole design are distributed across the training and test sets. With  $k = 8$ , each fold holds out an entire insole design (i.e., all available sizes of  $D_i$ ) as the test set, while the remaining seven designs are used for training.

Within each LOIO fold, hyperparameter tuning and type determination are performed strictly using the seven training designs to prevent data leakage. Specifically, the secondary hyperparameters of the SVR, KRR, and GPR models are re-tuned using an internal  $K$ -fold cross-validation ( $K = 3$ ) along the size axis, with only the training pool used for model selection. Likewise, for the PCA-based baseline models (PCA-Linear, PCA-KRR, PCA-GPR, and PCA-SVR), the number of retained principal components is dynamically selected within each LOIO training set to capture 95% of the cumulative explained variance. Concurrently, the hierarchical clustering ( $K = 4$ ) is reconstructed independently for each fold, ensuring that the geometric reference groups are derived solely from the training data. This strict isolation ensures that the final evaluation metrics—including the Hausdorff distance, Chamfer distance, and IoU—reflect genuine generalization to previously unseen insole designs.

The corresponding grid search spaces (or fixed settings for TPS) for the SVR, KRR, GPR, and TPS models, in both the tangent-normal and PCA latent representations, are summarized in Table 12.

**Table 12.** Hyperparameter search spaces used for grid search via 3-fold cross-validation along the size axis, within each LOIO training fold. Tangent-normal space targets are unwhitened physical displacements (mm), whereas PCA latent space targets are whitened (unit variance) principal components; the two representations therefore use separately calibrated search ranges for  $\epsilon$  and  $\gamma$ .

| Model | Hyperparameter        | Tangent-Normal Space                                                                            | PCA Latent Space                                                                       |
|-------|-----------------------|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| SVR   | $C$                   | $\{1, 10, 10^2, 10^3, 5 \times 10^3\}$                                                          | $\{1, 10, 10^2, 10^3\}$                                                                |
|       | $\epsilon$            | $\{0.1, 0.5, 1.0, 2.0, 5.0\}$                                                                   | $\{10^{-3}, 10^{-2}, 0.05, 0.1, 0.5\}$                                                 |
|       | $\gamma$              | $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 1.25 \times 10^{-3}, \text{'scale'}\}$ | $\{5 \times 10^{-4}, 1.25 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}, \text{'scale'}\}$ |
| KRR   | $\alpha$              | $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$                                                              | $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$                                                     |
|       | $\gamma$              | $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 1.25 \times 10^{-3}\}$                 | $\{5 \times 10^{-4}, 1.25 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$                 |
| GPR   | $\ell$ (length-scale) | $\{3, 5, 10, 20, 30\}$                                                                          | $\{1, 3, 5, 10, 20\}$                                                                  |
|       | Kernel                | $C(1) \cdot \text{RBF}(\ell) + \text{WK}(10^{-3}) + C(1) \cdot \text{DP}$                       | $C(1) \cdot \text{RBF}(\ell) + \text{WK}(10^{-4}) + C(1) \cdot \text{DP}$              |
| TPS   | Smoothing             | 0 (fixed)                                                                                       | —                                                                                      |

WK: WhiteKernel; DP: DotProduct kernel. All GPR fits used 10 optimizer restarts. TPS smoothing was fixed at 0, falling back to  $10^{-3}$  only when the unregularized interpolation matrix was numerically singular. TPS was not applied in the PCA latent space (interpolation baseline is defined only in the tangent-normal representation).

Across the eight-fold LOIO evaluation, the RMSE-based fallback threshold (Section 3.3,  $\theta = 3.5$ ) was triggered for three of the eight held-out designs (Insole02, Insole05, and Insole07; matched RMSE: 3.68–4.12). For these designs, the predictions were generated using ratio scaling because the matched reference did not satisfy the predefined RMSE criterion. The remaining five designs (Insole00, Insole01, Insole03, Insole04, and Insole06; matched RMSE: 1.21–3.29) were predicted using the type-specific tangent-normal SVR pathway. Therefore, the reported **Proposed\_Full** metrics reflect this mixed-strategy evaluation, consistent with the pipeline's intended fail-safe design (Algorithm 2, Phase 2).

Table 13 reports the individual Hausdorff, Chamfer, and IoU values for each held-out design, together with the selected prediction pathway and matched reference type.

**Table 13.** Per-design LOIO evaluation results for Proposed\_Full at  $N_{CP} = 40$ , with the query size (the design’s minimum available size) excluded from evaluation (Section 4.2). Each design was held out as the test fold. “Pathway” indicates whether the type-specific SVR pathway or the ratio scaling fallback was selected according to the RMSE-based type matching threshold ( $\theta = 3.5$ ) in Phase 2 of Algorithm 2.

| Held-Out Design         | Pathway                  | Matched RMSE | Hausdorff (mm)    | Chamfer (mm)      | IoU               |
|-------------------------|--------------------------|--------------|-------------------|-------------------|-------------------|
| Insole00                | SVR (matched: Insole04)  | 3.29         | 4.032             | 1.283             | 0.928             |
| Insole01                | SVR (matched: Insole03)  | 2.88         | 7.652             | 2.854             | 0.900             |
| Insole02                | Fallback (ratio scaling) | 4.12         | 7.459             | 3.072             | 0.889             |
| Insole03                | SVR (matched: Insole06)  | 1.21         | 3.164             | 1.007             | 0.961             |
| Insole04                | SVR (matched: Insole03)  | 3.14         | 3.180             | 1.157             | 0.952             |
| Insole05                | Fallback (ratio scaling) | 3.68         | 5.352             | 2.236             | 0.916             |
| Insole06                | SVR (matched: Insole03)  | 1.21         | 3.134             | 1.182             | 0.939             |
| Insole07                | Fallback (ratio scaling) | 3.72         | 3.836             | 1.313             | 0.948             |
| Mean $\pm$ SD (8 folds) | —                        | —            | $4.726 \pm 1.892$ | $1.763 \pm 0.831$ | $0.929 \pm 0.026$ |

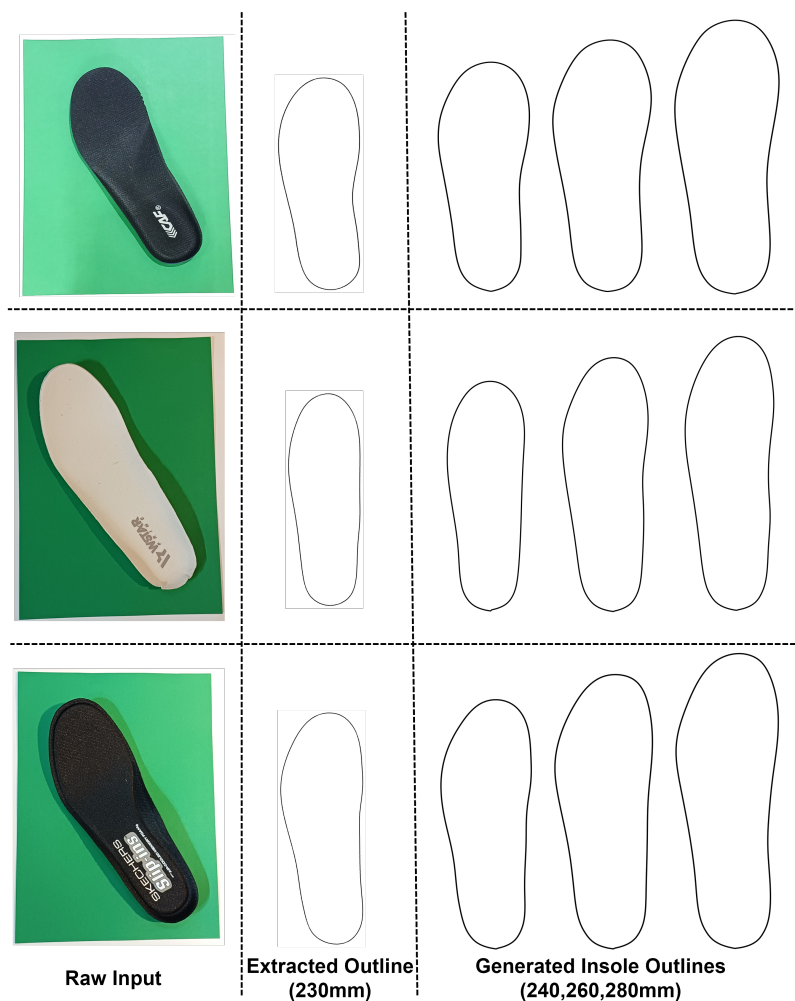
Finally, the quantitative results from all eight LOIO folds are aggregated to report the mean, standard deviation, and 95% confidence intervals, together with paired statistical tests to assess the consistency and statistical significance of the observed performance differences.

#### 4.3. Qualitative Comparison

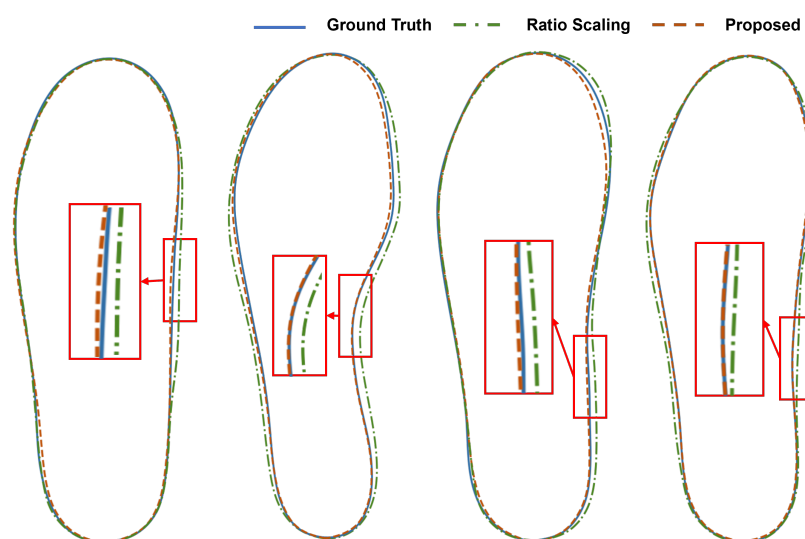
Following the eight-fold LOIO protocol described in Section 4.2, we first provide a qualitative comparison with the ratio scaling baseline, which represents the most direct scaling-based alternative to the proposed method. A quantitative comparison with the full set of baselines is presented in Section 4.5.

Figure 7 presents the insole outlines generated by the proposed method for three target sizes (240, 260, and 280). Each row corresponds to a different input insole sample. From left to right, the columns show the original input image, the extracted contour, and the synthesized outlines at the target sizes. The proposed method reflects local shape variations associated with size progression through tangent–normal-based displacement prediction. As the target size increases, the forefoot and rearfoot regions exhibit gradual and coherent geometric expansion while maintaining overall contour smoothness and structural continuity. Furthermore, the guardrail validation step suppresses non-physical deformation artifacts, ensuring geometric consistency across the generated size variations.

Figure 8 presents a visual comparison between the conventional ratio scaling approach and the proposed method across four different insole types. As highlighted in the enlarged regions (red boxes), ratio scaling (green dashed) exhibits noticeable deviations from the ground truth (blue solid), particularly in areas with pronounced curvature variation, such as the ball region. This discrepancy arises because global proportional scaling cannot adequately capture the non-uniform local deformation characteristics inherent in size progression. In contrast, the proposed method (orange dashed) produces outlines that more closely align with the ground truth across all four types. This behavior suggests that separating type-specific geometric structures and learning local displacements enables the more accurate modeling of non-uniform geometric transformations.



**Figure 7.** Generated insole outlines for target sizes 240, 260, and 280 using the proposed method. Each row corresponds to a different insole sample. From left to right: input image, extracted outline, and generated outlines.



**Figure 8.** Qualitative comparison between ratio scaling and the proposed method for four different insole types. Ratio scaling (green dashed) shows noticeable deviations in the ball region, whereas the proposed method (orange dashed) closely matches the ground truth (blue solid). Red boxes highlight regions with significant geometric differences.

#### 4.4. Sensitivity to Control-Point Density

To analyze the impact of the control-point density on the shape prediction performance, we varied the number of control points ( $N_{CP}$ ) from 20 to 50. For each configuration, identical training and inference conditions were maintained, and the geometric similarity between the predicted and ground truth shapes was quantitatively evaluated.

Three evaluation metrics were employed: the Hausdorff distance, Chamfer distance, and intersection over union (IoU).

The Hausdorff distance measures the maximum deviation between two point sets and is suitable for assessing worst-case local errors along the contour. The bidirectional Hausdorff distance is defined as

$$H(P, G) = \max \left\{ \max_{p \in P} \min_{g \in G} \|p - g\|, \max_{g \in G} \min_{p \in P} \|g - p\| \right\} \quad (12)$$

The Chamfer distance reflects the overall geometric similarity by summing the average nearest-neighbor distances between two point sets:

$$d_{CD}(P, G) = \frac{1}{|P|} \sum_{p \in P} \min_{g \in G} \|p - g\| + \frac{1}{|G|} \sum_{g \in G} \min_{p \in P} \|g - p\| \quad (13)$$

The IoU evaluates the area-based overlap between the predicted region and the ground truth region.

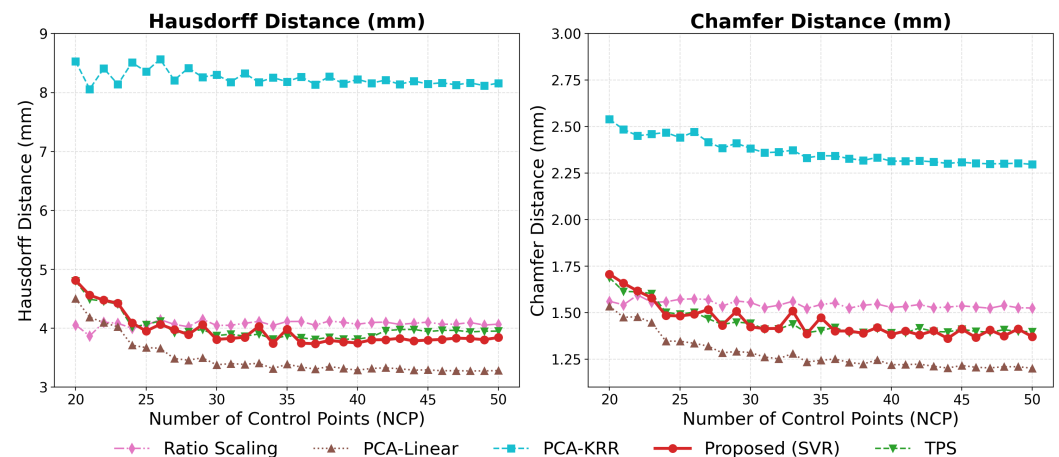
Using these three metrics as a unified evaluation framework, we conducted both a sensitivity analysis of the control-point density and a quantitative comparison of different regression models. The comparison methods are summarized as follows.

- **Ratio Scaling:** A simple global scaling baseline that uniformly enlarges or shrinks the contour according to the target size ratio, without considering local geometric characteristics.
- **PCA-Linear:** A global statistical shape model that projects B-spline control points into a low-dimensional latent space via PCA and models the relationship between latent coefficients and size using linear regression.
- **PCA-KRR:** A non-linear extension of the PCA-based model, where latent coefficients are predicted using kernel ridge regression with an RBF kernel to capture non-linear global shape variations.
- **GPR:** A probabilistic non-linear baseline that applies Gaussian process regression within the same tangent–normal displacement space as the proposed SVR model, providing a direct comparison of the non-linear regression performance in the same geometric representation as the proposed method.
- **TPS:** A non-linear interpolation baseline that applies thin plate spline interpolation within the same tangent–normal displacement space as the proposed SVR model, providing a direct comparison of the non-linear interpolation performance in the same geometric representation.
- **Proposed (Type-Specific SVR):** The proposed method, which performs cluster-based type classification and directly predicts tangent–normal local displacements for each type using SVR.

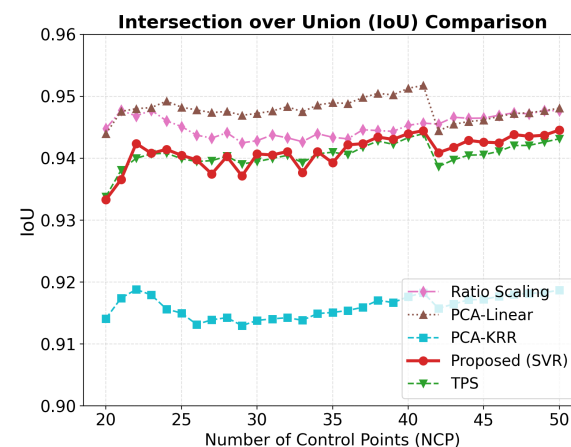
Three additional variants (PCA-GPR, PCA-SVR, and KRR) were also evaluated as supplementary baselines and are reported in Section 4.5 for completeness.

GPR, TPS, and KRR are evaluated directly in the tangent–normal displacement space without PCA-based dimensionality reduction. This allows the effect of the regression or interpolation method to be assessed while keeping the geometric representation consistent with that of the proposed SVR approach.

To investigate the effect of the B-spline control-point density on the geometric alignment accuracy, the number of control points ( $N_{CP}$ ) was increased from 20 to 50 in increments of one. Figures 9 and 10 illustrate the variations in the Hausdorff distance, Chamfer distance (error metrics), and IoU (similarity metric) with respect to  $N_{CP}$ .



**Figure 9.** Hausdorff distance and Chamfer distance according to control-point density.



**Figure 10.** Intersection over union (IoU) comparison.

Overall, TPS and the proposed method—which operate directly on the tangent–normal displacement representation—show more apparent performance improvements with increasing  $N_{CP}$ , with decreasing geometric errors (Hausdorff and Chamfer distances) and increasing IoU values. The improvement is particularly evident from  $N_{CP} = 20$  to approximately 30. This trend suggests that a certain level of control-point density is beneficial in representing the curvature and geometric characteristics of the insole contour. In contrast, ratio scaling and PCA-KRR remain comparatively insensitive to changes in  $N_{CP}$  across the evaluated range, although the underlying reasons differ and are discussed below.

The ratio scaling baseline shows negligible performance variation as  $N_{CP}$  increases. This behavior suggests that increasing representational density alone does not substantially improve a global proportional scaling approach, which does not explicitly model local geometric deformation.

PCA-Linear demonstrates a general improvement trend with increasing  $N_{CP}$ , although fluctuations are observed in certain intervals, potentially reflecting the limited expressive capacity of a global linear regression model in the latent space. PCA-KRR, despite introducing non-linear regression in the latent space, does not outperform PCA-Linear on this dataset and remains relatively stable across the entire  $N_{CP} = 20$ –50 range.

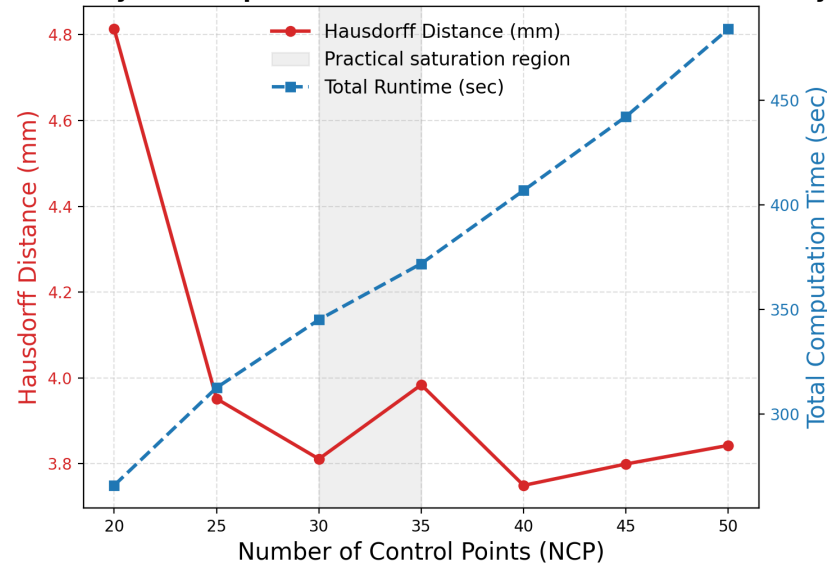
(Hausdorff error: 9.50–10.29 mm; IoU: 0.894–0.902). At  $N_{CP} = 40$ , averaged over the eight-fold LOIO evaluation, PCA-KRR shows a higher Hausdorff error (9.986 mm) than PCA-Linear (4.157 mm). Moreover, the Hausdorff error difference between PCA-KRR and the proposed method is statistically significant ( $p = 0.023$ ), whereas the corresponding difference between PCA-Linear and the proposed method is not statistically significant ( $p = 0.187$ ). These results may indicate that, given the limited number of samples available within each type pool, the additional flexibility of non-linear kernel regression in the low-dimensional PCA latent space does not necessarily lead to improved generalization. In contrast, the proposed method models local displacements directly in the tangent–normal representation without PCA-based dimensionality reduction. Notably, because the 95% cumulative explained variance criterion (Section 4.2) is satisfied with as few as  $n = 3$  principal components regardless of  $N_{CP}$ , the latent representation used by PCA-KRR, and consequently its prediction performance, remains relatively insensitive to the control-point density. This contrasts with TPS and the proposed method, which operate directly in the tangent–normal displacement space and therefore show greater sensitivity to changes in  $N_{CP}$ .

The proposed method achieves Hausdorff errors that make it statistically indistinguishable from PCA-Linear and comparable to or better than most baselines across the evaluated control-point configurations. It additionally provides type-specific local displacement modeling and explicitly enforces monotonic size progression, which are not directly incorporated into the compared baseline methods. Furthermore, the marginal improvement becomes less consistent beyond  $N_{CP} \approx 30$  (Table 14), with a temporary degradation from  $30 \rightarrow 35$  followed by an improvement from  $35 \rightarrow 40$ . Among the reported checkpoints, the lowest Hausdorff error is achieved at  $N_{CP} = 40$ , as detailed below.

As summarized in Table 14 and Figure 11, beyond  $N_{CP} \approx 40$ , the geometric error no longer shows a consistent improvement and instead exhibits slight fluctuations or degradation (e.g., +1.7% and +1.0% increases in Hausdorff distance from  $40 \rightarrow 45$  and  $45 \rightarrow 50$ , respectively), while the computation time continues to increase monotonically (+8.7% to +9.5% per step). This indicates that increasing the control-point density beyond this range provides no clear additional geometric benefit relative to its computational cost. We therefore select  $N_{CP} = 40$  as the representative operating point within the evaluated range. Among the reported checkpoints ( $N_{CP} = 20$ – $50$  in five-point increments; Table 14),  $N_{CP} = 40$  achieves a Hausdorff error that is effectively tied with the lowest observed value ( $N_{CP} = 30$ : 4.723 mm;  $N_{CP} = 40$ : 4.726 mm), with the 0.003 mm difference falling well within the observed fold-to-fold variability. Given the observed fold-to-fold variability and the non-monotonic changes in performance across nearby configurations,  $N_{CP} = 40$  is regarded as a practically representative operating point rather than a uniquely optimal configuration. This configuration is consistently applied across all baseline comparisons and ablation experiments reported in this paper.

**Table 14.** Marginal change in geometric error and computation time per five-point increase in  $N_{CP}$  (proposed method). Beyond  $N_{CP} \approx 40$ , changes in Hausdorff distance become small relative to the fold-to-fold variability, while the computation time continues to increase monotonically.

| $N_{CP}$ Range      | $\Delta$ Hausdorff (%) | $\Delta$ Time (%) |
|---------------------|------------------------|-------------------|
| 20 $\rightarrow$ 25 | −16.6                  | +17.6             |
| 25 $\rightarrow$ 30 | −1.3                   | +10.4             |
| 30 $\rightarrow$ 35 | +6.2 (noise)           | +7.8              |
| 35 $\rightarrow$ 40 | −5.8                   | +9.4              |
| 40 $\rightarrow$ 45 | +1.7                   | +8.7              |
| 45 $\rightarrow$ 50 | +1.0                   | +9.5              |

**Accuracy vs. Computational Cost across Control-Point Density**

**Figure 11.** Accuracy (Hausdorff distance) vs. total computation time across  $N_{CP}$ . The shaded region marks  $N_{CP} = 30\text{--}35$ , where the geometric error begins to show small fold-to-fold fluctuations while the runtime continues to increase.

#### 4.5. Comparison with Baseline Methods

Table 15 presents a pairwise comparison of Proposed\_Full against representative baseline methods.

**Table 15.** Pairwise comparison of Proposed\_Full against representative baseline methods ( $N_{CP} = 40$ , 8-fold LOIO, query size excluded from evaluation).  $p$ -values are from paired  $t$ -tests ( $n = 8$  folds). Bold indicates  $p < 0.05$ .

| Baseline               | Hausdorff (mm)    | Chamfer (mm)      | IoU               | $p$ (Haus.)  | $p$ (Chamfer) | $p$ (IoU)    |
|------------------------|-------------------|-------------------|-------------------|--------------|---------------|--------------|
| Proposed_Full (ours)   | $4.726 \pm 1.892$ | $1.763 \pm 0.831$ | $0.929 \pm 0.026$ | —            | —             | —            |
| Ratio Scaling          | $5.158 \pm 1.119$ | $1.947 \pm 0.523$ | $0.930 \pm 0.019$ | 0.431        | 0.419         | 0.892        |
| PCA-Linear             | $4.157 \pm 1.593$ | $1.563 \pm 0.760$ | $0.938 \pm 0.025$ | 0.187        | 0.159         | 0.114        |
| PCA-KRR                | $9.986 \pm 4.392$ | $2.854 \pm 0.867$ | $0.899 \pm 0.027$ | <b>0.023</b> | <b>0.020</b>  | <b>0.034</b> |
| GPR (Gaussian process) | $4.607 \pm 1.668$ | $1.693 \pm 0.820$ | $0.932 \pm 0.024$ | 0.486        | 0.198         | 0.306        |
| TPS                    | $4.791 \pm 1.492$ | $1.771 \pm 0.735$ | $0.929 \pm 0.022$ | 0.775        | 0.888         | 0.833        |

Additional baseline variants (PCA-GPR:  $p = 0.082$ ; PCA-SVR:  $p = 0.485$ ; KRR:  $p = 0.226$ ; all Hausdorff) show similarly non-significant differences from Proposed\_Full and are omitted here for brevity.

#### 4.6. Ablation Study

To evaluate the contributions of individual components within the proposed pipeline, we conducted an ablation study under the same LOIO cross-validation framework. The full proposed model (**Proposed\_Full**) was compared with the following variant configurations.

- **Ablation1 (NoClustering):** Bypasses the cluster-based type determination (Section 3.3) and applies a single global SVR model trained on all available designs. Because type-based grouping in the proposed pipeline is established through the clustering step, disabling clustering also removes the subsequent type-based grouping. Accordingly, this ablation evaluates the combined effect of cluster-based type determination and type-specific modeling.
- **Ablation2 (NoTangentNormal):** Disables the relative deformation transfer in the tangent–normal space and instead predicts displacements directly in the Cartesian  $X, Y$  coordinate space.

- Ablation3 (NoGuardrail): Omits the final monotonicity constraint (Phase 5 of Algorithm 2), thereby allowing the model predictions to remain unconstrained with respect to monotonic size progression.

Additionally, to further examine the reference selection strategies, we evaluated **Ablation1b (ClusterOnly)**, which exclusively applies Strategy A (nearest cluster centroid) regardless of the RMSE threshold, and Proposed\_Full\_NoFallback, which omits the global ratio scaling fallback, to examine the performance of the SVR pathway without this fallback mechanism.

As summarized in Table 16, neither variant showed a statistically significant difference from Proposed\_Full in terms of the Hausdorff distance ( $p > 0.4$  for both comparisons). The results for Ablation1b indicate that restricting the reference selection to Strategy A alone did not result in a statistically significant change in performance relative to the full strategy. Similarly, the results for Proposed\_Full\_NoFallback indicate that removing the ratio scaling fallback did not produce a statistically significant difference in the Hausdorff distance. These findings suggest that the evaluated reference selection variants did not result in statistically significant performance differences under the present LOIO evaluation setting, while their potential role in providing robustness to individual cases remains relevant to the overall design of the pipeline.

**Table 16.** Reference selection robustness check: forcing Strategy A only (Ablation1b) and disabling the ratio scaling fallback (Proposed\_Full\_NoFallback), compared against Proposed\_Full ( $N_{CP} = 40$ , 8-fold LOIO, mean  $\pm$  std, query size excluded from evaluation).  $p$ -values are from paired  $t$ -tests against Proposed\_Full ( $n = 8$  folds).

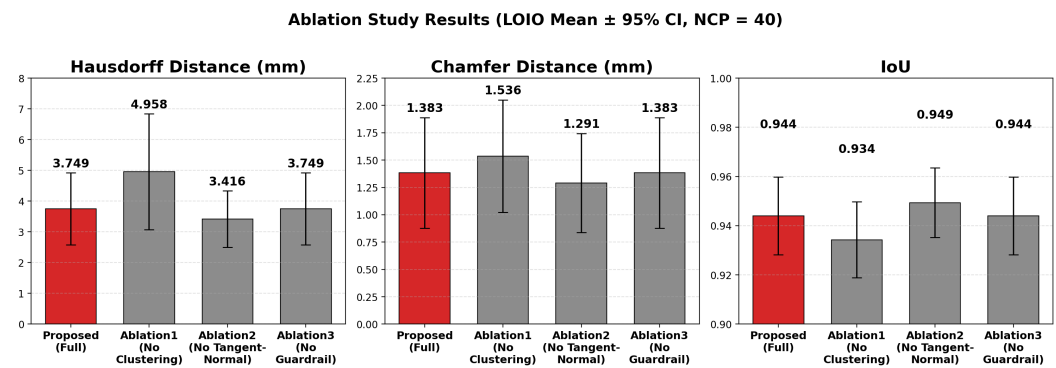
| Configuration            | Hausdorff (mm)    | Chamfer (mm)      | IoU               | $p$ (Hausdorff) |
|--------------------------|-------------------|-------------------|-------------------|-----------------|
| Proposed_Full            | $4.726 \pm 1.892$ | $1.763 \pm 0.831$ | $0.929 \pm 0.026$ | –               |
| Ablation1b (ClusterOnly) | $5.132 \pm 1.056$ | $1.832 \pm 0.589$ | $0.929 \pm 0.020$ | 0.435           |
| Proposed_Full_NoFallback | $4.758 \pm 2.097$ | $1.572 \pm 0.613$ | $0.932 \pm 0.022$ | 0.901           |

#### Quantitative Analysis of the Guardrail Mechanism

To characterize the contribution and behavior of the guardrail mechanism (monotonicity correction), its activation frequency was tracked across all predictions generated during the LOIO evaluation. This analysis was conducted to determine whether the guardrail materially affected the reported performance or primarily served as a constraint for preventing non-monotonic size progression.

As summarized in Table 17 and Figure 12, removing clustering (Ablation1) was the only variant that showed a statistically significant difference from **Proposed\_Full**, with a higher Hausdorff distance ( $p = 0.032$ ) and lower IoU ( $p = 0.033$ ). Disabling the tangent-normal decomposition (Ablation2) resulted in a numerically lower Hausdorff distance, but the difference was not statistically significant ( $p = 0.196$ ). Removing the guardrail (Ablation3) produced results identical to those of Proposed\_Full, as the guardrail was not activated at  $N_{CP} = 40$ . Its behavior under conditions that induce activation is examined separately below.

For the proposed configuration at  $N_{CP} = 40$ , the guardrail was activated in **0.0%** of the evaluated cases. Thus, the reported geometric accuracy was unaffected by guardrail correction under the standard LOIO evaluation setting. The guardrail instead functions as a safety mechanism for preventing non-monotonic size progression when such cases occur. Its behavior under conditions outside the standard evaluation range was further examined through the extrapolation stress test described below.



**Figure 12.** Ablation study results (LOIO mean  $\pm$  95% CI,  $N_{CP} = 40$ ).

**Table 17.** Ablation study results ( $N_{CP} = 40$ , 8-fold LOIO, mean  $\pm$  std, query size excluded from evaluation).  $p$ -values are from a paired  $t$ -test against Proposed\_Full ( $n = 8$  folds).

| Configuration               | Hausdorff (mm)    | Chamfer (mm)      | IoU               | $p$ (Hausdorff)  | $p$ (Chamfer)    | $p$ (IoU)        |
|-----------------------------|-------------------|-------------------|-------------------|------------------|------------------|------------------|
| Proposed_Full               | 4.726 $\pm$ 1.892 | 1.763 $\pm$ 0.831 | 0.929 $\pm$ 0.026 | –                | –                | –                |
| Ablation1 (NoClustering)    | 6.130 $\pm$ 3.060 | 1.899 $\pm$ 0.804 | 0.919 $\pm$ 0.025 | 0.032            | 0.272            | 0.033            |
| Ablation2 (NoTangentNormal) | 4.320 $\pm$ 1.618 | 1.651 $\pm$ 0.778 | 0.936 $\pm$ 0.025 | 0.196            | 0.149            | 0.087            |
| Ablation3 (NoGuardrail)     | 4.726 $\pm$ 1.892 | 1.763 $\pm$ 0.831 | 0.929 $\pm$ 0.026 | N/A <sup>†</sup> | N/A <sup>†</sup> | N/A <sup>†</sup> |

<sup>†</sup> The guardrail was never activated for this configuration (0.0% trigger rate at  $N_{CP} = 40$ ), so its output is numerically identical to that of Proposed\_Full.

To further characterize the guardrail behavior beyond the standard evaluation range, we conducted a stress test by extrapolating predictions to sizes up to 150 mm beyond the maximum training size (290 mm). Under this extrapolation setting, Proposed\_Full maintained a 0.0% activation rate (0/25 extreme-range and 0/55 near-range predictions). The 25 extreme-range predictions correspond only to the five designs routed to the type-specific SVR pathway under the standard LOIO protocol (Insole00, Insole01, Insole03, Insole04, Insole06;  $5 \times 5$  extrapolated sizes). The remaining three designs (Insole02, Insole05, Insole07) are handled by the ratio scaling fallback, which is monotonic by construction; these predictions are therefore not subject to the guardrail mechanism. In contrast, applying the same extrapolation range to Ablation1\_NoClustering, which does not employ a fallback mechanism and applies a single global SVR model to all eight held-out designs ( $8 \times 5 = 40$ ), resulted in guardrail activation in 10 of 40 extreme-range predictions, with corrections of up to 0.33 mm. These results indicate that the guardrail can become active under extrapolation conditions when the predicted size progression violates the prescribed monotonicity constraint, while remaining inactive for the evaluated predictions of the proposed configuration.

## 5. Conclusions and Discussion

This study proposes a data-driven parametric shape modeling framework for the automated size-based grading of 2D insole contours to support digital footwear design and AR/VR-based applications. The framework integrates image-based contour extraction, B-spline parametric representation, and type-specific SVR-based local displacement regression. Unlike conventional global ratio scaling or PCA-based latent-space regression approaches, the proposed method directly predicts the tangent-normal local displacements of control points without dimensionality reduction, enabling the modeling of non-uniform and non-linear geometric deformations.

Overall, the principal contribution of this work lies not in any single regression technique but in the integrated strategy that combines type-specific local modeling with a tangent-normal displacement representation, which together enable an explicit monotonic-

ity constraint and yield a geometrically interpretable, structurally constrained framework for size-based shape grading.

Quantitative experiments based on eight-fold LOIO cross-validation showed that the proposed method significantly outperformed the no-clustering ablation (Ablation1,  $p = 0.032$ ) and PCA-KRR ( $p = 0.023$ ) in terms of the Hausdorff distance, while showing no statistically significant differences from ratio scaling, PCA-GPR, GPR, and TPS (all  $p > 0.05$ ). PCA-Linear showed a small numerical advantage in the Hausdorff distance, but the difference from the proposed method was not statistically significant ( $p = 0.187$ ). These results suggest that global linear latent-space regression remains a competitive baseline for the present dataset. At the same time, the proposed method avoids the dimensionality reduction step of PCA-KRR by modeling local displacements directly in the tangent–normal representation. The guardrail mechanism further incorporates an explicit constraint on monotonic size progression, which is not imposed in the compared baseline methods.

The sensitivity analysis further demonstrated that increasing the control-point density beyond  $N_{CP} \approx 40$  did not yield consistent additional improvements in geometric accuracy, while the computational cost continued to increase. Consequently,  $N_{CP} = 40$  was identified as a practical operating point that balances geometric fidelity and the computational cost among the evaluated configurations. From an industrial perspective, these findings suggest that an appropriately designed parametric representation can provide favorable geometric fidelity without requiring excessively dense control-point configurations.

Unlike individualized 3D scan-based customization approaches, this work focuses on standard size-based digital grading automation. The proposed tangent–normal displacement decomposition and cluster-based type modeling provide a practical design strategy that preserves the local curvature characteristics while maintaining model interpretability.

The scope of these findings should be interpreted in the context of the dataset and evaluation setting. The proposed framework was developed and evaluated using 47 size-specific insole samples spanning eight distinct insole designs from a single manufacturer, providing a basis for assessing size-based grading across unseen insole designs under the LOIO protocol. However, broader validation across additional manufacturers, footwear categories, design types, and size ranges is needed to further assess the generalizability of the learned deformation patterns. The present study also focused on classical statistical and kernel-based regression approaches, including SVR, KRR, GPR, and TPS, which were appropriate given the scale of the available dataset and provided interpretable comparisons of different regression strategies. Data-intensive deep learning approaches, such as graph neural networks or implicit neural representations, were therefore outside the scope of the present evaluation.

The proposed framework provides a basis for future extension from 2D insole contours to full 3D shoe mesh grading. By modeling vertex coordinates as functions of size variables, it could facilitate the automation of grading for the entire shoe structure, including the upper geometry. With larger-scale datasets, integration with graph neural networks (GNNs) or implicit neural representation-based deformation models could further enhance the modeling capacity. Additionally, combining shape generation with foot type classification and fit prediction may enable the development of digital design support systems that consider both geometric and functional suitability.

**Author Contributions:** Conceptualization, S.L. and C.K.; methodology, C.K. and S.L.; software, G.E.L., M.K. and J.H.L.; validation, G.E.L., M.K. and J.H.L.; formal analysis, G.E.L., M.K. and J.H.L.; investigation, G.E.L., M.K., J.H.L. and J.K.; resources, J.K.; data curation, G.E.L. and M.K.; writing—original draft preparation, G.E.L.; writing—review and editing, S.L. and C.K.; visualization, G.E.L. and M.K.; supervision, C.K.; project administration, C.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government (MSIT) (IITP-2026-RS-2023-00260267).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data and source code supporting the findings of this study are publicly available at [https://github.com/Leegaeun1/Shoe\\_PBL\\_Project](https://github.com/Leegaeun1/Shoe_PBL_Project) (accessed on 7 August 2026).

**Conflicts of Interest:** Jiwon Kim is currently employed by MESSE ESANG. However, this study was conducted before his employment with the company, while he was a student. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

- Porter, M.E.; Heppelmann, J.E. Why every organization needs an augmented reality strategy. *Harv. Bus. Rev.* **2017**, *95*, 46–57.
- Tao, F.; Qi, Q. Make more digital twins. *Nature* **2019**, *573*, 490–491. [[CrossRef](#)] [[PubMed](#)]
- Nee, A.Y.C.; Ong, S.K.; Chrysosouris, G.; Mourtzis, D. Augmented reality applications in design and manufacturing. *CIRP Ann.* **2012**, *61*, 657–679. [[CrossRef](#)]
- Dionisio, J.D.N.; Burns, W.G.; Gilbert, R. 3D virtual worlds and the metaverse: Current status and future possibilities. *ACM Comput. Surv.* **2013**, *45*, 1–38. [[CrossRef](#)]
- Cheng, W.H.; Song, S.; Chen, C.Y.; Hidayati, S.C.; Liu, J. Fashion meets computer vision: A survey. *ACM Comput. Surv.* **2021**, *54*, 1–41. [[CrossRef](#)]
- Rahman, M.; Soliman, G.; Ahmed, F.; Islam, M.; Sayem, A.S.M. VR/AR and metaverse in fashion industry. In *Use of Digital and Advanced Technologies in the Fashion Supply Chain*; Springer: Singapore, 2025; pp. 25–39. [[CrossRef](#)]
- Nonyelum, N.P.; Ihensekhien, I.; Uko-Aviomoh, E.E. The rise of digital fashion: Virtual clothing and its market potentials. *Int. J. Educ. Psychol. Sci.* **2025**, *3*, 605–622. [[CrossRef](#)]
- Telfer, S.; Woodburn, J. The use of 3D surface scanning for the measurement and assessment of the human foot. *J. Foot Ankle Res.* **2010**, *3*, 19. [[CrossRef](#)] [[PubMed](#)]
- Tiwari, G.; Bhatnagar, B.L.; Tung, T.; Pons-Moll, G. Sizer: A dataset and model for parsing 3D clothing and learning size sensitive 3D clothing. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 1–18. [[CrossRef](#)]
- Chen, X.; Pang, A.; Yang, W.; Wang, P.; Xu, L.; Yu, J. TightCap: 3D human shape capture with clothing tightness field. *ACM Trans. Graph.* **2021**, *41*, 1–17. [[CrossRef](#)]
- Almond, K. Virtual 3D garment draping as a service. *Fash. Style Pop. Cult.* **2022**, *9*, 437–456. [[CrossRef](#)]
- Luximon, A.; Goonetilleke, R.S. Foot shape modeling. *Hum. Factors* **2004**, *46*, 304–315. [[CrossRef](#)] [[PubMed](#)]
- Schofield, N.A.; LaBat, K.L. Defining and testing the assumptions used in current apparel grading practice. *Cloth. Text. Res. J.* **2005**, *23*, 135–150. [[CrossRef](#)]
- Stankovic, K.; Booth, B.G.; Danckaers, F.; Burg, F.; Vermaelen, P.; Duerinck, S.; Sijbers, J.; Huysmans, T. Three-dimensional quantitative analysis of healthy foot shape: A proof-of-concept study. *J. Foot Ankle Res.* **2018**, *11*, 8. [[CrossRef](#)] [[PubMed](#)]
- Danckaers, F.; Stankovic, K.; Huysmans, T.; Booth, B.G.; Sijbers, J. Foot shape assessment techniques for orthotic and footwear applications: A methodological literature review. *Front. Bioeng. Biotechnol.* **2024**, *12*, 1416499. [[CrossRef](#)] [[PubMed](#)]
- Allan, J.J.; Munteanu, S.E.; Bonanno, D.R.; Buldt, A.K.; Choppin, S.; Bullas, A.; Pearce, N.; Menz, H.B. Methodological and statistical approaches for the assessment of foot shape using three-dimensional foot scanning: A scoping review. *J. Foot Ankle Res.* **2023**, *16*, 24. [[CrossRef](#)] [[PubMed](#)]
- Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. 3D-CODED: 3D correspondences by deep deformation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 230–246. [[CrossRef](#)]
- Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 165–174. [[CrossRef](#)]
- Hanocka, R.; Hertz, A.; Fish, N.; Giryas, R.; Fleishman, S.; Cohen-Or, D. MeshCNN: A network with an edge. *ACM Trans. Graph.* **2019**, *38*, 1–12. [[CrossRef](#)]
- Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; Jiang, Y.G. Pixel2Mesh: Generating 3D mesh models from single RGB images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 52–67. [[CrossRef](#)]

21. Pons-Moll, G.; Pujades, S.; Hu, S.; Black, M.J. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Trans. Graph.* **2017**, *36*, 1–15. [[CrossRef](#)]
22. Lahner, Z.; Cremers, D.; Tung, T. DeepWrinkles: Accurate and realistic clothing modeling. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 667–684. [[CrossRef](#)]
23. Patel, C.; Liao, Z.; Pons-Moll, G. TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7365–7375. [[CrossRef](#)]
24. Seul, A.; Viziteu, D.R.; Mihai, A.; Curteza, A.; Costea, M. Computer-generated shoe lasts using 3D scanned data and anthropometric measurements. In Proceedings of the International Conference Interdisciplinarity in Engineering, Târgu Mureş, Romania, 6–7 October 2022; pp. 129–143. [[CrossRef](#)]
25. Chen, Y.; Xie, R.; Yang, S.; Dai, L.; Sun, H.; Huo, Y.; Li, R. Single-view 3D garment reconstruction using neural volumetric rendering. *IEEE Access* **2024**, *12*, 49682–49693. [[CrossRef](#)]
26. Cootes, T.F.; Taylor, C.J.; Cooper, D.H.; Graham, J. Active shape models—Their training and application. *Comput. Vis. Image Understand.* **1995**, *61*, 38–59. [[CrossRef](#)]
27. Cootes, T.F.; Edwards, G.J.; Taylor, C.J. Active appearance models. In Proceedings of the European Conference on Computer Vision (ECCV), Freiburg, Germany, 2–6 June 1998; pp. 484–498. [[CrossRef](#)]
28. Blanz, V.; Vetter, T. A morphable model for the synthesis of 3D faces. In Proceedings of the SIGGRAPH, Los Angeles, CA, USA, 8–13 August 1999; pp. 187–194. [[CrossRef](#)]
29. Allen, B.; Curless, B.; Popovic, Z. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.* **2003**, *22*, 587–594. [[CrossRef](#)]
30. Anguelov, D.; Srinivasan, P.; Koller, D.; Thrun, S.; Rodgers, J.; Davis, J. SCAPE: Shape completion and animation of people. In Proceedings of the SIGGRAPH, Los Angeles, CA, USA, 31 July–4 August 2005; pp. 408–416. [[CrossRef](#)]
31. Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; Black, M.J. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.* **2015**, *34*, 248. [[CrossRef](#)]
32. Kim, M.; Pons-Moll, G.; Pujades, S.; Bang, S.; Kim, J.; Black, M.J.; Lee, S.H. Data-driven physics for human soft tissue animation. *ACM Trans. Graph.* **2017**, *36*, 54. [[CrossRef](#)]
33. Pumarola, A.; Corona, E.; Pons-Moll, G.; Moreno-Noguer, F. D-NeRF: Neural radiance fields for dynamic scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10318–10327. [[CrossRef](#)]
34. Tang, J.; Markhasin, L.; Wang, B.; Thies, J.; Nießner, M. Neural shape deformation priors. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 17117–17132. [[CrossRef](#)]
35. Duquesne, K.; Nauwelaers, N.; Claes, P.; Audenaert, E.A. Principal polynomial shape analysis: A non-linear tool for statistical shape modeling. *Comput. Methods Programs Biomed.* **2022**, *220*, 106812. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.