

Article

Tail-Preserving Shape Partitioning via Multi-Orientation Centroid-Line Extraction and Fuzzy Influence-Zone Assignment

Halit Nazli , Osman Yildirim  and Yasser Guediri 

Faculty of Engineering, Electrical and Electronics Engineering Department, Istanbul Aydin University, Beşyol, İnönü Cd. No:38, Küçükçekmece, Istanbul 34295, Türkiye; osmanyildirim@aydin.edu.tr (O.Y.); yasserguediri@stu.aydin.edu.tr (Y.G.)

* Correspondence: khaledalnazli@stu.aydin.edu.tr; Tel.: +90-5379760451

Abstract

Meaningful partitioning of 2D binary shapes remains a challenging problem in shape analysis because many existing methods rely mainly on local geometric rules or skeleton simplification, which often struggle to separate the main body of a shape from its protruding parts in a perceptually meaningful way. This limitation becomes more evident in shapes with thin limbs, branching structures, or irregular extensions, where preserving topology while achieving human-consistent decomposition is difficult. We present a fully automatic framework for the hierarchical partitioning of 2D binary shapes into semantically meaningful core bodies and protruding limbs (tails). The pipeline begins by generating candidate structural lines through multi-directional centroid tracking along horizontal, vertical, and diagonal ($\pm 45^\circ$) bands. Three direction-specific Sugeno fuzzy controllers first evaluate these lines based on normalized length, angular alignment, and minimum distance to the boundary. A second pair of fuzzy systems then classifies segments as either tails or core parts using thickness statistics derived from the distance transform. For ambiguous merged tail groups, iterative midpoint splitting is applied until stable labeling is achieved. High-curvature boundary corners are then detected via signed turning-angle analysis, and candidate cutting rays are assessed through exact region splitting, tail area measurement, and label purity analysis. An adaptive third-stage fuzzy controller ranks these candidates according to cut length, purity, and area. The highest-scoring non-overlapping cuts are executed iteratively, progressively peeling peripheral parts while preserving the overall topology and symmetry of the shape. The proposed framework is evaluated on a targeted subset of 32 categories from the 2D Shape Structure Dataset Results on this evaluated subset indicate that the method produces coherent and topologically consistent partitions, with competitive agreement with the available human-annotated references. This training-free framework provides an interpretable tool for 2D shape analysis, with potential applications in object recognition, computer animation, and symmetry studies.

Keywords: shape decomposition; binary shapes; fuzzy logic; core-tail partitioning; centroid lines; distance transform



Received: 23 March 2026

Revised: 19 April 2026

Accepted: 22 April 2026

Published: 27 April 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.

1. Introduction

Partitioning a 2D shape into semantically meaningful parts remains one of the most enduring challenges in computer vision and geometry processing. A particularly valuable resource for evaluating such partitioning is the 2D Shape Structure Dataset of Carlier et al. [1]. Built on the MPEG-7 collection, it contains more than 41,000 crowdsourced annotations

from 2861 participants across 1200+ shapes. These annotations provide hierarchical ground truth that distinguishes the main body from meaningful parts and fine details, offering the largest publicly available benchmark of human shape decomposition to date. Despite its richness, most existing algorithms are evaluated only on small subsets or on synthetic convexity scores, leaving their true alignment with human perception largely untested. Recent work has attempted to address over-segmentation through visibility-based spectral clustering [2]. By embedding boundary points into a spectral space derived from a visibility matrix and applying a recursive binary cut strategy guided by a novel visible protrusion strength signature, Li et al. achieved cleaner near-convex decompositions than classic baselines on MPEG-7 and the S&V dataset. Their method is elegant and parameter-light, yet it remains purely geometric: it optimizes local visibility and Euclidean proximity without explicitly modeling the perceptual distinction between thick core regions and thin protruding limbs. Consequently, on organic shapes with articulated tails or variable thickness, the resulting partitions can still diverge from human judgments captured in Carlier et al.'s annotations. Early geometric approaches relied on concave-point detection followed by divide-and-conquer strategies or integer linear programming [3–5]. Human observers, however, perform this task effortlessly, guided by perceptual rules such as the minima rule and the short-cut rule [6,7]. While more recent spectral and medial-axis methods improved stability [8,9], they often introduced redundant cuts in long curved branches or required careful parameter tuning. The result is a persistent gap between algorithmic output and human perceptual structure. We argue that the missing ingredient is a mechanism capable of reasoning about thickness, centrality, and directional consistency in a human-like, uncertainty-tolerant way. Fuzzy logic controllers (FLCs) are ideally suited for this task. They have been successfully applied to skeleton extraction, junction detection in microscopy, and hierarchical clustering problems where crisp thresholds fail [10–13]. Beyond fuzzy reasoning, several studies have also emphasized the importance of perceptual grouping, visibility structure, and hierarchical part organization in meaningful shape decomposition. Mi and DeCarlo [14] showed that relatability can support cleaner separation of visually coherent parts, while Liu et al. [15] introduced continuous visibility features that better capture boundary-based structural relationships. In parallel, Ren et al. [16] addressed decomposition through a minimum near-convex formulation, and Carlier et al. [17] further highlighted the importance of structured part organization and part similarity in annotated 2D shapes. More broadly, survey works on mesh and shape decomposition [18,19] have shown that meaningful partitioning depends not only on admissible cuts, but also on preserving the structural interpretation of the object. Additional fuzzy-based studies further support the usefulness of soft decision mechanisms in complex geometric and pattern-analysis problems [20–23]. Building on this foundation, we present a fully automatic, training-free framework that performs hierarchical core/tail partitioning through a cascade of six Sugeno fuzzy systems. The pipeline begins with multi-directional centroid tracking (horizontal, vertical, and both 45° diagonals) to generate candidate structural lines. Three direction-aware FLCs first score these lines on normalized length, angular deviation, and minimum boundary distance. A second pair of FLCs then classifies retained segments as tail or core using distance-transform thickness statistics (mean, gradient, coverage). Ambiguous merged tail groups undergo iterative midpoint splitting until stable labeling is reached. Finally, high-curvature corners are detected via signed turning angles, green-ray candidates are proposed and evaluated through exact region splitting and purity analysis, and an adaptive third-stage FLC ranks and executes the best non-overlapping cuts iteratively while preserving topology as shown in Figure 1. By explicitly incorporating perceptual thickness semantics and directional symmetry into the decision process, our method produces partitions that are not only near-convex but genuinely meaningful.

Quantitative evaluation on a targeted subset of the Carlier et al. benchmark demonstrates competitive alignment with human annotations compared with both classic geometric techniques [3–5] and the recent spectral approach [2]. The framework requires no training data, no deep learning, and only a handful of intuitive thresholds, making it immediately applicable to symmetry studies, shape analysis, and animation pipelines. The remainder of the paper is organized as follows. Section 2 reviews related geometric and fuzzy-based decomposition methods. Section 3 details the proposed three-stage fuzzy pipeline with pseudocode and membership functions. Section 4 presents experiments on the Carlier dataset alongside comparisons with state-of-the-art methods. Section 5 concludes with limitations and future directions.

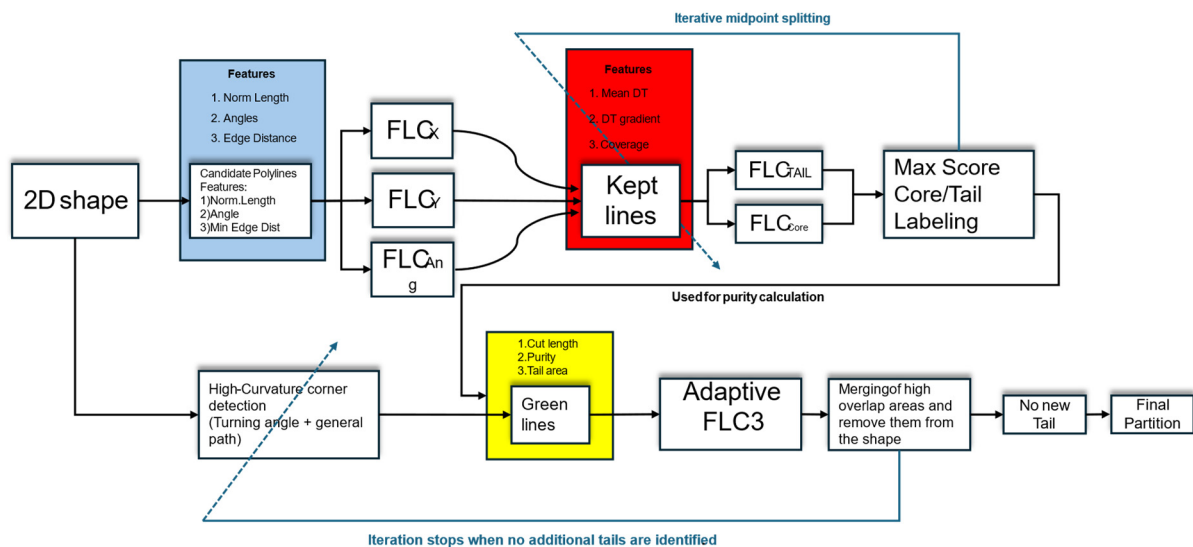


Figure 1. The entire pipeline of the proposed method. The light blue block indicates candidate-polyline extraction and first-stage geometric feature analysis, the red block indicates kept-line refinement and core/tail labeling using DT-based features, and the yellow block indicates green-line evaluation using cut length, purity, and tail area. Cyan arrows show the iterative process, while the white and gray boxes represent intermediate operations and final outputs.

2. Related Works

Meaningful decomposition of 2D shapes has long been regarded as an important step in shape analysis, object recognition, animation, and symmetry-related studies. A major advance in evaluation was introduced by Carlier et al. [1] through the 2D Shape Structure Dataset. This large-scale crowdsourced benchmark provides hierarchical human annotations of the main body, meaningful parts, and fine details for more than 1200 MPEG-7 shapes, making it the most comprehensive ground-truth resource currently available for perceptual shape decomposition. Even so, many existing methods are still evaluated only on limited subsets or through synthetic convexity measures, rather than by directly comparing their results with human perceptual judgments. More recently, Li et al. [2] revisited spectral clustering for near-convex decomposition. Their method builds a visibility-based affinity matrix, maps boundary points into a spectral space using the normalized Laplacian and then applies recursive binary partitioning. To reduce over-segmentation in branch-like regions, they introduced a new shape descriptor called Visible Protrusion Strength (VPS), which serves as an efficient stopping criterion. The method is elegant, relatively simple in terms of parameters, and achieves competitive convexity scores on the MPEG-7 and S&V datasets. However, like earlier geometric approaches, it relies mainly on visibility relations and Euclidean proximity, without explicitly accounting for perceptual thickness or directional symmetry. Earlier studies mainly focused on exact or approximate convex

decomposition. Lien and Amato [3] introduced the widely used Approximate Convex Decomposition (ACD) method, which iteratively removes the most significant concavity, or notch, until all resulting components satisfy a predefined concavity tolerance. Expanding on this direction, Liu et al. [5] formulated convex shape decomposition as an integer linear programming problem. Their method minimizes the total cut cost under explicit concavity constraints and produces near-optimal decompositions that can naturally handle shapes containing holes. Although effective, this optimization-based framework can still over-segment thin protruding regions and depends strongly on careful selection of the concavity threshold. Another line of research incorporates the short-cut rule more directly into the decomposition process. Luo et al. [4] proposed a computational model that generates candidate cuts through negative-curvature minima and selects the globally optimal set based on both convexity and cut length. This direction is supported by cognitive studies showing that human shape decomposition is guided by two main perceptual principles: the minima rule, which suggests that cuts tend to occur at negative-curvature minima, and the short-cut rule, which states that people generally prefer short cuts between nearby boundary points [6,7]. Taken together, these studies reveal a common limitation. Current geometric and spectral methods usually depend on hard thresholds and local visibility criteria, but they do not provide an explicit and flexible way to reason about thickness, centrality, and semantic importance in a more human-like manner. Beyond the classical convex and near-convex decomposition literature, several studies have explored more perceptually meaningful ways of partitioning 2D shapes. An earlier step toward perceptually meaningful decomposition was provided by Held and Abe [24], who examined the decomposition of binary shapes into meaningful parts and emphasized that the resulting partitions should reflect the perceived structural organization of the object rather than only geometric admissibility. Their work helped establish the idea that meaningful decomposition depends not only on where cuts can be placed, but also on whether the resulting parts remain visually coherent and semantically interpretable. Mi and DeCarlo [25] used relatability to support cleaner part separation, while Jiang et al. [26] and Ma et al. [27] explicitly framed decomposition as a perception-based problem rather than a purely geometric one. Fotopoulou and Psarakis [28] later showed that visibility-graph organization can also recover meaningful groups of boundary points. Likewise, Papanelopoulos and Avrithis [29] demonstrated that a simple medial-axis formulation can capture psychophysically motivated rules such as minima, short-cut preference, convexity, and symmetry, while Siddiqi et al. [30] extended decomposition toward multilevel shape structure and part similarity. These studies reinforce an important point: meaningful partitioning is not only a matter of finding admissible cuts but of preserving the internal organization of the shape in a way that remains consistent with human interpretation. The present work follows this direction but differs in that it embeds this reasoning into a fully automatic cascade of fuzzy controllers that progressively evaluates directional support, thickness-based tail/core evidence, and adaptive cut quality. By combining multi-orientation line extraction with iterative fuzzy classification and tail peeling, the proposed framework produces partitions that are not only near-convex but also more consistent with human perceptual structure, as demonstrated on the benchmark of Carlier et al.

3. Proposed Methodology

3.1. Centroids Extraction and Nearest Band Centroid Linking and Line Construction (Polyline)

This section presents the first stage of the proposed framework: extracting representative foreground centroids from a binary 2D shape using multi-directional band partitioning. The output is a set of centroid point clouds sampled along rows, columns, and the two principal diagonal orientations ($\pm 45^\circ$). These centroids act as compact geometric descriptors

for subsequent stages (e.g., graph construction, line linking, partitioning, or FLC-driven decisions). Define the input be a binary shape (foreground mask) as:

$$BW(r, c) \in \{0, 1\}. (r, c) \in \Omega = \{1, \dots, H\} \times \{1, \dots, W\}. \quad (1)$$

where r is the row index (vertical axis) and c is the column index (horizontal axis). Foreground pixels satisfy $BW(r, c) = 1$.

Our goal in this first stage is to obtain a compact, geometry-aware set of representative points (centroids) by partitioning the shape into directional bands and computing the centroid of each connected component inside each band. This produces centroid sets that describe the shape under multiple orientations (horizontal, vertical, and diagonals).

Connected components with 8-connectivity

Within each band-restricted mask, we extract connected components using 8-connectivity (Moore neighborhood). Two foreground pixels (r, c) and (r', c') are adjacent if:

$$\max\{|r - r'|, |c - c'|\} = 1. \quad (2)$$

Using 8-connectivity is important for thin/diagonal structures since a diagonal chain would break under 4-connectivity. Each connected component k is a set of pixels defined as Equation (3):

$$\mathcal{P}_k = \{(r_i, c_i)\}_{i=1}^{N_k}, \quad N_k = |\mathcal{P}_k|. \quad (3)$$

Each component $\mathcal{P}_k$ is a set of N_k foreground pixels that are mutually connected under Equation (2). In MATLAB, `bwconncomp(..., 8)` returns these components.

Centroid computation (image moments/center of mass)

For each connected component $\mathcal{P}_k$, the centroid is computed as the geometric centroid (also known as center of mass for uniform density). The centroid is computed assuming uniform density (each pixel contributes equally). Using the discrete mean:

$$\bar{c}_k = \frac{1}{N_k} \sum_{(r_i, c_i) \in \mathcal{P}_k} c_i, \quad \bar{r}_k = \frac{1}{N_k} \sum_{(r_i, c_i) \in \mathcal{P}_k} r_i. \quad (4)$$

where $\bar{c}_k$ is the mean column coordinate of all pixels in the component (x-coordinate), $\bar{r}_k$ is the mean row coordinate (y-coordinate), and intuitively, if you treat all pixels as equal masses located at their coordinates, then the centroid is the balance point. Equivalently, using raw spatial moments (named formulation: image moments):

$$m_{00} = \sum_{(r, c) \in \mathcal{P}_k} 1, \quad m_{10} = \sum_{(r, c) \in \mathcal{P}_k} c, \quad m_{01} = \sum_{(r, c) \in \mathcal{P}_k} r, \quad (5)$$

$$(\bar{c}_k, \bar{r}_k) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right). \quad (6)$$

where m_{00} is the area in pixels (count of pixels), so $m_{00} = N_k$. m_{10} and m_{01} accumulate coordinate sums. Dividing by m_{00} yields the mean coordinate; thus Equations (4) and (6) are equivalent. MATLAB `regionprops(CC, 'Centroid')` implements this centroid (internally via moments returns $[x, y] = [\bar{c}, \bar{r}]$.)

The shape may be complex (tails/branches). Instead of computing one centroid for the whole object, we compute multiple centroids by restricting the object to bands. Each band produces centroids of local components, giving a structured set of representative points.

A. Row bands (horizontal slabs)

Let the number of horizontal bands be B_r . We partition the image height H into contiguous, non-overlapping row intervals. Row band boundaries (integer, uniform partition)
For band index $b = 1, 2, \dots, B_r$, define the start and end rows:

$$r_b = 1 + \left\lfloor \frac{(b-1)H}{B_r} \right\rfloor, r_{b+1} = 1 + \left\lfloor \frac{bH}{B_r} \right\rfloor. \quad (7)$$

This creates B_r bands that cover all rows 1 to H . The floor operator $\lfloor \cdot \rfloor$ guarantees integer row indices and ensures no gaps and no overlaps between bands.

Row band index set:

$$\mathcal{R}_b = \{r \in \mathbb{Z} : r_b \leq r \leq r_{b+1} - 1\}. \quad (8)$$

where $\mathcal{R}_b$ is the set of rows belonging to band b . Row band restricted mask:

$$BW_b^{\text{row}}(r, c) = \begin{cases} BW(r, c), & r \in \mathcal{R}_b, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

We keep foreground pixels only inside the current horizontal band; everything else is forced to background. This makes the centroid extraction local rather than global. Connected components inside the row band.

Let $\{\mathcal{P}_{b,k}^{\text{row}}\}_{k=1}^{K_b^{\text{row}}}$ be the set of 8-connected foreground components inside BW_b^{row} , where each component is a set of pixels:

$$\mathcal{P}_{b,k}^{\text{row}} \subseteq \Omega, |\mathcal{P}_{b,k}^{\text{row}}| = N_{b,k}^{\text{row}}. \quad (10)$$

K_b^{row} is the number of components in band b . $N_{b,k}^{\text{row}}$ is the number of pixels in component k . Centroid of each component (center of mass/moment centroid)

$$\bar{c}_{b,k}^{\text{row}} = \frac{1}{N_{b,k}^{\text{row}}} \sum_{(r,c) \in \mathcal{P}_{b,k}^{\text{row}}} c, \quad \bar{r}_{b,k}^{\text{row}} = \frac{1}{N_{b,k}^{\text{row}}} \sum_{(r,c) \in \mathcal{P}_{b,k}^{\text{row}}} r. \quad (11)$$

B. Column bands (vertical slabs)

Let the number of vertical bands be B_c . We partition the width W into contiguous column intervals.

Column band boundaries

For $b = 1, 2, \dots, B_c$

$$c_b = 1 + \left\lfloor \frac{(b-1)W}{B_c} \right\rfloor, c_{b+1} = 1 + \left\lfloor \frac{bW}{B_c} \right\rfloor. \quad (12)$$

Column band index set

$$\mathcal{C}_b = \{c \in \mathbb{Z} : c_b \leq c \leq c_{b+1} - 1\}. \quad (13)$$

Column band restricted mask

$$BW_b^{\text{col}}(r, c) = \begin{cases} BW(r, c), & c \in \mathcal{C}_b, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Centroids inside the column band

Let $\{\mathcal{P}_{b,k}^{\text{col}}\}_{k=1}^{K_b^{\text{col}}}$ be the 8-connected components in BW_b^{col} . For each component:

$$\bar{c}_{b,k}^{\text{col}} = \frac{1}{N_{b,k}^{\text{col}}} \sum_{(r,c) \in \mathcal{P}_{b,k}^{\text{col}}} c, \bar{r}_{b,k}^{\text{col}} = \frac{1}{N_{b,k}^{\text{col}}} \sum_{(r,c) \in \mathcal{P}_{b,k}^{\text{col}}} r. \quad (15)$$

C. Diagonal bands (+45° and −45°)

Row/column bands capture horizontal/vertical structure, but tails and branches often lie diagonally. Therefore, we also partition the object by two diagonal coordinates.

+45° diagonal bands (constant $u = c - r$)

Define the +45° diagonal coordinate:

$$u(r, c) = c - r. \quad (16)$$

Pixels with the same u lie on the same +45° diagonal. This lets us band the shape along that diagonal direction. Let the number of +45° diagonal bands be B_+ . Define the global min/max:

$$u_{\min} = \min_{(r,c) \in \Omega} (c - r), u_{\max} = \max_{(r,c) \in \Omega} (c - r). \quad (17)$$

Define integer bin boundaries:

$$u_b = u_{\min} + \left\lfloor \frac{(b-1)(u_{\max} - u_{\min} + 1)}{B_+} \right\rfloor. \quad (18)$$

$$u_{b+1} = u_{\min} + \left\lfloor \frac{b(u_{\max} - u_{\min} + 1)}{B_+} \right\rfloor. \quad (19)$$

where $(u_{\max} - u_{\min} + 1)$ is the total count of integer diagonal levels. The formula splits these levels into B_+ contiguous groups.

Diagonal band mask:

$$BW_b^+(r, c) = \begin{cases} BW(r, c), & u_b \leq (c - r) \leq u_{b+1} - 1, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

Compute 8-connected components $\{\mathcal{P}_{b,k}^+\}_{k=1}^{K_b^+}$ and centroids:

$$\bar{c}_{b,k}^+ = \frac{1}{N_{b,k}^+} \sum_{(r,c) \in \mathcal{P}_{b,k}^+} c, \bar{r}_{b,k}^+ = \frac{1}{N_{b,k}^+} \sum_{(r,c) \in \mathcal{P}_{b,k}^+} r. \quad (21)$$

Here we do not crop the image; we mask diagonally; therefore, the computed centroid is already in global coordinates (no shift is required).

−45° diagonal bands (constant $v = c + r$)

Define the −45° diagonal coordinate:

$$v(r, c) = c + r. \quad (22)$$

Let the number of −45° diagonal bands be B_- (your numDiagN45). Define:

$$v_{\min} = \min_{(r,c) \in \Omega} (c + r), v_{\max} = \max_{(r,c) \in \Omega} (c + r). \quad (23)$$

Define bin boundaries:

$$v_b = v_{\min} + \left\lfloor \frac{(b-1)(v_{\max} - v_{\min} + 1)}{B_-} \right\rfloor. \quad (24)$$

$$v_{b+1} = v_{\min} + \left\lfloor \frac{b(v_{\max} - v_{\min} + 1)}{B_-} \right\rfloor. \quad (25)$$

Diagonal band mask:

$$BW_b^-(r, c) = \begin{cases} BW(r, c), & v_b \leq (c + r) \leq v_{b+1} - 1, \\ 0, & \text{otherwise.} \end{cases} \quad (26)$$

Components $\{\mathcal{P}_{b,k}^-\}$ and centroids:

$$\bar{c}_{b,k}^- = \frac{1}{N_{b,k}^-} \sum_{(r,c) \in \mathcal{P}_{b,k}^-} c, \bar{r}_{b,k}^- = \frac{1}{N_{b,k}^-} \sum_{(r,c) \in \mathcal{P}_{b,k}^-} r. \quad (27)$$

D. Output of Stage-I

The output of this stage is four centroid families

$$\mathcal{S}_{\text{cent}} = \left\{ \left\{ \begin{pmatrix} -\text{row} \\ \bar{c}_{b,k}^-, \bar{r}_{b,k}^- \end{pmatrix} \right\}, \left\{ \begin{pmatrix} -\text{col} \\ \bar{c}_{b,k}^-, \bar{r}_{b,k}^- \end{pmatrix} \right\}, \left\{ \begin{pmatrix} + \\ \bar{c}_{b,k}^+, \bar{r}_{b,k}^+ \end{pmatrix} \right\}, \left\{ \begin{pmatrix} - \\ \bar{c}_{b,k}^-, \bar{r}_{b,k}^- \end{pmatrix} \right\} \right\}. \quad (28)$$

Row band centroids capture how the object distributes from top to bottom. Column band centroids capture how it distributes from left to right. Diagonal centroids capture oblique structure, which is essential for tails/branches that are not axis-aligned. Together, these multi-directional centroids form a robust set of representative points for the next stages (e.g., linking centroids into curves/graphs, tail partitioning, or fuzzy decision inputs), Figure 2 shows the resulting centroids for all directions centroids.

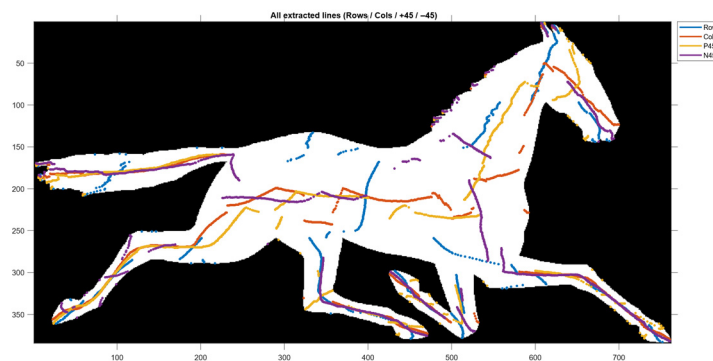


Figure 2. Extracted centroid sets obtained from the four directional band families.

The remaining operations of Stage I are summarized in Figure 3 and Algorithm 1 in the next section. The procedure is applied identically to the four orientations (row, column, $+45^\circ$, -45°). For one orientation $\mathcal{C}_b$ (e.g., rows), let there be B bands. Band $b \in \{1, \dots, B\}$ contains n_b centroids:

$$\mathcal{C}_b = \left\{ p_{b,i} \in \mathbb{R}^2 : i = 1, \dots, n_b \right\}, p_{b,i} = \begin{bmatrix} x_{b,i} \\ y_{b,i} \end{bmatrix}. \quad (29)$$

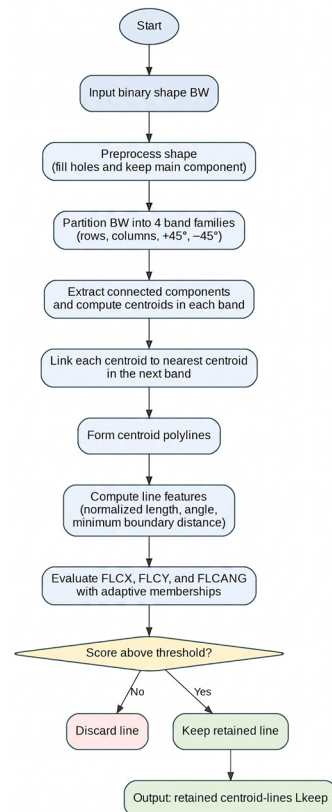


Figure 3. Flowchart summarization of Stage I in the TP-FLN framework for the multi-orientation centroid-line extraction and fuzzy line selection.

Nearest neighbor linking between consecutive bands
Let the maximum allowed linking distance to be:

$$d_{\max} = \maxDist.$$

For each centroid $\mathbf{p}_{b,i} \in \mathcal{C}_b$, we search for the closest centroid in the next band $\mathcal{C}_{b+1}$ using the Euclidean distance (also called the L_2 metric) as:

$$d(\mathbf{p}_{b,i}, \mathbf{p}_{b+1,j}) = \|\mathbf{p}_{b,i} - \mathbf{p}_{b+1,j}\|_2 = \sqrt{(x_{b+1,j} - x_{b,i})^2 + (y_{b+1,j} - y_{b,i})^2}. \quad (30)$$

where Squared distance is used because minimizing $\|\cdot\|_2$.

3.2. Feature Extraction from Centroid Lines (Stage 1)

As summarized above in Figure 3 and pseudocode 1 show below, after constructing the centroid chains (polylines) in the four orientations (Row, Column, $+45^\circ$, and -45°), Stage I proceeds with geometric feature extraction and fuzzy line selection. Each centroid line is described by three geometric measures: (i) polyline length, (ii) endpoint-based orientation angle, and (iii) minimum distance to the object boundary. These features provide a compact representation of line saliency, help distinguish core-like structures from boundary-adjacent tails, and supply normalized, shape-consistent inputs for the Stage I fuzzy logic controllers.

Algorithm 1. Stage I—Multi-orientation centroid-line extraction and fuzzy line selection

Input: Binary shape BW

Output: Retained centroid-lines Lkeep

- 1: Preprocess BW
- 2: Fill holes and keep the main connected foreground component
- 3: Compute boundary image and distance-to-boundary map
- 4: Partition BW into row, column, $+45^\circ$, and -45° band families
- 5: for each band family do
- 6: Extract 8-connected foreground components in each band
- 7: Compute the centroid of each component
- 8: Link each centroid to its nearest centroid in the next band
- 9: Form centroid polylines
- 10: end for
- 11: for each centroid polyline li do
- 12: Compute normalized length, endpoint angle, and minimum boundary distance
- 13: end for
- 14: Build adaptive membership parameters from pooled feature quantiles
- 15: Evaluate FLCX, FLCY, and FLCANG
- 16: Keep only lines whose fuzzy score exceeds the acceptance threshold
- 17: Return retained lines as Lkeep

To quantify how close each line lies to the object boundary, we compute a distance map using the Euclidean distance transform. Let ∂BW denote the boundary pixel set (obtained from a supplied edge map). The distance-to-boundary at pixel (r, c) is defined as:

$$D(r, c) = \min_{(r_b, c_b) \in \partial BW} \sqrt{(r - r_b)^2 + (c - c_b)^2}. \quad (31)$$

By construction, $D(r, c) = 0$ on boundary pixels and increases toward the interior. This map provides a robust geometric measure of “interiority” that is invariant to small contour perturbations as shown in Figure 4.

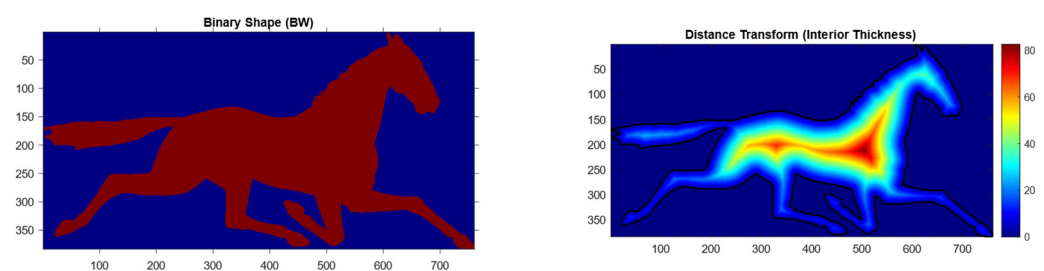


Figure 4. The thickness map.

(i) Polyline length (piecewise-linear arc length)

Given the ordered points of $\mathcal{L}_k$, the line length in pixels is computed as the sum of Euclidean distances between consecutive points:

$$\ell_k = \sum_{m=1}^{M_k-1} \sqrt{(x_{k,m+1} - x_{k,m})^2 + (y_{k,m+1} - y_{k,m})^2}. \quad (32)$$

where ℓ_k : polyline length of line k , measured in pixels, and k is the index of the extracted lines in the current orientation $k = 1, 2, \dots, n$, and M_k : number of centroid points in line k , m : index of a point along the line, $m = 1, \dots, M_k$, $(x_{k,m}, y_{k,m})$: the coordinates

of the m -th centroid point of line k , $(x_{k,m+1} - x_{k,m})$, $(y_{k,m+1} - y_{k,m})$: the horizontal and vertical displacement between consecutive points, $\sqrt{(\cdot)^2 + (\cdot)^2}$: Euclidean distance between consecutive points, and $\sum_{m=1}^{M_k-1}$: sums all segment lengths to obtain total arc length. This corresponds to the standard arc length of a piecewise-linear curve and is well suited for centroid chains because the band-based ordering ensures that successive vertices are spatially adjacent.

(ii) Endpoint-based orientation angle

To describe the direction of each line using a stable and simple descriptor.

$$p_{k,\text{low}} = (x_{\text{low}}, y_{\text{low}}), p_{k,\text{other}} = (x_{\text{other}}, y_{\text{other}}).$$

The lower endpoint is selected by:

if $y_{k,2} > y_{k,1}$, then $p_{k,\text{low}} = p_{k,2}$;

if $y_{k,2} < y_{k,1}$, then $p_{k,\text{low}} = p_{k,1}$;

if $y_{k,2} = y_{k,1}$, then the leftmost point (smaller x) is chosen as $p_{k,\text{low}}$.

Direction components (Cartesian-consistent)

We then define:

$$\Delta x_k = x_{\text{other}} - x_{\text{low}}, \Delta y_k = y_{\text{low}} - y_{\text{other}}. \quad (33)$$

where Δx_k : horizontal displacement from the chosen origin to the other endpoint, Δy_k : vertical displacement with sign flipped so that “up” is positive (Cartesian convention).

This flip is important because in images y increases downward, while in standard geometry y increases upward.

Angle using the two-argument arctangent

The endpoint angle (in degrees) is:

$$\theta_k = \text{atan2}(\Delta y_k, \Delta x_k). \quad (34)$$

where θ_k : orientation angle of line k , computed from its endpoints, $\text{atan2}(y, x)$: the two-argument arctangent function, which returns the angle of the vector (x, y) while handling quadrant correctly (unlike $\arctan(\frac{y}{x})$), Mapping the angle to $[0, 180)$ (undirected orientation).

To make the angle invariant to reversing endpoints (a line has the same orientation in both directions), we map it to $[0, 180)$:

$$\theta_k = \begin{cases} \theta_k + 180, & \theta_k < 0, \\ \theta_k, & \theta_k \geq 0. \end{cases}$$

where $\theta_k = 0^\circ$ corresponds to a horizontal line pointing right (in the standardized coordinate system), $\theta_k \approx 90^\circ$ corresponds to a vertical line. Using $[0, 180)$ means a line at 10° and the same line reversed are treated as the same orientation.

Degenerate case

If both endpoints coincide ($\Delta x_k = 0$ and $\Delta y_k = 0$), then the orientation is undefined and θ_k is set to NaN (ignored).

To obtain scale-consistent lengths, we normalize by the object bounding-box span.

Shape extents

Let the foreground pixel coordinates be $\{(r, c) : BW(r, c) = 1\}$. Define:

$$H_s = r_{\max} - r_{\min} + 1, W_s = c_{\max} - c_{\min} + 1. \quad (35)$$

where $r_{\min}, r_{\max}$: minimum and maximum row indices of foreground pixels, $c_{\min}, c_{\max}$: minimum and maximum column indices of foreground pixel, H_s : object height in pixels, and W_s : object width in pixels.

Maximum reference length (per orientation)

$$L_{MAXV} = \max(W_s, 1), L_{MAXH} = \max(H_s, 1),$$

$$L_{\max} = \max\left(\sqrt{W_s^2 + H_s^2}, 1\right). \quad (36)$$

where $L_{\max H}$: maximum feasible horizontal span, $L_{\max V}$: maximum feasible vertical span, $L_{\max}$: maximum feasible diagonal span (bounding-box diagonal), and $\max(.,1)$: avoids division by zero for very small objects.

Normalized length

$$\tilde{\ell}_k = \frac{\ell_k}{L_{\max}}, \quad (37)$$

where $L_{\max}$ is chosen according to the line orientation (Row/Col/Diag), $\tilde{\ell}_k$: normalized length, approximately in $[0, 1]$ for typical cases, ℓ_k : raw polyline length from the first length equation, and $L_{\max}$: orientation-specific maximum reference length.

(iii) Minimum boundary distance of a line

For each polyline $\mathcal{L}_k$, we evaluate the distance map at all its points and take the minimum value:

$$d_k^{\min} = \min_{m=1, \dots, M_k} D(\text{round}(y_{k,m}), \text{round}(x_{k,m})) \quad (38)$$

Rounding is applied because centroid coordinates may be subpixels. The minimum operator yields a conservative indicator: if any segment of the line approaches the boundary, then $d_k^{\min}$ becomes small. Hence, lines fully embedded in the object tend to produce larger $d_k^{\min}$, whereas boundary-adjacent branches and tails tend to produce smaller $d_k^{\min}$.

After extracting centroid lines in four orientations (Row, Column, $+45^\circ$, -45°) and computing their geometric features (normalized length, boundary distance, and endpoint orientation), we introduce an adaptive fuzzy scoring stage that selects the most plausible structural lines while suppressing noisy or spurious chains as described in summarized in Figure 3. The key idea is to construct the fuzzy membership functions automatically from the current shape using robust percentiles of the observed feature distribution, yielding a self-constructed fuzzy inference system (FIS) that generalizes across shapes without manual retuning.

Global pooling of line features across orientations

Let the four line families be:

$$LR, LC, LP, LN,$$

corresponding to Row, Column, $+45^\circ$, and -45° orientations, respectively. For each line we already computed:

Normalized length $\tilde{\ell} \in [0, 1]$,

Minimum distance-to-boundary $d^{\min}$ (pixels),

Endpoint orientation angle $\theta \in [0, 180)$ degrees.

To ensure a unified decision layer, we concatenate features from all orientations into global vectors:

$$\mathbf{d} = [d_1^{\min}, \dots, d_M^{\min}]^\top, \boldsymbol{\ell} = [\tilde{\ell}_1, \dots, \tilde{\ell}_M]^\top, \boldsymbol{\theta} = [\theta_1, \dots, \theta_M]^\top \quad (39)$$

where M is the total number of lines across all four families.

To keep angular values consistent, we apply:

$$\theta_i \leftarrow \theta_i \bmod 180,$$

so that $\theta_i \in [0, 180)$ for all lines.

Interpretation.

This pooling step allows the fuzzy membership functions to be derived from the entire set of candidate lines, rather than from a single orientation, improving robustness to variations in pose and shape topology.

Percentile-driven adaptive membership parameters:

Manual fuzzy membership tuning is often dataset-dependent and not stable across different object sizes and morphologies. Instead, we derive length thresholds directly from the current pooled length distribution using robust percentiles:

Let $\mathcal{X} = \left\{ \tilde{\ell}_i \mid \tilde{\ell}_i \text{ is finite} \right\}$. Define:

$$q_{20} = \text{Prctile}(\mathcal{X}, 20), q_{50} = \text{Prctile}(\mathcal{X}, 50), q_{80} = \text{Prctile}(\mathcal{X}, 80).$$

where q_{20} represents the typical boundary between short and moderate lines, q_{50} is the median length and reflects the “central tendency” of the shape’s line population, and q_{80} separates long lines from the rest and is less sensitive to outliers than using max length. These values are then used as shape-specific breakpoints to define membership functions for line length. As a result, the fuzzy system becomes adaptive by construction: for compact shapes, “long” is defined relative to that shape; for elongated shapes, the membership automatically expands.

Fuzzy inputs and design principle (three interpretable cues)

For each candidate line i , we define the following fuzzy inputs:

Normalized line length

$$L_i = \tilde{\ell}_i \in [0, 1].$$

Orientation angle/orientation compatibility

$$\Theta_i = |\theta_i| \in [0, 180).$$

Boundary proximity (minimum DT distance)

$$E_i = d_i^{\min} \in [0, 50] \text{ pixels.}$$

where L_i measures structural extent (longer chains are typically more reliable), Θ_i encodes directionality preference (horizontal/vertical/diagonal depending on the target), and E_i measures “interiority” using DT; lines closer to the boundary often correspond to thin tails or noisy segments, while core structures lie deeper.

Self-constructed Sugeno FIS for three “main-structure” hypotheses

We construct three Sugeno-type FIS modules as shown in Figure 1, each representing a different structural hypothesis:

FLC_X: favors horizontal/main X-direction connectivity,

FLC_Y: favors vertical/main Y-direction connectivity,

FLC_{ANG}: favors diagonal/angular skeleton hypotheses.

Each FIS produces a scalar score:

$$S \in [0, 1],$$

implemented as a zero-order Sugeno system (constant consequents), which is stable and computationally efficient for large line sets.

a. Length membership functions (adaptive)

Using (q_{20}, q_{50}, q_{80}) , we define three triangular membership functions:

$$\begin{aligned}\mu_{\text{short}}(L) &= \text{trimf}(L; 0, 0, q_{20}), \\ \mu_{\text{mid}}(L) &= \text{trimf}(L; q_{20}, q_{50}, q_{80}), \\ \mu_{\text{long}}(L) &= \text{trimf}(L; q_{50}, q_{80}, 1).\end{aligned}$$

where Triangular MFs provide smooth transitions and preserve interpretability. The break-points are estimated per-shape, making the FIS adaptive.

b. Edge-distance membership functions (fixed but interpretable)

We use three membership functions for DT-based proximity:

$$\begin{aligned}\mu_{\text{near}}(E) &= \text{trimf}(E; 0, 0, 5), \\ \mu_{\text{good}}(E) &= \text{trimf}(E; 3, 8, 15), \\ \mu_{\text{best}}(E) &= \text{trimf}(E; 10, 50, 50).\end{aligned}$$

These ranges reflect a practical geometric interpretation: lines touching the boundary are penalized (“near”), lines moderately inside are preferred (“good”), and lines deeply inside are strongly preferred (“best”).

c. Angle-compatibility membership functions (orientation dependent)

The angle memberships differ depending on which “main structure” we want:

In FLC_X (horizontal best), the system rewards angles near 0° (and near 180° after symmetry), while penalizing near 90° .

In FLC_Y (vertical best), the system rewards angles near 90° , and penalizes near 0° and 180° .

In FLC_{ANG} (diagonal best), the system rewards angles near 45° and 135° , while penalizing 0° , 90° , and 180° .

This is implemented through multiple triangular MFs over $[0, 180]$, e.g., for the diagonal case:

$$\mu_{\text{good}45}(\Theta) = \text{trimf}(\Theta; 25, 45, 65), \mu_{\text{good}135}(\Theta) = \text{trimf}(\Theta; 115, 135, 155).$$

with additional MFs to cover non-diagonal angles (e.g., “bad 0° ”, “bad 90° ”, “bad 180° ”) to discourage incompatible lines.

Rule base and scoring (interpretable, orientation-aware decision logic).

Each FIS uses a rule base of the form:

$$\text{IF } (L \text{ is } A) \text{ AND } (\Theta \text{ is } B) \text{ AND } (E \text{ is } C) \text{ THEN } S = s_0,$$

where A , B , and C are linguistic terms (e.g., short/mid/long, good/mid/bad, near/good/best), and s_0 is a constant Sugeno output level.

We use four crisp output levels:

$S \in \{0.10, 0.30, 0.60, 1.00\}$ (None, Low, Mid, High).

The rule base encodes intuitive geometric reasoning:

Longer lines are generally more trustworthy than short fragments.

Lines that are angle-consistent with the target hypothesis (horizontal/vertical/diagonal) are favored. Lines that remain inside the object core (larger DT distance) are favored over

boundary-hugging chains. The final score for each line i is obtained by evaluating the corresponding FIS:

$$S_i^X = \text{FIS}_X(L_i, \Theta_i, E_i), S_i^Y = \text{FIS}_Y(L_i, \Theta_i, E_i), S_i^{ANG} = \text{FIS}_{ANG}(L_i, \Theta_i, E_i).$$

We also identify the best-supported main line under each hypothesis:

$$i_X = \operatorname{argmax}_i S_i^X, i_Y = \operatorname{argmax}_i S_i^Y,$$

$$i_{ANG} = \operatorname{argmax}_i S_i^{ANG}.$$

To remove weak lines and retain only confident candidates, we apply a score threshold T :

$$\mathcal{K} = \{i \mid S_i > T\}.$$

In our implementation, $T = 0.5$.

Because each orientation is best assessed by a specific hypothesis, we use row lines scored by FLC_Y (vertical consistency criterion for row band chains), column lines scored by FLC_X (horizontal consistency criterion for column band chains), and diagonal lines scored by FLC_{ANG} .

Thus, the retained sets are:

$$\begin{aligned} \mathcal{L}_{\text{keep}}^R &= \{\mathcal{L}_i^R : S_i^Y > T\}, & \mathcal{L}_{\text{keep}}^C &= \{\mathcal{L}_i^C : S_i^X > T\}, \\ \mathcal{L}_{\text{keep}}^P &= \{\mathcal{L}_i^P : S_i^{ANG} > T\}, & \mathcal{L}_{\text{keep}}^N &= \{\mathcal{L}_i^N : S_i^{ANG} > T\}. \end{aligned}$$

In this stage, the fuzzy membership breakpoints are derived from the current shape using pooled feature quantiles rather than fixed global thresholds. This makes the line selection process adaptive to variations in scale and morphology. The three fuzzy inputs—normalized line length, boundary distance, and orientation compatibility—provide an interpretable basis for distinguishing dominant structural lines from weak or noisy candidates. The final retained lines for Shape 1 are illustrated in Figure 5.

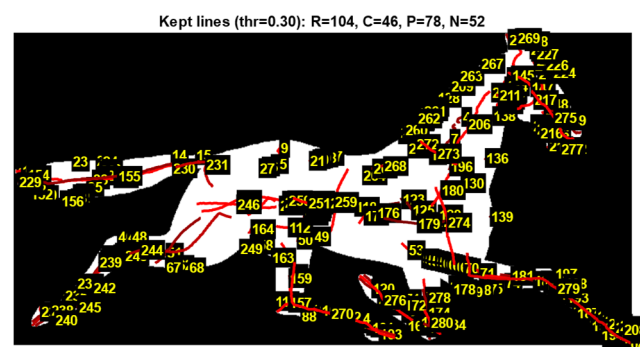


Figure 5. The Kept lines after stage 1 of shape 1.3.3. Thickness-Aware Fuzzy Controllers for Tail/Core Disambiguation (Stage 2).

The operations of Stage II are summarized in Figure 6 and the corresponding Algorithm 2 pseudocode. After Stage I retains the most reliable centroid lines, Stage II assigns each line a semantic role by evaluating whether it behaves more like a peripheral tail segment or a central core segment. To this end, two complementary thickness-aware fuzzy controllers, FLC_{TAIL} and FLC_{CORE} , are used. Their purpose is to compute interpretable tail-likeness and core-likeness scores from the thickness behavior observed along each retained centroid line. Rather than relying on fixed thresholds, Stage II derives its fuzzy memberships adaptively from shape-specific statistics, which makes the decision process

robust to variations in morphology, local irregularities, and junction complexity. The controller inputs are three EDT-derived descriptors: mean thickness, thickness instability, and normalized coverage. Together, these quantities provide complementary evidence about whether a line corresponds to a thin protruding structure or to a stable interior support line. Ambiguous merged tail groups are then refined through iterative midpoint splitting until stable labeling is reached, as summarized in Figure 6 and illustrated in Figure 7.

Algorithm 2. Stage II—Tail/core labeling with iterative midpoint splitting

Input: Retained centroid-lines L_{keep} , binary shape BW
Output: Final labeled lines L_{label} with labels {TAIL, CORE}
1: Compute the Euclidean distance transform DT inside BW
2: for each retained line li in L_{keep} do
3: Sample DT values along li
4: Compute mean thickness
5: Compute thickness instability (robust gradient statistic)
6: Compute coverage (normalized line extent)
7: end for
8: Derive adaptive fuzzy breakpoints from thickness, gradient, and coverage statistics
9: for each retained line li do
10: Evaluate FLCTAIL (mean thickness, instability, coverage)
11: Evaluate FLCCORE (mean thickness, instability, coverage)
12: Assign preliminary label by the larger score
13: end for
14: Merge nearby lines with the same winning label into candidate groups
15: while there exists an active ambiguous group do
16: Split the current line/group at its arc-length midpoint
17: Recompute features for both halves
18: Re-evaluate FLCTAIL and FLCCORE on both halves
19: Keep splitting only the half that remains ambiguous
20: Stop when both halves become stable or no meaningful split exists
21: end while
22: Optionally apply the same splitting logic in two passes: TAIL first, then CORE
23: Return the stable labeled line set L_{label}

Feature 1: Mean thickness along the line (Tail vs. Core “width” cue)

We sample the thickness field along the line using subpixel interpolation:

$$v_{g,m} = D_{\sigma}(y_{g,m}, x_{g,m}), m = 1, \dots, M_g.$$

where $v_{g,m}$: thickness sample at the line point m , and $D_{\sigma}(y, x)$: evaluated using interpolation (bilinear in practice).

The mean thickness is then:

$$\mu_g = \frac{1}{M_g} \sum_{m=1}^{M_g} v_{g,m}, T_g = \frac{\mu_g}{D_{\max}} \in [0, 1] \quad (40)$$

where T_g : set of valid sample indices, $M_g = |\mathcal{I}_g|$: number of valid samples, μ_g : average thickness along the line (pixels), and $T_g \in [0, 1]$: normalized mean thickness

Interpretation: tails $\rightarrow$ low T_g , cores $\rightarrow$ high T_g .

Tails are thin: they lie near the periphery where the inscribed radius is small, so T_g tends to be low. Cores are thick: they pass through the main body of the shape, where

thickness is higher, so T_g tends to be high. Normalizing by $D_{\max}$ makes the feature scale-free, allowing consistent fuzzy reasoning across different shapes and sizes.

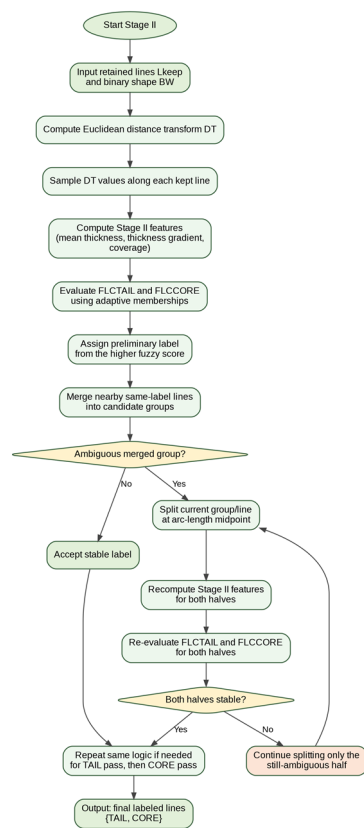


Figure 6. Stage II flowchart of Stage II for tail/core labeling with iterative midpoint splitting.

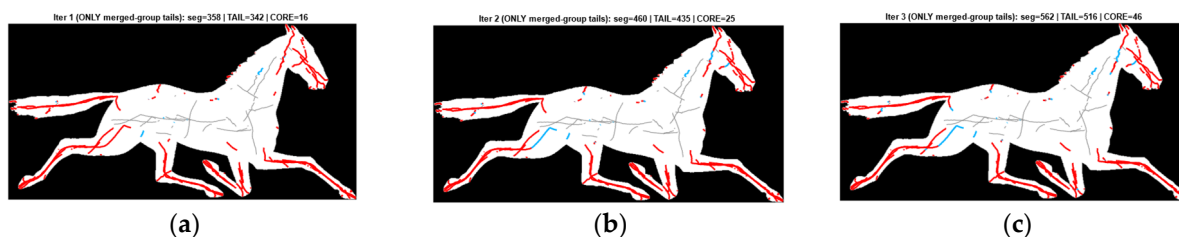


Figure 7. Examples of iterative midpoint splitting in Stage II: (a) first iteration, (b) second iteration, and (c) third iteration.

Feature 2: Thickness instability along the line (Detecting transitions and junctions)

A thin structure is not necessarily a tail if it represents a transition limb or noisy chain. Therefore, we explicitly measure how much thickness changes along the line. First define the local arc-length steps:

$$\Delta s_{g,m} = \sqrt{(x_{g,m+1} - x_{g,m})^2 + (y_{g,m+1} - y_{g,m})^2}. \quad (41)$$

where $\Delta s_{g,m}$: spatial step length (pixels).

Then define the discrete thickness change rate:

$$\gamma_{g,m} = \frac{|v_{g,m+1} - v_{g,m}|}{\Delta s_{g,m}}. \quad (42)$$

where $\gamma_{g,m}$: thickness change per pixel of arclength (units: pixels/pixel = dimensionless). We aggregate this robustly (less sensitive to outliers) using the median and normalize:

$$G_g = \frac{\text{median}_m(\gamma_{g,m})}{D_{\max}}. \quad (43)$$

where G_g measures how stable thickness is along the line. Tail structures tend to have smaller thickness fluctuations, whereas tail junctions and irregular limbs can produce larger values.

To obtain a bounded feature suitable for fuzzy membership design, we apply robust percentile scaling:

$$s_G = \max(\text{Prctile}(G, 90), \epsilon), G_g^{(01)} = \min\left(\frac{G_g}{s_G}, 1\right) \in [0, 1]. \quad (44)$$

A true tail typically exhibits a relatively consistent thin profile, i.e., low-to-moderate instability.

Junctions, merges, or noisy line candidates often show abrupt thickness changes (high instability).

Using the median and 90th-percentile scaling makes this feature robust to occasional sampling noise or local boundary irregularities.

This instability cue is critical: it allows Stage II to reject false tails that are thin only locally but unstable overall.

Feature 3: Coverage (Normalized line extent inside the shape)

Finally, we use the line coverage measure (already computed as the normalized length):

$$C_g = \tilde{\ell}_g \in [0, 1].$$

where C_g : how much of the object span the line covers (dimensionless).

A line that spans a meaningful part of the structure (higher C_g) is more likely to represent a real tail/core path rather than a short fragment. Coverage acts as a reliability cue, strengthening decisions when thickness statistics are ambiguous. The proposed Stage II inputs form a complementary and logically complete description of tail/core behavior:

“How thick is it?”— T_g (mean thickness)

“How stable is that thickness?”— $G_g^{(01)}$ (thickness instability)

“How much of the structure does it cover?”— C_g (coverage)

This triplet is not a generic feature set: it is explicitly designed for centroid-line semantic classification and is extracted from a continuous thickness field (EDT) rather than from fragile skeleton pixels.

$$\begin{aligned} t_{20}, t_{50}, t_{80} &= \text{Prctile}(t, 20, 50, 80) \\ c_{20}, c_{50}, c_{80} &= \text{Prctile}(c, 20, 50, 80) \\ g_{20}, g_{50}, g_{80} &= \text{Prctile}(G^{(01)}, 20, 50, 80) \end{aligned}$$

with monotonicity enforced via a small ϵ .

Membership functions.

We use triangular membership functions:

$$\text{trimf}(x; a, b, c) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ \frac{c-x}{c-b}, & b < x < c, \\ 0, & x \geq c. \end{cases} \quad (45)$$

where a, b, c : triangle support and peak parameters.

A. Thickness memberships (on $T \in [0,1]$)

$$\begin{aligned}\mu_{\text{short}}(T) &= \text{trimf}(T; 0, 0, t_{50}), \\ \mu_{\text{mid}}(T) &= \text{trimf}(T; t_{20}, t_{50}, t_{80}), \\ \mu_{\text{long}}(T) &= \text{trimf}(T; t_{50}, t_{80}, 1.1).\end{aligned}$$

“1.1” is used to ensure the triangle fully covers the upper boundary during implementation.

B. Instability memberships (on $G^{(01)} \in [0,1]$)

$$\begin{aligned}\mu_{\text{low}}(G) &= \text{trimf}(G; 0, 0, g_{50}), \\ \mu_{\text{mid}}(G) &= \text{trimf}(G; g_{20}, g_{50}, g_{80}), \\ \mu_{\text{high}}(G) &= \text{trimf}(G; g_{50}, g_{80}, 1.1).\end{aligned}$$

C. Coverage memberships (on C)

For the tail FLC input range $C \in [0, c_{80}]$ (implementation choice), the memberships are:

$$\begin{aligned}\mu_{\text{near}}(C) &= \text{trimf}(C; 0, 0, c_{50}), \\ \mu_{\text{good}}(C) &= \text{trimf}(C; c_{20}, c_{50}, c_{80}), \\ \mu_{\text{best}}(C) &= \text{trimf}(C; c_{50}, c_{80}, 1.1).\end{aligned}$$

Outputs of Stage II

Using the same three inputs $\left(\bar{t}_i, g_i, c_i\right)$

Each rule outputs one of four constants:

$S \in \{0.10, 0.36, 0.70, 1.00\}$. {NoneOut (reject), LowOut, MidOut, HighOut}

Tail likelihood:

$$S_g^{\text{tail}} = \text{FIS}_{\text{TAIL}}\left(T_g, G_g^{(01)}, C_g\right) \in [0, 1],$$

Core likelihood:

$$S_g^{\text{core}} = \text{FIS}_{\text{CORE}}\left(T_g, G_g^{(01)}, C_g\right) \in [0, 1].$$

Each of the 27 rules in FLC_{TAIL} and FLC_{CORE} outputs one of four Sugeno constants: NoneOut (0.10), LowOut (0.36), MidOut (0.70), HighOut (1.00).

The rule bases are deliberately complementary and monotonic: FLC_{TAIL} rewards low mean thickness + low instability + high coverage, while FLC_{CORE} rewards the opposite. Instability and coverage act as confidence modulators in both controllers, and the results are shown in Figure 7. These scores are then used in the subsequent iterative midpoint splitting of merged tail groups and in the final corner-aware peeling of Stage III, closing the loop between classification and progressive shape decomposition, as described more in Figure 8 at the next section.

The tail/core decision layer operates on continuous EDT-derived descriptors rather than directly on discrete skeleton heuristics. By deriving the fuzzy membership breakpoints from shape-specific statistics, the controller adapts to local geometric variation without relying on fixed global thresholds. These scores are then used in the subsequent iterative midpoint splitting stage.

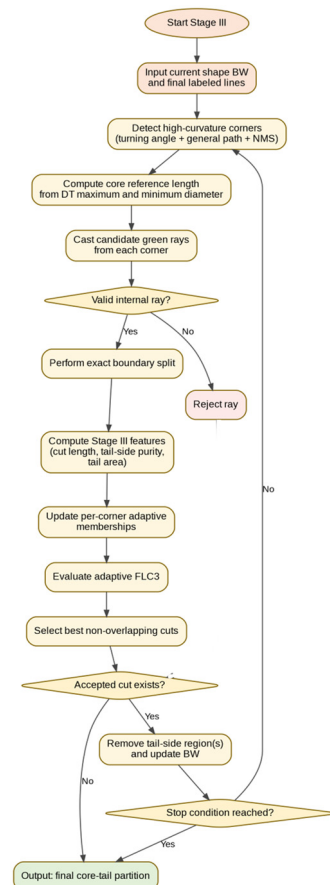


Figure 8. flowchart of Stage III for the adaptive fuzzy cut-ranking and iterative tail-peeling process.

3.3. Adaptive FLC3

The operations of Stage III are summarized in Figure 8 shown above and the corresponding Algorithm 3 pseudocode block below. After Stage II assigns semantic tail/core evidence to the retained centroid lines, Stage III determines where the actual shape cuts should be executed. To this end, a single adaptive Sugeno fuzzy controller, FLC_3 , is used to rank candidate cutting rays emitted from high-curvature boundary corners. The purpose of this controller is to assign a composite quality score to each candidate cut by jointly considering cut length, tail-side purity, and removed tail area. Rather than relying on fixed global thresholds, Stage III derives its fuzzy memberships adaptively from per-corner and per-iteration statistics. This makes the decision process robust to differences in protrusion size, local geometry, and the progressive topology changes that occur during peeling. After the highest-scoring non-overlapping cuts are executed, the tail-side regions are removed, the shape is updated, and the same corner-detection and fuzzy-ranking procedure is repeated on the reduced shape.

In this way, Stage III forms the final adaptive cut-selection layer of the TP-FLN framework. Its role is not only to rank candidate cuts but also to control the progressive, topology-preserving tail-peeling process, as summarized in Figure 8 and illustrated locally in Figure 9.



Figure 9. Example of green rays emitted from corner 35.

Algorithm 3. Stage III—Adaptive fuzzy cut ranking and iterative tail peeling

Input: Current binary shape BW, labeled lines Llabel, maximum peeling iterations T

Output: Final partition of BW into core and tail regions

```

1: Initialize BWcurrent ← BW
2: for t = 1 to T do
3:   Extract the outer boundary of BWcurrent
4:   Detect high-curvature corners using turning angle, general path, and NMS
5:   Compute DT on BWcurrent
6:   Estimate the core reference length Lref_core from the minimum diameter
7:   Cast candidate green rays from each accepted corner
8:   Keep only geometrically valid rays that stay mostly inside the shape
9:   for each candidate cut cj do
10:    Split BWcurrent exactly along cj
11:    Identify the smaller side as the candidate tail side
12:    Compute cut length, tail-side purity, and tail area
13:  end for
14:  Update adaptive per-corner FLC3 memberships from local quantiles
15:  Evaluate FLC3 for all candidate cuts
16:  Select the highest-scoring non-overlapping cuts
17:  if no accepted cut remains then
18:    break
19:  end if
20:  Remove accepted tail-side regions from BWcurrent
21: end for
22: Return the final partition given by the remaining core and removed tails

```

High-curvature points along the object boundary are identified using a turning-angle-based corner detector combined with the general path criterion. Let the ordered boundary sequence extracted from the largest 8-connected contour be denoted by:

$$P = \{p_i\}_{i=1}^N, p_i = (r_i, c_i), \quad (46)$$

where r_i and c_i are the row and column coordinates of the i -th boundary pixel, respectively. For each boundary point p_i , the local corner strength is estimated from the signed turning angle formed by two boundary direction vectors separated by a step size k :

$$\mathbf{v}_{\text{in}}(i) = \begin{bmatrix} c_i - c_{i-k} \\ r_i - r_{i-k} \end{bmatrix}, \mathbf{v}_{\text{out}}(i) = \begin{bmatrix} c_{i+k} - c_i \\ r_{i+k} - r_i \end{bmatrix}, \quad (47)$$

where circular indexing is used to preserve continuity along the closed contour. The signed turning angle at point i is then computed as

$$\theta_i = \text{atan2}\left(\det[\mathbf{v}_{\text{in}}(i), \mathbf{v}_{\text{out}}(i)], \mathbf{v}_{\text{in}}(i)^\top \mathbf{v}_{\text{out}}(i)\right), \quad (48)$$

where $\det[\cdot, \cdot]$ denotes the 2D determinant and $(\cdot)^\top$ denotes the transpose. In this form, θ_i preserves both the magnitude and orientation of the local boundary turn.

To isolate meaningful corners, the absolute turning angle $|\theta_i|$ is first thresholded using:

$$|\theta_i| \geq \theta_{\text{thr}}, \theta_{\text{thr}} = 40^\circ, \quad (49)$$

and the surviving candidates are further refined by non-maximum suppression within a local neighborhood. In addition to this local measure, a cumulative turning descriptor, referred to as the general path, is evaluated as

$$\text{GP}(i) = \sum_{j=1}^i \theta_j. \quad (50)$$

This accumulated angular variation provides a global consistency check and helps suppress isolated fluctuations caused by boundary noise or minor contour irregularities.

Finally, only stable and well-separated corners are retained by enforcing both a minimum circular index spacing and a minimum Euclidean distance between adjacent detections. Specifically, neighboring corners must satisfy a boundary-index separation of at least 12 samples and a spatial separation of at least 4 pixels. The resulting corner sets are evaluated as:

$$\mathcal{C} = \{\mathbf{c}_j\}_{j=1}^M \quad (51)$$

The resulting corner set defines the boundary locations from which the candidate green rays are emitted and later evaluated in Stage III. An example of these rays for one detected corner is shown in Figure 9.

Stage III Fuzzy Inputs For each candidate cut (green ray from a corner), Stage III computes three interpretable features:

Feature 1: Cut length (smaller is better)

$$L_c = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (52)$$

where (x_0, y_0) is the corner point and (x_1, y_1) is the opposite boundary intersection.

Feature 2: Tail-side label purity (larger is better) After exact boundary splitting, we measure the fraction of tail-labeled skeleton length that falls inside the smaller (tail) region:

$$\text{purity} = \frac{\text{length of TAIL segments in tail - side region}}{\text{total skeleton length in tail - side region}} (0 - 1) \quad (53)$$

Feature 3: Tail area (larger is better)

$$A_{\text{tail}} = \min(A_1, A_2) \quad (54)$$

(the pixel area of the smaller region created by the exact poly2mask split).

These three inputs form a logically complete decision space:

“How short is the cut?”— L_c

“How pure is the tail side?”—purity

“How large is the removed tail?”— A_{tail}

Reference scales: $L_{\text{ref_core}}$, L_{max} and A_{max} To guarantee full adaptivity at every iteration, the upper bounds of the fuzzy membership functions are derived directly from the current shape geometry.

The core reference length $L_{\text{ref_core}}$ is computed as follows:

Locate the point of maximum inscribed radius (the thickest point of the shape) using the Euclidean distance transform:

$$\left((x_0, y_0) = \arg \max_{(x,y)} D(x, y) \right) \quad (55)$$

From this point, compute the diameter in every direction $\theta = 0^\circ \dots 179^\circ$ and take the minimum diameter:

$$L_{\text{ref_core}} = \min_{\theta} \text{length of the chord through } (x_0, y_0)$$

This value represents the narrowest cross-section through the thickest core region.

We then define the adaptive upper bound for cut length:

$$L_{\text{max}} = L_{\text{ref_core}} \times 5 \quad (56)$$

Similarly, the upper bound for tail area is taken from the observed tail areas in the current iteration:

$$A_{\text{max}} = \max(A_{\text{tail}}) \times 1. \quad (57)$$

These two reference scales (L_{max} and A_{max}) are recomputed in every iteration and used to set the upper triangles of the “very long” and “large” membership functions, ensuring the fuzzy controller automatically adapts to the shrinking core as tails are progressively peeled.

Adaptive quantiles (per-corner and per-iteration breakpoints) To guarantee full adaptivity at every iteration, we derive corner-specific data-driven anchors directly from the green rays belonging to that corner on the current shape:

$L_{15}, L_{30}, L_{45}, L_{60}, L_{75}$ (15th to 75th percentiles of clamped lengths at that corner)

A_{20}, A_{50}, A_{80} (20th, 50th, 80th percentiles of tail areas at that corners)

These quantiles are recomputed in every iteration, so the fuzzy controller automatically adapts to the shrinking and simplifying geometry after each peel.

Membership functions Triangular membership functions whose parameters are set directly from the per-corner, per-iteration quantiles:

- A. Cut-length memberships (on L_c)
very short: $\text{trimf} [-0.1, L_{15}, L_{30}]$, short: $\text{trimf} [L_{15}, L_{30}, L_{45}]$, mid: $\text{trimf} [L_{30}, L_{45}, L_{60}]$
long: $\text{trimf} [L_{45}, L_{60}, L_{75}]$, very long: $\text{trimf} [L_{60}, L_{75}, L_{\text{max}}]$
 - B. Purity memberships (on $[0, 1]$)
low: $\text{trimf} [-0.1, 0, 0.45]$, mid: $\text{trimf} [0.25, 0.55, 0.85]$, high: $\text{trimf} [0.70, 1, 1.1]$
 - C. Tail area memberships (on A_{tail})
small: $\text{trimf} [-0.1, 0, A_{50}]$, mid: $\text{trimf} [A_{20}, A_{50}, A_{80}]$, large: $\text{trimf} [A_{50}, A_{80}, A_{\text{MAX}}]$
- Outputs of Stage III A single composite score:

$$\text{score}_{\text{FLC}_3} \in [0, 1]$$

Each of the 45 rules outputs one of five Sugeno constants: $\text{VL} = 0.10$, $\text{L} = 0.35$, $\text{M} = 0.60$, $\text{H} = 0.80$, $\text{VH} = 1.00$.

The 45 rules are monotonic and deliberately reward very short + high-purity + large-area cuts (up to VH) while heavily penalizing long or impure ones (down to VL). Low purity forces almost all rules to VL; high purity unlocks the full reward spectrum, with length acting as a soft penalty.

Iterative Execution The highest-scoring non-overlapping cuts (after simple tail area overlap merging) are executed in each iteration: the tail-side region is removed, the shape is updated, and stage 3 (corner detection → green rays → adaptive FLC₃) is re-run on the new, smaller shape. This closed-loop iteration continues for a user-specified number of steps (or until no high-scoring cuts remain), producing a progressive, topology preserving peeling that converges to a clean core. By integrating per-corner adaptive fuzzy ranking with iterative execution, Stage III updates its decision criteria after each peel and supports progressive topology-preserving tail removal on the evolving shape.

All membership functions and tables for the six Sugeno-type fuzzy logic controllers are detailed in the Appendix A section.

3.4. Novelty Summary

The novelty of the proposed TP-FLN framework lies primarily at the framework level, in the integration of three adaptive fuzzy stages into a single hierarchical, fully automatic, training-free pipeline for 2D core–tail partitioning. The method combines: (i) multi-orientation centroid-line extraction with fuzzy line selection, (ii) thickness-aware tail/core reasoning based on the Euclidean distance transform, and (iii) adaptive iterative cut ranking for progressive tail peeling. Rather than relying on manually fixed hard thresholds, the framework constructs its fuzzy decision boundaries from the statistics of the current shape, making the controllers data-driven and shape-adaptive at each stage. More specifically, Stage I derives its fuzzy memberships from pooled feature quantiles of the extracted centroid lines, allowing the line selection process to adapt to variations in scale, orientation, and morphology. Stage II uses continuous EDT-derived thickness descriptors to evaluate tail-like and core-like behavior through adaptive fuzzy reasoning rather than discrete skeleton pruning rules. Stage III updates its cut-ranking memberships from per-corner and per-iteration statistics as the shape evolves during successive peeling steps. In this way, the framework maintains a consistent adaptive decision process across extraction, semantic labeling, and cut execution. Accordingly, the contribution of TP-FLN is not claimed as the invention of each individual geometric or fuzzy operation in isolation but as their combination into a unified decomposition framework that avoids rigid threshold design and instead uses data-driven fuzzy inference throughout the full partitioning pipeline. This integrated design provides an interpretable and training-free approach for meaningful core–tail decomposition in binary shapes.

4. Results

The proposed TP-FLN framework was evaluated on the 2D Shape Structure Dataset of Carlier et al. [1], a public benchmark for perceptual 2D shape decomposition that provides hierarchical human annotations of the main body and meaningful peripheral parts. This dataset is appropriate for the present study because TP-FLN is specifically designed to separate a central core from protruding tail-like regions in binary shapes. In addition to the benchmark evaluation, several representative silhouettes and real-world binary masks were also used for qualitative analysis. For quantitative evaluation, two complementary protocols were used in this study. First, a controlled direct comparison with literature baselines was conducted only on shape instances that were explicitly reported in the original baseline papers and could be reliably matched to the Carlier et al. benchmark masks. This protocol was adopted because several baseline methods depend on method-specific

parameter settings and their official source code is not publicly available, making unofficial re-implementation under assumed settings methodologically unfair. Under this controlled protocol, TP-FLN was compared with five representative baselines: Approximate Convex Decomposition (ACD) [3], Convex Shape Decomposition (CSD) [5], the spectral near-convex decomposition method of Li et al. [2], the short-cut-rule-based decomposition model [4], and Minimum Near-Convex Shape Decomposition (MNCD) [16]. The comparative partitions shown in Figure 10 were therefore taken directly from the original publications [2–5,16], and the comparison was restricted to directly matched published examples. To support fair spatial evaluation, the published baseline partitions were registered to the corresponding benchmark masks, and a uniform semantic mapping was applied in which the largest component was treated as the core and the remaining components were grouped as tails. Second, a broader benchmark validation was conducted for TP-FLN against the Carlier et al. reference masks over a targeted subset of 32 categories, selected according to the decomposition objective of the present work. Specifically, the selection protocol retained categories whose shapes exhibit a discernible main-body/protrusion organization, making core-tail style partitioning visually meaningful and interpretable, while categories dominated by compact, nearly uniform, or weakly articulated shapes, for which a clear core-tail interpretation is not naturally defined, were excluded from the present analysis. This subset-based protocol was adopted in order to align the evaluation with the intended scope of the method rather than to maximize performance on arbitrary shape classes. Under this setting, the shapes illustrated in Figure 10 results are shown in Table 1, Table 2, and Figure 11 which present the tail count and segmentation performance comparisons across literature baselines. Additionally, Table 3 and Figure 12 summarize the TP-FLN-versus-benchmark categories evaluation. All experiments for TP-FLN were implemented in MATLAB 2024 using standard image-processing operations together with Sugeno fuzzy inference systems.

Table 1. Metrics for tail count agreement with the benchmark. Lower MAE and RMSE indicate better structural agreement, while higher exact-match and within-one accuracy indicate more reliable recovery of the benchmark tail configuration.

Method	Shapes Used	MAE	RMSE	Exact Match (%)	Within ± 1 (%)
ACD	7	9.29	12.97	0.00	0.00
CSD	7	6.86	10.77	0.00	0.00
MNCD	4	4.00	4.64	0.00	0.00
VR	7	4.14	5.41	28.57	28.57
RSD	7	5.00	6.31	14.29	28.57
TP-FLN	7	0.14	0.38	85.71	100.00

Table 2. Per-method Core and Tail Segmentation Performance: IoU and Dice Scores, averaged for the shapes in Figure 10. MNCD was calculated only from 4 available shapes.

Methods	Avg Core IOU	Avg Tail IOU	Avg Dice
ACD	40%	28%	52%
CSD	48%	35%	59%
MNCD	64%	43%	72%
VR	71%	53%	76%
RSD	74%	56%	78%
TP-FLN	91%	83%	92%

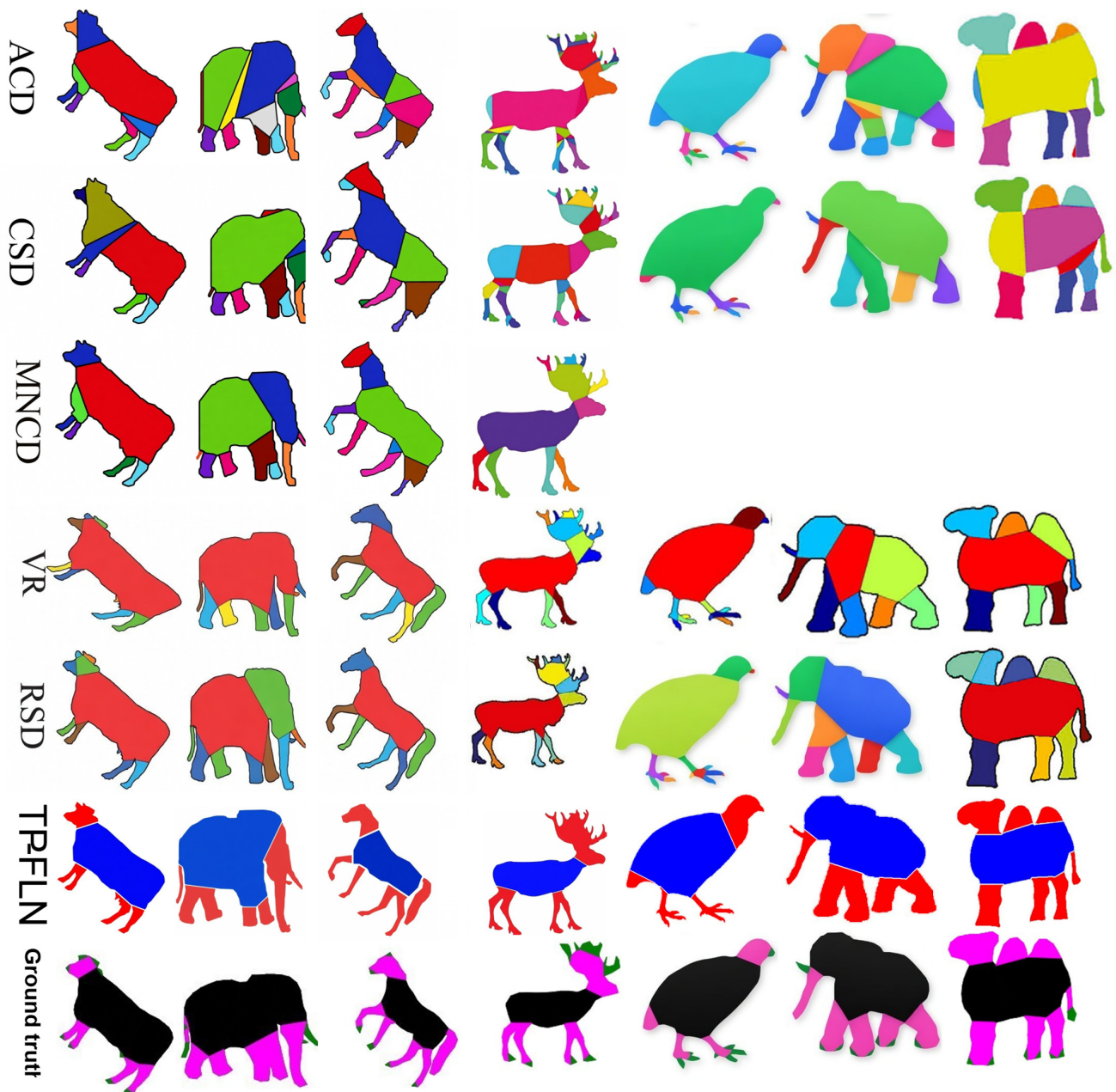


Figure 10. Visual results of our TP-FLN against the other methods with the ground-truth referenced in.

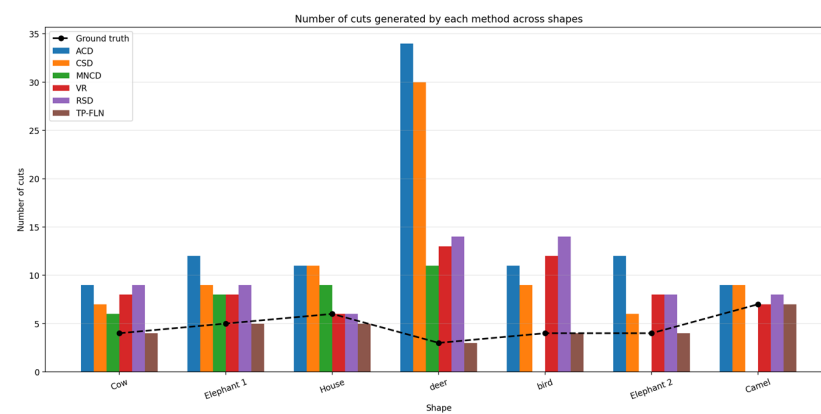


Figure 11. Tail count comparison across seven benchmark shapes. The dashed black line indicates the benchmark reference; TP-FLN shows the closest overall agreement.

Table 3. Per-category average Core/Tail IoU and Dice for TP-FLN versus the benchmark.

Category	Average Core IoU	Average Tail IoU	Dice
Lizard	94%	92%	96.4%
Fly	91%	84%	93.3%
Horse	95%	93%	97.2%
Jellyfish	94%	95%	96.4%
spoon	97%	95%	98.0%
Octopus	97%	97%	98.5%
Fork	98%	99%	99.2%
Camel	95%	94%	97.2%
Crown	98%	96%	98.5%
Deer	95%	94%	97.2%
turtle	97%	93%	97.4%
key	97%	98%	98.7%
dog	93%	85%	94.2%
cattle	96%	94%	97.4%
beetle	98%	97%	98.7%
ray	96%	94%	97.4%
guitar	95%	93%	96.9%
cup	97%	96%	98.2%
apple	98%	95%	98.2%
stars	99%	98%	99.2%
birds	94%	89%	95.6%
spring	95%	97%	98.0%
bottle	91%	86%	93.9%
chicken	95%	92%	96.6%
elephant	95%	89%	95.8%
fish	87%	83%	91.9%
flatfish	97%	99%	99.0%
frog	95%	89%	95.8%
glass	99%	98%	99.2%
pocket	97%	95%	98.0%
teddy	86%	86%	92.5%
tree	80%	77%	88.0%

Let N_i^{pred} = predicted tail count for shape i , N_i^{gt} = benchmark tail count for shape i , and M is the total number of shapes. The absolute error for one shape is

$$AE_i = |N_i^{\{pred\}} - N_i^{\{gt\}}| \quad (58)$$

$$MAE = \frac{1}{M} \sum_{i=1}^M |N_i^{pred} - N_i^{gt}| \quad (59)$$

$$RMSE = \sqrt{\left(\frac{1}{M} \sum_{i=1}^M |N_i^{pred} - N_i^{gt}| \right)} \quad (60)$$

Two discrete accuracy measures were also reported. The exact-match accuracy was defined as

$$Acc_{exact} = \frac{1}{M} \sum_{i=1}^M 1(N_i^{pred} - N_i^{gt}) \times 100, \quad (61)$$

and the within-one-tail accuracy was defined as

$$Acc_{\pm 1} = \frac{1}{M} \sum_{i=1}^M 1(|N_i^{pred} - N_i^{gt}| \leq 1) \times 100, \quad (62)$$

where $1(\cdot)$ is the indicator function. These tail count measures are useful because they summarize structural agreement at the object level and complement the region-based overlap scores.

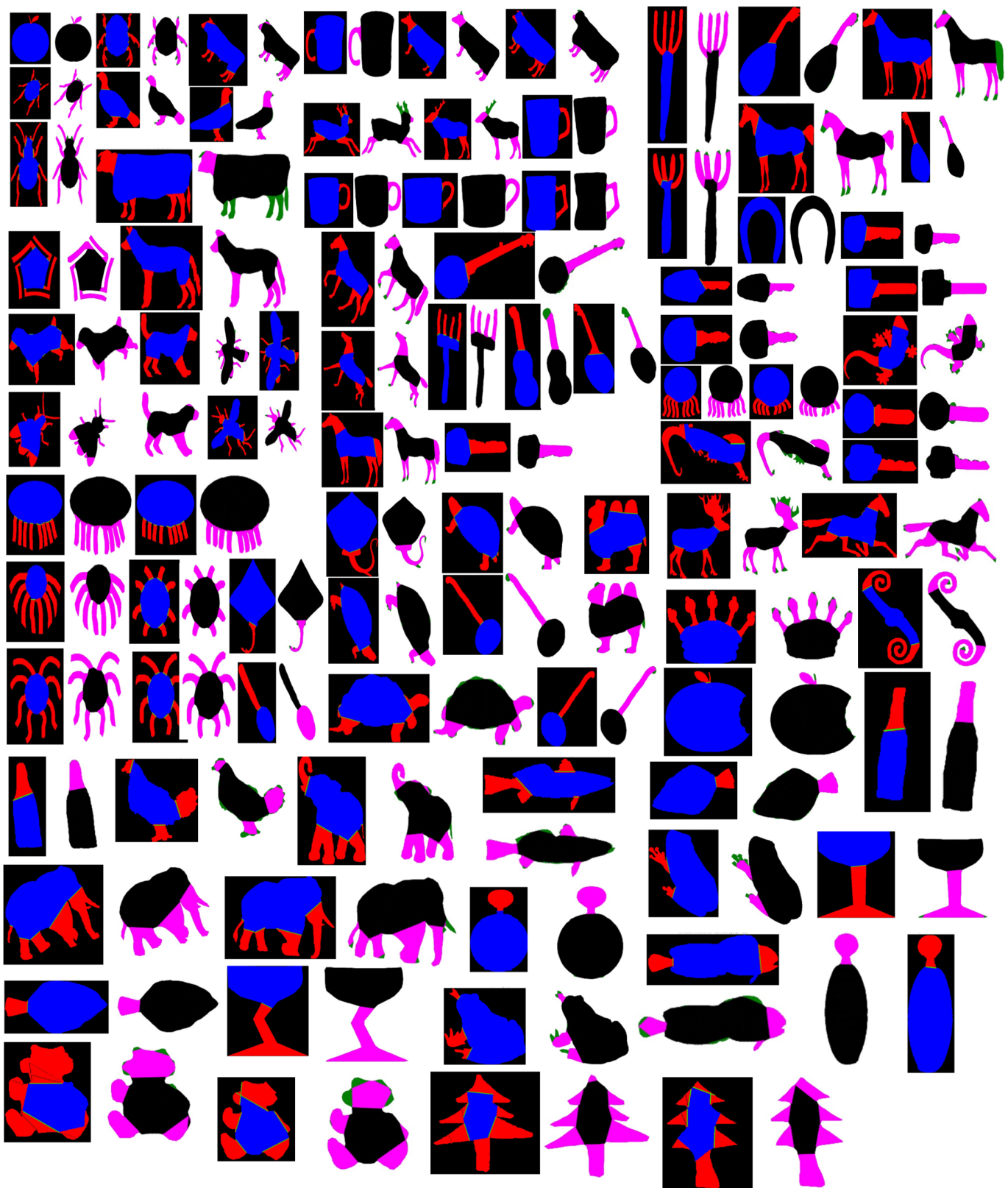


Figure 12. Representative examples from the TP-FLN-versus-benchmark category evaluation. Red and blue denote the TP-FLN tail and core regions, respectively, while pink and black denote the benchmark tail and core labels.

Complexity and Sensitivity Analysis

Because TP-FLN is designed as a largely self-adaptive framework, the sensitivity analysis was restricted to the small number of external control settings that are not derived directly from the input shape itself. In practice, the only fixed control parameter that remains externally set is the Stage I kept-lines threshold, which is used only to suppress weak centroid-line candidates before the later semantic and cut-selection stages. The remaining fuzzy membership breakpoints are not manually tuned; instead, they are computed automatically from shape-dependent statistics, such as pooled feature quantiles and per-iteration geometric measurements. Therefore, the framework does not rely on a large set of hard-coded thresholds but derives most of its decision boundaries directly from the current input shape. The sensitivity analysis showed that moderate variation in the kept-lines threshold produced only limited changes in the final overlap measures, with any differences largely confined to minor peripheral candidates rather than the final core-tail partition itself. This behavior supports the practical robustness of TP-FLN and confirms that its final decomposition is driven mainly by shape-adaptive fuzzy reasoning rather than by manual threshold adjustment.

From a computational perspective, the TP-FLN framework is training-free and consists of three sequential geometric–fuzzy stages. Stage I mainly involves connected-component extraction inside directional bands, centroid computation, nearest-band linking, and fuzzy scoring of the candidate lines. For an input mask with N foreground pixels and M extracted centroid-line candidates, this stage is approximately linear in the image traversal cost, with an additional pairwise linking cost between adjacent bands. Stage II operates only on the retained lines and evaluates line-wise thickness statistics along sampled points, making its cost proportional to the number of kept lines and their sampled lengths. Stage III is the most computationally demanding part, since candidate cuts are evaluated from detected corners and the procedure is repeated after each accepted peel. If C denotes the number of active corners, R the number of rays tested per corner, and I the number of peeling iterations, then the dominant cost is governed by this iterative cut-evaluation process.

The total computational cost of the proposed framework can be approximated as

$$T_{\text{TP-FLN}} = T_1 + T_2 + T_3,$$

where T_1 , T_2 , and T_3 denote the computational costs of Stages I, II, and III, respectively.

For the three stages, the complexity can be written approximately as

$$T_1 = \mathcal{O}(N + M), T_2 = \mathcal{O}(KS), T_3 = \mathcal{O}(ICRN)$$

and therefore, the overall complexity becomes

$$T_{\text{TP-FLN}} = \mathcal{O}(N + KS + ICRN).$$

Here, N is the number of foreground pixels in the binary shape, M is the number of candidate centroid lines extracted in Stage I, K is the number of retained lines after Stage I, S is the average number of sampled points per retained line in Stage II, C is the number of active corners in Stage III, R is the average number of candidate rays tested per corner, and I is the number of peeling iterations.

In practice, the third stage is the dominant term because the cut-evaluation process is repeated after each accepted peel. However, for typical 2D silhouettes, the numbers of retained lines, valid corners, and iterations remain limited, which keeps the method computationally feasible in practice.

5. Discussion

The results show that the proposed TP-FLN framework can produce shape partitions that are not only geometrically correct but also structurally meaningful. This is important because good shape partitioning is not only about finding possible cut locations. It also depends on whether the resulting parts match the way the object is naturally organized in perception. In this work, the combination of centroid-line extraction and fuzzy decision-making allows the partitioning process to go beyond strict geometric rules and better capture the difference between the main core region and its protruding parts. One of the main strengths of the proposed framework is its ability to handle shapes that have a central body with thin or elongated extensions. The multi-orientation centroid-line analysis helps capture the internal structure of the shape from different directions, while the fuzzy classifiers evaluate these candidates using clear geometric features. As a result, the final partition is guided by thickness, stability, support, and regional consistency instead of depending on a single hard threshold. This helps explain the strong agreement between the obtained partitions and the reference decomposition. Another important advantage of the method is that it is easy to understand and interpret. The whole pipeline is built on clear geometric measurements, and the fuzzy logic controllers combine them in a flexible and adaptive way. This makes the framework easier to analyze and explain than purely black-box approaches, while still allowing it to work well across different shape categories. In this way, the method provides a good balance between rule-based geometric reasoning and soft decision-making. From an application point of view, the proposed framework can be useful in several tasks related to structural shape analysis, such as silhouette understanding, part-based recognition, symmetry analysis, and graphics preprocessing. In many of these tasks, separating the main body of an object from its outer components is an important step before higher-level analysis can be performed. For this reason, the present work offers not only a new partitioning pipeline but also a practical and interpretable method for meaningful 2D shape decomposition. Overall, the results support the effectiveness of the proposed TP-FLN framework and show that structure-aware fuzzy reasoning can improve the quality of 2D shape partitioning. By combining multi-directional structural extraction with adaptive fuzzy evaluation, the method provides a coherent decomposition of complex binary shapes that remains consistent with human perception.

6. Conclusions

In this study, a new fuzzy-based framework, termed TP-FLN, was presented for meaningful 2D shape partitioning. The method combines multi-orientation centroid-line extraction, fuzzy tail/core classification, and adaptive cut selection to separate complex binary shapes into structurally meaningful parts. Unlike purely geometric partitioning approaches, the proposed framework is designed to consider both internal shape organization and perceptual part interpretation. Experimental results on a targeted subset of complex shapes indicate that the method produces coherent decompositions, demonstrating competitive alignment with the human-annotated reference partitions. The use of fuzzy logic also provides an important advantage in terms of interpretability, since the final decisions are based on clear geometric cues rather than on rigid thresholds or black-box learning. Overall, the proposed TP-FLN framework offers an effective, automatic, and interpretable solution for 2D shape decomposition. It provides a useful contribution to part-based shape analysis and may serve as a practical basis for future extensions in more advanced structural and perceptual shape understanding tasks.

Author Contributions: Conceptualization, H.N. and O.Y.; methodology, software. H.N. formal analysis, investigation, resources, data curation, O.Y.; writing—original draft preparation, H.N. and Y.G. writing—review and editing, H.N. and Y.G.; visualization and supervision, O.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study are publicly available, at <https://github.com/2dshapesstructure/dataset-processing>, <https://drive.google.com/drive/folders/1F1TEZOgShNSGZn2uBtbw0xYen25zsPn9?usp=sharing> (accessed on 3 February 2026).

Conflicts of Interest: Authors declare no conflicts of interest.

Appendix A. Fuzzy Rule Tables

The proposed method uses six Sugeno-type fuzzy logic controllers (FLCs): three controllers for the centroid-line selection stage, two controllers for tail/core discrimination, and one controller for adaptive cut evaluation. For clarity, the rule bases are reported below in compact tabular form. In all tables, the entry in each cell denotes the consequent output of the corresponding rule.

Appendix A.1. Output Labels Used by the FLCs

Table A1. Output constants used in the fuzzy controllers.

Controller	Output Labels	Sugeno Constant Values
FLC_X, FLC_Y, FLC_ANG	NoneOut, LowOut, MidOut, HighOut	0.10, 0.30, 0.60, 1.00
FLC_TAIL	NoneOut, LowOut, MidOut, HighOut	0.10, 0.36, 0.70, 1.00
FLC_CORE	NoneOut, LowOut, MidOut, HighOut	0.00, 0.25, 0.70, 1.00
FLC3	VL, L, M, H, VH	0.10, 0.35, 0.60, 0.80, 1.00

Appendix A.2. Rule Tables for the Centroid-Line Selection Stage FLCs

Appendix A.2.1. FLC_X (Horizontal Preference)

Inputs: lineLength = {Short, Mid, Long}; lineAngleErr = {Bad, Mid, Likemid, Good, Alsogood}; lineEdgeDist = {Near, Good, Best}

Table A2. FLC_X rules for lineEdgeDist = near.

	Bad	Mid	Likemid	Good	Alsogood
short	NoneOut	NoneOut	NoneOut	MidOut	MidOut
mid	NoneOut	LowOut	LowOut	MidOut	MidOut
long	LowOut	MidOut	MidOut	HighOut	HighOut

Table A3. FLC_X rules for lineEdgeDist = good.

	Bad	Mid	Likemid	Good	Alsogood
short	NoneOut	LowOut	LowOut	MidOut	MidOut
mid	LowOut	MidOut	MidOut	HighOut	HighOut
long	MidOut	HighOut	HighOut	HighOut	HighOut

Table A4. FLC_X rules for lineEdgeDist = best.

	Bad	Mid	Likemid	Good	Alsogood
short	LowOut	MidOut	MidOut	MidOut	MidOut
mid	MidOut	HighOut	HighOut	HighOut	HighOut
long	HighOut	HighOut	HighOut	HighOut	HighOut

Appendix A.2.2. FLC_Y (Vertical Preference)

Inputs: lineLength = {Short, Mid, Long}; lineAngleErr = {Bad0, Mid, Good, Likemid, Bad180}; lineEdgeDist = {Near, Good, Best}

Table A5. FLC_Y rules for lineEdgeDist = near.

	Bad0	Mid	Good	Likemid	Bad180
short	NoneOut	NoneOut	LowOut	NoneOut	NoneOut
mid	NoneOut	LowOut	MidOut	LowOut	NoneOut
long	LowOut	MidOut	HighOut	MidOut	LowOut

Table A6. FLC_Y rules for lineEdgeDist = good.

	Bad0	Mid	Good	Likemid	Bad180
short	NoneOut	LowOut	MidOut	LowOut	NoneOut
mid	LowOut	MidOut	MidOut	MidOut	LowOut
long	MidOut	HighOut	HighOut	HighOut	MidOut

Table A7. FLC_Y rules for lineEdgeDist = best.

	Bad0	Mid	Good	Likemid	Bad180
short	LowOut	MidOut	MidOut	MidOut	LowOut
mid	LowOut	HighOut	HighOut	HighOut	LowOut
long	LowOut	MidOut	HighOut	HighOut	LowOut

Appendix A.2.3. FLC_ANG (Diagonal Preference)

Inputs: lineLength = {Short, Mid, Long}; lineAngleErr = {Bad0, Bad90, Bad180, Mid70, Mid110, Good45, Good135}; lineEdgeDist = {Near, Good, Best}

Table A8. FLC_ANG rules for lineEdgeDist = near.

	Bad0	Bad90	Bad180	Mid70	Mid110	Good45	Good135
short	NoneOut	NoneOut	NoneOut	NoneOut	NoneOut	NoneOut	NoneOut
mid	NoneOut	NoneOut	NoneOut	LowOut	LowOut	MidOut	MidOut
long	LowOut	LowOut	LowOut	MidOut	MidOut	HighOut	HighOut

Table A9. FLC_ANG rules for lineEdgeDist = good.

	Bad0	Bad90	Bad180	Mid70	Mid110	Good45	Good135
short	NoneOut	NoneOut	NoneOut	LowOut	LowOut	MidOut	MidOut
mid	LowOut	LowOut	LowOut	MidOut	MidOut	HighOut	HighOut
long	MidOut	MidOut	MidOut	HighOut	HighOut	HighOut	HighOut

Table A10. FLC_ANG rules for lineEdgeDist = best.

	Bad0	Bad90	Bad180	Mid70	Mid110	Good45	Good135
short	LowOut	LowOut	LowOut	MidOut	MidOut	MidOut	MidOut
mid	MidOut	MidOut	MidOut	HighOut	HighOut	HighOut	HighOut
long	HighOut	HighOut	HighOut	HighOut	HighOut	HighOut	HighOut

Appendix A.3. Rule Tables for the Tail/Core Discrimination Stage

Inputs for both controllers: thickMean = {short, mid, long}; thickGrad = {low, mid, high}; covLen = {near, good, best}.

Table A11. FLC_TAIL rule table.

thickMean/thickGrad	covLen = Near	covLen = Good	covLen = Best
short, low	MidOut	HighOut	HighOut
short, mid	LowOut	MidOut	MidOut
short, high	NoneOut	LowOut	LowOut
mid, low	MidOut	MidOut	HighOut
mid, mid	MidOut	MidOut	MidOut
mid, high	NoneOut	NoneOut	LowOut
long, low	NoneOut	NoneOut	NoneOut
long, mid	NoneOut	NoneOut	NoneOut
long, high	NoneOut	NoneOut	NoneOut

Table A12. FLC_CORE rule table.

thickMean/thickGrad	covLen = Near	covLen = Good	covLen = Best
short, low	NoneOut	NoneOut	LowOut
short, mid	NoneOut	NoneOut	LowOut
short, high	NoneOut	NoneOut	NoneOut
mid, low	LowOut	LowOut	NoneOut
mid, mid	LowOut	LowOut	LowOut
mid, high	LowOut	LowOut	LowOut
long, low	MidOut	HighOut	HighOut
long, mid	MidOut	HighOut	HighOut
long, high	MidOut	MidOut	MidOut

*Appendix A.4. Rule Tables for the Adaptive Cut-Selection Stage***FLC3**

Inputs: cutLen = {Very Short, Short, Mid, Long, Very Long}; Purity = {Low, Mid, High}; tailArea = {Small, Mid, Large}. For Compact Presentation, the Rule Base Is Reported Separately for Each Purity Level

Table A13. FLC3 rules for purity = low.

cutLen\tailArea	Small	Mid	Large
very short	VL	VL	L
short	VL	VL	VL
mid	VL	VL	VL
long	VL	VL	VL
very long	VL	VL	VL

Table A14. FLC3 rules for purity = mid.

cutLen\tailArea	Small	Mid	Large
very short	L	M	H
Short	L	L	M
Mid	M	M	L
Long	L	L	L
very long	VL	VL	VL

Table A15. FLC3 rules for purity = high.

cutLen\tailArea	Small	Mid	Large
very short	M	H	VH
short	M	H	H
mid	L	L	M
long	VL	L	M
very long	VL	VL	VL

References

1. Carlier, A.; Leonard, K.; Hahmann, S.; Morin, G.; Collins, M. The 2D Shape Structure Dataset: A User Annotated Open Access Database. *Comput. Graph.* **2016**, *58*, 23–30. [\[CrossRef\]](#)
2. Li, Z.; Hu, J.; Stojmenovic, M.; Liu, Z.; Liu, W. Revisiting Spectral Clustering for Near-Convex Decomposition of 2D Shape. *Pattern Recognit.* **2020**, *105*, 107371. [\[CrossRef\]](#)
3. Lien, J.-M.; Amato, N.M. Approximate Convex Decomposition of Polygons. *Comput. Geom. Theory Appl.* **2006**, *35*, 100–123. [\[CrossRef\]](#)
4. Luo, L.; Shen, C.; Liu, X.; Zhang, C. A Computational Model of the Short-Cut Rule for 2D Shape Decomposition. *IEEE Trans. Image Process.* **2015**, *24*, 273–283. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Liu, H.; Liu, W.; Latecki, L.J. Convex shape decomposition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 97–104. [\[CrossRef\]](#)
6. Singh, M.; Seyranian, G.D.; Hoffman, D.D. Parsing Silhouettes: The Short-Cut Rule. *Percept. Psychophys.* **1999**, *61*, 636–660. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Shapira, L.; Shamir, A.; Cohen-Or, D. Consistent Mesh Partitioning and Skeletonisation Using the Shape Diameter Function. *Vis. Comput.* **2008**, *24*, 249–259. [\[CrossRef\]](#)
8. Papanelopoulos, N.; Avrithis, Y.; Kollias, S. Revisiting the Medial Axis for Planar Shape Decomposition. *Comput. Vis. Image Underst.* **2019**, *179*, 66–78. [\[CrossRef\]](#)
9. Asafi, S.; Goren, A.; Cohen-Or, D. Weak Convex Decomposition by Lines-of-sight. *Comput. Graph. Forum* **2013**, *32*, 23–31. [\[CrossRef\]](#)
10. Bloch, I. Fuzzy Skeleton by Influence Zones—Application to Image Segmentation. *Pattern Recognit. Lett.* **2008**, *29*, 203–213. [\[CrossRef\]](#)
11. Nedyalkova, M.; Simeonov, V.; Madjarova, G.; Simeonov, P. Fuzzy Divisive Hierarchical Clustering of Solvents Based on Abraham Descriptors. *Symmetry* **2020**, *12*, 1763. [\[CrossRef\]](#)
12. Radojević, M.; Žalik, B. Fuzzy-Logic Based Detection and Characterization of Junctions and Terminations in Fluorescence Microscopy Images of Neurons. *Comput. Biol. Med.* **2016**, *68*, 49–56. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Žalik, B.; Mongus, D.; Lukač, N. A Hierarchical Universal Algorithm for Geometric Objects' Reflection Symmetry Detection. *Symmetry* **2022**, *14*, 1060. [\[CrossRef\]](#)
14. Mi, X.; DeCarlo, D. Separating Parts from 2D Shapes using Relatability. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8. [\[CrossRef\]](#)
15. Liu, G.; Gingold, Y.; Lien, J.-M. Continuous Visibility Feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015; pp. 1182–1190. [\[CrossRef\]](#)
16. Ren, Z.; Yuan, J.; Liu, W. Minimum Near-Convex Shape Decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2546–2552. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Carlier, A.; Leonard, K.; Hahmann, S.; Morin, G.; Collins, M. A 2D Shape Structure for Decomposition and Part Similarity. In *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, Cancun, Mexico, 4–8 December 2016; IEEE: New York, NY, USA, 2016; pp. 301–306. [\[CrossRef\]](#)
18. Shamir, A. A Survey on Mesh Segmentation Techniques. *Comput. Graph. Forum* **2008**, *27*, 1539–1556. [\[CrossRef\]](#)

19. Theologou, P.; Pratikakis, I.; Theoharis, T. A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation. *Comput. Vis. Image Underst.* **2015**, *135*, 49–82. [[CrossRef](#)]
20. Kulkarni, A.D. Fuzzy Convolution Neural Networks for Tabular Data Classification. *IEEE Access* **2024**, *12*, 151846–151855. [[CrossRef](#)]
21. Ren, W.; Ma, O.; Ji, H.; Liu, X. Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches. *IEEE Access* **2020**, *8*, 135628–135639. [[CrossRef](#)]
22. Mehran, K. Takagi–Sugeno Fuzzy Modeling for Process Control. In *Fuzzy Logic and Applications*; Springer: Berlin/Heidelberg, Germany, 2008.
23. Yang, J.; Wang, Y.; Wang, T.; Yang, X. Fuzzy-Based Tracking Control for a Liu-Chen Four-Scroll Chaotic System. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 927–939.
24. Held, A.; Abe, K. On the decomposition of binary shapes into meaningful parts. *Pattern Recognit.* **1994**, *27*, 637–647. [[CrossRef](#)]
25. Mi, X.; DeCarlo, D.; Stone, M. Abstraction of 2D Shapes in Terms of Parts. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering (NPAR 2009), New Orleans, LA, USA, 1–2 August 2009*; Association for Computing Machinery (ACM Press): New York, NY, USA; pp. 15–24. [[CrossRef](#)]
26. Jiang, T.; Dong, Z.; Ma, C.; Wang, Y. Toward Perception-Based Shape Decomposition. In *Computer Vision—ACCV 2012; Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7725, pp. 188–201. [[CrossRef](#)]
27. Ma, C.; Dong, Z.; Jiang, T.; Wang, Y.; Gao, W. A Method of Perceptual-Based Shape Decomposition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2013; pp. 873–880. [[CrossRef](#)]
28. Fotopoulou, F.; Psarakis, E.Z. A Visibility Graph Based Shape Decomposition Technique. In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISAPP)*; IEEE: New York, NY, USA, 2014; pp. 515–522. [[CrossRef](#)]
29. Papanelopoulos, N.; Avrithis, Y. Planar Shape Decomposition Made Simple. In *Proceedings of the British Machine Vision Conference (BMVC)*; British Machine Vision Association (BMVA Press): Durham, UK, 2015; pp. 13.1–13.12. [[CrossRef](#)]
30. Siddiqi, K.; Kimia, B.B. Parts of Visual Form: Computational Aspects. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 239–251. [[CrossRef](#)]

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.