

© Copyright 2026

Dingyuan Xue

Training-Free Feature-Based Ranking for Multi-Candidate Mask Selection in Image
Segmentation

Dingyuan Xue

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2026

Reading Committee:

Kelvin Sung

Munehiro Fukuda

Clark Olson

Program Authorized to Offer Degree:
Computing & Software Systems

University of Washington

Abstract

Training-Free Feature-Based Ranking for Multi-Candidate Mask Selection in Image
Segmentation

Dingyuan Xue

Chair of the Supervisory Committee:

Kelvin Sung

Computing and Software Systems

Recent advances in neural-network-based and promptable image segmentation have improved the accuracy, robustness, and flexibility of object-mask generation. Under ambiguous visual conditions, however, these systems may produce multiple plausible candidate masks for the same prompted target, whereas downstream applications generally expect only one mask. This creates a candidate-selection problem: determining which candidate most accurately delineates the intended target. Existing approaches often rely on learned quality predictors, additional supervision, calibration data, reference datasets, or model-specific modification. These

dependencies increase implementation effort and limit portability when the candidate image segmentation system is pretrained, externally supplied, or otherwise fixed.

This thesis presents the ACES (Area, Center-location, Edge-proximity, and Silhouette) framework, a training-free, user-guided scoring system for inference-time candidate ranking. Inspired by classical binary-image analysis, ACES measures four corresponding geometric and morphological properties for each candidate mask and compares these observations with four target reference values representing the expected properties of the intended target. These comparisons produce four compatibility terms, which are summed to obtain the final ACES score. The highest-scoring candidate mask is recommended as the final output.

The ACES framework provides a versatile default set of target reference values derived through controlled experiments. These reference values can be modified based on application knowledge and the characteristics of the target geometry. An existing candidate set can therefore be re-ranked under different target references without repeating candidate mask generation. Since the ranking requires only candidate masks and image dimensions, the scoring computation can be applied to outputs from different segmentation pipelines as a lightweight post-processing stage.

The ACES framework is evaluated on the DAVIS 2016 benchmark using candidate masks generated by a Grounding DINO and SAM 2 pipeline. Two settings are examined. The first applies the default reference values to all sequences without target-specific adjustments. The second derives reference values based on the geometric characteristics of the intended target. In the default setting, ACES increases mean intersection over union (IoU) from 0.6175 for upstream confidence-based selection to 0.7308. The framework is most effective when the candidate set contains at least one mask that substantially overlaps the intended target and when the target’s scale, location, and shape remain sufficiently stable across the evaluated frames to be represented by a single set of reference values. Under these conditions, ACES can suppress distractor masks that receive higher upstream confidence and recommend a candidate that more closely matches the intended target. These results demonstrate that reference-based geometric scoring provides an effective, interpretable, and portable basis for multi-candidate mask selection while preserving the existing segmentation pipeline.

TABLE OF CONTENTS

List of Figures	iv
List of Tables	vi
Chapter 1. Introduction	1
Chapter 2. Related Work.....	4
2.1 Image Segmentation.....	4
2.2 Segmentation Quality Evaluation and Candidate Selection	5
2.3 Observations and Opportunities.....	6
Chapter 3. Method	8
3.1 Problem Formulation and Inference-Time Candidate Selection Pipeline	8
3.2 Candidate Properties and Compatibility Terms.....	11
3.2.1 Area Compatibility.....	12
3.2.2 Center-Location Compatibility	13
3.2.3 Edge-Proximity Compatibility.....	14
3.2.4 Silhouette Compatibility	16
3.3 Unified Reference-Matching Score and Candidate Selection	17
3.4 Limitations of the Proposed Framework.....	18
Chapter 4. Implementation.....	20
4.1 Implementation Requirements and Tool Selection.....	20
4.2 Property Measurement and Compatibility Computations.....	21
4.2.1 Area and Center Compatibility	21

4.2.2	Edge Proximity Compatibility	22
4.2.3	Contour and Connected-Component Analysis.....	22
4.2.4	Hole Detection and Silhouette Compatibility	23
4.2.5	Unified Score and Candidate Selection	24
4.3	Analysis Tools	25
Chapter 5. Systematic Reference Value Specification and Case Studies		28
5.1	A Systematic Procedure for Deriving the Values of a T-Vector	28
5.2	Bear Case Study	29
5.2.1	Area Reference Value (tA).....	29
5.2.2	Center-Location Reference Value (tC)	30
5.2.3	Edge-Proximity Reference Value (tE).....	31
5.2.4	Silhouette Reference Value (tS).....	32
5.2.5	ACES Score	32
5.3	Additional Examples for Edge and Silhouette Reference Values	33
5.4	Summary	36
Chapter 6. Results		37
6.1	Default Setting Evaluation	37
6.2	Partial GT Derived Reference Value Evaluation.....	41
6.2.1	Partial GT Reference Value Specification	41
6.2.2	Candidate-Set Screening and Overall Performance.....	42
6.2.3	Sequence-Level Results and Parkour Case Study	44
6.3	Limitations and Failure Analysis	47

6.4	Summary	48
Chapter 7. Conclusion and Future Work		50
7.1	Conclusion	50
7.2	Future Work	51
Bibliography		53

LIST OF FIGURES

Figure 1.1. Image segmentation assigns object regions at the pixel level, providing detailed information about object shape and spatial extent	1
Figure 3.1. Overview of the proposed inference-time candidate-mask selection framework. An upstream segmentation system generates multiple candidate masks, which ACES evaluates and ranks to select a final mask.	10
Figure 4.1. Analysis dashboard for inspecting candidate-mask property values, target reference values, compatibility terms, and the final unweighted ACES ranking.	27
Figure 5.1. Bear case study: (a) original image and (b) ground-truth mask of the intended target.	29
Figure 5.2. Visual approximation of the bear’s image-normalized area using an enclosing bounding box.	30
Figure 5.3. Visual estimation of the center-location reference value using the target center, image center, and half-diagonal reference distance.	31
Figure 5.4. Visual approximation of the bear’s edge-proximity reference value.	32
Figure 5.5. Contrasting edge-proximity configurations: (a) an interior foreground region and (b) a boundary-adjacent foreground region.	34
Figure 5.6. Silhouette-reference example: (a) original bicycle image, (b) a correct mask with $S = 0.157$, and (c) a candidate mask with $S = 0.575$ that fills the hollow interiors of the bicycle tires.	35
Figure 6.1. Per-sequence mean IoU comparison between SAM confidence-based selection and scores from the ACES reference function.	38
Figure 6.2. Representative candidate-selection improvements in multi-instance scenes: raw images (left), SAM confidence-based selections (middle), and ACES selections (right) for the flamingo and breakdance sequences.	39
Figure 6.3. Representative near-tie cases in which SAM confidence-based selection and ACES produce nearly identical masks for the camel, blackswan, and bear sequences.	40

Figure 6.4. Representative candidate-generation failure in the kite-surf sequence: (a) raw image, (b) ground-truth target, and (c–d) two incomplete candidates generated by the Grounding DINO and SAM 2 pipeline.	44
Figure 6.5. Per-sequence mean IoU comparison between the default ACES reference function and that based on partial GT derived reference values, excluding the 5 failure sequences.	45
Figure 6.6. Parkour candidate-selection example: (a) raw image, (b) ground-truth overlay, (c) SAM-confidence selection, (d) partial-GT ACES selection, and (e) default ACES selection.	45
Figure 6.7. Target-property variation in the motocross-jump sequence, from an early frame to a later frame.	47

LIST OF TABLES

Table 6.1. Overall candidate-selection performance from the DAVIS 2016 sequences. .	38
Table 6.2. Candidate-set quality and partial GT derived ACES performance before and after excluding upstream candidate-generation failures.....	43
Table 6.3. Image-weighted candidate-selection performance after excluding sequences with oracle candidate mean IoU below 0.60.....	44

Chapter 1. INTRODUCTION

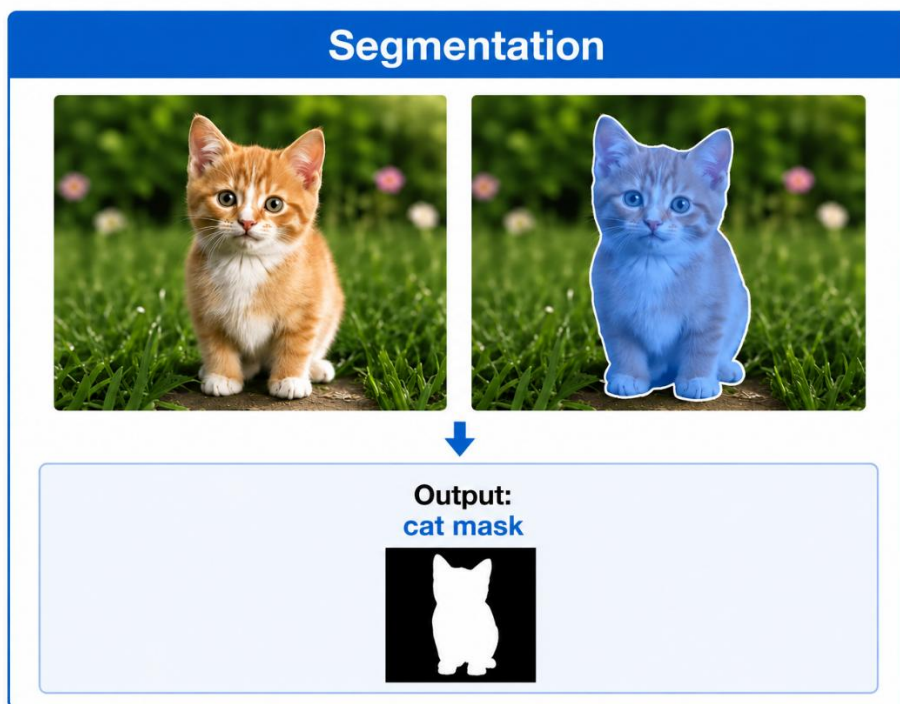


Figure 1.1. Image segmentation assigns object regions at the pixel level, providing detailed information about object shape and spatial extent

In computer vision, as illustrated in Figure 1.1, image segmentation assigns labels or object regions at the pixel level, providing detailed information about object shape and spatial extent. Because of the fine-grained representation, it has become an important component in many computer vision applications [1], [2]. The results of image segmentation, referred to as masks, often serve as intermediate representations for later processing stages. For example, masks may be used to separate manipulable objects from the background in robotic systems, isolate foreground regions in image-editing tools, or identify object geometry before estimating shape or structure in reconstruction pipelines [3], [4], [5], [6], [7].

Methods for computing segmentation masks have developed from traditional image-processing approaches to modern deep learning models. Earlier methods relied on image cues visible to humans, such as edges, color similarity, region continuity, and boundary smoothness [2], [8], [9], [10], [11], [12]. These methods perform well in many cases; however,

appearance variation, cluttered backgrounds, weak boundaries, and complex object structures can often pose challenges.

Modern neural network architectures, including FCN, U-Net, Mask R-CNN, and transformer-based segmentation models, address many of these limitations by learning hierarchical visual representations directly from training data [13], [14], [15], [16]. More recently, segmentation models such as the Segment Anything Model (SAM) support mask generation based on flexible prompts that specify or localize the target object, including points, bounding boxes, and external localization results [17]. Together, these advances have substantially improved the accuracy, robustness, and flexibility of image segmentation across a wide range of visual domains.

In real images, object boundaries may be weak, objects may be partially occluded, and foreground regions may resemble nearby background regions [1]. Under these conditions, a segmentation system may produce several plausible candidate masks for the object of interest, or the target object [17]. These candidate masks typically attempt to capture the same target in different ways, varying in object extent, boundary accuracy, fragmentation, or the inclusion of nearby distractor regions. Although multiple candidate masks can preserve useful segmentation hypotheses, downstream applications usually require a single final mask rather than a set of competing outputs. This thesis focuses on the candidate-selection problem resulting from modern segmentation models: when multiple plausible masks are available for the same target object, how can the most reliable mask be selected for downstream applications?

Existing work has explored approaches to estimate or improve mask quality, including learned quality prediction, confidence calibration, uncertainty estimation, and proxy-based evaluation methods [18], [19], [20], [21]. These methods can improve segmentation performance. However, these approaches require additional supervision, model retraining, architectural modifications, reference data, or additional evaluation procedures. These requirements introduce additional implementation effort and can complicate the deployment of a segmentation model in larger inference pipelines.

In contrast, this thesis considers an alternative setting: the upstream segmentation system is treated as a fixed candidate-mask generator where the task is to select among the produced masks during inference. A solution is proposed to examine explicit visual cues, based on results from classical segmentation methods, to directly evaluate candidate masks without modifying or retraining the upstream model.

Traditional segmentation methods were designed around explicit visual cues that are prevalent in modern segmentation outputs. For example, different candidate masks often vary in region continuity, boundary stability, internal completeness, and leakage into distractor regions [2], [8], [9], [10], [11], [12]. From this perspective, classical cue-based analysis complements neural-network-based mask generation: the neural model provides semantically plausible candidates. At the same time, explicit visual-cue evaluation can help distinguish candidates with stronger geometric and structural consistency. This observation motivated the examination of classical-style geometric and spatial measurements as a post-processing strategy for candidate mask selection.

A lightweight scoring system is proposed to rank candidate masks generated by an upstream segmentation model. Instead of learning a new quality predictor or modifying the segmentation model, the proposed system evaluates each candidate mask based on a set of interpretable geometric and spatial cues. Each cue measures one aspect of mask plausibility, such as foreground extent, centroid location, image-edge proximity, and silhouette structure. These cue-based features are converted into scores for each candidate mask, allowing the system to select the most plausible mask for downstream use while remaining training-free and model-independent.

To evaluate the proposed framework, experiments were conducted on the DAVIS 2016 benchmark [22] using candidate masks generated by Grounding DINO and the SAM 2 pipeline [23], [24]. The results show that the proposed scoring system performs well in the default training-free setting, supports manual adjustment when required, and can achieve further improvement with limited ground-truth annotations. However, because the method relies on lightweight geometric priors, its effectiveness may decrease when the handcrafted functions cannot capture geometric or topological structures, or when inconsistencies in these structures occur within a video sequence.

This thesis demonstrates that classical geometric and spatial cues can be effectively integrated with modern segmentation pipelines to support inference-time candidate-mask selection. The results show that this combination of traditional visual algorithms and modern mask generation can improve selection performance while remaining interpretable, training-free, and compatible with existing segmentation systems. Future work should further examine how such cue-based ranking strategies can be generalized to support more diverse object categories, dynamic scenes, and segmentation pipelines.

Chapter 2. RELATED WORK

This chapter reviews three areas of prior work that are relevant to the candidate-mask selection problem. First, Section 2.1 summarizes the development of image segmentation methods, from traditional to modern foundation segmentation systems capable of generating multiple candidate masks. Section 2.2 reviews existing approaches for evaluating mask quality and selecting segmentation outputs. Finally, Section 2.3 discusses geometric and spatial cues that have traditionally been used to assess the quality of regions and the structural consistency of masks. Together, these areas provide the foundation for the proposed feature-based candidate-ranking framework.

2.1 IMAGE SEGMENTATION

Image segmentation aims to partition an image into meaningful regions or object instances at the pixel level [1], [2]. The output of a segmentation process is commonly represented as a segmentation mask, where pixels belonging to the object or region of interest are labeled as foreground and otherwise as background. Because of pixel-level accuracy, segmentation has become an important component in many applications, including robotics, video analysis, image editing, augmented reality, and geometry-aware reconstruction systems [3], [4], [5], [6], [7]. In this context, segmentation performance refers to the quality of the resulting mask, specifically, whether the final mask accurately captures the intended foreground region, object extent, and boundary structure.

Early image segmentation methods are designed to identify handcrafted image cues, such as edge information, color similarity, texture consistency, region continuity, and spatial coherence [2], [8], [9]. Representative techniques included thresholding, watershed segmentation, graph-based segmentation, region growing, and active contour models [10], [11], [12]. These methods were especially effective when foreground-background contrast, boundary evidence, or region homogeneity was sufficiently well defined. However, they were not designed to fully address highly variable visual conditions, in which illumination changes, background clutter, ambiguous boundaries, and appearance variation can degrade the quality of the resulting segmentation masks [2].

Deep learning has substantially improved segmentation performance by enabling models based on learned visual representations. Architectures such as Fully Convolutional Networks, U-Net, Mask R-CNN, and transformer-based segmentation models have established strong foundations for semantic and instance segmentation through dense prediction, encoder-decoder feature learning, instance-aware prediction, and long-range contextual modeling [13], [14], [15], [16]. Building on this progress, foundation and promptable segmentation models such as the Segment Anything Model further increase segmentation flexibility by generating masks from prompts, including points, boxes, and external localization cues [17]. These models, with default support for diverse visual domains, have increased the accessibility of segmentation in open-world settings.

However, open-world deployment also introduces substantial variation in prompts and visual conditions. Under conditions such as occlusion, clutter, ambiguous boundaries, or visually similar foreground and background regions, promptable segmentation models may produce multiple plausible candidate masks for the same intended target [1], [17]. A confidence or quality estimate is often associated with each candidate mask to convey the approximated reliability. Since downstream pipelines typically require a final mask, this multimask output introduces a candidate-selection problem: the system must decide which mask is best.

2.2 SEGMENTATION QUALITY EVALUATION AND CANDIDATE SELECTION

Evaluating mask quality has become an important problem in practical segmentation pipelines [18]. Existing approaches generally estimate mask quality through learned quality prediction, uncertainty estimation, confidence calibration, or proxy-based evaluation methods [18], [19], [20], [21].

One major direction is to incorporate mask quality estimation directly into the segmentation model. Mask Scoring R-CNN introduced an additional prediction branch to estimate mask quality and recalibrate segmentation confidence scores using predicted MaskIoU values [18]. Related work has further explored confidence calibration [20] and uncertainty estimation [19]. These approaches aim to improve ranking reliability by learning stronger correlations between predicted confidence and segmentation quality. However, because of the requirements for additional supervision, calibration data, architectural modifications, or model-specific training procedures,

these methods are less suitable when the upstream segmentation model is inaccessible, and a separate process is required to generate candidates.

Another direction attempts to estimate segmentation quality without requiring target-image ground truth during inference. Reverse Classification Accuracy (RCA) indirectly predicts segmentation performance by comparing predicted results with reference datasets via proxy evaluation procedures [21]. More broadly, out-of-distribution detection and reliability estimation methods investigate how prediction confidence behaves under domain shift and uncertain visual conditions [19], [20]. RCA approaches require reference data with available ground truth and introduce an additional proxy evaluation step.

Although existing quality-estimation approaches can provide useful ranking information, most require additional supervision, retraining, architectural modification, or reference datasets [25], [26]. In practical deployment scenarios, segmentation systems often serve as pretrained or externally supplied candidate generators in larger pipelines, where additional training or model modifications may not be feasible. Additionally, learned confidence estimates may become less reliable under ambiguous conditions or domain shift [19], [20]. These limitations motivate exploring lightweight, model-independent alternatives for inference-time candidate evaluation and selection.

2.3 OBSERVATIONS AND OPPORTUNITIES

One can interpret modern image segmentation as two distinct steps. First, neural-network-based and promptable segmentation models generate semantically meaningful candidate masks. Second, a completely independent step selects the most appropriate mask from the candidate set for the intended target. In this second step, each mask contains explicit pixel-level properties, including foreground extent, spatial location, boundary structure, connected components, and possible interior holes. It is observed that candidate-mask selection can be examined as an image evaluation problem: determining which mask in the input set exhibits the most plausible representation of the target object.

The candidate masks resulting from the semantically based evaluation are binary images consisting solely of black and white pixel regions with distinct and measurable geometric and morphological properties. In the subsequent candidate-mask selection step, the evaluation requirements align well with those of classical segmentation and binary image analysis. Classical

methods have demonstrated the effectiveness of extracting explicit visual and geometric cues, such as region continuity, compactness, boundary smoothness, spatial coherence, and shape consistency, for computing segmentations [2], [8], [9], [10], [11], [12]. It is important to note that in this setting, classical-style measurements do not replace modern neural segmentation models. Instead, they are used to evaluate the outputs of these models and select the best candidate.

Many image feature measurements identified by classical methods are especially relevant to candidate-mask selection. Centroid of foregrounds is an effective way to describe the spatial location of a binary region [27]. Distance-transform analysis can assign each image pixel a distance to the nearest image boundary, thereby summarizing the spatial relationship between candidate foreground pixels and the image boundary [28]. Shape and morphology analysis further integrate convex-hull solidity, connected-component structure, and holes to describe binary objects [27]. The demonstrated effectiveness of extracting these geometric, spatial, and structural measurements complements the semantic candidate-generation capability of modern neural segmentation models.

These observations identify an opportunity for a new lightweight candidate-mask selection approach. Modern segmentation systems can produce a set of plausible masks as usual, while separate, explicit pixel-level measurements can provide interpretable criteria for comparing the masks. Building on this observation, the next chapter develops a scoring framework to rank candidate masks without modifying the upstream segmentation model.

Chapter 3. METHOD

3.1 PROBLEM FORMULATION AND INFERENCE-TIME CANDIDATE SELECTION PIPELINE

Modern promptable segmentation systems, particularly foundation segmentation models such as the Segment Anything Model (SAM) and Segment Anything Model 2 (SAM 2), can generate multiple candidate masks for a single prompted target under ambiguous visual conditions [17], [23]. Although multi-mask prediction preserves alternative segmentation hypotheses, many downstream applications require one final mask rather than a set of competing candidates [6]. This thesis defines the candidate-mask selection problem as follows: given multiple masks generated for the same target, select the candidate that best represents the intended target.

The problem begins after candidate generation. It does not include detecting or semantically identifying the target, generating new segmentation masks, or modifying the upstream segmentation model. Instead, candidate-mask selection focuses only on evaluating the available multi-mask outputs and determining which existing candidate to pass to the downstream application.

For the i -th image or video frame, let

$$M_i = \{M_{i,1}, M_{i,2}, \dots, M_{i,N_i}\} \quad (1)$$

denote the set of N_i candidate masks produced by an upstream segmentation system. Each candidate

$$M_{i,j} \in \{0,1\}^{H_i \times W_i}, \quad j = 1, \dots, N_i$$

is a binary mask aligned with an image of height H_i and width W_i . A foreground value of 1 represents the possible target region proposed by the upstream system, while a background value of 0 represents the remaining image region. The term *foreground*, therefore, refers only to the positive region of a candidate mask and does not imply that the region has been verified as ground truth.

Modern multi-mask segmentation systems have demonstrated that a scalar score can be used to rank alternative masks. Intersection over Union (IoU) measures the overlap between two masks

as the ratio of the area of the intersection to the area of the union, with higher values indicating greater overlap. SAM predicts an estimate of this IoU for each generated mask and uses the predicted values to order the candidates [17]. SAM 2 follows the same general mechanism and selects the candidate with the highest predicted IoU when ambiguity remains unresolved [23]. These scores are learned estimates produced from the internal image, prompt, and decoder representations of the segmentation model. Their use nevertheless establishes a useful principle: the candidate-mask selection problem can be addressed by assigning a comparable score to each candidate and selecting the highest-ranked result.

A second important characteristic concerns the relationship among the masks in M_i . Because they are generated for the same prompted target, the candidates commonly share coarse semantic alignment rather than representing unrelated segmentation tasks. However, they may differ substantially in foreground extent, relative location, leakage into nearby distractors, fragmentation, internal holes, thin structures, and silhouette coherence. These differences are encoded directly in the binary masks and can be measured through classical image-analysis operations.

Combining these two observations motivates the proposed solution. Existing multi-mask systems show that scoring can rank alternative masks, while the candidate masks themselves exhibit measurable geometric and morphological differences that can inform additional ranking efforts. This thesis, therefore, proposes ACES, a training-free scoring system that ranks candidates based on their compatibility with a t-vector.

For image i , the target reference, hereafter referred to as the t-vector, is denoted by

$$t_i = [t_{A,i}, t_{C,i}, t_{E,i}, t_{S,i}] \in [0,1]^4 \quad (2)$$

where the four values provide references for the target’s Area, Center-location, Edge-proximity, and Silhouette properties. The t-vector may be specified using knowledge of the intended target or the application context, or a default vector may be used when target-specific information is unavailable. The four reference values and their corresponding mask measurements are detailed in Section 3.2.

The subscript i indicates that the t-vector can, in general, differ between images or video frames. When the target properties remain sufficiently stable, the same vector may instead be applied to an entire image set or sequence, such that $t_i = t$. Using one shared vector is therefore a practical configuration choice rather than a requirement of the formulation.

For each candidate $M_{i,j}$, ACES derives a scalar ACES score

$$S_{i,j} = S(M_{i,j}; t_i)$$

which represents how closely the candidate properties correspond to the supplied target reference. The selected output for image i is

$$M_i^* = \arg \max_{M_{i,j} \in \mathcal{M}_i} S(M_{i,j}; t_i) \quad (3)$$

Thus, ACES solves the candidate-mask selection problem by deriving a score for each candidate, ranking the candidates according to those scores, and returning the highest-scoring mask. The individual compatibility functions are presented in Section 3.2, and their aggregation into the final ACES score is defined in Section 3.3.

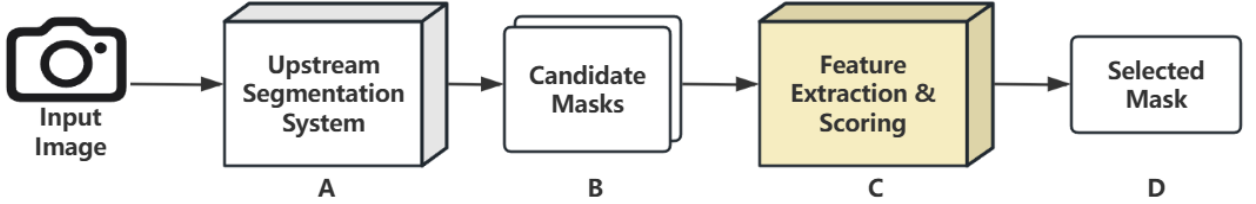


Figure 3.1. Overview of the proposed inference-time candidate-mask selection framework. An upstream segmentation system generates multiple candidate masks, which ACES evaluates and ranks to select a final mask.

Figure 3.1 presents the complete inference-time pipeline, consisting of an upstream segmentation system, a set of candidate masks, the proposed candidate-ranking module, and the selected output mask. These four stages correspond to Boxes A through D, respectively. The primary contribution of this thesis is Box C, which evaluates and ranks the masks produced by the preceding stages.

Box A represents the upstream segmentation system that generates the candidate masks. In this work, candidate generation is performed through text-guided localization followed by promptable segmentation. A localization model first identifies coarse regions associated with the target described by a text prompt, and then supplies these regions to a segmentation model to generate one or more masks [17]. This stage provides the semantic localization and mask generation capabilities required by the pipeline, but ACES does not modify it.

Box B contains the candidate-mask set $\mathcal{M}_i$ defined in Equation (1). Each mask is spatially aligned with the input image and represents one possible segmentation of the intended target. The candidates may differ in extent, position, connectedness, boundary interaction, and shape structure, providing the measurable variation used by the proposed ranking module.

Box C implements ACES. For each candidate, the module measures Area, Center-location, Edge-proximity, and Silhouette properties and compares them with the corresponding values in t_i . These comparisons produce four normalized compatibility terms, which are combined into the scalar score $S_{i,j}$. Candidates are then ranked according to this score. The detailed measurements and comparison functions are presented in Section 3.2, while the final score aggregation is defined in Section 3.3.

Box D is the selected output. After all candidates have been evaluated, the mask with the highest ACES score is returned according to Equation (3). The ranking module, therefore, converts multiple upstream segmentation hypotheses into one final output without generating a new mask or altering the existing candidates.

Although the candidate masks are spatially aligned with the source image, ACES does not analyze RGB appearance, texture, color, or internal neural-network representations. The image dimensions provide the coordinate system required to normalize mask properties, while the evaluation itself operates on the binary foreground regions. Consequently, ACES requires no additional training, architectural modifications, or access to the upstream model's internal features. The t-vector may also be updated to recompute the ranking of an existing candidate set without regenerating the masks, enabling the method to serve as a lightweight post-processing stage for different segmentation systems.

3.2 CANDIDATE PROPERTIES AND COMPATIBILITY TERMS

The proposed solution, ACES, evaluates candidate masks based on their geometric and spatial plausibility. Binary-mask analysis is a fundamental component of classical image processing. Existing work has used region area to characterize foreground extent [2], centroid-based descriptors to represent spatial location [27], distance transforms to quantify spatial relationships between foreground pixels and image boundaries [28], and connected components, convex hulls, and enclosed holes to describe region shape and topology [27]. Building on these established

measurements, ACES evaluates candidate masks along four corresponding ACES properties: Area, Center-location, Edge-proximity, and Silhouette.

- **Area compatibility (A)**, compares the foreground extent of a candidate mask with an expected target area using region-size information commonly considered in region-based segmentation and binary image analysis [2], [8], [9];
- **Center-location compatibility (C)**, compares the observed centeredness of the mask foreground with an expected centeredness using centroid-based region representation, a standard measurement in binary image analysis [27];
- **Edge-proximity compatibility (E)**, compares the mask foreground's proximity to the image boundary with an expected edge proximity using distance-transform analysis [28];
- **Silhouette compatibility (S)**, compares observed global shape coherence with an expected silhouette value using connected components, convex-hull solidity, and internal holes in binary shape and morphology analysis [27].

Each compatibility term lies in the range $[0, 1]$, where a higher value indicates closer agreement with the corresponding reference value. Because the four terms share the same scale, they are combined directly without adjustable feature weights, as described in Section 3.3.

3.2.1 *Area Compatibility*

Area is a basic property in classical image analysis and is commonly used to describe the spatial extent of foreground regions. Region-based segmentation and binary image analysis have long used size, continuity, and coherence to form, describe, or filter segmented regions [2], [8], [9]. Similarly, morphological processing commonly uses connected-component area to remove small, noisy components or suppress implausible regions [27]. These established uses of regional size motivate the proposed area observation and compatibility term: each candidate mask is first represented by its image-normalized foreground area, and this measurement is then compared with a user-specified target area via a soft compatibility function rather than a hard threshold. In this thesis, an observation denotes the normalized numerical value computed from a mask property. This design allows the framework to prefer masks whose foreground extent is close to the expected target scale while gradually penalizing larger deviations.

Let

$$r_i = \frac{|M_i|}{|I|} \quad (4)$$

denote the normalized area ratio of candidate mask M_i , where $|M_i|$ is the number of foreground pixels in the mask and $|I|$ is the total image area.

The area compatibility term score is defined as

$$A(M_i; t_A) = \max\left(0, 1 - \left(\frac{r_i - t_A}{t_A}\right)^2\right) \quad (5)$$

where t_A is the area reference value, with a default value used when no application-specific target scale is provided.

This formulation produces a parabola-shaped preference centered around t_A . Candidate masks whose area is close to the target scale receive a higher area compatibility value, while masks with r_i that are excessively smaller or larger than t_A are progressively penalized. Unlike hard thresholding, this soft formulation preserves flexibility across different target object categories and scene layouts while still suppressing candidates with implausible region sizes.

3.2.2 Center-Location Compatibility

The center-location property describes the relative position of a candidate mask using its foreground centroid. In binary image analysis, the centroid is a standard region descriptor that summarizes the spatial location of a connected component or mask region [27]. Prior work on visual attention and saliency has also reported center-bias effects, in which important objects or human fixations often appear closer to the image center across many natural image datasets [29], [30]. These references motivate using the image center as a stable spatial reference, but not as a universal preference. The proposed framework first computes the normalized distance between the mask centroid and the image center and converts it into a normalized centeredness value.

Let

$$(x_i, y_i)$$

denote the centroid of candidate mask M_i , and let

$$(x_c, y_c)$$

denote the image center. The normalized centroid distance is computed as

$$D_i = \frac{\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}}{0.5\sqrt{W^2 + H^2}}$$

where W and H denote image width and height.

The observed centeredness is then defined as

$$c_i = 1 - \min(D_i, 1) \quad (6)$$

In this way, c_i is a normalized centeredness measurement in the range $[0,1]$, where values approaching 0 indicate that the mask centroid is progressively farther from the image center, and a value of 1 indicates that the centroid coincides with the image center.

Let $t_c \in [0, 1]$ denote the center-location reference value, where 1 represents a target at the image center and lower values represent progressively greater offsets. The center-location compatibility term is defined as

$$C(M_i; t_c) = 1 - |t_c - c_i| \quad (7)$$

Candidate masks therefore receive higher center-location compatibility values when their observed centeredness is closer to the reference value, whether that target is centered or otherwise. The image center serves only as the coordinate reference for this comparison. This preserves the practical benefit of a center prior when appropriate, while allowing lower reference values for objects expected to appear near image boundaries.

3.2.3 *Edge-Proximity Compatibility*

The edge-proximity property describes the relationship between a candidate mask and the image boundary. The implementation constructs an image-border distance map using distance-transform analysis [28]. In this map, each image pixel is assigned its Euclidean distance to the nearest image border. The resulting observation describes where the candidate foreground lies within the image rather than the distance from foreground pixels to the candidate mask's own contour.

Let $d(p)$ denote the Euclidean distance from image pixel p to the nearest image border. Larger values indicate pixels farther inside the image, while smaller values indicate pixels close to an image edge. For candidate mask M_i , the framework summarizes the distance values at its foreground pixels using a fixed quantile statistic.

The normalized image-edge distance is defined as

$$d_i^{\text{edge}} = \frac{Q_q(\{d(p): p \in M_i\})}{d_{\max}} \quad (8)$$

where:

- $Q_q(\cdot)$ denotes the q -th quantile of image-border distance values at foreground pixels of the candidate mask;
- $d_{\max}$ is the maximum achievable distance-transform value used for normalization.

The value d_i^{edge} is clipped to $[0,1]$, where larger values indicate that the candidate lies farther from the image boundary. The edge-proximity observation is then defined as

$$e_i = 1 - d_i^{\text{edge}} \quad (9)$$

Let $t_E \in [0, 1]$ denote the edge-proximity reference value, where 1 indicates a target near an image edge, and 0 indicates a target far from the image boundary. The edge-proximity compatibility term is defined as

$$E(M_i; t_E) = 1 - |t_E - e_i| \quad (10)$$

The use of a quantile statistic is motivated by robust-statistics practice, in which quantiles reduce sensitivity to extreme observations [31]. In this setting, a single foreground pixel, whether near or far from the image boundary, would not determine the entire candidate measurement. The quantile q is therefore treated as a fixed internal measurement parameter, while t_E is the user-specified edge-proximity reference value. The specific quantile-based formulation is a design choice of this thesis.

Although center-location and edge-proximity are related, they describe different spatial properties: t_C refers to the expected position of the mask centroid, whereas t_E refers to how the foreground region is distributed relative to the image boundary. A compact object near the image center may have high t_C and low t_E , while an off-center object adjacent to an image boundary may have low t_C and high t_E . Both values may also be high for an extended or ring-shaped region whose centroid remains near the image center while part of the foreground approaches the image boundary. Therefore, the two terms are retained separately because centroid location alone does not determine the foreground region's relationship with the image boundary.

The edge-proximity term is particularly useful when candidates correspond to similar semantic regions but differ in leakage toward image boundaries or in whether the foreground lies at the expected distance from the image edge.

3.2.4 Silhouette Compatibility

The silhouette property describes the global shape and topological coherence of a candidate mask. This feature is inspired by standard binary shape analysis and mathematical morphology, where connected components, convex hulls, and holes are commonly used to describe the structure of binary regions [27]. Candidate masks generated under ambiguous conditions may exhibit connectivity, hole structure, or boundary irregularities that differ from those expected for the target, even when they remain semantically related to it. The proposed framework, therefore, first computes a silhouette-coherence observation s_i from three complementary geometric measurements.

First, solidity measures how completely a foreground region fills its convex hull. This is a standard contour property, commonly defined as the ratio of the contour area to the convex hull area [32]. Let Γ_i^* denote the external contour enclosing the largest connected foreground component of the candidate mask M_i . The mask-contour area and its convex-hull area are defined as

$$\begin{aligned} A_{\text{mask},i} &= \text{Area}(\Gamma_i^*) \\ A_{\text{hull},i} &= \text{Area}(\text{ConvHull}(\Gamma_i^*)) \end{aligned}$$

The solidity observation is then defined as

$$\text{Solidity}_i = \begin{cases} \frac{A_{\text{mask},i}}{A_{\text{hull},i}}, & A_{\text{hull},i} > 0 \\ 0, & A_{\text{hull},i} = 0 \end{cases}$$

Higher solidity values indicate that the largest foreground component more completely fills its convex hull.

Second, fragmentation evaluates whether the foreground region is dominated by one main connected component or distributed across multiple disconnected components. Connected-component analysis is a standard operation in digital picture processing and binary image analysis [27], [33]. Let

$$A_{\text{max},i}$$

denote the largest connected-component area among all components, and let

$$A_{\text{total},i}$$

denote the total area of those components. Fragmentation is defined as

$$Fragmentation_i = 1 - \frac{A_{max,i}}{A_{total,i}}$$

Masks dominated by a single substantial component receive lower fragmentation observations, whereas masks distributed across multiple substantial components receive higher fragmentation observations.

Third, the internal hole structure is evaluated using a flood-fill-based hole estimation procedure. Flood filling and morphological hole filling are standard binary image operations for identifying enclosed background regions inside foreground objects [34]. Let

$$A_{hole,i}$$

denote the total area of enclosed holes inside the mask region. The hole ratio is computed as

$$R_{hole,i} = \frac{A_{hole,i}}{A_{mask,i}}$$

The observed silhouette-coherence value is then defined as

$$s_i = Solidity_i \cdot (1 - Fragmentation_i) \cdot (1 - R_{hole,i}) \quad (11)$$

In this way, the silhouette-term value s_i is high for masks that are compact, minimally fragmented, and contain few internal holes. It is not treated as a universally preferable quality value. Let $t_S \in [0, 1]$ denote the silhouette reference value. The silhouette compatibility term is defined as

$$S(M_i; t_S) = 1 - |t_S - s_i| \quad (12)$$

Because some real objects naturally contain hollow structures, thin components, or complex topology, a corresponding lower t_S can be used in these cases. However, the silhouette observation remains a coarse summary and cannot fully represent category-specific topology. The limitations of this handcrafted representation are further discussed in Chapter 6.

3.3 UNIFIED REFERENCE-MATCHING SCORE AND CANDIDATE SELECTION

The four terms described in Section 3.2 each measure compatibility for a different mask property.

- Area compatibility compares foreground extent,
- Center-location compatibility compares centroid position,
- Edge-proximity compatibility compares the mask's relation to the image boundary, and
- Silhouette compatibility compares global shape and topological coherence.

Together, these four terms are referred to as the ACES compatibility terms in this thesis.

Since the goal is to select one mask from several alternatives, candidate-mask selection can be viewed as a multi-criteria ranking problem. Additive aggregation is commonly used to combine normalized criteria into a single score for each alternative [35]. Because all four ACES compatibility terms have the same $[0, 1]$ range, the proposed framework uses an unweighted additive score based on t-vector.

For a candidate mask M_i and t-vector $\mathbf{t}$, the final ACES score is defined as

$$\text{Score}(M_i; \mathbf{t}) = A(M_i; t_A) + C(M_i; t_C) + E(M_i; t_E) + S(M_i; t_S) \quad (13)$$

where:

- $A(M_i; t_A)$ denotes the area compatibility term;
- $C(M_i; t_C)$ denotes the center-location compatibility term;
- $E(M_i; t_E)$ denotes the edge-proximity compatibility score;
- $S(M_i; t_S)$ denotes the silhouette compatibility score.

When application-specific values are unavailable, the framework can use a fixed default t-vector. The quantile q in the edge-distance formulation of Equation (8) remains a fixed internal measurement parameter as 0.25 and is not part of the user input.

Because the four terms are normalized to the same range and are added with equal contribution, the total score lies in $[0, 4]$. Differences in ranking arise from how closely each candidate matches the approximated geometry specified in a given t-vector.

The ACES score is computed for each candidate, and the candidates are ranked accordingly. As indicated in Equation (3), the mask with the highest ACES score is selected as the final candidate.

The reference-based formulation provides flexibility across deployment settings. Different applications may expect different target areas, locations, edge relations, or silhouette characteristics. The proposed framework allows these target values to be adjusted while preserving the same measurement and ranking mechanism. Parameter selection and empirical analysis of the reference values are discussed further in Chapter 5.

3.4 LIMITATIONS OF THE PROPOSED FRAMEWORK

The proposed framework is subject to several important limitations:

Dependence on relatively stable and accurately specified target properties. The ACES scoring function is based on the area, relative location, edge proximity, and silhouette characteristics of the target mask. When extracting masks from multiple images in a video, the supplied target values must remain reasonably representative across all frames. When an object moves substantially, changes scale, undergoes strong deformation, or changes its relation to the image boundary, a fixed t-vector may become inaccurate. Under these conditions, the framework may favor incorrect candidates.

Reliance on handcrafted descriptors and user-specified reference values. Although the t-vector can be adjusted for off-center, thin, hollow, or structurally complex objects, each ACES property is represented by a coarse scalar observation. Distinct masks may have similar area, centeredness, edge proximity, or silhouette statistics even when only one is semantically correct. In addition, an inaccurate user reference value can systematically favor the incorrect candidate. The framework, therefore, cannot replace semantic reasoning.

These limitations arise directly from the design choice to rely on lightweight geometric measurements rather than on learned semantic reasoning. The proposed method is intended as a simple, user-guided inference-time candidate-ranking framework rather than a universal mask-quality estimator.

Chapter 4. IMPLEMENTATION

4.1 IMPLEMENTATION REQUIREMENTS AND TOOL SELECTION

The implementation of the framework introduced in Chapter 3 must support reliable mask-level computations, including foreground-area calculation, centroid estimation, image-border distance analysis, contour extraction, convex-hull computation, connected-component analysis, and enclosed-hole detection. Additionally, for a given mask, recording both the ACES score and the corresponding term values is necessary for debugging, sequence-level analysis, and interpretation of why a particular mask is selected.

Independently, during development, proper support for interactive visual inspection is essential. Since the framework is based on matching geometric properties of masks to the values in the target t-vector, the implementation must allow a user to inspect candidate masks while manipulating the four individual values in the t-vector. Furthermore, when ground truth is available for analysis, the tool must record candidate IoU values and compare them with those resulting from the ACES-selected mask.

Based on these requirements, Python was selected as the primary implementation language for its support for numerical computation, image processing, experimental scripting, and rapid interface development. The following APIs are also chosen.

- NumPy [36] is used for binary-array operations, coordinate statistics, quantile calculation, and score aggregation.
- OpenCV [37] provides the distance transform, contour, convex-hull, connected-component, and flood-fill operations.
- COCO [38] run-length encoding is used to store and recover masks efficiently.
- Pandas [39] is used to export image- and sequence-level diagnostics.
- Streamlit [40] dashboard supports interactive inspection.

Together, these tools implement the ACES ranking framework and provide the analytical support required for the experiments in Chapters 5 and 6.

4.2 PROPERTY MEASUREMENT AND COMPATIBILITY COMPUTATIONS

This section describes how the ACES terms are computed in practice. Each candidate is converted to a binary array aligned with the input image. The implementation first measures the candidate's geometric properties and then compares those observations with the target reference value defined in Chapter 3. All four compatibility terms are clipped to the range $[0, 1]$. Empty candidates are treated as invalid and should receive a compatibility score of 0 before ranking.

4.2.1 *Area and Center Compatibility*

As shown in Listing 4.1, the area and center terms are computed from the same foreground mask. Lines 1-4 convert the candidate to a Boolean mask and return zero-valued outputs when the foreground is empty. Lines 5-6 compute the normalized area observation and its compatibility with the area reference value. Lines 7-15 compute the foreground centroid, normalize its displacement by half of the image diagonal, and compare the resulting centeredness observation with the center-location reference value. Line 16 returns both observations and compatibility terms for later diagnostic analysis.

Listing 4.1. Area and Center Compatibility

Input: candidate mask B; image height H; image width W; reference values t_area and t_center

Output: $area_ratio$; $area_match$; $centeredness$; $center_match$

```

1  B <- BooleanMask(B)
2  foreground_count <- CountForeground(B)
3  If foreground_count = 0:
4      Return 0 for all: area_ratio, area_match, centeredness, center_match
5  area_ratio <- foreground_count / (W * H)    // image-normalized foreground
   area
6  area_match <- clip(1 - ((area_ratio - t_area) / t_area)^2, 0, 1)
7  centroid_x <- mean(x-coordinates of foreground pixels in B)
8  centroid_y <- mean(y-coordinates of foreground pixels in B)
9  x_offset <- centroid_x - (W / 2)    // image center in x
10 y_offset <- centroid_y - (H / 2)    // image center in y
11 center_distance <- hypot(x_offset, y_offset)
12 max_center_distance <- 0.5 * hypot(W, H)    // half of the image diagonal
13 normalized_distance <- center_distance / max_center_distance
14 centeredness <- 1 - clip(normalized_distance, 0, 1)
15 center_match <- clip(1 - abs(t_center - centeredness), 0, 1)
16 Return area_ratio, area_match, centeredness, center_match

```

4.2.2 *Edge Proximity Compatibility*

Listing 4.2 implements the image-edge term using the border-distance map shared by all candidates from the same image. Lines 1-2 return zero-valued outputs when the foreground is empty. Lines 3-4 obtain the q -th quantile of the precomputed border-distance values under the candidate foreground and normalize the result by the maximum map value. Lines 5-6 invert the normalized distance to obtain the edge-proximity observation and compare it with the edge-proximity reference value. Line 7 returns the normalized distance, the edge-proximity observation, and the compatibility term. The experiments use $q = 0.25$, which preserves sensitivity to foreground regions extending toward an image boundary without allowing a single extreme pixel to determine the measurement.

Listing 4.2. Image-Edge Proximity Compatibility

Input: candidate mask B ; border-distance map D ; maximum map value d_max ; quantile q ; reference value t_edge

Output: $edge_distance$; $edge_proximity$; $edge_match$

```

1  If CountForeground(B) = 0:
2    Return 0 for all: edge_distance, edge_proximity, edge_match
3  edge_distance <- Quantile(D at foreground pixels in B, q) / d_max    //
   normalized image-border distance
4  edge_distance <- clip(edge_distance, 0, 1)
5  edge_proximity <- 1 - edge_distance
6  edge_match <- clip(1 - abs(t_edge - edge_proximity), 0, 1)
7  Return edge_distance, edge_proximity, edge_match

```

4.2.3 *Contour and Connected-Component Analysis*

As detailed in Listing 4.3, solidity and fragmentation are computed through separate contour and connected-component procedures. Lines 1-3 handle the case in which no external contour exists. Lines 4-10 select the largest contour, compute the area of its convex hull, and derive the solidity observation. Lines 11-12 obtain the areas of 8-connected components and discard components smaller than 1 percent of the total foreground area. Lines 13-16 derive the fragmentation observation when more than one substantial component remains. Line 17 returns the contour area, solidity, and fragmentation observations.

Listing 4.3. Solidity and Fragmentation Observations

Input: candidate mask B

Output: *contour_area*; solidity; fragmentation

```

1  contours <- FindExternalContours(B)
2  If contours is empty:
3      Return 0 for all: contour_area, solidity, fragmentation.
4  main_contour <- contour in contours with the largest area    // dominant
   foreground component
5  contour_area <- ContourArea(main_contour)
6  hull_area <- ContourArea(ConvexHull(main_contour))
7  If hull_area <= 0:
8      solidity <- 0
9  Else:
10     solidity <- clip(contour_area / hull_area, 0, 1)
11     component_areas <- PixelAreas(FindConnectedComponents(B, connectivity = 8))
12     valid_areas <- Filter(component_areas, area >= 0.01 * sum(component_areas))
   // ignore very small components
13     If Count(valid_areas) <= 1:
14         fragmentation <- 0
15     Else:
16         fragmentation <- clip(1 - max(valid_areas) / sum(valid_areas), 0, 1)
17     Return contour_area, solidity, fragmentation.
```

OpenCV's `findContours`, `contourArea`, and `convexHull` functions implement lines 1 and 4-6 of Listing 4.3, while `connectedComponentsWithStats` supplies the component areas used in lines 11-12 [37]. Keeping the two procedures separate preserves their distinct roles: solidity describes how completely the dominant foreground component fills its convex hull, whereas fragmentation measures substantial foreground outside that component.

4.2.4 Hole Detection and Silhouette Compatibility

Listing 4.4 shows the flood-fill procedure used to complete the silhouette term. Lines 1-2 return zero-valued outputs when no valid contour is available. Lines 3-5 invert and pad the mask, remove the exterior-connected background by flood-filling from the padded corner, and retain only enclosed background pixels. Line 6 converts these pixels into the hole-ratio observation. Lines 7-8 combine solidity, fragmentation, and hole ratio into the silhouette observation and compare it with the silhouette reference value. Line 9 returns the hole ratio, silhouette observation, and compatibility term.

Listing 4.4. Hole Ratio and Silhouette Compatibility

Input: candidate mask B; contour_area; solidity; fragmentation; reference value t_silhouette

Output: hole_ratio; silhouette; silhouette_match

```

1  If contour_area <= 0:
2      Return 0 for all: hole_ratio, silhouette, silhouette_match
3  padded_background <- Pad(logical inverse of B, width = 1, value = 1)
4  FloodFillFromCorner(padded_background)    // remove exterior-connected
      background
5  holes <- background pixels remaining after cropping to the original image
      // enclosed holes
6  hole_ratio <- clip(CountTrue(holes) / contour_area, 0, 1)
7  silhouette <- clip(solidity * (1 - fragmentation) * (1 - hole_ratio), 0, 1)
8  silhouette_match <- clip(1 - abs(t_silhouette - silhouette), 0, 1)
9  Return hole_ratio, silhouette, silhouette_match

```

The intermediate values returned on line 9 of Listing 4.4 are retained in each diagnostic record. They show whether a low silhouette compatibility value results from low solidity, substantial fragmentation, a high hole ratio, or a mismatch between the silhouette observation and the silhouette reference.

4.2.5 Unified Score and Candidate Selection

Listing 4.5 integrates the four compatibility computations into the final ranking procedure. Lines 1-5 precompute the image-border distance map and initialize the selection state. Lines 6-13 evaluate each candidate by calling the procedures in Listings 4.1-4.4, summing the four compatibility terms, and storing one diagnostic record. Lines 14-16 update the current winner only when the new score is greater than the current best score. Line 17 returns the selected mask together with all per-candidate records.

Listing 4.5. ACES Candidate Ranking

Input: candidate masks {M₁, ..., M_N}; image size H x W; reference values t_area, t_center, t_edge, and t_silhouette; quantile q

Output: selected_mask; per-candidate records

```

1  D <- ComputeImageBorderDistanceMap(H, W)
2  d_max <- max(D)
3  best_score <- -infinity
4  best_index <- undefined
5  records <- empty list
6  For i <- 1 to N:
7      B <- BooleanMask(M_i)
8      area_result <- AreaCenterCompatibility(B, H, W, t_area, t_center)    //
      Listing 4.1
9      edge_result <- EdgeProximityCompatibility(B, D, d_max, q, t_edge)    //
      Listing 4.2
10     shape_result <- SolidityFragmentation(B)    // Listing 4.3

```

Listing 4.5. ACES Candidate Ranking

Input: candidate masks $\{M_1, \dots, M_N\}$; image size $H \times W$; reference values t_{area} , t_{center} , t_{edge} , and $t_{\text{silhouette}}$; quantile q

Output: selected_mask; per-candidate records

```

11 silhouette_result <- HoleSilhouetteCompatibility(B, shape_result,
    t_silhouette) // Listing 4.4
12 score <- area_result.area_match + area_result.center_match +
    edge_result.edge_match + silhouette_result.silhouette_match
13 Append BuildRecord(i, area_result, edge_result, shape_result,
    silhouette_result, score) to records
14 If score > best_score:
15     best_score <- score
16     best_index <- i
17 Return M_best_index and records

```

4.3 ANALYSIS TOOLS

After the property-measurement and compatibility functions were implemented, an interactive analysis tool was developed to inspect their behavior on actual candidate masks. The tool supports the inspection and validation of individual-mask results before the large-scale evaluation in Chapters 5 and 6.

The primary controls correspond directly to the t-vector: target area t_A , expected centeredness t_C , expected image-edge proximity t_E , and expected silhouette value t_S . A separate advanced control exposes the border-distance quantile $q = 0.25$ for implementation analysis, while making clear that q defines how the edge observation is measured rather than what property the user expects.

For every candidate, the dashboard displays the raw observations, the four compatibility terms, the total ACES score, and the resulting rank. Viewing each candidate alongside these values helps determine whether a ranking change is due to area mismatch, center-location mismatch, edge-proximity mismatch, or silhouette mismatch. When ground truth is available, the dashboard also displays candidate IoU values and compares the ACES-selected mask with the candidate selected by the upstream SAM confidence score. This comparison separates candidate-generation failure from candidate-selection failure.

The tool was also used to verify the direction and scale of the default reference values used in the experiments: $t_A = 0.25$, $t_C = 1.0$, $t_E = 0.0$, $t_S = 1.0$. These values describe a target occupying approximately one-quarter of the image, located near the image center, away from the image boundary, and with a highly coherent silhouette. The dashboard was used to confirm how

departures from these reference values changed the selected mask; the quantitative evaluation of the default and sequence-specific references is reported in Chapter 6.

Figure 4.1 shows the Streamlit interface developed with assistance from an AI coding agent. The agent was used to accelerate dashboard layout, interactive controls, and result-display components, while the property definitions, compatibility equations, and ranking logic remained those specified in Chapters 3 and 4. Representative prompts for the interface included:

“Build a Streamlit interface that loads an image and multiple binary candidate masks, displays each mask beside its observed area, centeredness, edge proximity, and silhouette values, and provides sliders for the four corresponding target reference values.”

“When a reference-value slider changes, recompute the four compatibility terms and their unweighted sum, update the ranking table, and display the selected mask beside the SAM-confidence baseline.”

The generated interface scaffolding was then inspected and modified to match the actual mask-storage format, function signatures, diagnostic fields, and experimental output files. This workflow reduced interface-development time without changing the implemented ACES computations.

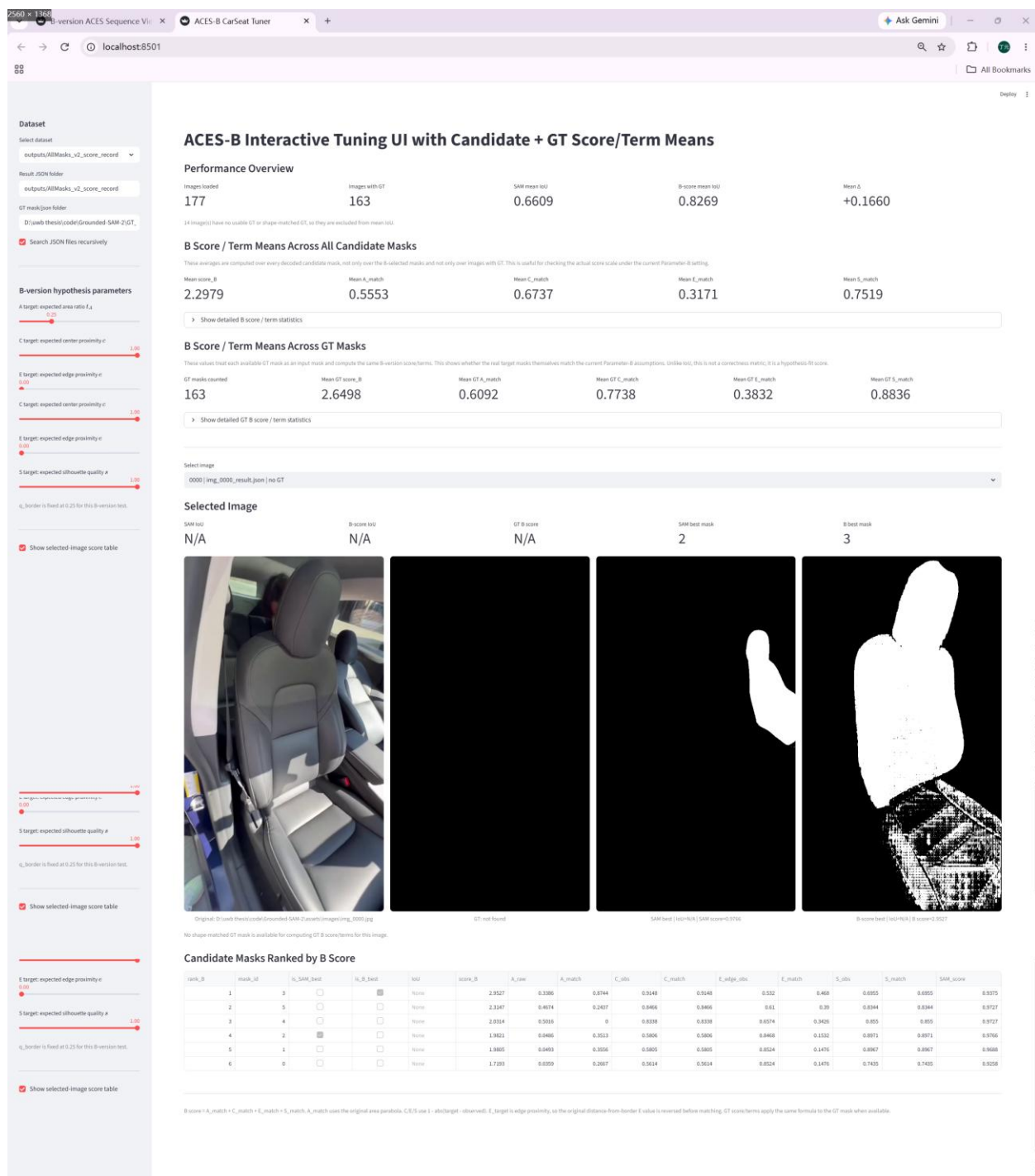


Figure 4.1. Analysis dashboard for inspecting candidate-mask property values, target reference values, compatibility terms, and the final unweighted ACES ranking.

Chapter 5. SYSTEMATIC REFERENCE VALUE SPECIFICATION AND CASE STUDIES

ACES ranks candidate masks based on geometric and morphological property scores computed using the user-specified values in the t-vector. Its effectiveness, therefore, depends on whether the values in the t-vector reasonably represent the intended target. Although the four t-vector values are scalar parameters, they can be estimated visually from the input image or specified based on application knowledge.

This chapter presents a systematic procedure for deriving the four values of an ACES t-vector based on the properties of a target. A generic input image of a bear is first used as a case study to illustrate how these values are estimated. Two additional examples then provide further clarification of the edge-proximity and silhouette terms, which can be more difficult to interpret visually.

5.1 A SYSTEMATIC PROCEDURE FOR DERIVING THE VALUES OF A T-VECTOR

Given an input image with a visually identifiable target, the four values of the ACES t-vector, $[t_A, t_C, t_E, t_S]$, can be approximated according to the following steps.

1. t_A (Area): The approximated fraction of the image occupied by the target.
2. t_C (Center): The target's approximated centroid displacement from the image center, normalized by half of the image diagonal to convert into centeredness.
3. t_E (Edge): The target's expected image-edge proximity, approximated from the normalized 25th-percentile distance between its foreground pixels and the nearest image border.
4. t_S (Silhouette): The approximated connectedness, solidity, fragmentation, and internal-hole structure of the target.

These t-vector element values are expected to be estimates and do not require pixel-level accuracy. Their purpose is to provide an approximate geometric description of the target rather than to reproduce the ground-truth mask. The approximation can be applied only when the original image and general target knowledge are available.

Note that the t-vector is an approximation of the target geometry and does not indicate the relative importance of the four ACES terms. For example, a value of 0 for the t_C term indicates

that the expected target center of mass is located towards the corner of the image and does not diminish the importance of the Center-location compatibility term. In this way, the value of each t-vector element identifies the expected observation at which the corresponding compatibility term reaches maximum. Continuing the previous example $t_c = 0$, in this case, candidate masks with targets located at the corner of an image will receive a high value for the Center-location compatibility term from Equation (7). The four compatibility terms are subsequently combined through the unweighted sum defined in Section 3.3.

5.2 BEAR CASE STUDY

Figure 5.1 presents the original image and the ground-truth mask used for this case study. The intended target is the bear in the lower-left region of the image.



Figure 5.1. Bear case study: (a) original image and (b) ground-truth mask of the intended target.

5.2.1 Area Reference Value (t_A)

The area reference value describes the fraction of the image occupied by the target foreground. A practical visual approximation can be obtained by drawing an axis-aligned bounding box around the target, as illustrated in Figure 5.2.



Figure 5.2. Visual approximation of the bear's image-normalized area using an enclosing bounding box.

When comparing the red bounding box to the entire image, it appears that the bounding box's width and height each occupy approximately 40% of the image's dimensions. The area of the enclosing box is therefore approximately $0.4 * 0.4 = 0.16$. Now, since the bear occupies only part of the enclosing rectangle, its foreground-area ratio can be estimated to be approximately $t_A = 0.10$. Since ACES uses the values in the t-vector to compute rankings of model-generated candidate masks rather than to reconstruct the exact target mask, those values need not be measured precisely. Coarse but informed estimates based on visual inspection or application knowledge are sufficient.

5.2.2 *Center-Location Reference Value (t_C)*

The center-location reference value approximates the target's centeredness. A value of 1 indicates that the target centroid is located at the image center, while lower values indicate progressively greater displacement from the image center. A value of 0 indicates that the target centroid is located at one of the image corners.

As illustrated in Figure 5.3, the target center can be estimated visually and compared with the image center. The distance between these two points is normalized by half of the image diagonal. The centeredness can therefore be approximated as

$$t_C = 1 - \frac{\text{estimated distance from target center to image center}}{\text{half image diagonal}} \quad (14)$$



Figure 5.3. Visual estimation of the center-location reference value using the target center, image center, and half-diagonal reference distance.

In the example shown in Figure 5.3, the bear is located to the left of the image center with a red circle indicating its approximated centroid. The normalized displacement is the ratio of the lengths of the red over the white line, and can be visually estimated to be approximately 0.3. The corresponding centeredness is therefore approximately

$$t_C = 1 - 0.3 = 0.7$$

5.2.3 Edge-Proximity Reference Value (t_E)

The edge-proximity reference value describes the relationship between the target foreground and the boundaries of the entire image. Recall that a distance-transform analysis is performed and each foreground pixel is assigned its Euclidean distance to the nearest image border. The 25th percentile of these distances, d_q , is chosen to be normalized by the maximum value, d_{max} , of the image-border distance map. For this reason, a straightforward approach to approximate the t_E value would be

$$t_E = 1 - \frac{d_q}{d_{max}} \quad (15)$$

Figure 5.4 provides a visual approximation in which the red circle and line represent the distance corresponding to the 25th percentile of the foreground-pixel distances. The white circle and line represent the maximum. Since d_q is about half that of d_{max} , the edge-proximity reference value can be approximated as $t_E = 0.50$, indicating an intermediate expected edge proximity: the target is not expected to touch the image boundary, but part of its foreground is still located relatively close to it.



Figure 5.4. Visual approximation of the bear's edge-proximity reference value.

5.2.4 Silhouette Reference Value (t_s)

The silhouette reference value describes the structural coherence of the target. In the formulation, the silhouette term combines three measurements: solidity relative to the convex hull; fragmentation across connected components; and the proportion of enclosed internal holes.

A value near 1 represents a largely connected, solid, and hole-free foreground region. In contrast, a value near 0 represents a mask with substantial fragmentation, low solidity, and large internal holes.

The bear example represents a largely connected foreground object. Its body has an irregular, natural outline due to its head, legs, and fur. However, it does not contain significant internal holes or disconnected foreground fragments. The target is therefore structurally coherent without being perfectly convex or geometrically regular. For these reasons, a relatively high but less than maximal silhouette reference value would be appropriate. The silhouette reference value chosen is $t_s = 0.80$.

5.2.5 ACES Score

The visual approximations described in Sections 5.2.1–5.2.4 produce the following t-vector for the bear:

$$t_{\text{bear}} = [t_A, t_C, t_E, t_s] = [0.10, 0.70, 0.50, 0.80]$$

To examine whether these visually estimated values provide a reasonable geometric description of the target, the ground-truth bear mask in Figure 5.1(b) is evaluated using the four ACES compatibility functions. Let M_{bear} denote this mask. This evaluation is not intended to perform

candidate selection; it is used only to determine whether the visually approximated reference values are compatible with the known target mask.

Using the area compatibility function defined in Equation (5), M_{bear} produces

$$A(M_{bear}; t_A = 0.10) = 0.9657$$

The Center-location function in Equation (7) gives

$$C(M_{bear}; t_C = 0.70) = 0.9731$$

Similarly, the Edge-proximity function in Equation (10), with the image-border-distance quantile fixed at $q = 0.25$, produces

$$E(M_{bear}; t_E = 0.50) = 0.9812$$

and the Silhouette function in Equation (12) gives

$$S(M_{bear}; t_S = 0.80) = 0.9734$$

Using the unweighted additive formulation in Equation (13), the resulting ACES score is

$$\begin{aligned} ACES(M_{bear}; t_{bear}) &= A(M_{bear}) + C(M_{bear}) + E(M_{bear}) + S(M_{bear}) \\ &= 0.9657 + 0.9731 + 0.9812 + 0.9734 = 3.8933 \end{aligned}$$

The resulting term values and total score show that a t-vector estimated through visual inspection can provide a reasonable compatibility evaluation for the intended target. This supports the feasibility of approximating the target's Area, Center-location, Edge-proximity, and Silhouette values when target-image information is available. In the candidate-selection setting, the same t-vector is supplied as the four parameters of the ACES functions, which are then applied to each candidate generated by the upstream pipeline to compute its ranking score.

5.3 ADDITIONAL EXAMPLES FOR EDGE AND SILHOUETTE REFERENCE VALUES

Area and center location can usually be estimated directly from an object's visible extent and position. Edge proximity and silhouette structure require more careful interpretation because they summarize distributions and topological properties rather than a single visible property. This section provides two additional examples to illustrate how these reference values can be estimated.

5.3.1 *Edge-Proximity Example*

Figure 5.5 shows two synthetic masks with broadly similar foreground areas and centroids but

substantially different relationships with the image boundary.

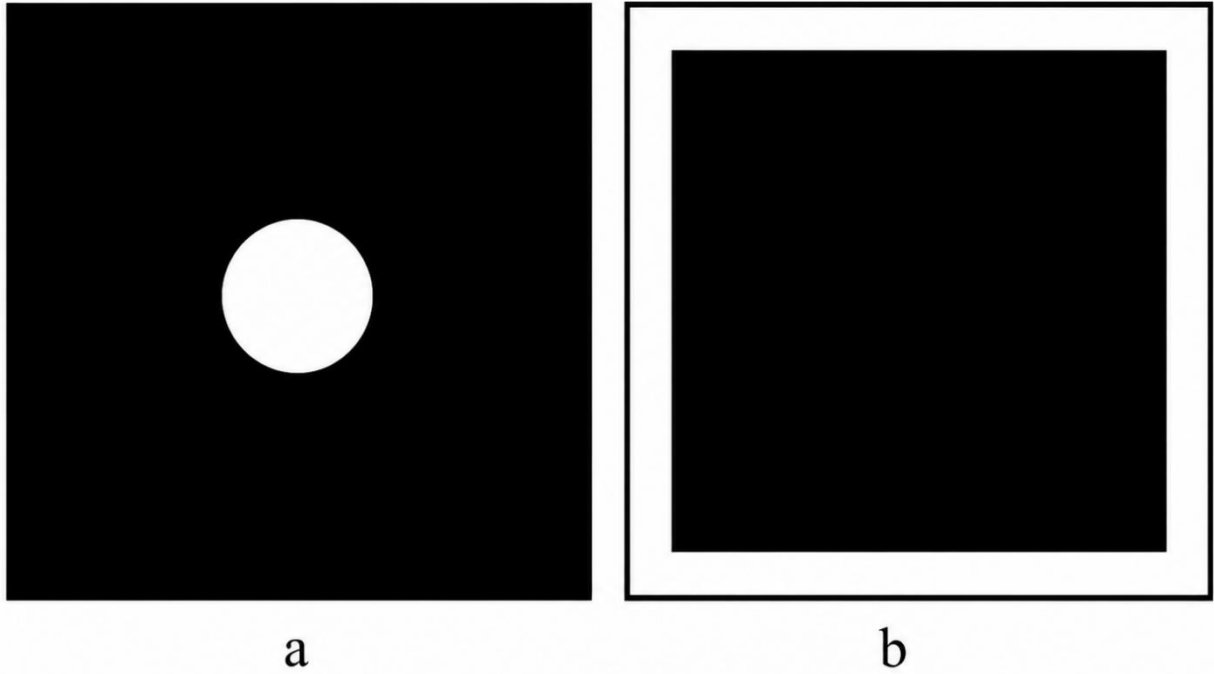


Figure 5.5. Contrasting edge-proximity configurations: (a) an interior foreground region and (b) a boundary-adjacent foreground region.

The foreground region in Figure 5.5(a) is located well inside the image. Its foreground pixels have relatively large distances to the nearest image border, where the 25th-percentile foreground pixel distance, d_q , would be close to that of the max distance, d_{max} . An estimation based on Equation (15) would result in t_E having a value of close to 0.

In contrast, Figure 5.5(b) contains a target distributed along the image boundary. Its 25th-percentile foreground pixel distance is close to zero. An t_E estimation based on Equation (15) would result in a value that is close to 1. A target expected to touch or follow the image boundary should therefore use a value of t_E close to 1.

This comparison illustrates that the edge reference value does not measure the distance from foreground pixels to the mask's own contour. It measures the relationship between the candidate foreground and the boundary of the entire image. Consequently, t_E is useful when candidate masks

have similar area and center location but differ in boundary contact or leakage toward image edges.

5.3.2 Silhouette Example

Figure 5.6 presents a bicycle example in which the desired mask should cover the visible bicycle structure, including the tires, while preserving the empty regions enclosed by the tires. In other words, the interior of each wheel should remain background rather than being incorrectly filled as foreground.



Figure 5.6. Silhouette-reference example: (a) original bicycle image, (b) a correct mask with $S = 0.157$, and (c) a candidate mask with $S = 0.575$ that fills the hollow interiors of the bicycle tires.

The bicycle example illustrates an important case in which a geometrically correct target does not have a solid or hole-free silhouette. Each wheel forms an approximately ring-shaped foreground structure surrounding a large background region. These enclosed background regions are identified as internal holes by the silhouette measurement described in Section 3.2.4.

According to the ACES silhouette formulation Equation (11),

a larger internal-hole ratio reduces the observed silhouette value. Consequently, a candidate that correctly preserves the hollow wheel interiors may have a lower silhouette observation than a candidate that incorrectly fills the complete wheel regions.

For example, consider two candidates with similar areas, locations, and overall bicycle coverage. The first candidate accurately segments the wheels as ring-shaped structures and leaves the enclosed wheel interiors as background. The second candidate fills the wheel interiors, producing two solid foreground disks. Although the second candidate is less accurate, filling the holes increases its silhouette coherence by reducing R_{hole} . A high silhouette reference value could favor the incorrectly filled candidate.

For a bicycle target whose wheel interiors are expected to remain empty, t_s should be reduced to reflect the target's naturally hollow topology. Lowering t_s increases compatibility with candidates whose silhouette observations are reduced by the legitimate holes inside the wheels.

This can allow the correct hollow-wheel candidate to outrank an alternative that fills the wheel interiors as shown in Figure 5.6(c).

This example demonstrates that silhouette coherence should not be interpreted as a property that must always be maximized. A high t_S is suitable for targets expected to form solid, connected, and largely hole-free regions. A lower t_S is appropriate when the intended target naturally contains substantial enclosed holes, as in bicycle wheels.

5.4 SUMMARY

This chapter presented a structured visual procedure for estimating the values of the ACES t-vector. The bear case study demonstrated how approximate observations of the target area, relative location, edge proximity, and silhouette structure can be translated into t-vector values and used to score a mask. The supplementary examples further clarified that the edge reference value describes the target's relationship with the image boundary. In contrast, the silhouette reference value should reflect expected properties such as connectedness, solidity, fragmentation, and internal holes.

When image-specific target information is unavailable, the same reasoning can be applied to define an idealized default target: a medium-sized object located at the image center, away from the image boundary, with a coherent, largely hole-free silhouette. The resulting default t-vector is

$$[t_A, t_C, t_E, t_S] = [0.25, 1.00, 0.00, 1.00]$$

The image-border-distance quantile is fixed separately at $q = 0.25$. This configuration is applied unchanged to all sequences in the default training-free experiments reported in Chapter 6.

Chapter 6. RESULTS

This chapter evaluates whether ACES can improve candidate-mask selection when applied to real segmentation outputs. The experiments use candidate masks generated by the same Grounding DINO and SAM 2 pipeline for all compared selection methods. The upstream localization and mask-generation stages, therefore, remain unchanged, and the evaluated methods differ only in how they select one final mask from the available candidate set.

Section 6.1 presents the evaluation process based on fixed default values for the t-vector, as introduced in Section 5.4, in which the scoring function can be evaluated for each candidate mask without requiring image-specific adjustments or ground-truth information. The resulting ranking is compared with the candidate selected based on the upstream SAM confidence score. Section 6.2 examines how the reference values of the t-vector can be adjusted when additional target information is available and analyzes representative success and failure cases.

6.1 DEFAULT SETTING EVALUATION

This section evaluates ACES under its default no-user-input configuration. The default t-vector defined in Section 5.4, with the image-border-distance quantile fixed at $q=0.25$, is

$$[t_A, t_C, t_E, t_S] = [0.25, 1.00, 0.00, 1.00]$$

Accordingly, Equation (13) becomes

$$\begin{aligned} \text{Score}(M_i; \mathbf{t}) &= A(M_i; t_A) + C(M_i; t_C) + E(M_i; t_E) + S(M_i; t_S) \\ &= A(M_i; 0.25) + C(M_i; 1.0) + E(M_i; 0.0) + S(M_i; 1.0) \end{aligned}$$

where M_i is a candidate mask. In the rest of this chapter, this ACES scoring function with the default t-vector values is referred to as the ACES reference function.

The same ACES reference function is evaluated for every image in each sequence. No ground-truth masks are used to recompute the reference values of the t-vector or select the final candidate.

The central comparison is between two selection rules applied to the same candidate masks. The SAM baseline selects the candidate with the highest upstream confidence score, whereas ACES selects the candidate with the highest score from the ACES reference function. Consequently, the difference in IoU results from the candidate-selection rule rather than from a change in candidate generation.

The evaluation is based on the DAVIS 2016 dataset, which contains 2,798 images from 40 sequences. Table 6.1 reports the mean IoU across all the images.

Selection Rule	Mean IoU	Gain over SAM	Notes
SAM confidence	0.6175	0.0000	upstream SAM confidence score
ACES default	0.7308	+0.1133	Fixed training-free reference values for the t-vector

Table 6.1. Overall candidate-selection performance from the DAVIS 2016 sequences.

The fixed ACES configuration increases the mean IoU from 0.6175 to 0.7308, corresponding to a gain of 0.1133 over the SAM confidence-based selection. Because both methods receive the same candidate set, this improvement indicates that the highest-confidence SAM candidate is not always the candidate with the greatest ground-truth overlap. In a substantial number of cases, better candidates exist in the generated set but are ranked lower by the upstream confidence scores.

Figure 6.1 presents the mean IoU of both selection methods for each sequence. The sequences are ordered from the largest ACES improvement on the left to the largest decrease on the right.

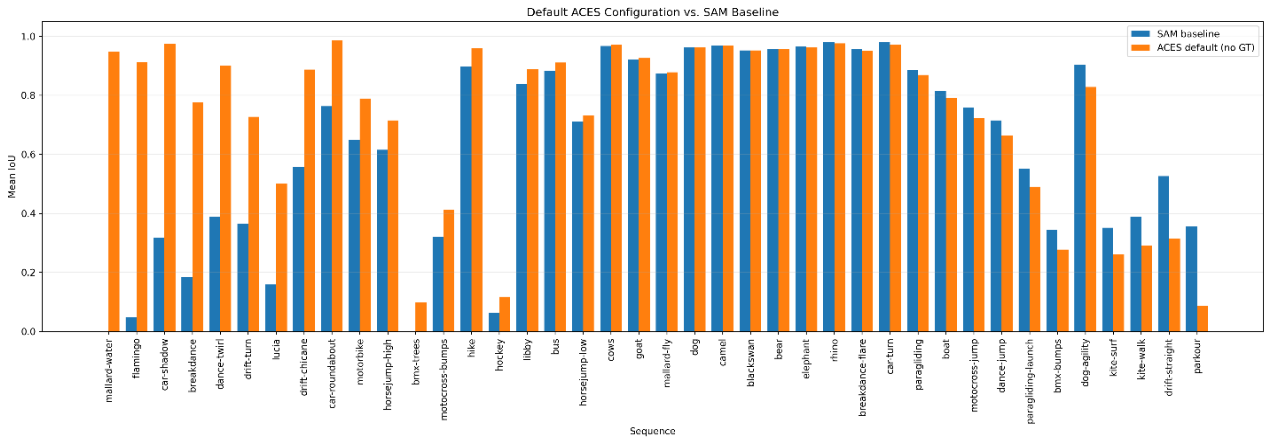


Figure 6.1. Per-sequence mean IoU comparison between SAM confidence-based selection and scores from the ACES reference function.

The sequence-level results show that the overall improvement is not uniform. ACES improves the mean IoU on 23 of the 40 evaluated sequences and decreases on 17. Many of these differences are relatively small and statistically insignificant. A smaller group of sequences has significant positive or negative differences. The overall gain in Table 6.1 is driven primarily by sequences in which the two selection rules select substantially different candidate masks.

The largest improvements of ACES occur in the following sequences: mallard-water, flamingo, car-shadow, breakdance, dance-twirl, drift-turn, lucia, and drift-chicane. For example, the mean IoU increases from approximately 0.0000 to 0.9466 on mallard-water, from 0.0479 to 0.9131 on

flamingo, from 0.3167 to 0.9749 on car-shadow, and from 0.1833 to 0.7763 on breakdance. These large improvements demonstrate candidate-selection failures rather than candidate-generation failures. Because both methods are based on the same candidate set, the high-IoU masks selected by ACES were already present in the upstream output.

Analysis of the cases suggests that the improvements commonly occur when the candidate set contains multiple plausible object regions. For example, as illustrated in Figure 6.2, sequences such as the flamingo (top row) and breakdance (bottom row) contain multiple visually plausible foreground instances. Under these conditions, the upstream confidence score may select a distractor or a non-target instance, while the ACES reference function favors a candidate whose scale, location, edge relationship, and silhouette are more consistent with the intended target. It is important to reiterate that the improvements result from re-ranking the existing candidates rather than from modifying SAM 2 or generating new masks.



Figure 6.2. Representative candidate-selection improvements in multi-instance scenes: raw images (left), SAM confidence-based selections (middle), and ACES selections (right) for the flamingo and breakdance sequences.

A second group of sequences shows little difference between the two methods. Examples include cows, goat, mallard-fly, dog, camel, blackswan, bear, elephant, rhino, and breakdance-flare. In many of these sequences, the SAM baseline already achieves a high mean IoU, leaving limited room for improvements. For example, as illustrated in Figure 6.3, the SAM and ACES results are nearly identical for camel (top row), blackswan (middle row), and bear (bottom row). In such cases, the two methods either select the same candidate or choose candidates with very similar ground-truth overlap. These near-ties indicate that ACES generally preserves performance when the upstream confidence ranking is already reliable, although minor positive or negative

differences can still occur.

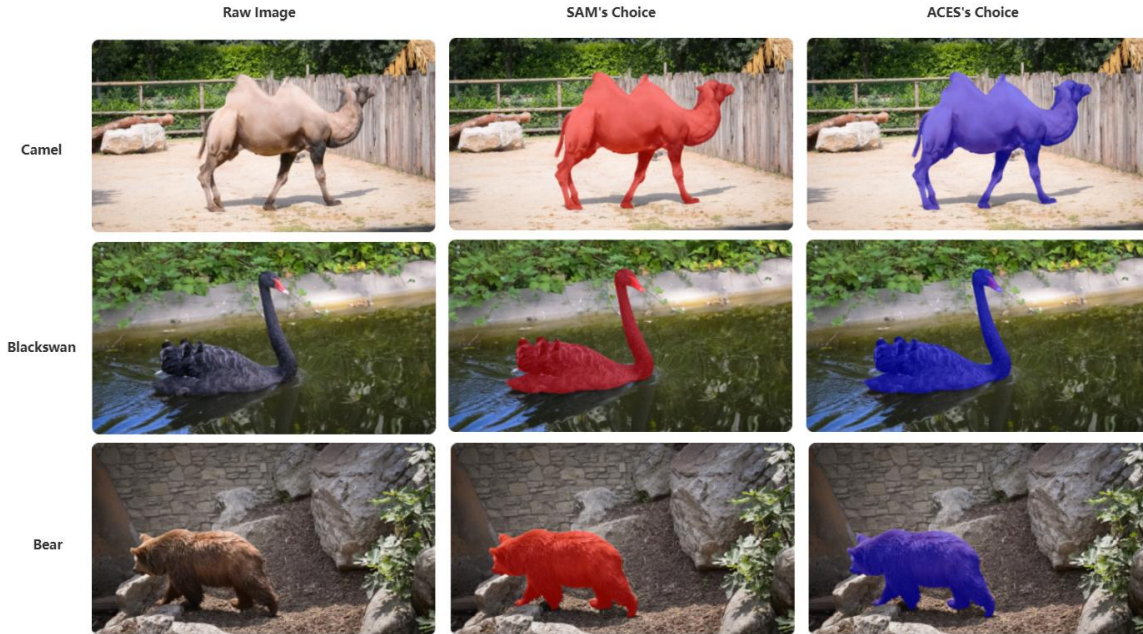


Figure 6.3. Representative near-tie cases in which SAM confidence-based selection and ACES produce nearly identical masks for the camel, blackswan, and bear sequences.

The right side of Figure 6.1 contains the main failure sequences. The largest decreases occur in parkour, drift-straight, kite-walk, kite-surf, dog-agility, and bmx-bumps. These results are consistent with the limitations identified in Section 3.4. A set of fixed reference values for the t -vector is not suitable for describing the geometric characteristics of all targets. For example, as illustrated previously in Figure 5.6, the bicycle in bmx-bumps contains thin structures and hollow wheels, whereas the default value $t_5 = 1$ describes a coherent and largely hole-free target. In this case, a candidate that incorrectly fills the wheel interiors may match the default silhouette reference value more closely than a topologically correct hollow-wheel candidate.

Lastly, a low ACES result may occur because no sufficiently accurate candidate is available, because the fixed reference values do not describe the target (as discussed above), or because the ACES reference function identifies a candidate with an incorrect semantic object. These alternatives are distinguished later through qualitative inspection of representative cases. Nevertheless, Figure 6.1 establishes where the fixed configuration succeeds and where image- or sequence-specific target information may be required.

Overall, the experiment with the default ACES reference function supports two conclusions. First, a fixed, training-free ACES configuration can substantially improve candidate selection

without modifying the upstream segmentation pipeline and does not require referencing ground truth during selection. Second, the default values for t-vector are not universally appropriate: when the target geometry or topology differs substantially from the canonical reference values, the same compatibility functions may favor an incorrect candidate. This motivates the adjusted reference value evaluation in the following section.

6.2 PARTIAL GT DERIVED REFERENCE VALUE EVALUATION

The experiment in Section 6.1, using the default ACES reference function, assumes that no target-specific information is available. Note that the default t-vector of the ACES reference function represents only one idealized target configuration: a medium-sized, centered, interior, and structurally coherent foreground object. When the intended target consistently differs from this configuration, the default reference values may favor geometrically plausible distractors rather than the target.

This section evaluates whether customized reference values for the t-vector can improve candidate selection performance. The purpose is not to optimize feature weights or modify the ACES scoring functions. Instead, limited ground-truth masks are used to estimate the t-vector values that a user could specify by following the structured procedure described in Chapter 5. The following experiments examine the ACES performance when the target’s geometric properties are available.

6.2.1 *Partial GT Reference Value Specification*

For each sequence, 20% of the annotated images are selected as the partial ground-truth (GT) subset. To avoid concentrating the samples within one portion of the video, the sequence is divided into temporal buckets, and one image is randomly sampled from each bucket using a fixed random seed. This produces a temporally distributed set of representative target masks.

For every sampled ground-truth mask G_j , the same four observation functions used for candidate masks are applied, restating Equations (4), (6), (9), and (11) from Chapter 3,¹

¹ The definition of the symbols can be found in the corresponding sections in Chapter 3.

$$\begin{aligned}
r(G_j) &= \frac{|G_j|}{HW} \\
c(G_j) &= 1 - \min(D(G_j), 1) \\
e(G_j) &= 1 - d_{\text{edge}}(G_j) \\
s(G_j) &= \text{Solidity}(G_j)[1 - \text{Fragmentation}(G_j)][1 - R_{\text{hole}}(G_j)]
\end{aligned}$$

where these are the functions that evaluate the given masks for values representing normalized foreground area, centeredness, image-edge proximity, and silhouette coherence, with the image-border-distance quantile remaining fixed at $q=0.25$.

The sequence-specific reference values of t-vector are then computed as the mean of the sampled ground-truth mask values:

$$\mathbf{t}_{GR} = [t_A, t_C, t_E, t_S] = \frac{1}{m} \sum_{j=1}^m [r(G_j), c(G_j), e(G_j), s(G_j)]$$

where m is the number of sampled ground-truth masks, in this way, the resulting t-vector is actually the average description of the selected masks' geometric characteristics. The values from this t-vector are used in Equation (13) for computing the ACES scores of all candidate masks in the corresponding sequence.

This experiment should be interpreted as a target-reference value specification study rather than a label-free deployment result or a strictly held-out generalization experiment. Its purpose is to test the central claim developed in Chapter 5: when the user can provide more accurate values describing the expected target geometry, the same ACES measurement and ranking mechanism can be improved without modifying the upstream segmentation system.

6.2.2 Candidate-Set Screening and Overall Performance

ACES is designed to select among candidate masks generated by an upstream pipeline such as Grounding DINO and SAM 2 pipeline. If no candidate adequately represents the intended target, updating the t-vector values cannot recover a correct mask. It is therefore necessary to distinguish candidate-selection failures from candidate-generation failures before evaluating the benefit of target-specific reference values.

For each image, an oracle candidate IoU is computed as the maximum IoU among all available candidates. The sequence-level oracle candidate mean IoU consequently represents the performance that an ideal selection rule could achieve from the existing candidate set. A sequence

is classified as a candidate-generation failure when its oracle candidate mean IoU is below 0.60.

Table 6.2 summarizes candidate-set quality for all cases in the first row, the separation based on success in candidate generation in the second row, and sequences of failure in candidate generation in the third row.

Group	Sequences	Images	SAM IoU	Partial-GT ACES IoU	Gain over SAM	Mean ACES Score	Oracle Candidate IoU	Candidate Failure Ratio
All sequences	40	2798	0.6175	0.7781	+0.1606	3.6549	0.8365	0.1362
Oracle candidate mean IoU (≥ 0.60)	35	2424	0.6781	0.8569	+0.1787	3.7353	0.9141	0.0276
Oracle candidate mean IoU (< 0.60)	5	374	0.2242	0.2675	+0.0433	3.1336	0.3334	0.8396

Table 6.2. Candidate-set quality and partial GT derived ACES performance before and after excluding upstream candidate-generation failures.

The first row of Table 6.2 reports the results across all 40 sequences and 2,798 evaluated images. Over this complete set, SAM confidence-based selection achieves a mean IoU of 0.6175, whereas ACES using partial GT derived reference values achieves 0.7781, producing a gain of 0.1606. The corresponding mean ACES score is 3.6549, while the oracle candidate mean IoU is 0.8365, indicating that substantially better masks are often present in the candidate set than those selected by SAM; however, 13.62% of the images contain no candidate with an IoU of at least 0.60.

The second row reports the 35 sequences with oracle candidate mean IoU of at least 0.60, covering 2,424 images. In this group, ACES improves mean IoU from 0.6781 to 0.8569, with a higher mean ACES score of 3.7353, an oracle candidate mean IoU of 0.9141, and a low candidate-failure ratio of 0.0276. The third row reports the five sequences classified as candidate-generation failures, comprising 374 images, where both SAM and ACES perform substantially worse, with mean IoU values of 0.2242 and 0.2675, respectively. Their low oracle candidate mean IoU of 0.3334 and high candidate-failure ratio of 0.8396 confirm that the main limitation in this group is the absence of sufficiently accurate masks in the upstream candidate set rather than the candidate-ranking rule.

The five failure sequences in the third row of Table 6.2 are hockey, bmx-trees, kite-surf, bmx-bumps, and kite-walk. Figure 6.4 illustrates a representative candidate-generation failure from the kite-surf sequence, in which the available masks cover only partial components of the intended target. Because no candidate sufficiently represents the complete target, ACES cannot recover the correct mask through re-ranking. For this reason, these five sequences are excluded from the following discussion.



Figure 6.4. Representative candidate-generation failure in the kite-surf sequence: (a) raw image, (b) ground-truth target, and (c–d) two incomplete candidates generated by the Grounding DINO and SAM 2 pipeline.

Table 6.3 compares the three selection settings on the remaining 35 sequences.

Selection Rule	Mean IoU	Gain over SAM	Gain over Default
SAM confidence	0.6781	0.0000	—
ACES default, no GT	0.8115	+0.1333	0.0000
ACES with partial GT derived reference values	0.8569	+0.1787	+0.0454

Table 6.3. Image-weighted candidate-selection performance after excluding sequences with oracle candidate mean IoU below 0.60.

As discussed, the fixed default configuration improves mean IoU from 0.6781 to 0.8115, corresponding to a gain of 0.1333 over SAM confidence selection. Replacing the default t-vector values with reference values derived from the sampled ground-truth masks further increases the mean IoU to 0.8569. This produces an improvement of 0.1787 over SAM and 0.0454 over the default ACES configuration.

The improvement over the default configuration demonstrates that the reference values meaningfully influence candidate ranking. The four compatibility functions remain unchanged; only their expected target values are updated. When these values more accurately represent the object’s scale, position, edge relationship, and silhouette structure, ACES can more reliably distinguish the intended target from geometrically different distractors.

6.2.3 Sequence-Level Results and Parkour Case Study

Figure 6.5 compares the ACES reference function with that based on the partial GT derived reference values for each of the 35 retained sequences. The sequences are ordered according to the improvement produced by the partial GT configuration relative to the default configuration.

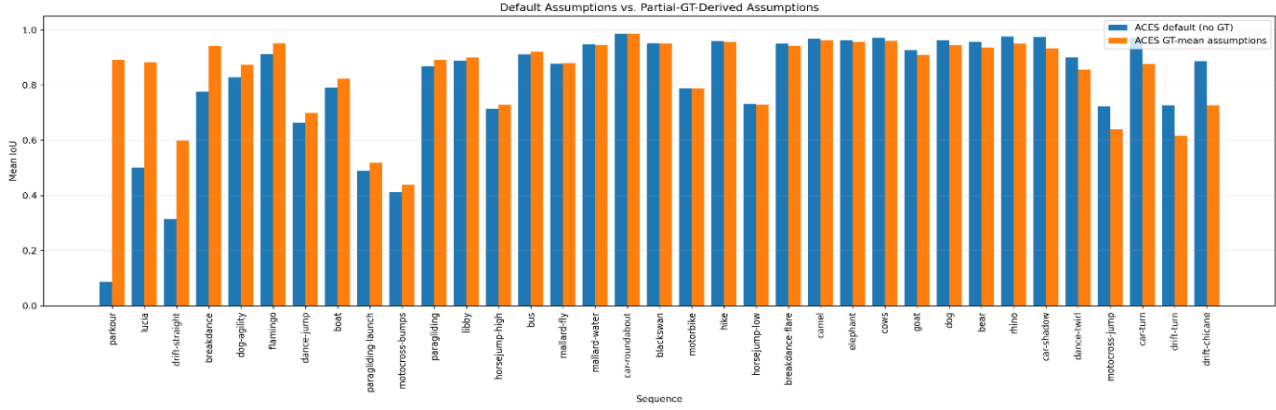


Figure 6.5. Per-sequence mean IoU comparison between the default ACES reference function and that based on partial GT derived reference values, excluding the 5 failure sequences.

The performance gain is not uniform across the sequences. The partial GT derived version outperforms the default configuration in 15 sequences and produces lower mean IoU in 20 sequences. Nevertheless, the image-weighted overall mean increases by 0.0454 because several severe failures are corrected.

The largest improvement occurs in parkour, where the mean IoU increases from 0.0866 under the reference ACES function to 0.8905 with the partial GT derived version. Substantial improvements also occur in lucia (from 0.4997 to 0.8820), drift-straight (from 0.3140 to 0.5995), and breakdance (from 0.7763 to 0.9412). These sequences contain targets whose expected properties differ considerably from those described by the canonical default t-vector. The results indicate that target-specific adjustment is particularly valuable when the intended object is systematically smaller, less centered, closer to an image boundary, or less structurally coherent than the default t-vector would describe.

The parkour sequence provides the clearest example of this behavior. Figure 6.6 presents one representative frame.



Figure 6.6. Parkour candidate-selection example: (a) raw image, (b) ground-truth overlay, (c) SAM-confidence selection, (d) partial-GT ACES selection, and (e) default ACES selection.

As shown in Figure 6.6(a) and Figure 6.6(b), the intended target is the jumping person. The geometry of the target differs substantially from that described by the default t-vector used in Section 6.1. In this case, the target occupies only a small fraction of the image, is displaced from the exact image center, lies closer to the top image boundary, and has an articulated, non-convex silhouette with lower solidity than a compact foreground region.

The 20 sampled ground-truth images produce the following reference value t-vector:

$$[t_A, t_C, t_E, t_S] = [0.0500, 0.7828, 0.4985, 0.6677]$$

Figure 6.6(c) shows that the SAM confidence score selects a foreground bollard rather than the jumping person. This is a confidence-ranking failure, not an absence of a suitable candidate: Figure 6.6(d) shows that the same candidate set contains a high-quality mask of the intended target.

Using the ACES function that is based on the partial GT derived t-vector, ACES selects the person mask in Figure 6.6(d). The selected candidate receives an ACES score of 3.45. Its small area, moderately off-center position, intermediate edge proximity, and articulated silhouette are consistent with the sequence-specific reference values.

The sequence-level results reinforce this interpretation. The SAM confidence baseline achieves a mean IoU of 0.3565 on parkour, while the default ACES reference function achieves 0.0866. With the ACES function based on a partial GT derived t-vector, an IoU of 0.8905 is achieved. The oracle candidate mean IoU is 0.9593, confirming that high-quality target candidates are available in many frames in the sequence and that many suboptimal selections are attributable to candidate ranking rather than candidate generation.

This example demonstrates that reference value adjustment may be necessary for targets that clearly violate the canonical default geometric assumptions. The partial GT procedure does not introduce a new scoring model. It supplies the existing scoring function with a more accurate description of the target, analogous to a user estimating the target characteristics through the process presented in Chapter 5.

The adjusted reference values do not improve every sequence. In several cases, performance decreases relative to the default vector, even though they are derived from sampled ground-truth masks. These decreases indicate limitations associated with temporal variation, scalar averaging, candidate ambiguity, and the handcrafted property representation. They are examined in the following section.

6.3 LIMITATIONS AND FAILURE ANALYSIS

Although both the default and partial GT derived configurations improve the overall performance of candidate selection, the improvement is not uniform across all sequences. ACES attempts to represent the geometry of the target using four scalar reference values that describe area, centeredness, image-edge proximity, and silhouette coherence. This compact representation is effective when the target properties remain reasonably stable, but it becomes less reliable when the target changes substantially across a sequence.

The motocross-jump sequence provides a representative example. As illustrated in Figure 6.7, the apparent size and pose of the motorcycle-and-rider target change substantially as it approaches the camera. In the earlier portion of the sequence, the target occupies a small region of the image, while in the later portion, the same target occupies a much larger region and is observed from a different angle.

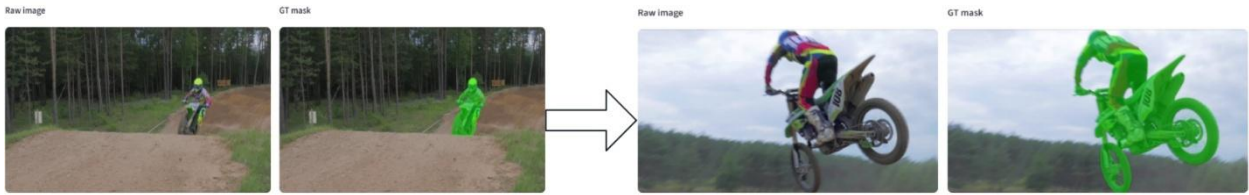


Figure 6.7. Target-property variation in the motocross-jump sequence, from an early frame to a later frame.

The foreground-area observation reflects the magnitude of this variation. Across the sampled ground-truth images, the area ratio ranges from 0.0390 to 0.5073, with a mean of $t_A = 0.1895$ and a standard deviation of 0.1561. In this case, there exists no appropriate t_A that can properly capture the foreground area of the target in the entire sequence. Additionally, in this case, the changes in target position, viewing angle, and articulation can also invalidate the center-location, edge-proximity, and silhouette observations, although area variation is the most pronounced mismatch.

The quantitative results are consistent with this failure mode. On motocross-jump, the SAM confidence baseline achieves a sequence mean IoU of 0.7586. The fixed default ACES configuration decreases the result to 0.7236, and the partial GT derived version further decreases the IoU to 0.6394. The oracle candidate mean IoU is 0.7955, indicating that usable target candidates are present. The failure, therefore, arises from the ranking process.

This case also demonstrates the limitation of relying exclusively on binary mask geometry.

SAM retains learned image features, prompt information, appearance cues, and semantic representations that can associate a candidate with the requested motorcycle-and-rider target despite large geometric changes. ACES does not observe the source image or the semantic identity of the foreground region. When any candidate matches the supplied geometric reference values more closely than the intended target, ACES has no other information to reject it. Geometric compatibility should not be interpreted as semantic correctness.

A possible future extension is to amend the target reference values with reliability weights:

$$S_{weighted} = w_A A_{match} + w_C C_{match} + w_E E_{match} + w_S S_{match}$$

The reference values and reliability weights would serve different purposes. The target values would continue to describe the expected geometric properties, whereas the weights would determine how strongly each property affects the final ranking. When a property varies substantially across a sequence, its weight could be reduced. For example, the high variance of the area in the motocross-jump sequence could justify a smaller w_A , preventing an unreliable area reference value from exerting the same influence as the more stable properties. Such weights could be specified manually or estimated from the variance of the sampled ground-truth masks.

ACES also depends on the quality of the upstream candidate set. It can only re-rank masks that have already been generated and cannot reconstruct a missing target. As shown by the five low-oracle sequences identified in Table 6.2, when Grounding DINO or SAM 2 fails to produce a valid candidate, even an ideal selection rule cannot achieve a high IoU. These cases should be addressed by modifying the upstream prompt, localization procedure, segmentation model, or candidate-generation strategy, and cannot be addressed by adjusting the ACES reference values.

The current results identify two distinct requirements for a successful ACES application. First, the candidate set must contain at least one sufficiently accurate representation of the target. Second, in the case of a sequence, the geometric properties of the target must remain relatively stable across the sequence. When either requirement is violated, the selection performance will become unstable.

6.4 SUMMARY

This chapter evaluated ACES based on real candidate masks generated by the Grounding DINO and SAM 2 pipeline. Using the default ACES reference function without ground-truth information, ACES increased the image-weighted mean IoU from 0.6175 for SAM confidence-based selection to 0.7308 across 2,798 evaluated images. The sequence-level results showed that this improvement

was concentrated in cases where SAM assigned high confidence to incorrect candidates while a substantially better target mask already existed in the candidate set.

The partial ground-truth experiment examined whether more representative target reference values could further improve the ACES selection performance. Five sequences were excluded because their sequence-level oracle candidate mean IoU was below 0.60, indicating persistently poor upstream candidate-set quality. Across the remaining 35 sequences, the SAM baseline achieved a mean IoU of 0.6781, the default ACES configuration achieved 0.8115, and the partial GT derived reference values achieved 0.8569. The ACES function, based on the derived reference values, therefore improved the mean IoU by 0.1787 over SAM and by 0.0454 over the fixed default configuration. The parkour case demonstrated that target-specific reference values can correct severe failures when the intended object differs substantially from the canonical default target.

The results also identified two principal limitations. ACES cannot recover a target that is absent from the upstream candidate set, and a single reference t-vector may become unreliable when target properties vary substantially within a sequence.

Overall, the experiments show that ACES can substantially improve candidate selection when at least one sufficiently accurate target candidate is available and the reference values in the t-vector remain representative.

Chapter 7. CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

This thesis investigated the inference-time candidate-mask selection problem in modern image segmentation pipelines. Promptable segmentation systems may generate multiple plausible masks for the same intended target, while downstream applications generally require one final mask. The problem occurs when the set of plausible masks generated by the upstream segmentation system includes a correct representation of the target, but the associated confidence scores do not rank that mask highly enough for selection. This thesis differentiated candidate generation and candidate selection as separate stages and focused on selecting among masks produced by a fixed upstream segmentation system.

Inspired by classical image-processing techniques for analyzing binary masks and extracting geometric and morphological properties, this thesis proposed ACES, a user-guided scoring framework that operates directly on binary candidate masks and image dimensions to rank candidate masks without retraining, additional annotations during inference, architectural modifications, or access to internal neural network representations. ACES defines four property-specific functions that measure the Area, Center-location, Edge-proximity, and Silhouette of each binary candidate mask, each expressed as a normalized value. Each function also accepts a target reference parameter that characterizes the expected value of the corresponding property for the intended target and measures the compatibility between the candidate-mask value and the corresponding target reference value, producing one normalized compatibility term. The four terms are combined through an unweighted additive score, and the highest-scoring candidate is selected as the final output.

The four parameters for the functions, the t-vector $t = [t_A, t_C, t_E, t_S]$, describe the target in terms of the expected Area, Center-location, Edge-proximity, and Silhouette of the intended target. These values can be modified without regenerating the candidate masks, allowing the same candidate set to be re-ranked for different targets or application requirements. This thesis proposes a straightforward procedure for specifying target-specific reference values and defines a fixed default vector for cases where detailed target information is unavailable.

The evaluation demonstrates that ACES improves candidate selection over upstream

confidence-based selection. The fixed default t-vector provides a practical, training-free configuration that improves candidate selection without requiring target-specific adjustments. In many cases, adjusting the t-vector for target-specific reference values can significantly improve the ranking results. By separating candidate ranking from candidate generation, ACES provides an interpretable and model-independent mechanism that can be applied without modifying the upstream segmentation system. Overall, this thesis establishes explicit reference-based geometric and morphological scoring as an effective and portable solution to the modern candidate-mask selection problem.

7.2 FUTURE WORK

One important direction for future work is the development of dynamic reference values. The current experiments apply a single t-vector to the entire sequence, even though the formulation allows the vector to vary across images. A future system could update the reference values over time using target tracking, recently selected masks, temporal smoothing, or predicted changes in motion and scale. This would reduce the type of reference mismatch observed in the motocross-jump sequence, where the target area and pose changed substantially as the object approached the camera.

A second direction is to introduce property-specific reliability weights while preserving the distinction between reference values and feature importance. The reference values should continue to describe the expected target properties, whereas reliability weights would determine how strongly each compatibility term contributes to the final ACES score. When a property of the target varies substantially across images, the corresponding weight could be reduced. For example, large area variation could justify assigning less influence to the Area compatibility term while retaining more stable spatial or silhouette information.

Future work could also combine ACES with semantic evidence. The present framework deliberately excludes RGB appearance and internal model representations to remain training-free and model-independent. However, a hybrid ranking system could combine geometric compatibility with upstream confidence, text-image similarity, object embeddings, or vision-language reasoning. Such a system could use ACES to reject geometrically inconsistent candidates while using semantic evidence to distinguish between different objects with similar mask geometry.

Finally, the framework should be evaluated across additional datasets, target categories, and candidate-generation pipelines. The current study uses DAVIS 2016 candidate masks generated by Grounding DINO and SAM 2. Testing ACES with other promptable segmentation systems, still-image datasets, medical images, robotic perception tasks, and image-editing pipelines would provide a broader assessment of its portability. Additional work could also investigate methods for improving candidate diversity and candidate-set quality, since no ranking procedure can select a correct mask when the upstream system fails to generate one.

BIBLIOGRAPHY

- [1] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3059968.
- [2] S. R. Vantaram and E. Saber, "Survey of contemporary trends in color image segmentation," *J. Electron. Imaging*, vol. 21, no. 4, p. 040901, Oct. 2012, doi: 10.1117/1.JEI.21.4.040901.
- [3] B. Tapasvi, B. S. Kandula, C. Sowmya, T. S. Kalyan, S. Salma, and B. P. Kalyan, "Comprehensive Survey of Semantic Segmentation Techniques Powered by Deep Learning," in *2025 7th International Conference on Intelligent Sustainable Systems (ICISS)*, Mar. 2025, pp. 1208–1216. doi: 10.1109/ICISS63372.2025.11076345.
- [4] Shih-Fu Chang and H. Sundaram, "Structural and semantic analysis of video," in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, New York, NY, USA: IEEE, 2000, pp. 687–690. doi: 10.1109/ICME.2000.871455.
- [5] B. Yuan and D. Zhao, "A Survey on Continual Semantic Segmentation: Theory, Challenge, Method and Application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10891–10910, Dec. 2024, doi: 10.1109/TPAMI.2024.3446949.
- [6] *colmap/colmap*. (Jul. 27, 2025). C++. *colmap*. Accessed: Jul. 27, 2025. [Online]. Available: <https://github.com/colmap/colmap>
- [7] Y. Wang, X. He, S. Peng, H. Lin, H. Bao, and X. Zhou, "AutoRecon: Automated 3D Object Discovery and Reconstruction," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 21382–21391. doi: 10.1109/CVPR52729.2023.02048.
- [8] J. Qi, H. Yang, and Z. Kong, "A review of traditional image segmentation methods," in *5th International Conference on Computer Information Science and Application Technology (CISAT 2022)*, SPIE, Oct. 2022, pp. 400–406. doi: 10.1117/12.2656653.
- [9] K. K. Brar *et al.*, "Image segmentation review: Theoretical background and recent advances," *Inf. Fusion*, vol. 114, p. 102608, Feb. 2025, doi: 10.1016/j.inffus.2024.102608.
- [10] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991, doi: 10.1109/34.87344.
- [11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004, doi: 10.1023/B:VISI.0000022288.19776.77.
- [12] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988, doi: 10.1007/BF00133570.
- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 18, 2015, *arXiv*: arXiv:1505.04597. doi: 10.48550/arXiv.1505.04597.
- [14] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017, doi: 10.1109/TPAMI.2016.2572683.

- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” Jan. 24, 2018, *arXiv*: arXiv:1703.06870. doi: 10.48550/arXiv.1703.06870.
- [16] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,” Oct. 28, 2021, *arXiv*: arXiv:2105.15203. doi: 10.48550/arXiv.2105.15203.
- [17] A. Kirillov *et al.*, “Segment Anything,” Apr. 05, 2023, *arXiv*: arXiv:2304.02643. doi: 10.48550/arXiv.2304.02643.
- [18] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, “Mask Scoring R-CNN,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 6402–6411. doi: 10.1109/CVPR.2019.00657.
- [19] A. Kendall and Y. Gal, “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?,” Oct. 05, 2017, *arXiv*: arXiv:1703.04977. doi: 10.48550/arXiv.1703.04977.
- [20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 1321–1330. Accessed: Apr. 29, 2026. [Online]. Available: <https://proceedings.mlr.press/v70/guo17a.html>
- [21] V. V. Valindria *et al.*, “Reverse Classification Accuracy: Predicting Segmentation Performance in the Absence of Ground Truth,” *IEEE Trans. Med. Imaging*, vol. 36, no. 8, pp. 1597–1606, Aug. 2017, doi: 10.1109/TMI.2017.2665165.
- [22] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV: IEEE, Jun. 2016, pp. 724–732. doi: 10.1109/CVPR.2016.85.
- [23] N. Ravi *et al.*, “SAM 2: Segment Anything in Images and Videos,” Oct. 28, 2024, *arXiv*: arXiv:2408.00714. doi: 10.48550/arXiv.2408.00714.
- [24] S. Liu *et al.*, “Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection,” Jul. 19, 2024, *arXiv*: arXiv:2303.05499. doi: 10.48550/arXiv.2303.05499.
- [25] S. He *et al.*, “Computer-Vision Benchmark Segment-Anything Model (SAM) in Medical Images: Accuracy in 12 Datasets,” May 05, 2023, *arXiv*: arXiv:2304.09324. doi: 10.48550/arXiv.2304.09324.
- [26] “Manipulation of a Complex Object Using Dual-Arm Robot with Mask R-CNN and Grasping Strategy | Journal of Intelligent & Robotic Systems | Springer Nature Link.” Accessed: Apr. 29, 2026. [Online]. Available: <https://link.springer.com/article/10.1007/s10846-024-02132-0>
- [27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2008.
- [28] G. Borgefors, “Distance transformations in digital images,” *Comput. Vis. Graph. Image Process.*, vol. 34, no. 3, pp. 344–371, Jun. 1986, doi: 10.1016/S0734-189X(86)80047-0.
- [29] A. Borji and J. Tanner, “Reconciling Saliency and Object Center-Bias Hypotheses in Explaining Free-Viewing Fixations,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1214–1226, Jun. 2016, doi: 10.1109/TNNLS.2015.2480683.
- [30] T. Judd, K. Ehinger, F. Durand, and A. Torralba, “Learning to predict where humans look,” in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 2106–2113. doi: 10.1109/ICCV.2009.5459462.

- [31] “Robust statistics for outlier detection - Rousseeuw - 2011 - WIREs Data Mining and Knowledge Discovery - Wiley Online Library.” Accessed: Jun. 10, 2026. [Online]. Available: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.2>
- [32] “OpenCV: Contour Properties.” Accessed: Jun. 10, 2026. [Online]. Available: https://docs.opencv.org/4.13.0/d1/d32/tutorial_py_contour_properties.html
- [33] A. Rosenfeld and J. L. Pfaltz, “Sequential Operations in Digital Picture Processing,” *J. ACM*, vol. 13, no. 4, pp. 471–494, Oct. 1966, doi: 10.1145/321356.321357.
- [34] “Flood-Fill Operations - MATLAB & Simulink.” Accessed: Aug. 03, 2026. [Online]. Available: <https://www.mathworks.com/help/images/flood-fill-operations.html>
- [35] “Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey | Springer Nature Link.” Accessed: Jun. 10, 2026. [Online]. Available: <https://link.springer.com/book/10.1007/978-3-642-48318-9>
- [36] “NumPy.” Accessed: Jul. 18, 2026. [Online]. Available: <https://numpy.org/>
- [37] “OpenCV - Open Computer Vision Library,” OpenCV. Accessed: Jul. 10, 2026. [Online]. Available: <https://opencv.org/>
- [38] “COCO - Common Objects in Context.” Accessed: Jul. 18, 2026. [Online]. Available: <https://cocodataset.org/#home>
- [39] “pandas - Python Data Analysis Library.” Accessed: Jul. 18, 2026. [Online]. Available: <https://pandas.pydata.org/>
- [40] “Streamlit • A faster way to build and share data apps.” Accessed: Jul. 18, 2026. [Online]. Available: <https://streamlit.io/>