# Intelligent Scissors: Interactive Image Segmentation

Zhiyuan Song
Student ID: 12012503

July 9, 2025

**Abstract**

This report presents an implementation of the Intelligent Scissors interactive image segmentation tool, based on the work by Mortensen and Barrett [1, 2]. We detail the technical aspects of image feature extraction for both grayscale and RGB images, the formulation of the local cost function using Laplacian, Gradient Magnitude, and Gradient Direction features with their associated mathematical formulas, and the efficient Dijkstra-based graph search leveraging a bucket queue for O(N) performance. Core interactive features such as the live-wire path visualization, Cursor Snap for precise interaction, and Path Cooling for automatic seed placement via history tracking are described. The evaluation is primarily visual, demonstrating the tool's ability to accurately trace object boundaries on example images.

## 1 Introduction

Image segmentation, the process of partitioning an image into meaningful regions, is a fundamental task in computer vision and image editing. While fully automated segmentation remains a challenge due to the variability of image content, manual tracing is often tedious and lacks precision. The Intelligent Scissors algorithm, introduced by Mortensen and Barrett, offers an interactive solution that leverages human guidance for high-level decisions while automating the precise boundary tracing along image edges. This report describes our implementation of the core Intelligent Scissors functionality, focusing on image feature extraction, the local cost formulation, the efficient graph search, and key interactive features including the live-wire visualization, Cursor Snap, and Path Cooling.

## 2 Implementation Details

Our implementation follows the modular structure described in the original work. The system consists of several interconnected components responsible for processing the image, calculating costs, finding optimal paths, and managing the interactive user interface.

The `Image` class handles loading images and storing pixel data. The `ImageProcessor` computes essential edge-detection features using convolution kernels. For **RGB images**, features are computed for each channel and then combined to create single-valued feature maps. The `CostCalculator` uses these maps to determine the local cost $l(p, q)$ for a link from pixel $p$ to neighbor $q$:

$$l(p, q) = \omega_Z f_Z(q) + \omega_G f_G(p, q) + \omega_D f_D(p, q)$$

The weights $\omega_Z, \omega_G, \omega_D$ (using values $0.3, 0.3, 0.1$ from [1]) determine the influence of each feature. The feature costs are:

1. $f_Z(q)$: Laplacian Zero-Crossing cost, binary.
2. $f_G(q)$: Gradient Magnitude cost, an inverse linear ramp of the gradient magnitude.
3. $f_D(p, q)$: Gradient Direction cost, based on the alignment of the link direction and edge direction.

The `IntelligentScissorsSearch` class performs a Dijkstra-like search using a **bucket queue** to achieve O(N) performance [2].

## 2.1 Interactive Features

Several features enhance the interactive boundary definition:

1. **Live-Wire:** As the mouse moves, the optimal path from the cursor back to the current seed is displayed in real-time.
2. **Cursor Snap:** The cursor automatically "snaps" to the pixel with the maximum gradient magnitude within a small neighborhood, aiding precise placement.
3. **Path Cooling:** This feature automates seed placement on stable segments by tracking pixel history on the live wire.

# 3 Evaluation Framework and Qualitative Results

A comprehensive evaluation requires both quantitative analysis and qualitative assessment. This section first details the methodology for a formal user study and then presents the qualitative results obtained during this project.

## 3.1 Methodology for a Formal User Study

A rigorous scientific evaluation is structured around a formal experimental design to test clear, falsifiable hypotheses.

### 3.1.1 Hypotheses

The experiment is designed to test the following hypotheses:

1. **Accuracy Hypothesis:** The tool produces boundaries significantly closer to a ground-truth boundary than manual tracing.
2. **Efficiency Hypothesis:** Users can define boundaries significantly faster using the tool.
3. **Reproducibility Hypothesis:** Boundaries produced by the tool are significantly more consistent.

### 3.1.2 Experimental Setup and Procedure

**Materials** An Image Dataset, Ground-Truth Boundaries, a Participant Group, and a Baseline Tool for manual tracing.

**Procedure** Each participant would trace objects multiple times with both tools. Data logged would include boundary coordinates, completion time, and seed points.

### 3.1.3 Quantitative Metrics and Statistical Analysis

**Metric 1: Accuracy** The Mean Boundary Distance would be analyzed using a **two-tailed paired-samples t-test** or its non-parametric equivalent, the **Wilcoxon signed-rank test**.

**Metric 2: Efficiency** The Mean Time to Completion would be analyzed using the same statistical tests.

**Metric 3: Reproducibility** The Mean Boundary Distance between user-generated paths would be analyzed using a **one-way ANOVA** or a **Kruskal-Wallis test**, followed by appropriate post-hoc tests.

## 3.2 Results of Qualitative Evaluation

While a full quantitative study was beyond the project's scope, a qualitative evaluation was performed to validate the tool's core functionality.

As shown in Figure 1, the tool robustly traces high-contrast contours. In regions of high curvature or low contrast, the live-wire occasionally deviated, highlighting the need for user guidance via seed points to constrain the search.

Figure 2 demonstrates performance on a complex color photograph where object and background share similar features. In these ambiguous regions, more frequent user intervention was necessary, confirming that the tool functions as an interactive assistant.

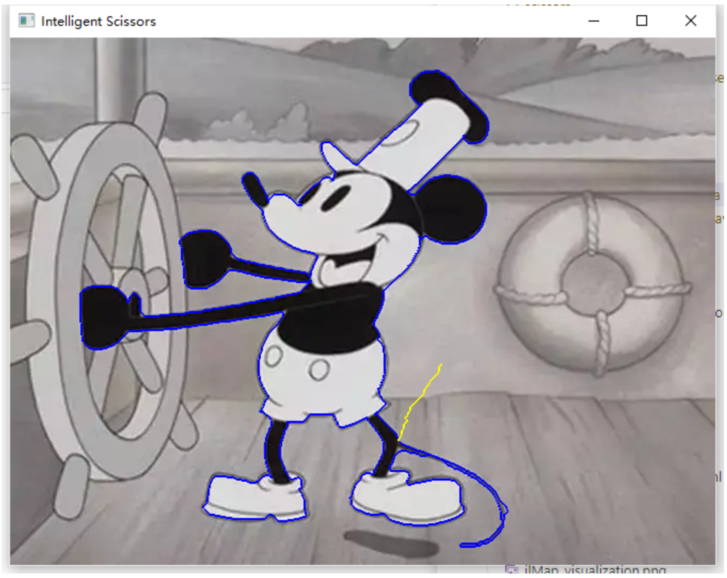
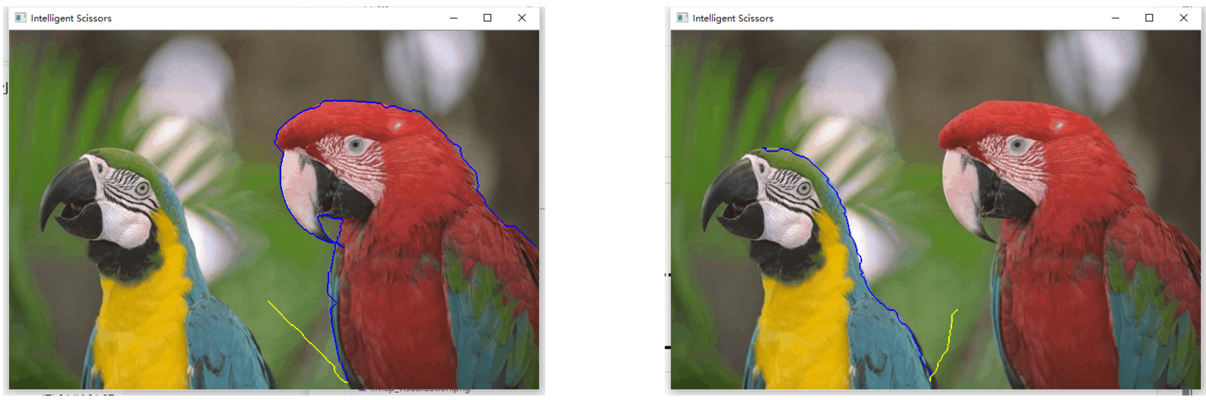



Figure 1: Intelligent Scissors output tracing the outline of Mickey Mouse (grayscale).




(a) Red Parrot outline.

(b) Green Parrot outline.

Figure 2: Intelligent Scissors output on a complex color image, demonstrating challenges in regions of color and texture ambiguity.

## 4 Conclusion and Future Work

We have successfully implemented the core interactive features of the Intelligent Scissors image segmentation tool. This includes image feature extraction for both grayscale and RGB formats, a tunable local cost function, an efficient O(N) graph search using a bucket queue, and key interactive elements: the real-time live-wire visualization, Cursor Snap, and Path Cooling.

Our qualitative evaluation serves as a successful proof-of-concept. It confirms that the tool is functional and behaves as described in the literature, effectively tracing clear object boundaries while relying on user input to resolve ambiguity. The visual results, while insightful, are inherently subjective.

The immediate next step is to conduct the formal user study detailed in Section 3.1. This quantitative analysis is essential for producing statistically defensible claims about the tool's performance relative to manual methods. It would provide the objective evidence required to scientifically validate the significant improvements in accuracy, efficiency, and reproducibility that the Intelligent Scissors tool is designed to offer.

## References


[1] William A. Barrett and Eric N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331–341, 1997.

[2] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pages 191–198, New York, NY, USA, 1995. ACM.