An Implementation of the “Live-Wire” Image Segmentation Tool

Based on the papers “Interactive live-wire boundary extraction” and “Interactive Segmentation with Intelligent Scissors” by W. Barrett and E. Mortensen

James Gomez

CS 269

Final Project Report

Professor Terzopoulos

*Abstract*—Fully automated image segmentation for general imagery is still an unsolved problem. On the other hand, manual segmentation is tedious, time-consuming, and often inaccurate. **In their paper “Interactive live-wire boundary extraction”, Barrett and Mortensen introduced live-wire, an interactive tool for fast, accurate, and reproducible image segmentation via mouse gestures. This paper outlines an implementation of the live-wire tool using Java and the OpenCV computer vision library.**

# Introduction

Image segmentation via manual boundary tracing is tedious, time-consuming, and often inaccurate. On the other hand, accurate, fully-automated general image segmentation is still an unsolved problem. As a result, it is desirable to have interactive tools available that enable users to extract segments of interest from images. These types of tools are especially desirable in the medical field, where extraction of specific objects from medical imagery is of high interest and importance. In 1996, Barrett and Mortensen introduced Live-wire, an interactive tool for fast and reproducible image segmentation. This tool freed users from the tedious job of extracting

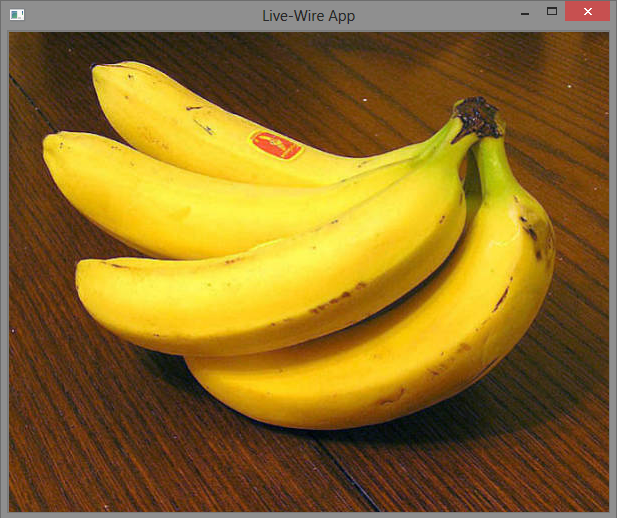
The remainder of this paper is organized as follows: in Section II, the previous work by Barrett and Mortensen is presented. In section III, the methodology and implementation details of this work are presented. In section IV, results of the implementation are presented and discussed. Finally, in section V, the paper is concluded.

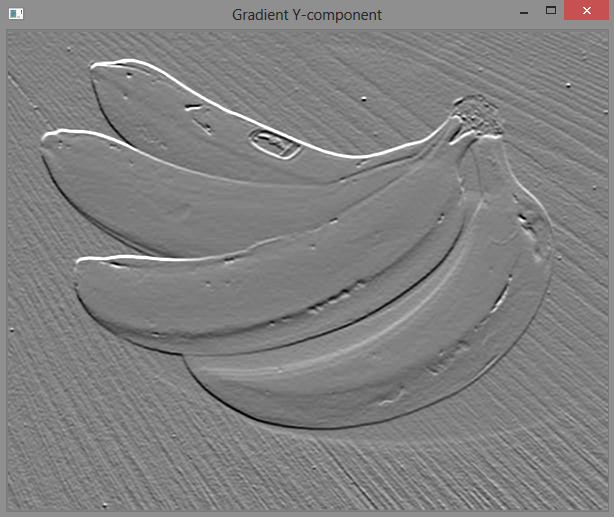
# Previous Work

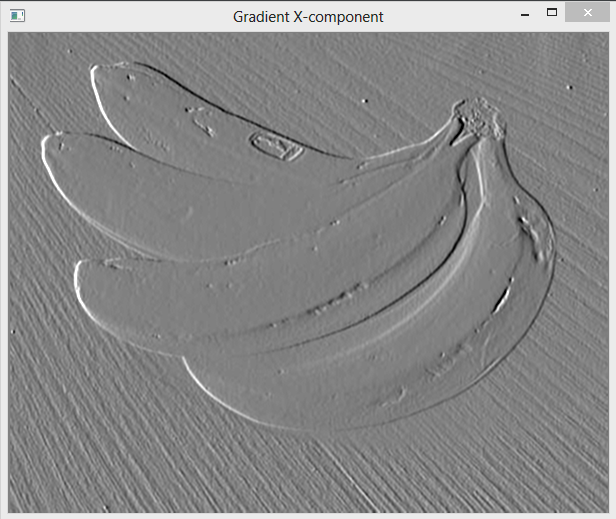
# Methodology and Implementation

In order to keep from re-inventing the wheel, this implementation of the live-wire tool utilizes the OpenCV computer vision library. It was desirable early on to use a language and environment that provides both good application performance (especially during graph expansion) and implementation speed and safety. OpenCV, though powerful, has a standard C++ API interface. Though C++ has the quality of creating high performance native code, it provides a tedious implementation experience. Thus, this live-wire implementation is based in Java and interfaces the OpenCV libraries via the JavaCV wrapper APIs. Java provides the performance needed for this application while offering a safer and higher speed implementation environment.

## Feature Extraction

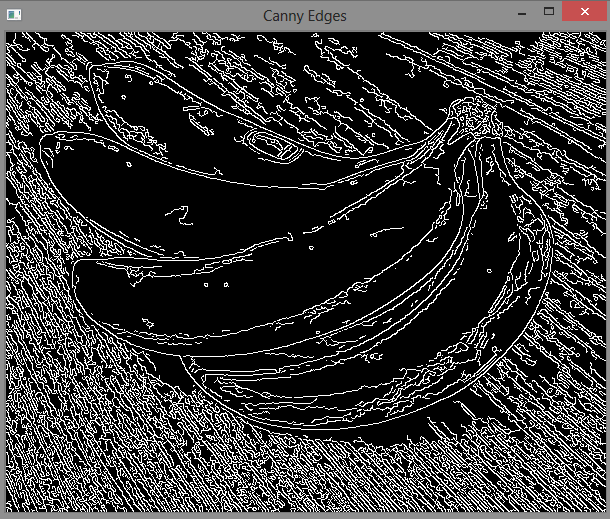


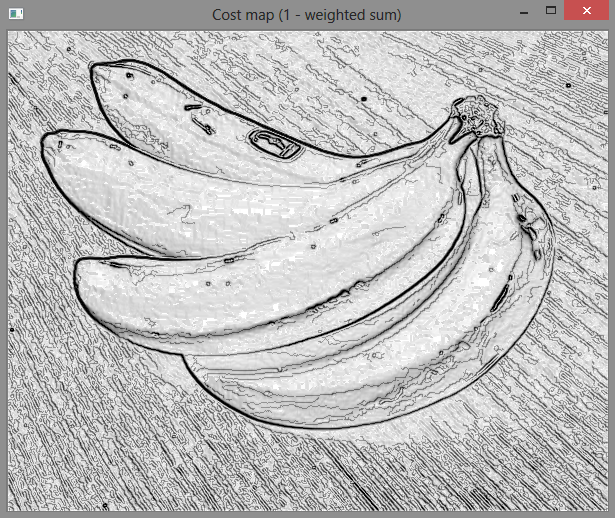












## Local Costs

TODO

## Graph Expansion

TODO

## Live-wire boundary

TODO

## Closed Boundary Detection

TODO

## Boundary and Segment Extraction

TODO

# Results

# Conclusion

# References

|  |  |
| --- | --- |
| [1] | W. A. Barrett and E. N. Mortensen, "Interactive Segmentation with Intelligent Scissors," *Graphical Models and Image Processing,* pp. 349-384, 1998. |
| [2] | W. A. Barrett and E. N. Mortensen, "Interactive live-wire boundary extraction," *Medical Image Analysis,* vol. 1, pp. 331-341, 1996/7. |