I. Introduction

Shape and color recognition algorithms and the software applications which implement them are traditionally very elaborate, and often extremely complex. I propose to develop a software application which efficiently implements a high-quality recognition algorithm. Unlike facial recognition algorithms, the technology industry does not have a standard open-source shape and color recognition toolkit. Therefore, Silhouette will attempt to implement a known shape detection algorithm and package it into a powerful and easy to use open-source application!

I.A Project Description

Upon its completion, Silhouette will consist of an application which recognizes and identifies known shapes and colors from a network-camera as programmed by the user. Unlike other applications which simply extend the power of a library to implement specialized computer vision, Silhouette will use its own recognition algorithm. In other words, the application will not rely on any libraries or external dependencies to operate. Ideally, Silhouette will implement a powerful, yet purely conceptual shape recognition algorithm which has not yet been translated to Java code. I am currently considering several algorithms on which to base my exploration: Canny edge detection algorithm, and the Sobel edge detection algorithm. Both are extremely popular but have not yet been ported to the Java platform and integrated into a unique application.

The Silhouette interface is designed to allow users to configure an incoming network-camera stream, recognition tolerance, and specify which shapes to identify. The main region of the interface consists of a window pane displaying the live network-camera JPG stream, automatically refreshing at a given interval. The bottom region of the application interface consists of a series of mini-canvases, in which users can “scribble” a shape they would like the algorithm to search for. For example, a user may draw a triangle on one of the canvases. Instantly, the

application recognizes the drawn image as a triangle using edge detection, and proceeds to search all incoming JPG-frames for a triangle. When a triangle is recognized, a wire frame shape is superimposed over the real-time streaming image, indicating to the user that a shape was found. Directly to the right of the main window region, will be a set of tools designed to allow the user to configure the tolerance, refresh rate, and other configuration settings of the application and algorithm. Please see the user interface mock-up for more information.

Finally, one of the biggest components of my project consists of manipulating an image stream from a network-camera. Therefore, I feel it is appropriate to explain the difference between a standard web-camera, and an Axis network-camera. A typical web-camera is usually connected directly to a computer via a USB or IEEE 1394 Firewire connection, and often requires third-party web-cam software to properly interface with the hardware. Additionally, most standard web-cameras have a limited resolution and are often not used for high quality imagery. On the other hand, an Axis network-camera is a self-sustaining device which connects directly to a local network. Designed for high performance security surveillance, Axis network cameras stream high quality video and motion-JPG images to any standard web-browser without third party software. In other words, the network-camera uses a self sustained embedded Linux kernel to manage the video stream. An Axis network-camera provides the perfect foundation for which Silhouette will capture live streaming images!

I.B Project Justification

Without question, shape and color recognition is one of the most intriguing and "sexy" applications of computing. By harnessing the power of shape detection, color detection, pattern detection, and gesture recognition, humans continue to grow closer and more intimate with powerful technologies. In essence, Silhouette is an opportunity to explore the foundations of such technologies, while focusing on the development of a custom toolkit which detects shapes and colors in a given image. By developing a custom recognition algorithm, I will expose myself to a unique and fundamentally important segment of computing. Additionally, developing a small set of rough shape recognition algorithms will help me understand the fundamental principles behind other, more powerful algorithms. In conclusion, Silhouette is a great opportunity to expand my knowledge of computing while exploring a popular and potentially powerful technology!