

## V. **Deployment and Delivery**

This section contains all documentation relating to application deployment and final software delivery of the Silhouette Project. It contains the following documents:

- **Deployment Diagram** - Section V.A
- **Build Process** - Section V.B
- **Code Inventory** - Section V.C
- **README File** - Section V.D
- **Known Bugs List** - Section V.E
- **Final Presentation Slides** - Section V.F
- **User Manual w/ Detailed Screenshots** - Section V.G

## V.A Deployment Diagram

The Deployment Diagram for the Silhouette Project is shown in Figure V.A.1 below. Essentially, the deployment of the project itself is very straightforward; only the application and web-camera are required for proper deployment and execution.

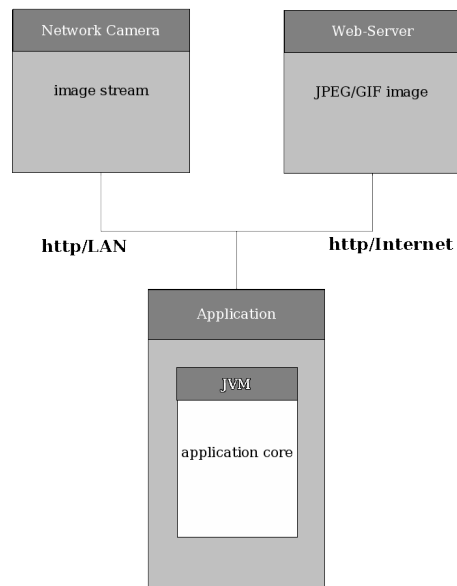


Figure V.A.1. Silhouette Project Deployment Diagram.

## V.B Build Process

In general, the build process is fairly straightforward and should take no more than a few minutes to compile and run Silhouette. No libraries or additional plug-in support is needed! The CD provided with this binder, contains all code and other project related documents. To compile Silhouette, please insert the included CD into your CD-ROM drive. The root of the CD should contain a "README" file and two directories:

- **README.txt** - The Silhouette application README file. Contains compilation and execution instructions, as well as basic information about using the application structure.
- **sources/** - The sources/ directory contains the final copy of the project source-code
- **documents/** - The documents/ directory contains a copy of all project related documents, including the project proposal, all analysis documents, all design documents, all testing related documents, and all Powerpoint slide sets.

To compile the source files available on the included CD, simply follow the set of basic instructions below:

1. **Copy Source Files** - To properly compile the source files available on the CD, you must first copy all files in the sources/ directory to a location on your local hard-drive. Typically, UNIX/Linux/Mac users often copy the sources/ directory to their personal home-directory. Windows users however, often copy the sources/ directory to their windows Desktop. The UNIX/Linux/Mac based command-line instruction for copying the sources/ directory to your home directory is as follows:

```
[mark@trriage cdrom]$ cp -R sources/ ~
```

- 2. Build Java Source Files** - We must now compile the Java source files using the native 'javac' Java Compiler. Please be sure the 'javac' command is correctly installed in the path of your command-line shell. It's important to note that all source files are organized into packages. A description of each package can be found on page 3 of the Product Design section of this notebook (Section III.A.2). To build the application, navigate into the newly copied sources/ directory and issue the following command:

```
[mark@trriage sources]$ javac edu/lmu/cs/kolich/algorithm/*.java
    edu/lmu/cs/kolich/shape/*.java
    edu/lmu/cs/kolich/camera/*.java
    edu/lmu/cs/kolich/silhouette/*.java
    edu/lmu/cs/kolich/utils/*.java
```

For readability, the commands above have been separated with newlines. When entering the commands on the command line, no newlines and tabs are necessary. Simply use a single-space, to space each argument to 'javac'.

- 3. Execute Application Class** - If all sources files compiled correctly, you can now run the application! To run the application, issue the following command from within the sources/ folder:

```
[mark@trriage sources]$ java
    edu.lmu.cs.kolich.silhouette.Silhouette
```

The application should now be open on your screen! For an in-depth user manual and basic HOWTOs, see Section V.G.

**V.C Code Inventory**

The following pages of this section contain hard-copy print-outs of all Silhouette Project source code in alphabetical order.

## V.D README File

The following file excerpt is the contents of the README file found on the included CD-ROM:

Silhouette  
Version 1.0

Mark S. Kolich

Loyola Marymount University  
Department of Computer Science  
Spring 2005

The Silhouette Project is free software.

### OVERVIEW

-----

The overall goal of the Silhouette project is to successfully implement a working application which accepts an image stream from a live network-camera, and applies a series of algorithms to the image aimed at identifying obvious shapes. Silhouette will attempt to recognize known shapes in the real-time camera stream.

While this may seem like a simple task, it is actually more difficult than one may expect. For example, the human mind is clearly capable of determining the difference between an apple and banana, or a cardboard box and a pool-cue. However, training a computer to recognize shapes is a daunting task, and has been studied by many computer engineers since the birth of computer graphics and machine vision.

### COMPILATION

**CMSI 402 SENIOR PROJECT LAB**  
Deployment/Delivery - Silhouette Project - Mark Kolich

-----

In general, the build process is fairly straightforward and should take no more than a few minutes to compile and run Silhouette. No libraries or additional plug-in support is needed! The CD provided with this binder, contains all code and other project related documents. To compile Silhouette, please insert the included CD into your CD-ROM drive. The root of the CD should contain a "README" file and two directories:

README.txt - The Silhouette application README file. Contains compilation and execution instructions, as well as basic information about using the application structure.

Sources/ - The sources/ directory contains the final copy of the project source-code

Documents/ - The documents/ directory contains a copy of all project related documents, including the project proposal, all analysis documents, all design documents, all testing related documents, and all Powerpoint slide sets.

To compile the source files available on the included CD, simply follow the set of basic instructions below:

1. Copy Source Files - To properly compile the source files available on the CD, you must first copy all files in the sources/ directory to a location on your local hard-drive. Typically, UNIX/Linux/Mac users often copy the sources/ directory to their personal home-directory. Windows users however, often copy the sources/ directory to their windows Desktop. The UNIX/Linux/Mac based command-line instruction for copying the sources/ directory to your home directory is as follows:

```
[mark@trriage cdrom]$ cp -R sources/ ~
```

2. Build Java Source Files - We must now compile the Java source files using the native 'javac' Java Compiler. Please be sure the 'javac' command is correctly installed in the path of your command-line shell.

## CMSI 402 SENIOR PROJECT LAB

### Deployment/Delivery - Silhouette Project - Mark Kolich

It's important to note that all source files are organized into packages. A description of each package can be found on page 3 of the Product Design section of this notebook (Section III.A.2). To build the application, navigate into the newly copied sources/ directory and issue the following command:

```
[mark@trriage sources]$ javac edu/lmu/cs/kolich/algorithm/*.java
edu/lmu/cs/kolich/shape/*.java edu/lmu/cs/kolich/camera/*.java
edu/lmu/cs/kolich/silhouette/*.java edu/lmu/cs/kolich/utils/*.java
```

3. Execute Application Class - If all sources files compiled correctly, you can now run the application! To run the application, issue the following command from within the sources/ folder:

```
[mark@trriage sources]$ java edu.lmu.cs.kolich.silhouette.Silhouette
```

The application should now be open on your desktop!

## V.E Known Bugs

At this time, there are no known "bugs." However, there are a few small components of the application which have been disabled. For example, the Triangle

and Diamond buttons on the Shape Selector tool have been temporarily disabled because their implementations are unavailable. Once the functionality for diamond and triangle detection has been implemented, the buttons can be easily enabled.

## **V.F Final Presentation Slides**

The final presentation slide set for the Silhouette Project is included on the next several pages of this section. A digital copy of the slide set, in Powerpoint

format, is also available on the included project CD-ROM.

## **V.G User Manual**

Any professional software product must have a clean and intuitive graphical user interface. The Silhouette Project is no exception! The Silhouette interface was

designed to be sleek, clean, intuitive, and simple. Frankly, there are no bells and whistles and other fancy GUI gadgets integrated into the application. As a result, more development and design effort was placed on perfecting the look and feel of the interface, and ensuring all components interact seamlessly with one another.

This section is designed to be a quick reference user-guide to the Silhouette interface and the various application components.

### **V.G.1      The Core Application Desktop**

The Silhouette project is built using the powerful Java GUI component, `JInternalFrame`. A Java internal frame allows windows and other application GUI

components to be opened as unique windows on the application desktop. For example, in Figure V.G.1.1 below, the Camera Manager is an internal frame/window which is available on the core application desktop. Similar to a standard window based GUI, it can be closed, minimized, and opened at anytime.

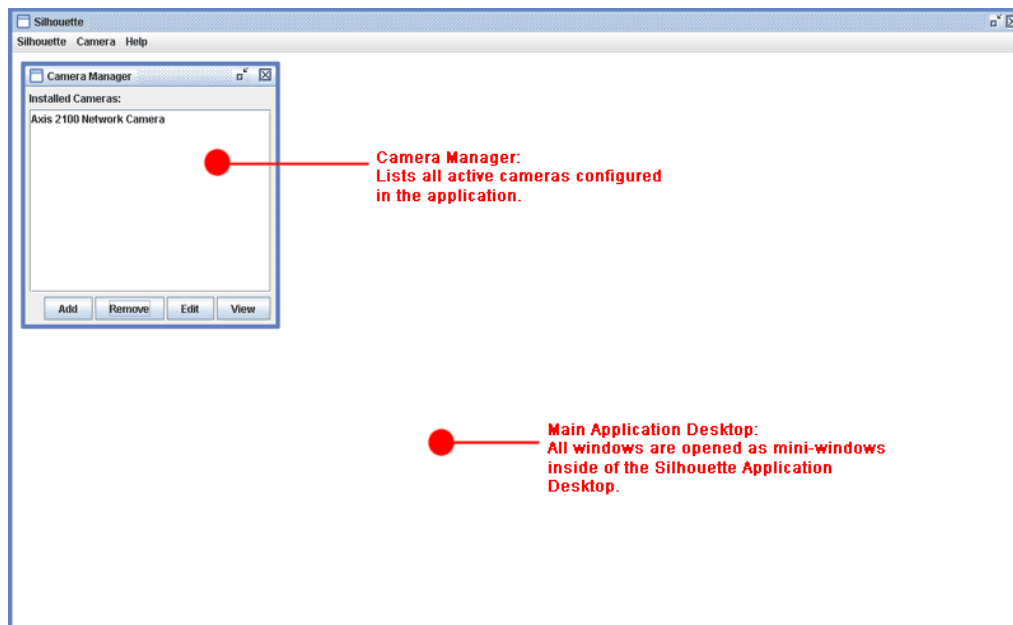


Figure V.G.1.1. The core application desktop.

The Camera Manager manages a list of all current cameras available to the user, and to the application framework. Users can add, remove, edit, and view a camera and its information by selecting the camera from the Camera Manager list, and clicking on the desired action button.

## V.G.2 The Silhouette Menu Bar

Before we introduce the various application components, let's first look at the primary application menu. The menu bar available at the top of the desktop window

is designed to help the user manage configured cameras and application settings. Each menu bar and its components are explained in detail below.

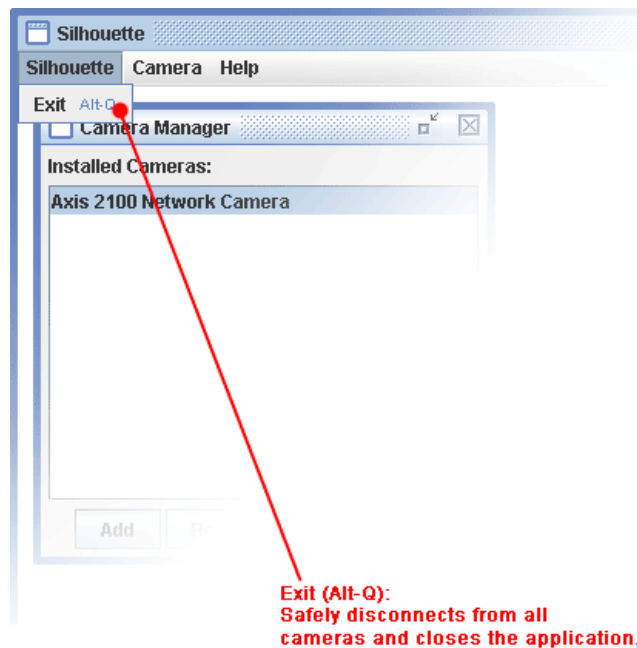


Figure V.G.2.1. The Silhouette menu.

As shown in Figure V.G.2.1, the user may gracefully close the application by selecting Silhouette, Exit from the main application menu bar. As a shortcut, a user may also press Alt-Q to quit at anytime.

Similar to the Silhouette menu, the Camera menu allows a user to add a camera, open the Camera Manager, and open the Shape Selector tool. As shown in Figure V.G.2.2 below, the user can also use several shortcuts:

- **Alt-A:** Opens the Add Camera dialog window.
- **Alt-M:** Opens the Camera Manager dialog window.
- **Alt-S:** Opens the Shape Selector Tool.

Each dialog and tool window are explained in detail later in this section.

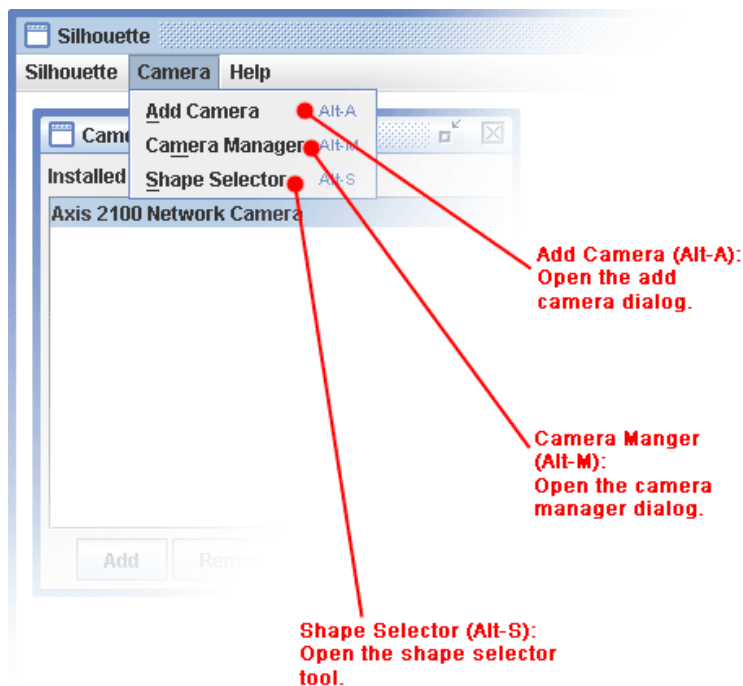


Figure V.G.2.2. The Silhouette Camera menu.

Finally, the Silhouette Help menu allows the user to open the application About Dialog, which provides a general description of the application and its main focus. As shown in Figure V.G.2.3 below, the user can also press Alt-H to open the About Dialog window.

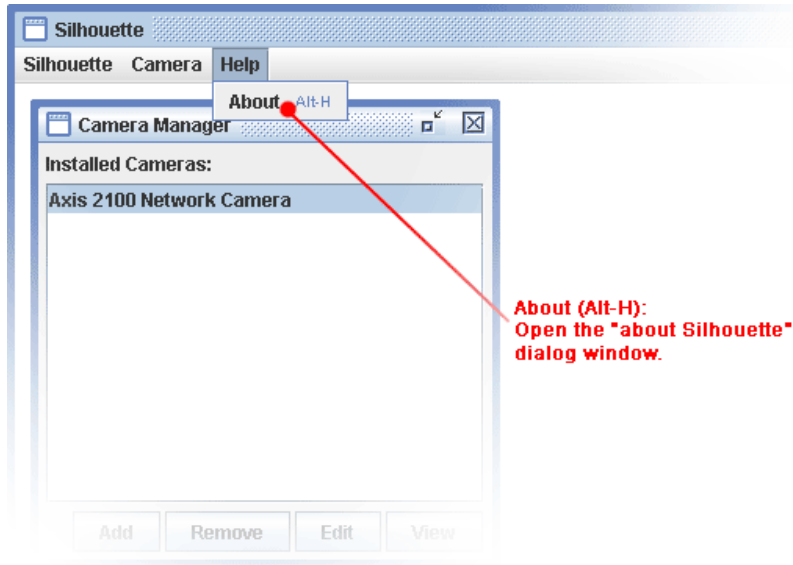


Figure V.G.2.3. The Silhouette Help menu.

### V.G.3 The Camera Manager

The Camera Manager, shown in Figure V.G.3.1 below, allows the user to add, remove, modify, or view a camera and its connectivity information. Each button and

their corresponding actions are shown below in red.

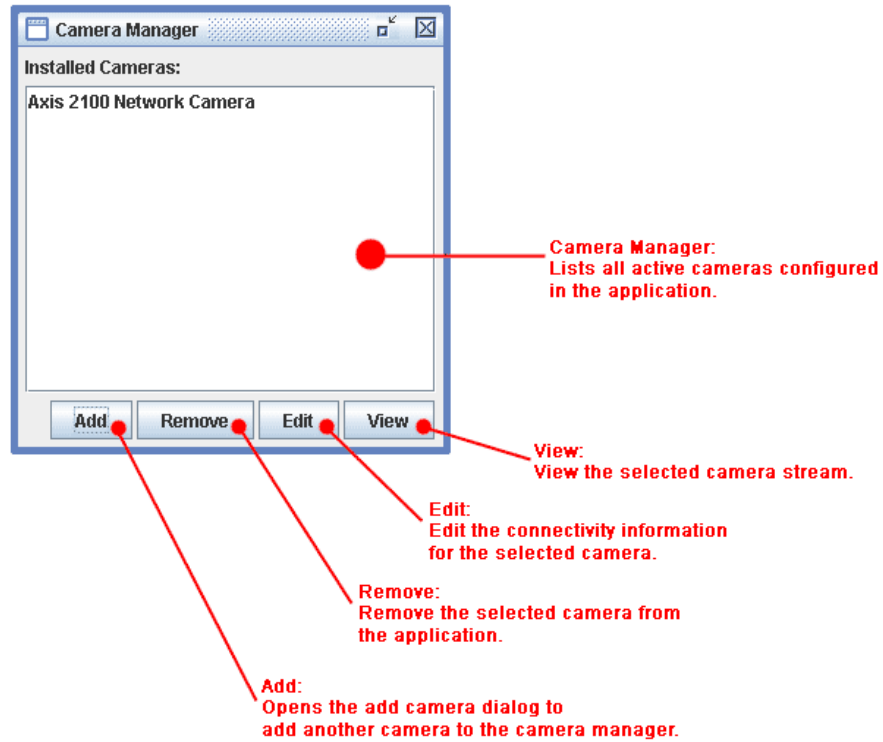


Figure V.G.3.1. The Silhouette Camera Manager.

#### V.G.4 The Camera Creator - Adding a Camera

Silhouette is a very flexible framework, which can connect to any camera stream using HTTP. Therefore, the Camera Creator dialog allows users to add a

camera to the Camera Manager. Once a camera is added to the Camera Manager, the user can then view the stream and apply various edge detection and shape recognition algorithms to the camera's image feed. Figure V.G.4.1 below highlights the important components of the Camera Creator dialog.

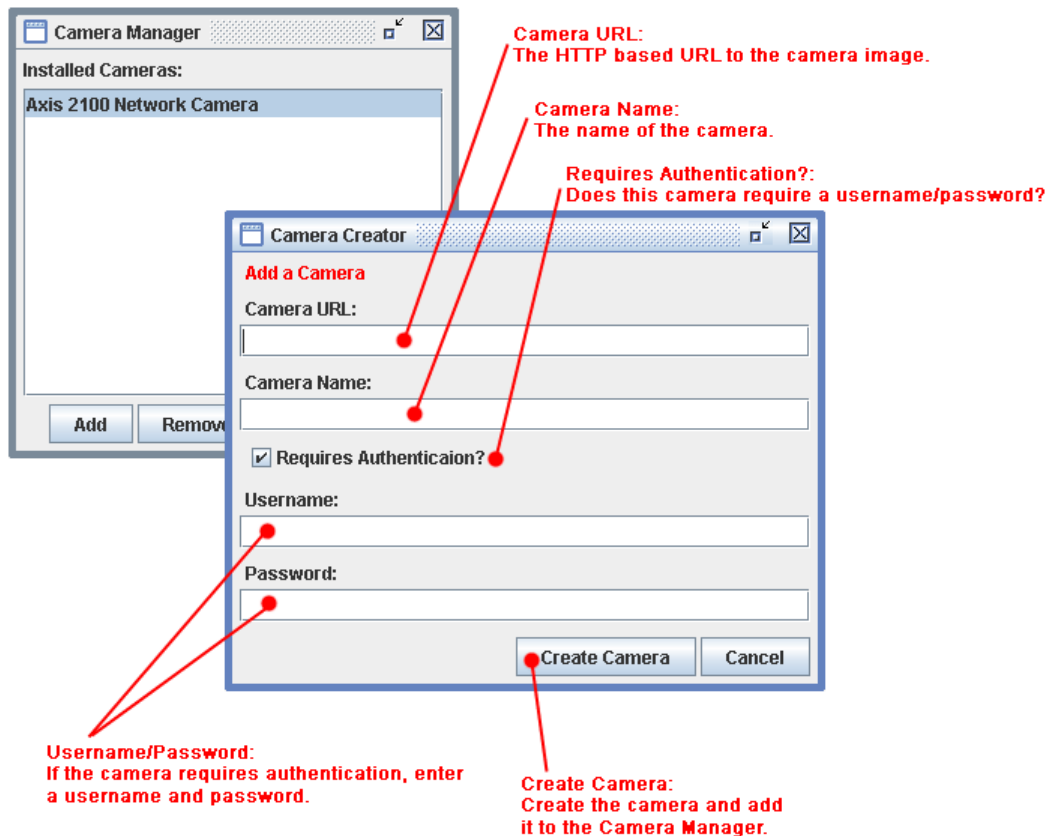


Figure V.G.4.1. The Silhouette Camera Creator.

To add a camera to the Camera Manager, the user must provide five important pieces of information to the application:

- **Camera URL** - The URL of the JPEG/GIF camera stream. When viewing the

camera stream, this is the URL the application will connect to. A URL must start with http://. The application structure does not support HTTPS.

- **Camera Name** - Each camera listed in the Camera Manager must have a name. Camera names do not have to be unique, however it is encouraged to assign a unique and meaningful name to each camera.
  
- **Requires Authentication?** - Some cameras are secured, and require a username and password to successfully connect and obtain a live video feed. If your camera requires a username and password, you must check the "Requires Authentication" check-box and provide a username and password in the correct fields.
  
- **Username** - If your camera requires a username, you must enter it in the username box. Blank usernames and passwords are not permitted.
  
- **Password** - If your camera requires a password, you must enter it in the password box. Blank passwords are not permitted.

Once all information is entered in the correct fields, click the "Create Camera" button to add it to the Camera Manager. When cameras are added, removed, or modified using the Camera Manager, the connection details are saved locally on the host computer. Therefore, the user is not required to re-enter all camera connectivity information each time the application is opened. Each available camera is automatically loaded into the Camera Manager upon startup.

#### **V.G.5            The Camera Editor - Editing a Camera**

On occasion, a user may want to edit a camera's connectivity information. For example, a user may want to change the Camera Name. The Camera Editor

allows a user to modify the camera URL, camera name, camera username, and camera password. As shown in Figure V.G.5.1 below, one can edit a camera's information by selecting a camera from the Camera Manager, and clicking the "Edit" button. This will open the Camera Editor dialog window. When the user has finished making all changes, they must click the "Save" button.

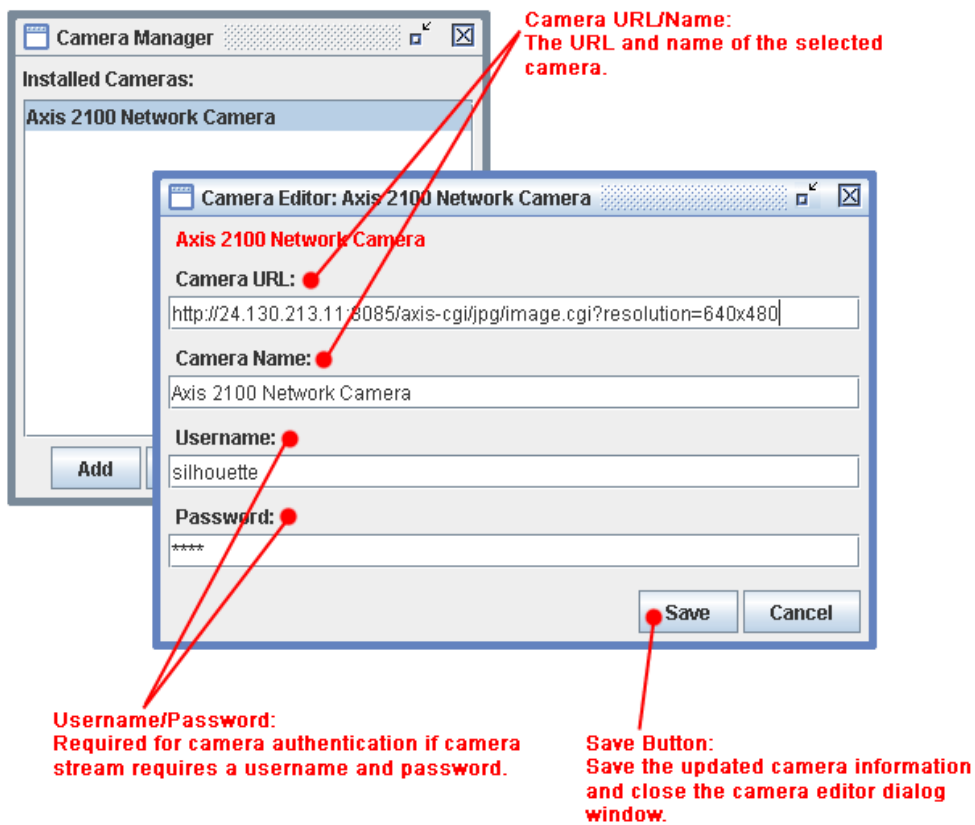


Figure V.G.5.1. The Silhouette Camera Editor.

## V.G.6 Viewing a Camera Stream

Viewing a camera stream is very simple once a camera has been added to the Camera Manager. To view a camera stream, select a camera from the Camera

Manager a click the “View” button. This will spawn a camera view window, and the shape selection tool. The camera view window, as shown in Figure V.G.6.1 below, contains multiple controls designed to help the user manage the camera they are viewing.

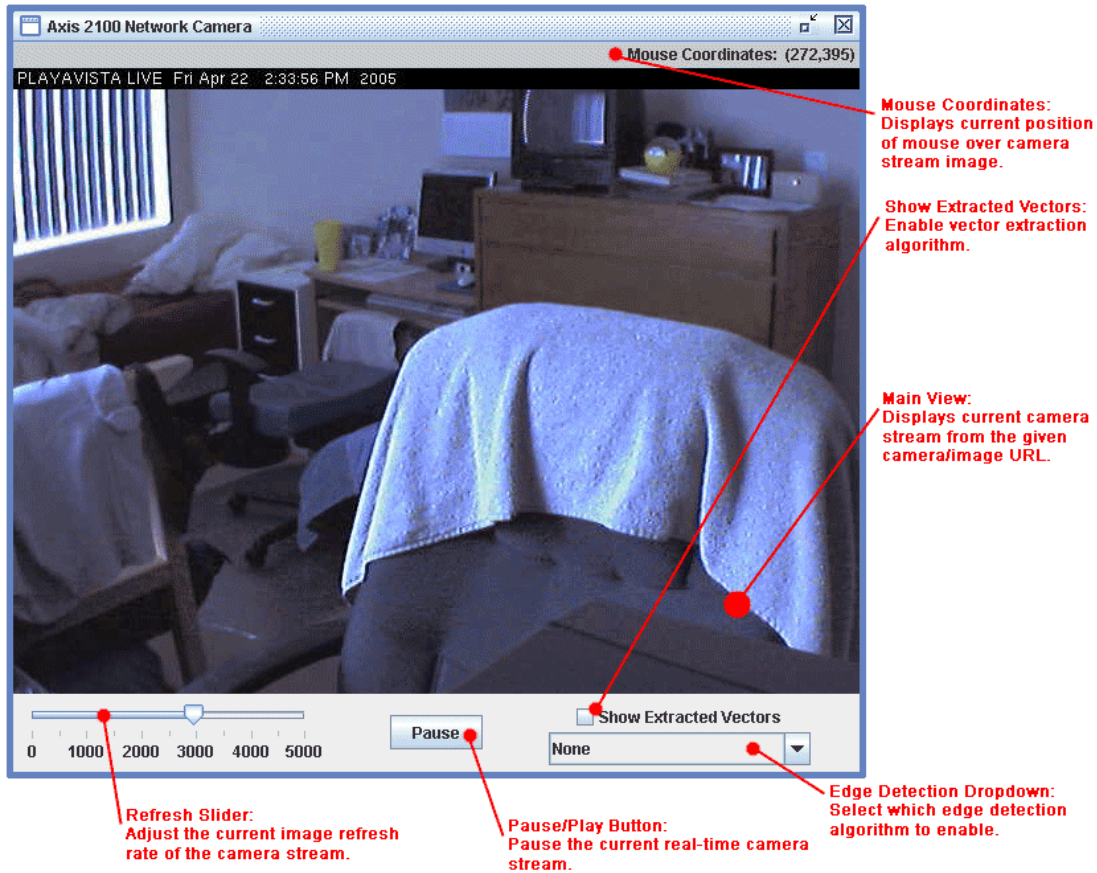


Figure V.G.6.1. The camera view window and controls.

### V.G.7 The Shape Selection Tool

Scanning for shapes is a key component of the Silhouette Project. To enable shape recognition, the user must click the button corresponding to the shape they

would like to search for in the shape selection window. For example, if the user would like to scan for circles, they must click the “Circle” button in the Shape Selection tool. The Shape Selection Tool/Window is shown in Figure V.G.7.1 below.

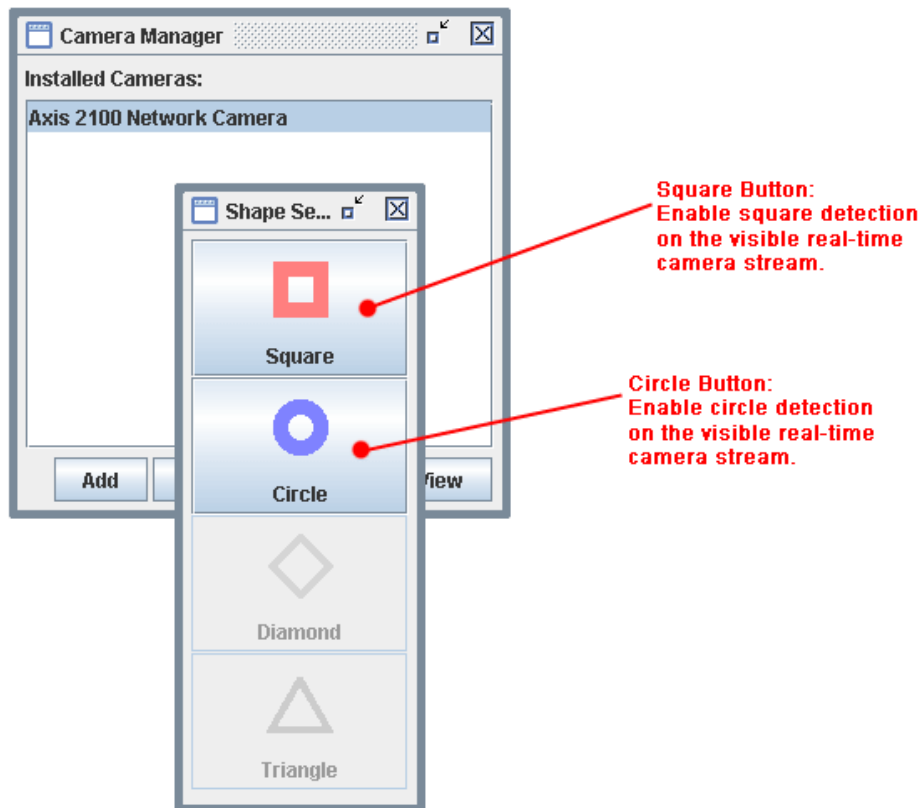


Figure V.G.7.1. The Silhouette Shape Selection Tool.

### V.G.8 Enabling Edge Detection and Shape Recognition

Enabling edge detection and shape recognition is a quick three-step process. First, the user must ensure that they have added a camera to the Camera Manager

and is successfully viewing the live feed from the camera. Second, the user must select "Sobel Edge Detection" from the Edge Detection drop-down box in the View Camera Window (see Section V.G.6, Viewing a Camera Stream). Third, the user must select which shape they'd like to scan for using the Shape Selection Tool.

Due to the intense CPU requirements of edge detection and shape recognition, the user can only scan for one shape at a time. Figure V.G.8.1 below shows the full application desktop with an open camera stream.

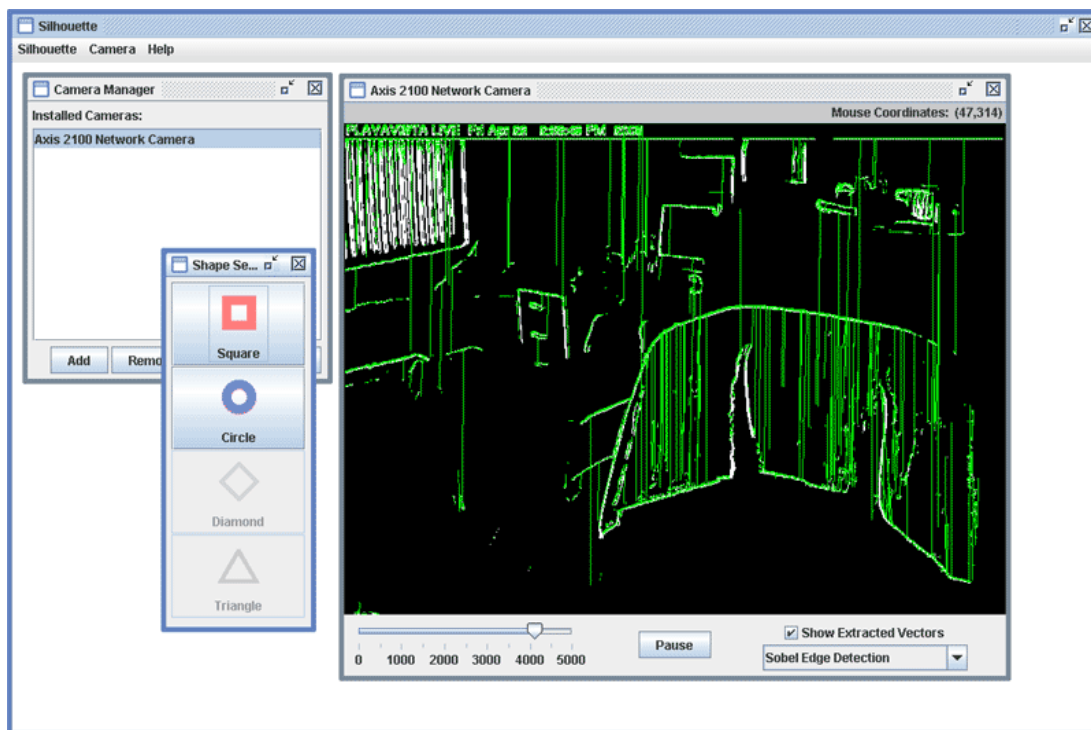


Figure V.G.8.1. Main application desktop with an open camera stream and visible vectors (in green).