System Requirements Specification

Electric-Eye

Nichelous Herndon
Adis Kovacevic
Zack Davis
Juan Sifuentes
Rommel Alonzo
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Revision History</td>
<td>7</td>
</tr>
<tr>
<td>1. Product Concept</td>
<td>8</td>
</tr>
<tr>
<td>1.1 Purpose and Use</td>
<td>8</td>
</tr>
<tr>
<td>1.2 Intended Audience</td>
<td>8</td>
</tr>
<tr>
<td>2. Product Description and Function Overview</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Features and Functions</td>
<td>10</td>
</tr>
<tr>
<td>2.2 External Inputs and Outputs</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Product Interfaces</td>
<td>10</td>
</tr>
<tr>
<td>3. Customer Requirements</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Robot Operating System (ROS) Compatible</td>
<td>11</td>
</tr>
<tr>
<td>3.2 Mapping Operations</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Localization Operations</td>
<td>12</td>
</tr>
<tr>
<td>3.4 Operate Under Vibrations</td>
<td>12</td>
</tr>
<tr>
<td>3.5 Power Requirement</td>
<td>13</td>
</tr>
<tr>
<td>3.6 Module Sleep Command</td>
<td>13</td>
</tr>
<tr>
<td>3.7 RGB-D Camera</td>
<td>14</td>
</tr>
<tr>
<td>3.8 On-board Storage</td>
<td>14</td>
</tr>
<tr>
<td>4. Packaging Requirements</td>
<td>15</td>
</tr>
<tr>
<td>4.1 Single Case Housing</td>
<td>15</td>
</tr>
<tr>
<td>4.2 Automatic Rotating Dome (ARD) Attachment</td>
<td>15</td>
</tr>
<tr>
<td>4.3 Physical Mounting</td>
<td>16</td>
</tr>
<tr>
<td>5. Performance Requirements</td>
<td>17</td>
</tr>
</tbody>
</table>
5.1 Response Time
5.2 Processing Time
5.3 Operating Temperature

6. Safety Requirements
6.1 Heat Safety
6.2 Power Surge Protection
6.3 No Sharp Edges

7. Maintenance and Support Requirements
7.1 Code Documentation
7.2 Testing
7.3 Maintenance

8. Other Requirements
8.1 Documentation in American English
8.2 Cost Effective
8.3 Mode Switch

9. Acceptance Criteria
9.1 Verify ROS Compatible Communication
9.2 Verify Mapping and Localization
9.3 Verify Power Usage
9.4 Product Packaging Verification

10. Use Cases
10.1 Operating Mode
10.2 Mapping Mode
11. Feasibility Assessment

11.1 Scope Analysis
11.2 Research
11.3 Technical Analysis
11.4 Cost Analysis
11.5 Resource Analysis
11.6 Schedule Analysis

12. Future Items

12.1 Packaging Requirement: 4.2 Automated Rotating Dome (ARD)

13. Glossary

13.1 Robot Operating System (ROS)
13.2 Point Cloud Library (PCL)
13.3 Transformation Matrix (6DOF)
13.4 Priority Levels

Document Last Updated on 11/8/2012
List of Figures

Figure 1—1: Product Concept .................................................................................................................. 9

Figure 10—1: Operating Mode of the Electric Eye .................................................................................. 28

Figure 10—2: Initiating Mapping Mode ................................................................................................ 29
List of Tables

Table 9-1: ROS Compatible Communication Requirements .......................................................... 25
Table 9-2: Mapping and Localization Requirements ........................................................................ 26
Table 9-3: Power Usage Requirements ......................................................................................... 26
Table 9-4: Product Packaging Requirements ............................................................................... 27
Table 11-1: Skills Assessment ....................................................................................................... 32
Table 11-2: Function Points .......................................................................................................... 33
Table 11-3: Jones’ First Order Influence Multipliers ...................................................................... 34
Table 11-4: Effort Estimation based on Jones’ First Order ............................................................ 35
Table 11-5: COCOMO Estimation .................................................................................................. 35
## Document Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Revision Date</th>
<th>Description</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>9/26/2012</td>
<td>Initial First Draft Start</td>
<td>Compiling all known subsections</td>
</tr>
<tr>
<td>0.2</td>
<td>10/6/2012</td>
<td>First Draft Finish</td>
<td>Finishing all incomplete subsections</td>
</tr>
<tr>
<td>0.3</td>
<td>10/9/2012</td>
<td>Finishing 11.6</td>
<td>Finished section 11.6</td>
</tr>
<tr>
<td>0.4</td>
<td>10/29/2012</td>
<td>Final Draft Revisions</td>
<td>Beginning our final draft revisions</td>
</tr>
<tr>
<td>1.0</td>
<td>11/8/2012</td>
<td>Baseline SRS Revision</td>
<td>Making final changes to SRS</td>
</tr>
</tbody>
</table>
1. Product Concept

This section describes the purpose, use and intended user audience for the Electric-Eye. The Electric-Eye is a module that will map a room in 3D offline and then perform localization when mounted onto a robot. Users of this product will be able to integrate this product into their robots that use the robot operating system (ROS) and use the features provided by the module.

1.1 Purpose and Use

Many robot designers wish for their robots to operate at peak performance. Sometimes, this means that these designers are forced to become experts in fields that drag them away from the real purpose of designing robots. The Electric-Eye is meant to reduce the learning curve that they must go through. Our product is meant to take the worry of learning computer vision techniques applied to the robot so the engineer can spend more time focusing on the robot in general and not its intricacies.

1.2 Intended Audience

Our initial audience was for our first sponsor, Chris McMurrough, who wanted us to design a system that can provide computer vision operations to a compatible robot. As research began into its feasibility, we discovered that our product’s audience can be extended. The Electric-Eye is for engineers who wish to skip the details that go into robot designs. By taking these worries off the engineer, they can focus on more important aspects and making sure their robot works as an entire system.
Figure 1—1: Product Concept
2. Product Description and Function Overview

This section provides the reader with an overview of the primary features and functions that the Electric-Eye will provide. The operational aspects of the product are defined here from the perspective of end users, maintainers and administrators. The key features and functions found in the product, as well as critical user interactions and user interfaces are described in detail.

2.1 Features and Functions

Electric-Eye will have two usage modes. The first mode will be ‘mapping’ mode which will map out an environment before beginning ‘operating’ mode. The second mode will be operating mode, during this mode the module will be attached to the robot and will perform localization so that the robot knows where it is inside of the environment.

The module itself will be a combination of several hardware components. There will be a motherboard component that will have a hard drive connected to it for storing the information gained during mapping mode. The motherboard will also have a RGB-D camera connected via USB port. The Electric-Eye will connect with the robot through an Ethernet port via cat5 connection. This is all shown in the conceptual drawing, Figure 1-1. One of the external sources not shown in the figure is the power source; the module will be receiving power from the robot that it is mounted on.

2.2 External Inputs and Outputs

External inputs for the module will be the RGB-D camera. The frames from the camera will be sent to the module’s memory storage and processed into a 3D map. Afterwards, the module will be able to access the map and perform localization. The module will then publish the transformation matrix with ROS compatible output via Ethernet connection.

2.3 Product Interfaces

There will be an LED indicator light on the module. The light will turn on depending on which mode the module is in. When the module is in mapping mode the indicator light will be blinking green. When the module is in operating mode the indicator light will be solid green. When the module is powered off the light will be turned off.
3. **Customer Requirements**

In order to provide great products to our customers, our sponsor has requested several features to be present. These features are to allow simple integration with any compatible robot design while still provide efficient and accurate localization information.

### 3.1 Robot Operating System (ROS) Compatible

**3.1.1 Description:** The module must communicate with the ROS and send the robot the transformation matrix data via Ethernet.

**3.1.2 Source:** Chris McMurrough

**3.1.3 Constraints:** The System must communicate with ROS with the appropriate ROS protocol.

**3.1.4 Standards:** ROS Protocol

**3.1.5 Priority:** 1 – Critical

### 3.2 Mapping Operations

**3.2.1 Description:** Using locally stored frames, the Electric-Eye will build a world map of the surrounding environment.

**3.2.2 Source:** Chris McMurrough

**3.2.3 Constraints:** The upper limit of the area that the Electric-Eye can map at once is an average sized room.

**3.2.4 Standards:** N/A

**3.2.5 Priority:** 1 – Critical
3.3 Localization Operations

3.3.1 Description: Using the built world map the Electric-Eye must be able to calculate where it is located inside of the map and which direction it is facing or tilting. The data calculated will be a 6DOF.

3.3.2 Source: Chris McMurrrough

3.3.3 Constraints: N/A

3.3.4 Standards: N/A

3.3.5 Priority: 1 – Critical

3.4 Operate Under Vibrations

3.4.1 Description: The module shall be able to operate under an environment where vibrations may occur, such as when the robot travels through rugged terrain.

3.4.2 Source: Chris McMurrrough

3.4.3 Constraints: The physical limitations of the Electric-Eye.

3.4.4 Standards: N/A

3.4.5 Priority: 2 - High
3.5 Power Requirement

3.5.1 **Description:** The Electric-Eye should connect to a robot with a 2-terminal connection.

3.5.2 **Source:** Chris McMurrough

3.5.3 **Constraints:** The module must not draw more than 150 watts of power.

3.5.4 **Standards:** N/A

3.5.5 **Priority:** 1 - Critical

3.6 Module Sleep Command

3.6.1 **Description:** The robot will send a command via ROS to the module that will put the Electric-Eye in a low power mode.

3.6.2 **Source:** Chris McMurrough

3.6.3 **Constraints:** N/A

3.6.4 **Standards:** N/A

3.6.5 **Priority:** 3 - Moderate
3.7 **RGB-D Camera**

3.7.1 **Description:** The module will have a RGB-D Camera. The camera will read in frames from the surrounding environment.

3.7.2 **Source:** Chris McMurrough

3.7.3 **Constraints:** The camera must be USB powered.

3.7.4 **Standards:** N/A

3.7.5 **Priority:** 1 - Critical

3.8 **On-board Storage**

3.8.1 **Description:** The module will be able to store frames read in from the RGB-D camera.

3.8.2 **Source:** Chris McMurrough

3.8.3 **Constraints:** The minimum capacity of the storage device will be 8GB.

3.8.4 **Standards:** N/A

3.8.5 **Priority:** 1 – Critical
4. Packaging Requirements

Since our product is required to be connected to the robot, an analysis of how the module is physically built is also required. This is to provide the end-user with a simple implementation of our product, while still providing efficient and accurate localization information to the connected robot.

4.1 Single Case Housing

4.1.1 Description: The Electric-Eye must be placed inside of a secure casing.

4.1.2 Source: Chris McMurrough

4.1.3 Constraints: The casing must be able to hold all necessary hardware components and must provide minimal interference with the robot.

4.1.4 Standards: N/A

4.1.5 Priority: 1 - Critical

4.2 Automatic Rotating Dome (ARD) Attachment

4.2.1 Description: The Electric Eye’s ARD can be attached along with the module. The attached add-on will allow the camera to rotate and get an all-around view of the environment.

4.2.2 Source: Power Vision

4.2.3 Constraints: N/A

4.2.4 Standards: N/A

4.2.5 Priority: 5 – Future
4.3 Physical Mounting

4.3.1 Description: The Electric Eye will be in a simple shape that facilitates easy mounting.

4.3.2 Source: Power Vision

4.3.3 Constraints: The method of mounting the Electric-Eye is up to the customer.

4.3.4 Standards: N/A

4.3.5 Priority: 1 – Critical
5. Performance Requirements

The following analysis provides a list of requirements that will allow our product to produce results in a timely manner. Performance requirements include calculation time, response time and efficiency.

5.1 Response Time

5.1.1 Description: The module must respond to ROS commands and send transformation matrix data in a timely manner.

5.1.2 Source: Chris McMurrough

5.1.3 Constraints: N/A

5.1.4 Standards: CAT 5, ROS Protocol

5.1.5 Priority: 2 - High

5.2 Processing Time

5.2.1 Description: The module must be able to process the frames read in from the camera and construct a world map in a timely manner.

5.2.2 Source: Chris McMurrough

5.2.3 Constraints: The CPU processing power will be the main limiting factor.

5.2.4 Standards: N/A

5.2.5 Priority: 3 – Moderate
5.3 Operating Temperature

5.3.1 Description: The module must be able to operate within a reasonable temperature range. Any use of the Electric-Eye outside of this temperature range is not recommended by Power Vision and is not guaranteed to work.

5.3.2 Source: Power Vision

5.3.3 Constraints: Operating temperatures will be greater than 50°F and less than 100°F.

5.3.4 Standards: N/A

5.3.5 Priority: 3 – Moderate
6. Safety Requirements

This section describes the safety requirements for the Electric-Eye module.

6.1 Heat Safety

6.1.1 Description: The module must not reach a dangerous level of heat; if the module overheats it will automatically shut off.

6.1.2 Source: Power Vision

6.1.3 Constraints: This is determined by the quality of the hardware purchased.

6.1.4 Standards: N/A

6.1.5 Priority: 1 – Critical

6.2 Power Surge Protection

6.2.1 Description: Helps keep the entire module system from starting small fires due to power complications.

6.2.2 Source: Chris McMurrrough

6.2.3 Constraints: The amount of power the module consumes.

6.2.4 Standards: N/A

6.2.5 Priority: 2 - High
6.3 No Sharp Edges

6.3.1 Description: The module must not have any sharp edges.

6.3.2 Source: Chris McMurrough

6.3.3 Constraints: The packaging material used will affect this requirement.

6.3.4 Standards: N/A

6.3.5 Priority: 2 - High
7. **Maintenance and Support Requirements**

This section establishes the maintenance and support requirements for Electric-Eye. This includes the code documentation, testing and maintenance of the module.

### 7.1 Code Documentation

**7.1.1 Description:** Electric-Eye’s source code will be well documented to allow future development teams, with programming experience, to upgrade and add to the module’s program.

**7.1.2 Source:** Power Vision

**7.1.3 Constraints:** N/A

**7.1.4 Standards:** N/A

**7.1.5 Priority:** 2 - High

### 7.2 Testing

**7.2.1 Description:** The Electric Eye will be tested to ensure all the required functionality is present and working correctly.

**7.2.2 Source:** Power Vision

**7.2.3 Constraints:** N/A

**7.2.4 Standards:** N/A

**7.2.5 Priority:** 1 - Critical
7.3 Maintenance

7.3.1 Description: Power Vision will not be available to provide maintenance after May 2013.

7.3.2 Source: Power Vision

7.3.3 Constraints: N/A

7.3.4 Standards: N/A

7.3.5 Priority: 1 – Critical
8. Other Requirements

This section will provide requirements for the Electric-Eye that do not pertain to the previous requirements categories above. These requirements include documentation standards, cost effectiveness and the ability to switch between modes.

8.1 Documentation in American English

8.1.1 Description: All documentation on the Electric-Eye will be done in American English.

8.1.2 Source: Power Vision

8.1.3 Constraints: N/A

8.1.4 Standards: N/A

8.1.5 Priority: 1 – Critical

8.2 Cost Effective

8.2.1 Description: Purchasing Electric-Eye will be more efficient than producing it individually.

8.2.2 Source: Power Vision

8.2.3 Constraints: N/A

8.2.4 Standards: N/A

8.2.5 Priority: 3 - Moderate
8.3 Mode Switch

8.3.1 Description: The Electric-Eye will have a switch that allows the module to switch between mapping mode and operating (localization) mode.

8.3.2 Source: Power Vision

8.3.3 Constrains: N/A

8.3.4 Standards: N/A

8.3.5 Priority: 2 - High
9. Acceptance Criteria

This section describes the acceptance criteria for the Electric-Eye module. Based on the following criteria the sponsor will determine if the product’s requirements are complete or incomplete.

9.1 Verify ROS Compatible Communication

9.1.1 Requirement(s) addressed:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>ROS Compatibility</td>
<td>The module must communicate with the ROS and send the robot the transformation matrix data via Ethernet.</td>
</tr>
<tr>
<td>3.6</td>
<td>Module Sleep Command</td>
<td>The robot will send a command via ROS to the module that will put the Electric-Eye in a low power mode.</td>
</tr>
<tr>
<td>5.1</td>
<td>Response Time</td>
<td>The module must respond to ROS commands and send transformation matrix data in a timely manner.</td>
</tr>
</tbody>
</table>

Table 9-1: ROS Compatible Communication Requirements

9.1.2 Verification Procedure: The Electric-Eye module will have a testing program that sends a message to ROS and receives an acknowledgment message to verify the module is successfully communicating with ROS.
9.2 Verify Mapping and Localization

9.2.1 Requirement(s) addressed:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>Mapping Operations</td>
<td>Using locally stored frames, the Electric-Eye will build a world map of the surrounding environment.</td>
</tr>
<tr>
<td>3.3</td>
<td>Localization Operations</td>
<td>Using the built world map the Electric-Eye must be able to calculate where it is located inside of the map and which direction it is facing or tilting. The data calculated will be a 6DOF.</td>
</tr>
</tbody>
</table>

Table 9-2: Mapping and Localization Requirements

9.2.2 Verification Procedure: Power Vision will visually inspect that the mapping and localization is functioning correctly based on the robot's response and movements.

9.3 Verify Power Usage

9.3.1 Requirement(s) addressed:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>Power Requirement</td>
<td>The Electric-Eye should connect to a robot with a 2-terminal connection.</td>
</tr>
</tbody>
</table>

Table 9-3: Power Usage Requirements

9.3.2 Verification Procedure: Test power consumption at peak performance and verify that it uses less than 150 watts of power.
9.4 Product Packaging Verification

9.4.1 Requirement(s) addressed:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>RGB-D Camera</td>
<td>The module will have a RGB-D Camera. The camera will read in frames from the surrounding environment.</td>
</tr>
<tr>
<td>3.8</td>
<td>On-board storage</td>
<td>The module will be able to store frames read in from the RGB-D camera.</td>
</tr>
<tr>
<td>4.1</td>
<td>Single Case Housing</td>
<td>The Electric-Eye must be placed inside of a secure casing.</td>
</tr>
</tbody>
</table>

Table 9-4: Product Packaging Requirements

9.4.2 Verification Process: Power Vision will verify that all components of the module are included and packaged securely in the module’s case.
10. Use Cases

This section will simulate and describe scenarios that the module may encounter. These scenarios will test various aspects of the Electric-Eye including but not limited to: Performance, Compatibility, and Resource Management.

10.1 Operating Mode

10.1.1 Scenario: The module sends the transformation matrix to ROS periodically. At the same time it listens for ROS commands.

10.1.2 Actor(s): ROS

![Figure 10—1: Operating Mode of the Electric Eye](image)
10.2 Mapping Mode

10.2.1 Scenario: A user turns on the module and puts it into mapping mode. After the user is done mapping the room, the user will turn off mapping mode. User can turn off the module.

10.2.2 Actor(s): End User

Figure 10—2: Initiating Mapping Mode
11. Feasibility Assessment

The purpose of this section is to describe and assess the feasibility of the Electric-Eye module. This section includes multiple different project analyses that include scope, research, technical, resource, schedule and cost analysis. In summary, the Electric-Eye module is feasible and will be completed by team Power Vision by the end of Senior Design 2.

11.1 Scope Analysis

Based on our skills, time, budget, and information resources, we predict that we can get a working prototype within our two semester period. Our main challenge will be to learn how to utilize ROS, analyze and apply mapping/localization algorithms and finally implement communication between different hardware to transfer localization data. We do not have the current skills but we do have to knowledge and resources to learn how to implement all these features. Our main focus will be towards requirements concerning ROS compatibility, mapping operations, and localization operations.

11.2 Research

Our sponsor, Chris McMurrough, has given us guidance over what is required to successfully complete this project. This includes suggested hardware, software libraries, algorithms, and possible risks. The hardware required for this project is available. Further research has also shown that most of the software needed to complete this project has already been implemented and is openly available. We have deducted that our future research priority will be analyzing the required complex mapping and localizing algorithms.

11.3 Technical Analysis

Technical aspects of our project will include hardware design, algorithm implementation, and programming language implementation.

The nature of implementing mapping and localizing algorithms are complex and will most likely be the most time consuming aspect of this project. Implementing these algorithms will require analyzing and correctly executing them under a limited computational power environment. We have two computer science undergraduates that are familiar with these computer algorithms.

Communicating with the robot will require using ROS and running the module using C++ or Python programming languages. The ROS language has certain procedures that must be learned as well. We
are familiar with C++ and only need to use ROS to communicate messages between the robot and the module.

The hardware aspect has been simplified with the availability of the necessary components. This allows us to concentrate more on how to best fit the physical structure and provide the most cost effective hardware.

### 11.4 Cost Analysis

Based on sponsor and team judgment, we predict that we will be able to complete this project with our $800 budget. Most of the hardware is available to buy and in several specifications. This gives us a lot of flexibility in terms of choosing between power and cost effectiveness. Overall, our budget will determine how powerful our module will execute.

### 11.5 Resource Analysis

Our team consists of 2 computer scientists, 1 software engineer, and 2 computer engineers. Even though each of us has skills that complement each other, we also have certain skills that set us apart. Some of our individual skills include one member who has a history with robot designing and engineering. Also, another member specializes in 3D computer vision techniques that can give us insight into several algorithms that we may be using. As the year progresses, our team will be further developing the required skills to complete this project including PCL, ROS and basic networking.
Table 11-1: Skills Assessment

11.6 Schedule Analysis

Our team used three methods of time estimation to create a feasible schedule that can ensure all requirements are met and completed in time. Using function points, the COCOMO Model, and Jones First Order Estimation we have estimated the size and effort needed to complete this project by the given deadline of May 2013.
### 11.6.1 Function Points

<table>
<thead>
<tr>
<th>Program Characteristics</th>
<th>Low Complexity</th>
<th>Medium Complexity</th>
<th>High Complexity</th>
<th>Function Point Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Inputs</td>
<td>$2 \times 3$</td>
<td>$2 \times 4$</td>
<td>$2 \times 6$</td>
<td>26</td>
</tr>
<tr>
<td>Number of Outputs</td>
<td>$0 \times 4$</td>
<td>$3 \times 5$</td>
<td>$1 \times 7$</td>
<td>22</td>
</tr>
<tr>
<td>Inquiries</td>
<td>$0 \times 3$</td>
<td>$1 \times 4$</td>
<td>$0 \times 6$</td>
<td>4</td>
</tr>
<tr>
<td>Logical Internal Files</td>
<td>$0 \times 7$</td>
<td>$0 \times 10$</td>
<td>$1 \times 15$</td>
<td>15</td>
</tr>
<tr>
<td>External Interface Files</td>
<td>$0 \times 5$</td>
<td>$0 \times 7$</td>
<td>$0 \times 10$</td>
<td>0</td>
</tr>
<tr>
<td>UAF Total</td>
<td></td>
<td></td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>Adjustment Factor</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>67</td>
</tr>
</tbody>
</table>

*Table 11-2: Function Points*
### Jones First Order Estimation Influence Multiplier

<table>
<thead>
<tr>
<th>Category</th>
<th>Influence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Communications</td>
<td>1</td>
</tr>
<tr>
<td>Distributed Data Processing</td>
<td>0</td>
</tr>
<tr>
<td>Performance</td>
<td>1</td>
</tr>
<tr>
<td>Heavily Used Configuration</td>
<td>1</td>
</tr>
<tr>
<td>Transaction Rate</td>
<td>1</td>
</tr>
<tr>
<td>Online Data Entry</td>
<td>0</td>
</tr>
<tr>
<td>End-User Efficiency</td>
<td>1</td>
</tr>
<tr>
<td>Online Update</td>
<td>0</td>
</tr>
<tr>
<td>Complex Processing</td>
<td>1</td>
</tr>
<tr>
<td>Reusability</td>
<td>1</td>
</tr>
<tr>
<td>Installation Ease</td>
<td>0</td>
</tr>
<tr>
<td>Operational Ease</td>
<td>0</td>
</tr>
<tr>
<td>Multiple Site</td>
<td>0</td>
</tr>
<tr>
<td>Facilitate Change</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>7</strong></td>
</tr>
</tbody>
</table>

Table 11-3: Jones’ First Order Influence Multipliers

**Value Adjustment Factor:** \[0.65 + (0.05 \times 7) = 1.0\]

**Adjusted Function Point:** \[67 \times 1.0 = 67\]
11.6.3 Effort Estimation

<table>
<thead>
<tr>
<th></th>
<th>Best in Class</th>
<th>Average</th>
<th>Worst in Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLOC</td>
<td>67^{43}</td>
<td>67^{45}</td>
<td>67^{48}</td>
</tr>
<tr>
<td>Calendar Months</td>
<td>6.10</td>
<td>6.63</td>
<td>7.53</td>
</tr>
</tbody>
</table>

*Table 11-4: Effort Estimation based on Jones’ First Order*

11.6.3 COCOMO 81 Model

The COCOMO 81 model is a very popular model used in many companies today. We use this model to get another perspective of how much time would be required for project completion.

SLOC: 5000 (rough estimate)  
Software Development Mode: Semi-Detached

- **Product Attributes:**
  - Required Reliability: **Very High**
  - Database Size: **Very Low**
  - Product Complexity: **Very High**

- **Computer Attributes:**
  - Execution Time Constraint: **High**
  - Main Storage Constraint: **High**
  - Virtual Machine Volatility: **Very Low**
  - Computer Turnaround Time: **Very Low**

- **Personnel Attributes:**
  - Analyst Capabilities: **Nominal**
  - Applications Experience: **Nominal**
  - Programmer Capability: **High**
  - Virtual Machine Experience: **Nominal**
  - Programming Language Experience: **High**

- **Project Attributes:**
  - Modern Programming Practices: **Very High**
  - Use of Software Tools: **High**
  - Required Development Schedule: **Very High**

*Table 11-5: COCOMO Estimation*
12. Future Items

This section will list the items that will be implemented in the future. The items listed in this section will most likely not be delivered with the working prototype of the Electric-Eye.

12.1 Packaging Requirement: 4.2 Automated Rotating Dome (ARD)

12.1.1 Description: The dome will be able to rotate the view angle for the camera to provide 3D data from different perspectives.

12.1.2 Constraints: This feature is more luxury than necessity for project completion.
13. Glossary

In this section we will go into further detail about terms that are used throughout this document. Terms such as ROS, PCL and priority levels will be defined here. Reference websites will also be provided for readers that wish for further details.

13.1 Robot Operating System (ROS)

The Robot Operating System, or ROS, is an operating system that can be installed on top of Linux or Microsoft Windows. This operating system provides libraries and tools to help software developers create robot applications. Some of the features that ROS offers are hardware abstraction, device drivers, message-passing and more. More information about ROS can be found at: www.ros.org.

13.2 Point Cloud Library (PCL)

Point Cloud Library, or PCL, is a large scale open project for 2D/3D image and point cloud processing. PCL contains many algorithms that are useful for 3D image processing such as noise filtering. PCL is open source software that is free for commercial and research use. More information about PCL can be found at: www.pointclouds.org.

13.3 Transformation Matrix (6DOF)

The transformation matrix, or 6DOF, is a term used throughout the document. The transformation matrix is a set of data that will be sent to ROS from the Electric-Eye which will include the pitch, roll, yaw, x, y and z data of the current location that the robot is located in respect to the camera.
13.4 Priority Levels

Throughout this document there are many requirements with various different priority numbers attached to them. These numbers also have a brief description of what the number means. This section of the glossary will go more in depth on the different priority levels:

13.4.1 Critical – A critical level priority means that Power Vision shall successfully implement this feature into the Electric-Eye.

13.4.2 High - A high level priority means that the requirement shall be implemented but it is not critical.

13.4.3 Moderate - A moderate level priority means that the feature will most likely be implemented.

13.4.4 Low - A low priority means that the feature will be implemented if we have time in our schedule.

13.4.5 Future - A future priority level means that the feature will not be implemented in the prototype delivered unless all other priorities have been implemented and we have time.