Determining the probability of detection for specified probability of false alarm and signal-to-noise ratio.

Software Requirements Specification

1.0

23 January 2018

Group L

Naman, Surya, Nitin, Ayush, Dharna

Prepared for

CS 258 Software Engineering

Spring 2018

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 23-01-18 | 1.0 | Ayush Prasad | Initial Proposal |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction i

1.1 Purpose i

1.2 Scope i

1.3 Definitions, Acronyms, and Abbreviations i

1.4 References i

1.5 Overview i

2. General Description ii

2.1 Product Perspective ii

2.2 Product Functions ii

2.3 User Characteristics ii

2.4 General Constraints ii

2.5 Assumptions and Dependencies ii

2.6 Operating Environment ii

2.7 Technology to be used ii

3. Specific Requirements ii

3.1 External Interface Requirements iii

3.1.1 User Interfaces iii

3.1.2 Hardware Interfaces iii

3.1.3 Software Interfaces iii

3.1.4 Communications Interfaces iii

3.2 Functional Requirements iii

3.3 Non-Functional Requirements iii

3.4 Design Constraints iv

3.5 Logical Database Requirements iv

# 1. Introduction

## The client requires two softwares to be developed, of which the first one presents a graphical representation based on three critical parameters which are the probability of detection, the probability of false alarm, signal-to-noise ratio. And the other software can be used to detect targets from a set of infrared images and later track the target.

## 1.1 Purpose

## The purpose of this SRS is to provide the end user the details about the operation of the products and all the specifications of the products. It is a comprehensive guide for anybody who is interested in knowing about the software in detail and wishes to pursue developing them.

## 1.2 Scope

The softwares to be developed are PoD and Target Tracking System.

The PoD will plot a graph between probability of detection and signal-to-noise ratio having different probability of false alarm. The client should be able to get the probability of detection corresponding to a specified probability of false alarm and signal-to-noise ratio and vice versa.

Target Tracking System deals with measurement extraction and their processing in the case where a detection is a set of cells or pixels in which received signal indicates the existence of a target. It is used to detect targets using centroid estimation technique and segmentation. The targets are tracked based on the tracking filters designed using white noise measurement and autocorrelation.

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| Probability of detection | The probability that the search object will be detected under given conditions if it is in the area searched. |
| False alarm | An erroneous radar target detection decision caused by noise or other interference exceeding the detection threshold. |
| Signal-to-noise ratio (SNR) | Signal-to-noise ratio (abbreviated **SNR** or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is **defined** as the ratio of signal power to the noise power, often expressed in decibels. |
| MATLAB | **MATLAB** is a fourth-generation programming language and numerical analysis environment. Uses for **MATLAB** include matrix calculations, developing and running algorithms, creating user interfaces (UI) and data visualization. |
| Image Segmentation | In computer vision, **image segmentation** is the process of partitioning a digital **image** into multiple segments (sets of pixels, also known as super-pixels). The goal of **segmentation** is to simplify and/or change the representation of an **image** into something that is more meaningful and easier to analyse. |
| Centroid Tracker | Tracker which tracks the calculated centroids of clusters. |
| Forward looking infrared imaging sensor (FLIR sensor) | It senses infrared radiation and are installed in forward looking infrared cameras. |
| Autocorrelation | Correlation of a signal with a delayed copy of itself as a function of delay. |

## 

## 1.4 References

*Following are the resources used :*

1. *INTRODUCTION TO ELECTRO-OPTICAL IMAGING AND TRACKING SYSTEMS by Khalil Seyrafi and S.A. Hovanessian*
2. *Tracking and Data Fusion: A Handbook of Algorithms by Peter K.Willett, Xin Tian, and Yaakov Bar-Shalom*
3. *Wikipedia*
4. *<https://www.twi-global.com/capabilities/integrity-management/non-destructive-testing/probability-of-detection/>*
5. *<https://definedterm.com/>*
6. *<to be updated soon>*

## 1.5 Overview

The rest of the SRS contains all the details of the projects. It starts with the general description of the project and all of its products. It also specifies the factors that affect the products. Then followed by specific requirements in detailed terms which describe all the software and hardware requirements of the products and description of the different system interfaces.

# 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

*This subsection of the SRS puts the product into perspective with other related products or*

*projects. (See the IEEE Guide to SRS for more details).*

## 2.2 Product Functions

This subsection of the SRS should provide a summary of the functions that the software will perform.

## 2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

## 2.4 General Constraints

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

## 2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

## 2.6 Operating Environment

**2.7 Technology to be used**

# 3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.

Each requirement in this section should be:

* Correct
* Traceable (both forward and backward to prior/future artifacts)
* Unambiguous
* Verifiable (i.e., testable)
* Prioritized (with respect to importance and/or stability)
* Complete
* Consistent
* Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

3.2.1.5 Error Handling

### 3.2.2 <Functional Requirement or Feature #2>

…

## 3.3 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc. ).

### 3.3.1 Performance

### 3.3.2 Reliability

### 3.3.3 Availability

### 3.3.4 Security

### 3.3.5 Maintainability

### 3.3.6 Portability

## 3.4 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

## 3.5 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.