Software Requirements Specification for Image Feature Correspondences for Camera Calibration

Kiran Singh

2025-02-24

Contents

Revision History

Date	Version	Notes
2022-02-007	1.0	Initial Release

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Not applicable, as the project is entirely based on image processing properties in software such as pixel intensity and image resolution. The scope of this software is not to utilize properties such as camera extrinsics themselves which do use SI units for distance.

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with robotic vision literature. The symbols are listed in alphabetical order.

symbol	**unit**	**description**
$\mathbf{A}$	3×4 Matrix	SE(3) equivalent transformation of a camera with respect to the target feature frame
$\mathbf{B}$	3×4 Matrix	SE(3) equivalent transformation of the end-effector in the robot-base frame
c	Unitless	Camera instance
d_k	Unitless	Indexed pixel or binary descriptor
$d_{Hamming}$	Unitless	Hamming Distance from bit-wise comparison of binary objects
$\mathcal{D}_{i,j}$	$n \times 2$ matrix	Detected keypoint for pose i and camera k
i	Unitless	Robot pose instance
j	Unitless	camera instance
$\mathcal{I}_{i,j}$	$u \times v$ matrix	Image matrix
I_k	Unitless	Pixel intensity
k	Unitless	Target feature instance
t	Initless	Pixel intensity threshold
u	Unitless	horizontal pixel position
v	Unitless	horizontal pixel position
X	$u \times v$ matrix	2-dimensional representation of image coordinates
$\mathbf{X}$	3×4 Matrix	SE(3) equivalent transformation of the target feature in the robot-base frame
$\mathbf{Y}$	3×4 Matrix	SE(3) equivalent transformation of the camera in the end-effector frame

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
BRIEF	Binary Robust Independent Elementary Features
DD	Data Definition
IFC	Image Feature Correspondences for Camera Calibration
FAST	Features from Accelerated Segment Test
FOV	Field-of-View
GD	General Definition
GS	Goal Statement
HERW	Hand-Eye Robot-World Formulation
IM	Instance Model
LC	Likely Change
ORB	Oriented FAST and Rotated BRIEF
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
SURF	Speeded-Up Robust Features
TM	Theoretical Model
VnV	Verification and Validation
XOR	Bitwise Exclusive-OR operation

1.4 Mathematical Notation

Unless specified otherwise, the following notation should be assumed to be the standard convention for the SRS document.

- Matrixes are capitalized and are bolded, i.e. $\mathbf{X}, \mathbf{Y}$
- Column vectors are lowercase and are bolded, i.e. $\mathbf{s}, \mathbf{t}$
- Scalars are lowercase and are not bolded, i.e. a, b

2 Introduction

Camera sensors are a common choice of sensor for many applications in robotics due in part to their low cost and ease of integration. Prior to their use, each camera must be calibrated such that collected data in collected imagery can be aligned with the 3D world. This process is essential to prepare the system so that imagery data can be correctly captured and processed for downstream operations.

Camera calibration consists of two aspects; intrinsic calibration and extrinsic calibration. Intrinsic calibration focuses on mapping the 2D camera image to the 3D camera frame, Extrinsic calibration converts the 3D camera frame to a global world frame. Extrinsic calibration is of significant interest as operators may need to reposition cameras on a robotic platforms for any number of operational needs.

The following section outlines the Software Requirements Specification (SRS) for a calibration algorithm that calculates the extrinsic parameters for a multi-camera robotic platform. The program may be refered to as Image Feature Correspondences, or IFC.

2.1 Purpose of Document

This document is the primary resource for the user to outline the desired characteristics of the user, the required system interfaces, and desired integrated behaviour of the IFC algorithm. The goals and key assumptions of the desired software are outlined, in addition to the required definitions, theoretical models and instance models required to support its development. Specifically, theoretical models are outlined to provide a framework to promote development such that a specific design solution is not imposed at an early stage of development. The SRS is abstract - it bounds what problems need to be solved by the system, rather than how it needs to be achieved.

Following a standard waterfall development model, this document will be used as a stepping to support the development of several additional documents, each of which demonstrates successive growth in the understanding and maturation of the software product. These documents include:

1. The Design Specification: An outline of the architectural decisions that details how the requirements will be realized in the system. This is inclusive of the choice of operating environment, system interfaces with the user and its environment, and the numerical methods that shall be implemented.

2. The Verification and Validation (V&V) Plan: An outline of the specific processess to be used to assess the implementation of the code as developed from the Design Specification. Verification assessments will be used to assess whether the system has been built to the specified requirements from the SRS. Validation tests may also be outlined to ensure that the software correctly addresses the problem as defined in build confidence that the design has satisfied the outlined requirements per 4.1.

2.2 Scope of Requirements

The outlined requirements includes conventional imagery processing algorithms. When supplied with the permissible inputs, the IFC software shall scan imagery data to and identify match candidates amongst each image for various cameras and robot poses.The entire document is written under the assumption that that the imagery scene is free of significant changes in ambient illumination during imagery capture. Camera intrinsics are expected to be known prior to compile time.

2.3 Characteristics of Intended Reader

Reviewers of this document should have a , and a strong understanding of image processing algorithms. A 4th year undergraduate or Master's level course in Computer Vision algorithm is strongly recommended. The reviewer should have an understanding of robot mechanics per a 3rd or 4th year undergraduate course. The developer should also have an background in introductory-level statistics.

2.4 Organization of Document

The remainder of the document uses a top-down structure that outlines, in order, the goals, assumptions, theories, definitions, and instance models. These components are then used to derive the functional and non-functional requirements. Goal statements and assumptions are sequentially distilled into theoretical models, definitions, instance models, and finally to requirements.

The reader can glean value through review of the instance models prior to the theoretical models, as the instance models outlne an operational specification of system behaviour, rather than a descriptive specification. This enables flexibility in what theoretical models are applied as they may be selected as inputs to the systems as a functional program.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Figure 1 depicts the system context. A rectangle represents the IFC software itself, whereas circles depict interactions with stakeholders, namely the user. Arrowheads are used to demonstrate the sequential flow of data between the software system and the environment.

- User Responsibilities:

Figure 1: System Context

  - Provide camera configuration data and imagery data to the system
  - Verify that the input data is contained in the correct data structures
  - Configure the type of feature-handler algorithms to be used with corresponding threshold criteria

- Image Feature Correspondences for Camera Calibration Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  - Assess whether inputs constitute a fully defined physical setup
  - Calculate required outputs
  - Provide a quality metric of confidence in the calculated outputs

This software is not intended for use in safety-critical applications. If it is required for applications that are deemed to be safety-critical, then the software will need to undergo a new review and verification cycle.

3.2 User Characteristics

The end user of the IFC software should have an undergraduate understanding of introductory mechanics of robots, statistics, and computer vision algorithms.

3.3 System Constraints

The IFC software shall be compatible with Python 3.1 libraries, such as OpenCV.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

Image Feature Correspondences for Camera Calibration is intended to evaluate how imagery data from robot-based cameras can can be manipulated to define and align features between separate images in support of downstream operations for extrinsic camera calibration.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- **Features:** Distinctive patterns or structures in an image that are identifiable and useful for matching between images
- **Keypoints:** Specific pixel locations in an image that represent significant and repeatable features.
- **Correspondences:** Pairs of keypoints between two images that represent represent the same real-world point.
- **Extrinsic Parameters:** The transform between the 3D camera frame to the 3D world frame.
- **Intrinsic Parameters:** camera parameters that pertain to the transform of the 2D image plane frame to the 3D camera frame.
- **Hand-eye:** the relation between the robot end-effector to the camera frame
- **Robot-world:** the relation between the robot base frame to the world frame
- **Pose:** refers to the position and orientation of an object, sensor, or robot within a given reference frame.
- **Patch:** A square region of an image of an assigned size.

4.1.2 Physical System Description

The physical system of Image Feature Correspondences for Camera Calibration, as shown in Figure 2, outlines case of a single-camera, single target configuration. This outlines the frames of interest for 4.1.2. This configuration can be extrapolated to the multi-camera, multi-target case, as shown in 3. A representation of equivalent keypoints between images is shown in 4.

PS1: Frames for the robot base, hand, camera, and target landmark
PS2: Known base-to-hand transform.
PS3: Projected camera images from distinct camera poses.

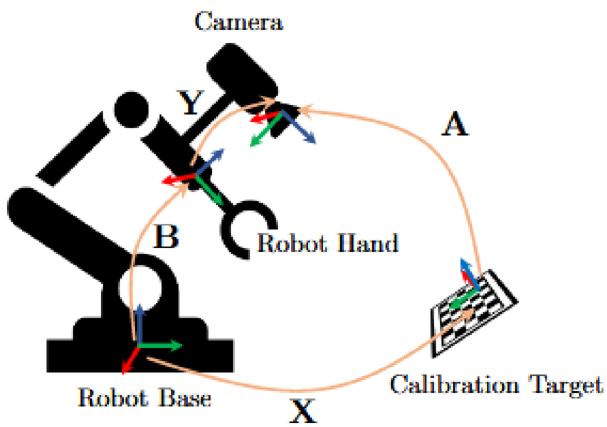


Figure 2: Single-camera robotic manipulator robot-world hand eye configuration. Modified from Wang et al. (2022).

.

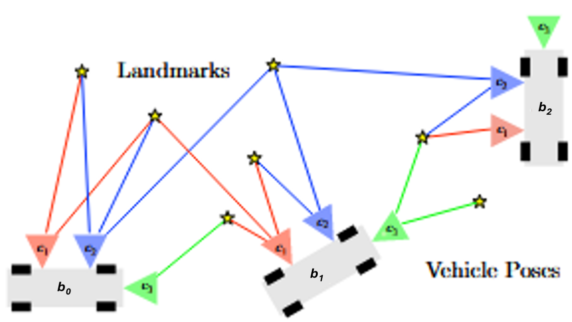


Figure 3: Multi-camera mobile robotic platform. Modified from Kaveti et al. (2024).

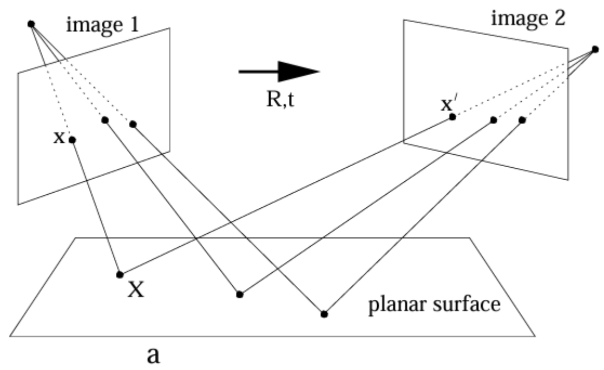


Figure 4: Multi-camera mobile robotic platform. Modified from Hartley and Zisserman (2000).

.

4.1.3 Goal Statements

The system goals follow.

GS1: Define the method(s) used to search for feature correspondences between each image frame.

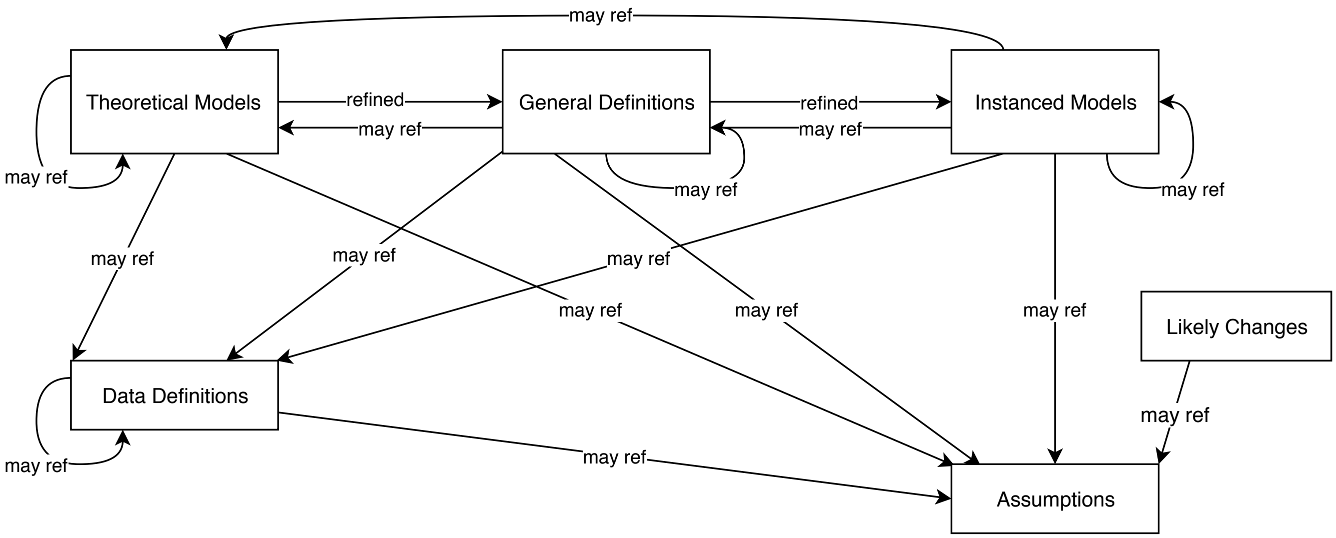
GS2: Define the method(s) used to search for feature correspondences between each image frame.

GS3: Identify a collection of features from each input image frame.

GS4: Identify feature correspondences amongst each input image.

GS5: Generate a report of the identified feature correspondences.

4.2 Solution Characteristics Specification



The instance models that govern Image Feature Correspondences for Camera Calibration are presented in Subsection 4.2.8. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Types

Omitted for Rev 1.0 release.

4.2.2 Scope Decisions

Control of the ambient illumination conditions falls outside the scope of this software. It is the responsibility of the user to verify that the ambient lighting conditions do not change to a significant degree during the image capture process.

4.2.3 Modelling Decisions

Omitted for Rev 1.0 release.

4.2.4 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [TM], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: Imagery shall be provided by at least one camera (GD4).

A2: All supplied imagery is produced by a pinhole model, affine camera (TM2, TM3, GD1).

A3: All imagery will be input as greyscale data (GD1).

A4: The solution is not limited to memory constraints observed in hardware used for real-time applications (TM3).

4.2.5 Theoretical Models

This section focuses on the general equations and laws that Image Feature Correspondences for Camera Calibration is based on.

Number	TM1
Label	**N-Dimensional Gaussian Kernel**
Equation	$G_{N-Dim}(\overrightarrow{x}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\overrightarrow{x}}{2\sigma^2}}$
Description	The above equation represents the distribution of data when defined as a n-dimensional Gaussian Distribution. $\overrightarrow{x}$ is the collection of n-dimensions in the Gaussian distribution. σ is the standard deviation of the Gaussian distribution.
Notes	All variables are unitless.
Source	Chung
Ref. By	GD1
Pre-conditions for TM1:	None
Derivation for TM1:	None

Number	TM2
Label	**Features from Accelerated Segment Test (FAST)**
Equation	$S_{p\to x} = \begin{cases} d, , I_{p\to x} \leq I_p - t & \text{(darker)} \\ s, , I_{p\to x} \leq I_p + t & \text{(similar)} \\ b, , I_p + t \leq I_{p\to x} & \text{(brighter)} \end{cases}$
Description	For the 2D image from an pinhole, affine camera model (A2), x represents the 16 contiguous pixels that surround pixel p within the image plane, or $x \in \{1, ..., 16\}$. $S_{p\to x}$ represents the comparison of pixel intensity for p to x, with an assignment of d, s, or b to represent that pixel p is brighter, darker, or similar to its neighbours. If the pixel is classified as either a b or a d, then the pixel is defined as a keypoint for edge detection.
Notes	All variables are unitless.
Source	Rosten and Drummond (2006)
Ref. By	GD2, LC1
Pre-conditions for TM2:	None
Derivation for TM2:	None

Number	TM3
Label	**Binary Robust Independent Elementary Features (BRIEF)**
Equation	$d_k = \mathbf{1}(I(p_k) < I(q_k))$
Description	For a selected patch within an 2D image from a pinhole camera (A2), and d_k represents the kth bit in a descriptor, pixels p_k and q_k are randomly sampled.
Notes	This theory is computationally expensive and may not be suited to real-time applications (A4).
Source	OpenCV Contributors (2024)
Ref. By	GD3, LC2
Pre-conditions for TM3:	None
Derivation for TM3:	None

Number	TM4
Label	**Hamming Distance**
Equation	$d_{Hamming}(d_{k,\alpha}, d_{k,\beta}) = \sum_{i=0}^{n-1}(d_{k,\alpha} \bigoplus d_{k,\beta})$
Description	The Hamming distance is used to compare two binary numbers by each succesive bit, and return the sum, $d_{Hamming}$, for the binary descriptors α and β. The Hamming Distance does not represent a physical distance. represent a physical distance$i$ represents the index within the total of n descriptors. $\bigoplus$ represents the bitwise exclusive-OR (XOR) operation. Both descriptors are assumed to originate from separate images (A1).
Notes	All variables are unitless.
Source	OpenCV (2021)
Ref. By	IM4, LC3
Pre-conditions for TM4:	None
Derivation for TM4:	None

4.2.6 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**2-Dimensional Gaussian Kernel**
SI Units	Unitless
Equation	$G_{2D}(u, v, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$
Description	G_{2D} represents the gaussian kernel transform that is applied to the 2D greyscale image (A2, A3) given by horizonal pixel u and vertical pixel v, both of which are unitless. σ represents the allowable standard deviation of the two dimensional distribution for a given image. The Gaussian kernel is used to smooth an image to reduce noise amongst the individual pixels. $\vec{x}$ is the collection of n-dimensions in the Gaussian distribution. σ is the standard deviation of the Gaussian distribution.
Source	TM1
Ref. By	IM1

Number	GD2
Label	**FAST Implementation**
Units	Unitless
Equation	$\sum_{k \in x} (\lvert I_k - I(u, v) \rvert > t) \geq N$
Description	The threshold count of $x \in \{1 \ldots 16\}$ is concretely defined as N. $I(u, v)$ represents the pixel intensity of pixel p. I_k represents the grayscale image intensity of the k^{th} pixel in the circle around pixel p. t represents the user-defined threshold to define an allowable range of pixel intensity.
Source	TM2
Ref. By	IM1

Number	GD3
Label	**Rotated Brief**
Units	Unitless
Equation	$d_k = I(p_k^{'}) < I(q_k^{'})$
Description	For a defined patch size around a selected keypoint, pixels p and q are randomly selected. d_k represents the kth bit in a descriptor, pixels $p_k^{'}$ and $q_k^{'}$ are randomly sampled. This ensures that BRIEF can be implemented in a manner that is rotation invariant to an image within a plane.
Source	OpenCV Contributors (2024)
Ref. By	IM3

Detailed derivation of Rotated Brief Transform

p_k and q_k are calculated by successive increments of a 12°. The transform between $p_k^{'}$ and $q_k^{'}$ is outlined below, where θ represents the increase in rotation by a denomination of 12°.

$$\begin{bmatrix} p_k' \\ q_k' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_k \\ q_k \end{bmatrix}$$

4.2.7 Data Definitions

No data definitions are outlined for the current release.

4.2.8 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.7 to replace the abstract symbols in the models identified in Sections 4.2.5 and 4.2.6.

Number	IM1
Label	**Gaussian-Smoothed Image,** $\mathcal{I}'_{i,j}(u,v,\sigma)$
Input	$\mathcal{I}_{i,j}(u,v)$, σ from GD1
Output	$\mathcal{I}'_{i,j}(u,v,\sigma)$ from TM1
Description	$\mathcal{I}'_{i,j}(u,v,\sigma)$ is the smoothed output image. $\mathcal{I}_{i,j}(u,v,\sigma)$ is the unfiltered input image. σ is the standard deviation of the 2D Gaussian Distribution.
Sources	Chung
Ref. By	GS3, GS4,IM2

Number	IM2
Label	**Image Keypoint Detection,** $\mathcal{D}_{i,j}(u,v)$
Input	$\mathcal{I}_{i,j}(u,v,\sigma)$ from IM1, t
Output	$\mathcal{D}_{i,j}(u,v)$, such that D is an n-dimensional 2D matrix of the horizontal and vertical pixel coordinates in the form of (u,v).
Description	$\mathcal{D}_{i,j}(u,v)$ defined by the mapping transform, $S_{p\rightarrow x}$, as outlined in TM2. $\mathcal{I}_{i,j}(u,v,\sigma)$ is the smoothed image (m x n) from IM1. t is the pixel intensity threshold.
Sources	Rosten and Drummond (2006)
Ref. By	GS1, GS3, IM3

Number	IM3
Label	**Produce Feature Descriptors,** D_{bin}
Input	$\mathcal{D}_{i,j}(u, v)$ from IM2, p_s, l_{bin}
Output	D_{bin} as as variable size vector
Description	$\mathcal{D}_{i,j}$ is the vector of pixels (u,v) of identified keypoints, per IM2. p_s is the size of pixel patch to draw samples, as an integer. l_{bin} is the size of the binary descriptor, in bits, per TM3. D_{bin} is the output vector of identified descriptors, per TM3.
Sources	OpenCV Contributors (2024)
Ref. By	GS2, GS4, IM4

Number	IM4
Label	**Inter-Image Descriptor Comparison,** $dist_{Hamming}$
Input	d_a, d_b from IM3
Output	$dist_{Hamming}$
Description	$dist_{Hamming}$ is the sum of bitwise **XOR** binary descriptors as outlined in TM3. d_a, d_b are binary descriptors as defined by IM3.
Sources	OpenCV (2021)
Ref. By	GS4, GS5

4.2.9 Input Data Constraints

Table 2 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 2 are listed in Table **??**.

Table 2: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
$\mathcal{I}_{i,j}(u,v)$	N/A	$0 \leq \mathcal{I}_{i,j}(u,v) \leq 255$	N/A	N/A
σ	$\sigma > 0$	$\sigma > 0$	0.05	N/A
i	$1\imath > 1$	$\leq i \leq 250$	40	N/A
j_{max}	$1 \leq j_{max} \leq 10$	$1 \leq j \leq 10$	3	N/A

(*) $\mathcal{I}_{i,j}(u,v)$ is limited to being an 8-bit unsigned integer

(*) standard deviation, σ, is constrained to be positive and real

(*) the robot pose, i, must be greater than 1, but should be no greater than 250 to avoid excessive runtimes

(*) the total number of cameras, j_{max}, should not exceed 10 as this exceeds the expected operational needs of the system

4.2.10 Properties of a Correct Solution

A correct solution must exhibit high recall. Recall can be described by the comparison of true positive (TP) over the total predicted results, which is the sum of all true positive and false negative (FN) outcomes. A specific performance metric may be assigned [LC5000].

$$Recall = \frac{TP}{TP + FN}$$

This table should be addressed for a subsequent revision. As it standard, there are currently no identified physical constraints on the system.

Table 4: Output Variables

Var	Physical Constraints

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: The IFC software shall accept updates to the variance in image noise, σ, upon user input.

R2: The IFC software shall accept updates to the image intensity threshold, t, upon user input.

R3: The IFC software shall accept updates to the patch size, p_s, upon user input.

R4: The IFC software shall accept updates to the bin size, l_{bin}, upon user input.

R5: The IFC software shall use the default noise suppresion parameters if no option is specified by the user.

R6: The IFC software shall use the default feature detection method if no option is specified by the user.

R7: The IFC software shall use the default keypoint descriptor method if no option is specified by the user.

R8: The IFC software shall use the default descriptor matching method if no option is specified by the user.

R9: The IFC software shall implement noise reduction on an input greyscale image per a prescribed standard deviation (from IM1).

R10: The IFC software shall, given a 2D greyscale image, define a set of keypoints per the alloted threshold criteria (from IM2).

R11: The IFC software shall define feature descriptors from identified keypoints (from IM3)..

R12: The IFC software shall identify matches between descriptors that originate from separate images (from IM4).

R13: The IFC software shall provide a .csv file that outlines the respective feature matches.

R14: The IFC software shall verify that all features are matched to features that originate from separate images.

R15: The IFC software shall verify that all feature correspondences are uniquely defined by the two respective sets of cameras, image frame, and associated pixels.

5.2 Nonfunctional Requirements

NFR1: **Usability** The IFC software shall be compatible with Python 3.1 libraries, such as OpenCV.

5.3 Rationale

The scope and assumptions of this document are intended to encapsulate a realistic problem formulation software. This software is intended for use in laboratory experimentation and configurations, rather than as an off-the-shelf product for applications that involve more risk, such as those that are safety-critical or mission critical in scope.

6 Likely Changes

LC1: The theoretical model of keypoint detection will be abstracted to expand the types of implementation models that can be used.

LC2: The theoretical model of assigning feature descriptors will be abstracted to expand the types of implementation models that can be used.

LC3: The theoretical model of descriptor matching will be abstracted to expand the types of implementation models that can be used.

7 Unlikely Changes

LC4: It is unlikely that the design will need to be adjusted to perform computation on RGB data instead of grayscale data.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 7 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 8 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	A1	A2	A3	A4
TM1			X	
TM2		X		
TM3		X		X
TM4	X			
GD1		X		
GD2				
GD3				
IM1				
IM2				
IM3				
IM4				
LC1				
LC2				
LC3				

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

	TM1	TM2	TM3	TM4	GD1	GD2	GD3	IM1	IM2	IM3	IM4	LC1	LC2	LC3
TM1					X			X						
TM2						X			X			X		
TM3							X			X			X	
TM4											X			X
GD1								X						
GD2									X					
GD3										X				
IM1									X					
IM2										X				
IM3											X			
IM4														
LC1						X			X					
LC2							X			X				
LC3											X			

Table 7: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	IM3	IM4	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
IM1		X			X				X				X						
IM2			X			X				X				X					
IM3				X			X	X			X				X				
IM4												X				X	X	X	X
R1	X												X						
R2		X												X					
R3			X												X				
R4			X												X				
R5													X						
R6	X													X					
R7		X													X				
R8			X													X			
R9	X												X						
R10		X																	
R11			X																
R12				X															
R13				X															
R14				X															
R15		X	X	X															

Table 8: Traceability Matrix Showing the Connections Between Requirements and Instance Models

9 Development Plan

This section may be expanded upon as part of a future release.

10 Values of Auxiliary Constants

No auxillary constants are defined for this software.

References

M. K. Chung. The gaussian kernel. https://pages.stat.wisc.edu/~mchung/teaching/MIA/reading/diffusion.gaussian.kernel.pdf. [Online; accessed 2-Feb-2025].

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, USA, 2000. ISBN 0521623049.

Pushyami Kaveti, Matthew Giamou, Hanumant Singh, and David M. Rosen. Oasis: Optimal arrangements for sensing in slam. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13818–13824, 2024. doi: 10.1109/ICRA57147.2024.10611644.

OpenCV. Feature matching with flann, 2021. URL https://docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html. Accessed: 2025-02-06.

OpenCV Contributors. Orb (oriented fast and rotated brief) - opencv tutorial, 2024. URL https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html. Accessed: 2025-02-05.

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33833-8.

Yifu Wang, Wenqing Jiang, Kun Huang, Sören Schwertfeger, and Laurent Kneip. Accurate calibration of multi-perspective cameras from a generalization of the hand-eye constraint. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1244–1250, 2022. doi: 10.1109/ICRA46639.2022.9811577.

End of Document.