Software Requirements Specification for : Image Feature Detection System

Gaofeng Zhou

April 12th,2024

Contents

Revision History

Date	Version	Notes
February 5	1.0	Initial revision
April 12	2.0	Second revision

1 Reference Material

This section records information for easy references.

1.1 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 the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
img_i	none	The $i-th$ image.
$M_(i)$	none	The matrix of $Image_i$
$I_(x)$	none	Image(x)
$G_(x)$	none	Gaussian Transform or Gaussian Filtering

1.2 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
IFDS	Image Feature Detection System
TM	Theoretical Model
BRISK	Binary Robust Invariant Scale Key-points

2 Introduction

Image features play a very foundational role in the area of image processing or, in another words,computer vision. Especially, When we decide to try 3d image reconstruction or 2d image stitching. Some image features can be detected by specific image process algorithms, and some of these features can be used for distinguishing whether they are suitable for further processing visually.
We develop this IFDS system to showcase the typical features of images such as edge points, corner points, BRISK points to make these feature points of images visually sensible to users who are interested in this area.
This IFDS system contains functions to do transformation and features detection using classical algorithms from OpenCV library.

2.1 Purpose of Document

The primary purpose of this document is to record the requirements and goals of IFDS, processing steps, algorithms selection. This document is also the basis for future development of IFDS.

2.2 Scope of Requirements

In this system, the images used for inputs should be 2d RGB images with the format constrained as JPEG,PNG,BMP,TIFF.
And this system will be distributed on Windows OS platform only.

2.3 Characteristics of Intended Reader

Readers of this document should have a basic understanding of image features. At least, they should know what these feature points stand for and how they will be used in future work.

2.4 Organization of Document

This document provides the goals, theories, definitions, road maps and assumptions.

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

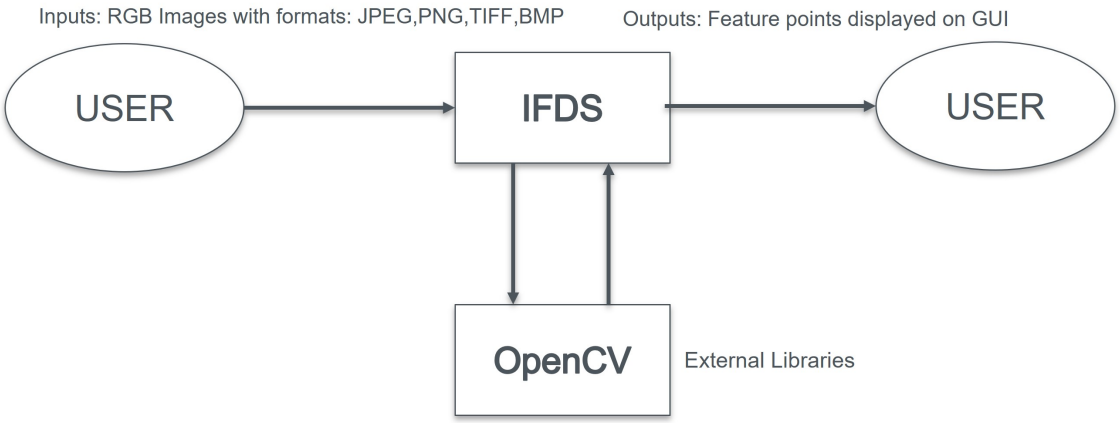

Figure 1: System Context

- User Responsibilities:
  - Input RGB images with formats as the system required.
- IFDS Responsibilities:
  - Import images and transform RGB images into Grayscale images.
  - Detect corner points.
  - Detect edge feature points.
  - Detect BRISK feature points.
  - Display corresponding feature points.
  - Export feature points image.
- External Library Responsibilities:
  - Provide classical feature detection functions for IFDS.

3.2 User Characteristics

The end users of IFDS should be able to install software on Windows OS. And they can understand what those image features are meant to do.

3.3 System Constraints

- This IFDS must be able to be run on Windows OS platform with distribution installer.and the Windows version should be updated to be newer than Windows 10.
- This IFDS will be developed with C++ and with the C++ libraries in OpenCV.
- This IFDS must have a GUI to control the system and display the images and the feature points.

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

IFDS is intended to solvedetecting image feature points within one system to make them visually sensible.

4.1.1 Terminology and Definitions

- Feature points: a feature is a piece of information about the content of an image; typically about whether a certain region of the image has certain properties.
- Corner points: In object detection, the intersection of two perpendicularly intersecting edgesyujin (2021).
- Edge points:The position of pixels in an image where certain quantities or properties (such as brightness) change rapidly.The pixel position in the image where the gradient is greater than a given thresholdyujin (2021).
- BRISK feature points:a novel method for keypoint detection, description and matchingLeutenegger et al. (2011).It can be used to replace the SIFT algorithm to do the Scale Invariant Feature Transformation.

4.1.2 Goal Statements

The goal statements are:

- GS1: Given a RGB image, this system can transfer the RGB image into Grayscale image first. and then it can do corner detection, edge detection and BRISK feature points detection.

- GS2: Given an imported image by users, this system would display this image with its original image and grayscale image on a GUI, and the feature points should be displayed on the grayscale image.
- GS3: This system should be able to save those feature points images.

4.2 Solution Characteristics Specification

The instance models that govern IFDS are presented in Subsection **??**. 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 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: Input images must be RGB images with formats limited among "JPEG, PNG, BMP, TIFF".
- A2 This system Only detects corner points, edge points and BRISK points.
- A3: The system will be only installed on Windows OS platform.

4.2.2 Theoretical Models

The IFDS system uses typical image processing methods to do feature points detection. Mainly, it derived from the theory of matrix and gradients operation.

RefName: **Gradients Calculation**

Label: Gradients Calculation

Equation: $\nabla I_{(x)} = \left[\frac{\partial I_{(x)}}{\partial x}, \frac{\partial I_{(x)}}{\partial y}\right]$

Description: Here, the $\nabla I_{(x)}$ stands for the gradients of image $I_{(x)}$.

Notes: This formula is only a brief demonstration to the principle of image processing.As to different images, we need to calculate their gradients on different dimensions and scales.

Source: https://en.wikipedia.org/wiki/Image_gradient

Ref. By: none

Preconditions for Gradients Calculation: None

Derivation for Gradients Calculation: Not Applicable

Not applicable

4.2.3 General Definitions

Not applicable

4.2.4 Input Data Constraints

All the input images must be RGB images with format limited among "JPEG, PNG, BMP, TIFF".

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: Verify the input images must be with the formats as specified in the assumptions.

R2: The RGB images must be transformed into Grayscale image when those images were imported into IFDS system.

R3: The feature points must be displayed onto the grayscale image.

R4: Those feature points image must can be saved into user's device.

5.2 Nonfunctional Requirements

NFR1: **Usability** The program should be easily understood and usable to end users. This will be tested by random volunteers.

NFR2: **Portability** This system is developed mainly for Windows OS platform, So it is not assured that it can be used on other OS platforms.

NFR3: **Maintainability** The effort required to make any of the likely changes listed for IFDS should be less than FRACTION of the original development time. FRACTION is then a symbolic constant that can be defined at the end of the report.

5.3 Rationale

The assumptions in this documentation have limited the most common format used for images. And those feature points such as corner, edge, BRISK are also typical feature points. Through this system, we meant to showcase those feature points intuitively and visually sensible using classical image processing algorithms. These algorithms include Harris corner points detection, gradients edge points detection, BRISK feature points detection. All those algorithms can be found in literature very easily.

6 Likely Changes

LC1: We may change the image detection algorithms depending on the practical performance. If one method has been proved not feasible, then we may adapt another practical method to realize them.

LC 2: The layout of GUI may be changed according to user feedback.

7 Unlikely Changes

UC1: The OpenCV library will not be changed.

UC2: The coding language will be not changed.

UC3: The OS platform which it will be distributed to will be changed.

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 1 shows the dependencies of likely changes on the assumptions.

9 Values of Auxiliary Constants

Show the values of the symbolic parameters introduced in the report.

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

The value of FRACTION, for the Maintainability NFR would be given here.

	A1	A2	A3
R1			
R2 R3			
R4			
LC1	X		
LC2			X

Table 1: Traceability Matrix Showing the Connections Between Assumptions and other items

References

Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, 2011.

Zhang yujin. *Handbook of Image Engineering*. Springer, 2021.