Project Title: System Verification and Validation Plan for Image Features Detection System

Gaofeng Zhou

April.14,2024

Revision History

Date	**Version**	**Notes**
Feb.18, 2024	1.0	Initial version
April.14,2024	2.0	Second revision

Contents

1 Symbols, Abbreviations, and Acronyms iii

2 General Information 1
2.1 Summary 1
2.2 Objectives 1
2.3 Relevant Documentation 1

3 Plan 1
3.1 Verification and Validation Team 2
3.2 SRS Verification Plan 2
3.3 Design Verification Plan 2
3.4 Verification and Validation Plan Verification Plan 2
3.5 Implementation Verification Plan 2
3.6 Automated Testing and Verification Tools 2
3.7 Software Validation Plan 3

4 System Test Description 3
4.1 Tests for Functional Requirements 3
4.1.1 GUI test 3
4.2 Tests for Nonfunctional Requirements 5
4.2.1 Accuracy test 5
4.2.2 Usability survey 7
4.3 Traceability Between Test Cases and Requirements 7

5 Appendix 9
5.1 Usability Survey Questions? 9

1 Symbols, Abbreviations, and Acronyms

symbol	description
T	Test
I	Image
G(x)	Gaussian Transform or Guassian Filtering
L(x)	Laplacian Transform
DoG	Differential of Gaussian Transform of Gaussian Transform
LoG	Laplacian Transform of Gaussian Transform of Gaussian Transform
SIFT	Scale Invariant Feature Transform
SRS	Software Requirements Specification

2 General Information

2.1 Summary

The Image Features Detection System will be tested according to this plan here. This system can be seen as a preliminary step before doing 2d image stitching or 3d image reconstruction. It will use one image as the inputs, and produce image feature points such as corner points, Edge Points,and BRISK Feature Points. All these feature points will be used to have first visual judgement before further image processing.

2.2 Objectives

The objectives of this system is to use one RGB image as input and then show the feature points of the RGB as the outputs generated by those image processing algorithms. There would be a GUI display setting for this system. On this GUI, there would be different buttons which indicate different functions to do image features detection. There should be two windows to show the input images and two windows to their features points on this GUI. With each click of these buttons, the result will be displayed on the GUI.

Besides those basic objectives of this system, it also should suffice the requirements of accuracy and usability. The accuracy should be in accordance with verified standards as which here we use MATLAB. The usability means the layout of this system should be user friendly. And the processing time should be in a high efficiency.

2.3 Relevant Documentation

SRS document SRS

3 Plan

The plan to do verification and validation will consist of SRS VnV plan, Unit VnV plan, Integration and system VnV plan.

3.1 Verification and Validation Team

Name	Role
Valerie Vreugdenhil	Domain Expert and Primary Reviewer
Seyed Ali Mousavi	Second reviewer for SRS
Hossain Al Jubair	Second reviewer for VnV plan
Xinyu Ma	Second reviewer for MG +MIS

3.2 SRS Verification Plan

For the SRS verification plan, we will apply the experts review as the main method to do this.

3.3 Design Verification Plan

For this part, we will use this checklist as below to do this VnV process 1. Is this design theoretically workable? 2. Is this design technically feasible? 3. Is this design the best way to solve this problem? 4. What do we need to complete this system as the design plan required? Is the cost worthy?

3.4 Verification and Validation Plan Verification Plan

We will apply expert review to do this plan.

3.5 Implementation Verification Plan

For the code part, we will use copilot to the codes check.
For the algorithm part, we will use MATLAB to verify.
For the GUI and integration part, we will use Qt test to achieve the result.

3.6 Automated Testing and Verification Tools

Here we may use github actions to do the CI/CD monitoring as we will let this project open sourced on github.

3.7 Software Validation Plan

Experts reviews will be considered here. and the algorithm part will be compared with MATLAB.

4 System Test Description

The whole system test will be processed in two parts. One is the functional requirements test, and the other is the non-function requirements test.

4.1 Tests for Functional Requirements

The functional requirements consist of GUI test, buttons function test, and display test.

4.1.1 GUI test

GUI here should be generate with Qt platform. and the GUI with integrated functions inside should be working accurately and well organized.

Button test

1. test-1-import button

   Control: Manual

   Initial State: first step

   Input: One RGB image with good precondition as required.

   Output: One RGB image will be displayed on the left window of the GUI, and its Grayscale Image will be displayed on the right side of the GUI, after the "import" button was clicked.

   How test will be performed: This test will be performed manually with the import button.

2. test-2- Corner button

   Control: Manual

   Initial State: After image was transformed into gray scale images

   Input: one grayscale image as the test 1.

   Output: There would be corner points detected on the grayscale image and these corner points will be displayed on the grayscale image of the right side window.

   How test will be performed: After the corner button was clicked, the corner points will be displayed on the grayscale image after we did test1, which means the sytem has transformed the RGB image into grayscale image and displayed them on the GUI windows.

3. test-3- Edge button

   Control: Manual

   Initial State: After the image was imported

   Input: the grayscale image

   Output: There would be edge points drawn onto the grayscale image displayed on the GUI windows after the ”edge” button was clicked.

   How test will be performed: After the RGB image was transformed into grayscale images successfully, we click the ”edge” button. Then there would be edge points appeared on the grayscale image displayed on the GUI window if this button worked well.

4. test-4- BRISK button

   Control: Manual

   Initial State: After images were transformed into grayscale image.

   Input: Grayscale image after the import button was clicked.

   Output: There would be BRISK points drawn onto the grayscale image displayed on the GUI window after the ”BRISK” button clicked.

   How test will be performed: After the RGB images were transformed into grayscale image successfully, we click the ”BRISK” button. Then there would be BRISK points appeared on the Grayscale image displayed on the GUI window if this button worked well.

5. test-5- Export button

   Control: Manual

   Initial State: After feature points was draw onto grayscale image.

   Input: feature points image.

   Output: Feature points image saved to user's computer.

   How test will be performed: After these feature points were detected successfully, if the export button was clicked, the feature points image could be saved to users' computer, if this button worked well.

6. test-6- Close button

   Control: Manual

   Initial State: Any state.

   Input: Any state.

   Output: This system will be closed, if this button worked successfully.

   How test will be performed: We can click this button at any time we want. if the system was closed after this button was clicked, then it means this button worked successfully.

4.2 Tests for Nonfunctional Requirements

Accuracy: we will compare the points we got with this system with the outcome we got with MATLAB.

Usability: we will do a survey to review the usability of this system.

4.2.1 Accuracy test

Algorithm accuracy verification

1. test-1-Corner points accuracy test

   Type: Manual

   Initial State:

   Input/Condition: First best 50 points generated by our system and MATALB respectively.

   Output/Result: the average absolute error value of this 50 points.

   How test will be performed: Firstly, we got 50 best edge points with the edge detection algorithm in our system, and then we would get 50 best edge points with edge detection algorithm in MATLAB. After that, we calculate these the differences of these 50 pairs of best edge feature points. We sum them up and then let the summation divided by 50,then we can get the value we needed.

2. test-2-Edge points accuracy test

   Type: Manual

   Initial State:

   Input/Condition: First best 50 points generated by our system and MATALB respectively.

   Output/Result: the average absolute error value of this 50 points.

   How test will be performed: Firstly, we got 50 best contour points with the edge detection algorithm in our system, and then we would get 50 best contour points with edge detection algorithm in MATLAB. After that, we calculate these the differences of these 50 pairs of best contour feature points. We sum them up and then let the summation divided by 50,then we can get the value we needed.

3. test-3-BRISK points accuracy test

   Type: Manual

   Initial State:

   Input/Condition: First best 50 points generated by our system and MATALB respectively.

Output/Result: the average absolute error value of this 50 points.

How test will be performed: Firstly, we got 50 best corner points with the corner detection algorithm in our system, and then we would get 50 best corner points with corner detection algorithm in MATLAB. After that, we calculate these the differences of these 50 pairs of best corner feature points. We sum them up and then let the summation divided by 50,then we can get the value we needed.

4.2.2 Usability survey

We will do a usability survey with a questionnaire as the appendix showed.

4.3 Traceability Between Test Cases and Requirements

Table 1: Traceability Matrix

	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**
Images Imported	X					
Grayscale image transformed	X					
Corner detection button		X				
Edge detection button			X			
BRISK detection button				X		
Export button					X	
Close button						X

References

5 Appendix

5.1 Usability Survey Questions?

1. Do you like the GUI layout of this system? Rate it in the range of 0-10, which 10 stands for best.
2. Does the display windows look good of this system? Rate it in the range of 0-10, which 10 stands for best.
3. Do you like the designs of these buttons of this system? Rate it in the range of 0-10, which 10 stands for best.
4. Do you like the way with which these feature points displayed of this system? Rate it in the range of 0-10, which 10 stands for best.
5. If you have any advice on this system, please write in the blank.