

Verification and Validation Report: IFDS

Gaofeng Zhou

April 2024

Contents

1 Revision History	ii
2 Symbols, Abbreviations and Acronyms	iii
3 Brief	1
3.1 Introduction	1
3.2 Verification and Validation Team	1
4 Functional Requirements Evaluation	2
4.1 GUI Test	2
5 Nonfunctional Requirements Evaluation	10
5.1 Usability	10
5.2 Accuracy	10
6 Changes Due to Testing	14
7 Trace to Requirements	14
8 Code Coverage Metrics	14

1 Revision History

Date	Version	Notes
April 18	1.0	Initial Version

2 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

3 Brief

3.1 Introduction

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.

3.2 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
Yileng Chen	Usability advisor

4 Functional Requirements Evaluation

4.1 GUI Test

- **Import Button Test with Different Formats of Images:** As we expected, the Import Button and function works well with all those formats of images.

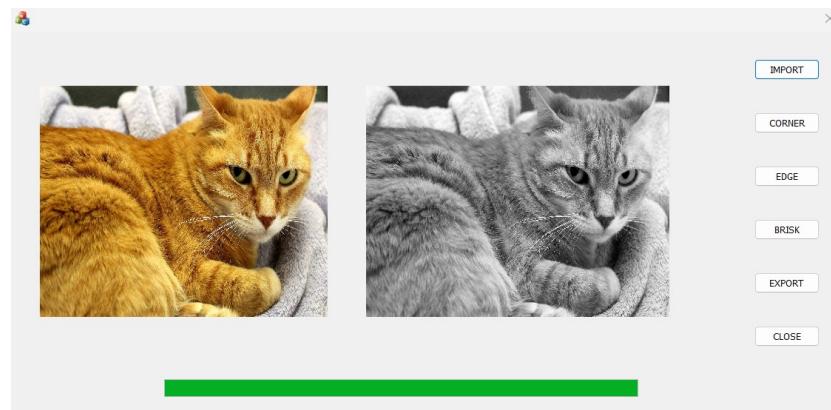


Figure 1: Import Button with JPEG

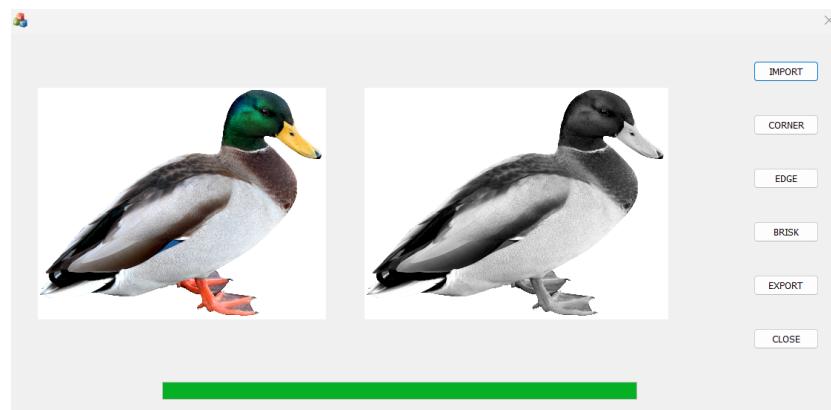


Figure 2: Import Button with PNG



Figure 3: Import Button with TIFF

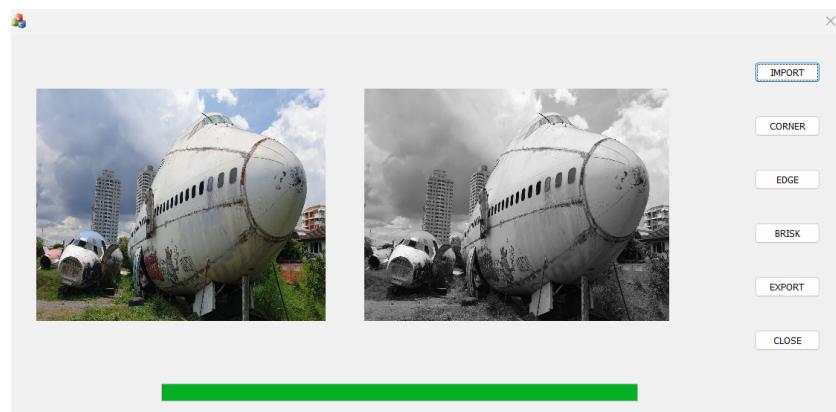


Figure 4: Import Button with BMP

- **Corner Button Test:** As the test showed this button and function works well with different formats of images.

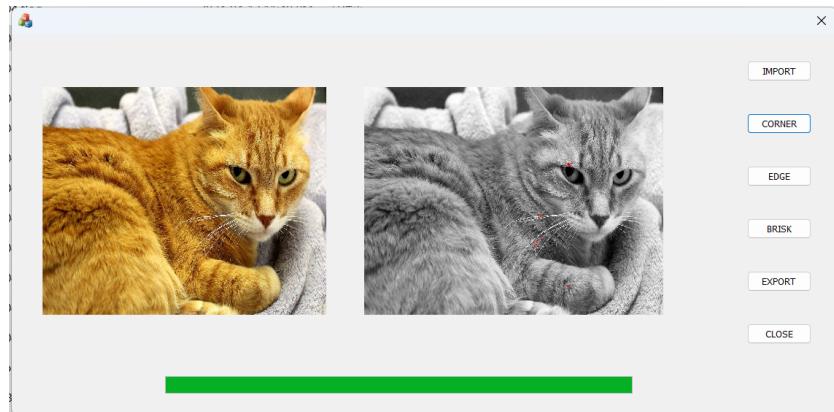


Figure 5: Corner Button with JPEG

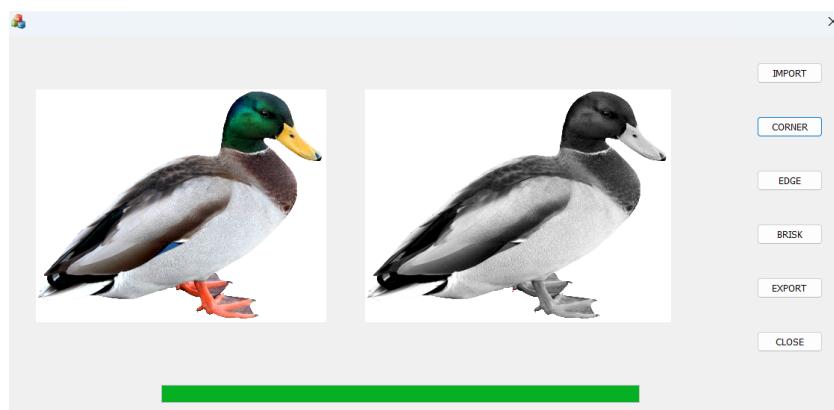


Figure 6: Corner Button with PNG

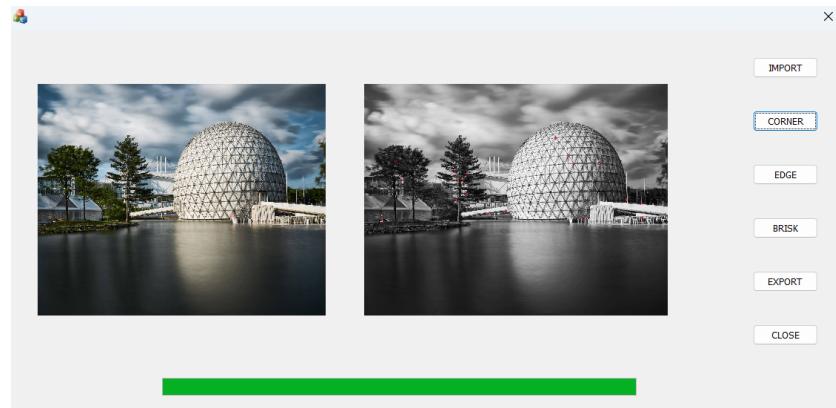


Figure 7: Corner Button with TIFF



Figure 8: Corner Button with BMP

- **Edge Button Test:** As the result showed, this button and function works well.

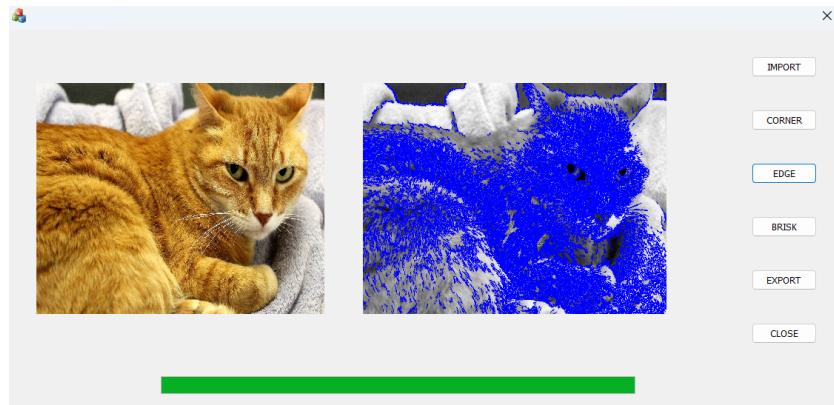


Figure 9: Edge Button with JPEG



Figure 10: Edge Button with PNG

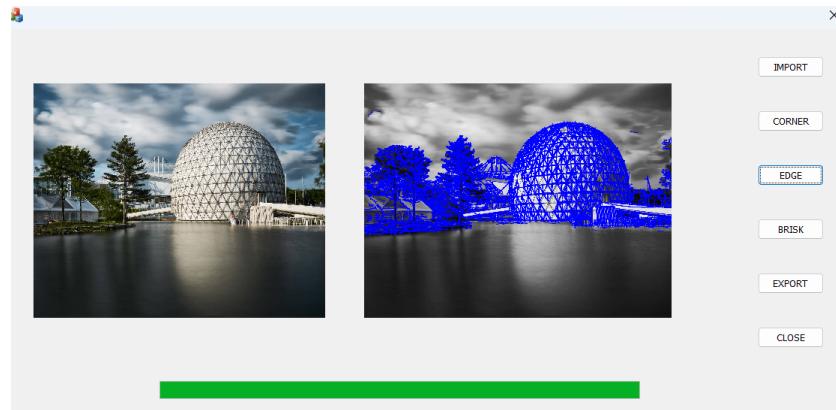


Figure 11: Edge Button with TIFF



Figure 12: Edge Button with BMP

- **BRISK Button Test** As the test results showed, this button and function works well too.

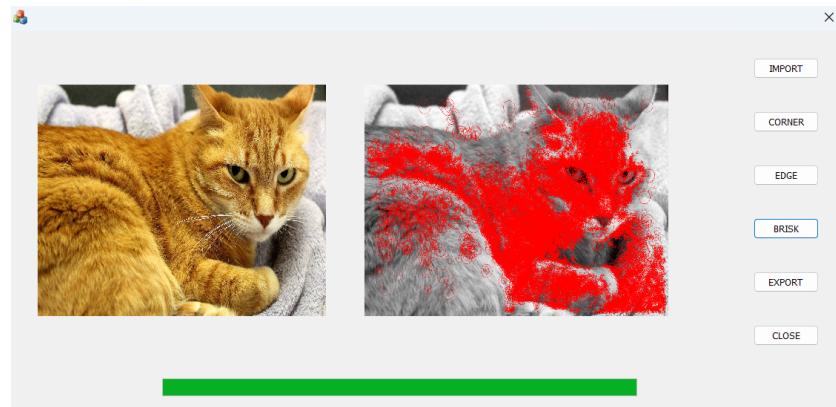


Figure 13: BRISK Button with JPEG

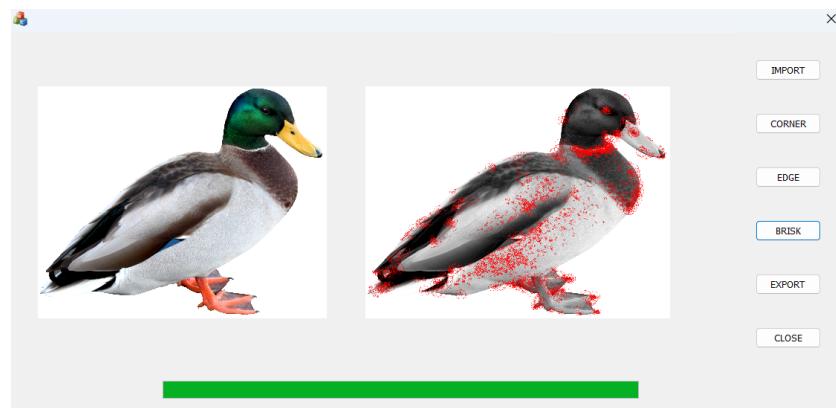


Figure 14: BRISK Button with PNG

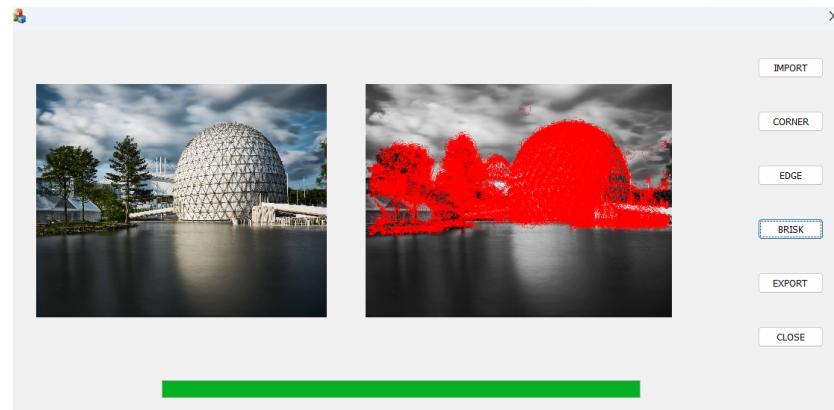


Figure 15: BRISK Button with TIFF



Figure 16: BRISK Button with BMP

5 Nonfunctional Requirements Evaluation

5.1 Usability

Due to the time limitation, we only did several user survey to the usability of this system, and the feedback is as bellow:

Question	Feedback
Do you like the GUI layout of this system? Rate it in the range of 0-10, which 10 stands for best?	9
Does the display windows look good of this system? Rate it in the range of 0-10, which 10 stands for best.	10
Do you like the designs of these buttons of this system? Rate it in the range of 0-10, which 10 stands for best.	8
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.	9
If you have any advice on this system, please write in the blank.	1. It would be better if the IFDS title was added to the toolbar. 2. Integrate the import and export button together to the toolbar. 3.Add a locale of Help.

5.2 Accuracy

We use Matlab to do the exact feature detection, and compare them with IFDS. We can see that visually most of those points are located in the same section. For more detailed test, we need to do further research on it.

- Corner points with Matlab,

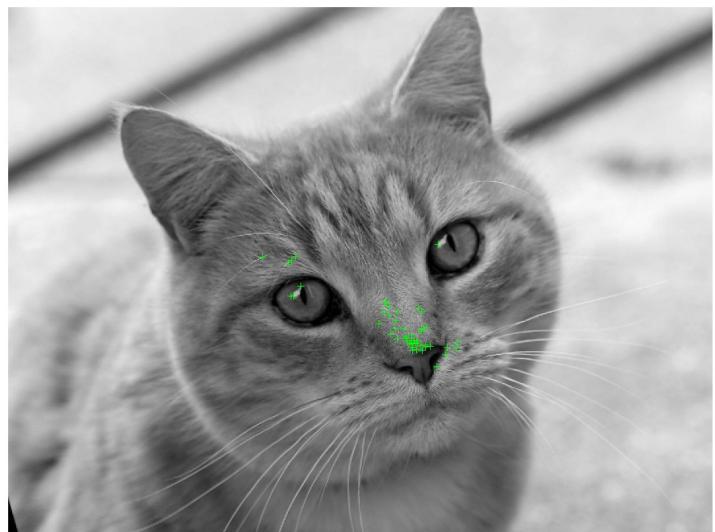


Figure 17: Corner points with Matlab

- Edge points with Matlab



Figure 18: Edge points with Matlab

- BRISK points with Matlab



Figure 19: BRISK points with Matlab

6 Changes Due to Testing

- Due to the first test of Qt platform, there are link problems, So we moved this project to visual studio and generated the GUI using MFC.
- Due to the visual effect of the GUI display, we changed the number of windows for displaying images from 4 to 2.
- Due to the version problems of OpenCV, we changed the SIFT feature points to a more advanced feature points as BRISK.

7 Trace to Requirements

Table 1: Traceability Matrix

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

8 Code Coverage Metrics

For the check of code review, the codes has been measured through Visual Studio compiling, running and debugging. It showed a strong robustness and complete coverage to the requirement of the SRS

References