Project Title: System Verification and Validation Plan for AortaGeomRecon

Jingyi Lin

June 8, 2023

1 Revision History

Date	Version	Notes
2023-04-30	1.0	First draft of VnV plan
Date 2	1.1	Notes

Contents

1 Revision History **i**

2 Symbols, Abbreviations and Acronyms **iii**

3 General Information **1**
- 3.1 Summary . . . 1
- 3.2 Objectives . . . 1
- 3.3 Relevant Documentation . . . 2

4 Plan **2**
- 4.1 Verification and Validation Team . . . 2
- 4.2 SRS Verification Plan . . . 2
  - 4.2.1 Functional Requirements . . . 2
  - 4.2.2 Nonfunctional Requirements . . . 3
- 4.3 Design Verification Plan . . . 3
- 4.4 Implementation Verification Plan . . . 3
- 4.5 Automated Testing and Verification Tools . . . 3
- 4.6 Software Validation Plan . . . 4

5 System Test Description **4**
- 5.1 Tests for Functional Requirements . . . 4
  - 5.1.1 Area of Testing1 . . . 4
  - 5.1.2 Area of Testing2 . . . 5
- 5.2 Tests for Nonfunctional Requirements . . . 5
  - 5.2.1 Area of Testing1 . . . 5
  - 5.2.2 Area of Testing2 . . . 6
- 5.3 Traceability Between Test Cases and Requirements . . . 6
- 5.4 Traceability Between Test Cases and Modules . . . 6

6 Appendix **7**
- 6.1 Symbolic Parameters . . . 7
- 6.2 Usability Survey Questions? . . . 7

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

(Lin, 2023b)

The following section provides an overview of the Verification and Validation (V&V) Plan for program AortaGeomRecon. This section explains the purpose of this document, the scope of the system, common definitions that are used in the document. Throughout this document, we refer to the terms that have been already explained in the document Software Requirements Specification for Program AortaGeomRecon.

3 General Information

3.1 Summary

AortaGeomRecon is tested for its segmentation module. The segmentation algorithm can be split into 3 parts, the descending aorta segmentation, the ascending aorta segmentation and the sagittal segmentation.

The descending aorta segmentation takes an original volume and a descending aorta centre coordinate as the inputs, and perform the segmentation on the volume, finally returns the result label volume indicate which voxel belongs to the descending aorta.

Smilarly, the ascending aorta segmentation takes an original volume, an ascending aorta centre coordinate, and the descending aorta segmentation result label volume as the inputs, and perform the segmentation on the original volume, finally returns the result label volume indicate which voxel belongs to the descending or ascending aorta.

Finally, the sagittal segmentation is used to fill in any missing voxel that is potentially belong to the aorta in the label volume.

3.2 Objectives

The main objective of the verification and validation is to ensure the segmentation algorithm correctness, while optimizing the algorithm for the readability and the efficiency. The segmentation algorithm has been tested with 6 samples and the results are verified by the domain experts. Continuous integration testing can ensure the software correctness for each modification of the existing code.

3.3 Relevant Documentation

Lin (2023a) Lin (2023b)

4 Plan

This section provides a description of the software that is being tested, the team that will perform the testing, and the milestones for the testing phase/

4.1 Verification and Validation Team

Name	Role description
Jingyi Lin	Design and implement testing modules.
Dr. Spencer Smith	Code review and provide test cases.

4.2 SRS Verification Plan

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.

4.2.1 Functional Requirements

The functional requirements have been already explained in the Software Requirements Specifications for Program AortaGeomRecon. As a reference, the names of the requirements are as follows:

- Input
- Region of Interest
- Output
- Visualization

4.2.2 Nonfunctional Requirements

Considering the use of this program in the research and academia nowadays, as well as keeping an eye on its future use in the medicine and clinical practice, the priority nonfunctional requirements are correctness, understandability, reliability, and usability. The usability Nonfunctional requirements are as follow:

- Interface
- 3D Rendering

4.3 Design Verification Plan

The GitHub issues lists the requirements and smaller tasks to fullfil the goal of the software requirements.

4.4 Implementation Verification Plan

[You should at least point to the tests listed in this document and the unit testing plan. —SS]

[In this section you would also give any details of any plans for static verification of the implementation. Potential techniques include code walk-throughs, code inspection, static analyzers, etc. —SS]

4.5 Automated Testing and Verification Tools

We use GitHub Actions to implement automated tests. GitHub Actions is a new feature to automate workflows that is build a cloud instance, which can be Windows, Linux or macOS.

GitHub Actions reads a yml file to deploy the repository on a temporary cloud instance, then execute the listed commands in the yml file. We built two automated tests with GitHub Actions: Linter and Continous Integration tests. Lint and CI tests will be executed every time when there is new code pushed on GitHub.

Linter is a static code analysis tool to flag programming errors, bugs and code format.

Continous Integration tests ensure the software correctness by comparing the result calculated with the new code to the result calculated by the old code. The old results are verified by testers and are set to the current ground truth. If the new result is very different from the old result then there is probably something wrong in the new code.

4.6 Software Validation Plan

[If there is any external data that can be used for validation, you should point to it here. If there are no plans for validation, you should state that here. —SS]

5 System Test Description

5.1 Tests for Functional Requirements

[Subsets of the tests may be in related, so this section is divided into different areas. If there are no identifiable subsets for the tests, this level of document structure can be removed. —SS]

[Include a blurb here to explain why the subsections below cover the requirements. References to the SRS would be good. —SS]

5.1.1 Area of Testing1

[It would be nice to have a blurb here to explain why the subsections below cover the requirements. References to the SRS would be good. If a section covers tests for input constraints, you should reference the data constraints table in the SRS. —SS]

Title for Test

1. test-id1

   Control: Manual versus Automatic

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

2. test-id2

Control: Manual versus Automatic

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

5.1.2 Area of Testing2

...

5.2 Tests for Nonfunctional Requirements

[The nonfunctional requirements for accuracy will likely just reference the appropriate functional tests from above. The test cases should mention reporting the relative error for these tests. —SS]

[Tests related to usability could include conducting a usability test and survey. —SS]

5.2.1 Area of Testing1

Title for Test

1. test-id1

Type:

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

5.2.2 Area of Testing2

...

5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

5.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

References

Jingyi Lin. Module guide. https://github.com/smiths/aorta/blob/main/docs/Design/MG/MG.pdf, 2023a.

Jingyi Lin. System requirements specification. https://github.com/smiths/aorta/blob/main/docs/SRS/SRS.pdf, 2023b.

6 Appendix

This is where you can place additional information.

6.1 Symbolic Parameters

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

6.2 Usability Survey Questions?

[This is a section that would be appropriate for some projects. —SS]