Requirements Document Revision 0
Liquid Rescaling

Team 35 - Marshiel
Lab 03
Marlee Roth
Daniel Wolff
Harsh Shah

October 27, 2017

Contents

1 Revision History **3**

2 Tools **3**

2.1 Type of Tests 3

3 Test Cases **3**

3.1 Test Cases Include: 3

3.2 Exception Testing 5

4 Proof of Concept **5**

5 Non Functional Requirements **5**

6 Test Plan Schedule **6**

List of Tables

1 Revisions 3

1 Revision History

Table 1: Revisions

N/A	N/A
N/A	N/A

2 Tools

The tests that will be used are Visual Studio's built-in unit testing framework to perform automated testing on features that have a definite output given a certain input. For example, if an image is scaled to have a width of 500 and a height of 600, the resulting image must have a width of 500 and a height of 600.

2.1 Type of Tests

The types of tests that will be performed are automated testing (see above), manual testing (for visual acceptance of scaled images). The automated tests will be designed using the white box technique (to cover all edge cases of the features), and the manual tests will be black box technique (since the implementation of the visual output of the scaled images are not known).
Automated tests will be run using Visual Studio. Manual tests will consist of images being scaled using our application and GIMP's Liquid Rescaling plugin. The output of the two will be compared, and should be very similar. A visual rating will be assigned by the testers, and a certain quality will have to be met with each test.

3 Test Cases

3.1 Test Cases Include:

- **RS1**
  - Test scaling, preservation, amplification and removal on simple black and white images which consist of a solid coloured background with 2 or 3 objects of different shades. Objects should preferably be simple solid shapes such as a square, rectangle, circle, sphere, cylinder and/or cube.*
- **RS2**
  - Test scaling, preservation, amplification and removal on simple coloured images which consist of a solid coloured background with 2 or 3 objects of different colours. Objects should preferably be simple solid

shapes such as a square, rectangle, circle, sphere, cylinder and/or cube.*

- **RS3**
  - Test scaling, preservation, amplification and removal on simple black and white images which consist of a solid coloured background with only 1 object of different shades. Objects should preferably be simple solid shapes such as a square, rectangle, circle, sphere, cylinder and/or cube.*
- **RS4**
  - Test scaling, preservation, amplification and removal on simple coloured images which consist of a solid coloured background with only 1 object of different colours. Objects should preferably be simple solid shapes such as a square, rectangle, circle, sphere, cylinder and/or cube.*
- **RS5**
  - Test scaling , preservation, amplification and removal on images which consist of a solid background with only 1 object. Such as an image of the ocean and the sky with one bird or a small island.*
- **RS6**
  - Test scaling, preservation, amplification and removal on images which consist of a solid background with only 1 complex object. Such as an image of a person or animal walking on the beach or in the water of the ocean.*
- **RS7**
  - Test scaling, preservation, amplification and removal on images which consist of a solid background with multiple complex objects. Such as an image of a group of people playing volleyball on the beach.**
- **RS8**
  - Test scaling, preservation, amplification and removal on colorful images which consist of multiple complex objects. Such as an image of the inside/outside of the a busy shopping center which has lots of festive decorations.**
- **Notes**
  - * The solid coloured background allows the team to test the behaviour of the energy function and as a result the program is expected to give clear cut removals, amplifications and preservation of objects.

  - ** The complex objects and colours allows the team to challenge the effectiveness of the algorithm's energy function. The program is expected to see slight distortions.

3.2 Exception Testing

The program shall generate an exception if the user inputs a file with an unacceptable file format (i.e. not an image file).
The program shall generate an exception if the user requests to save the output file in an a non-existing directory.

4 Proof of Concept

The proof of concept will be testing the applications ability to accurately scale and preserve an image. The application should be able to perform content aware image resizing on any image chosen. The main aspect that need to be tested are the images clarity, and likeness to the original image. This cannot be done using automated testing therefore it will need to be done manually. The tests done on the application are specified in section (Non-functional section).

5 Non Functional Requirements

To ensure the program is working at its best, non-functional requirements of the program will be tested.
The first requirement is the program's usability, it will be tested by having multiple different users from different age groups perform a series of tasks using the application without guidance. If the users are able to perform the tasks without much trouble then the application will be very user friendly.
The second requirement is the safety requirements. To test this images can be saved with the same filename as an existing file to see if it is rewritten, the file can be given the wrong file ending to see if it still saves properly. The image can also be edited but not saved to make sure the program does not save images unless specified.

The third requirement is performance. To test the performance of the program multiple test can be done.
To test the programs start up speed, the application will be launched and timed to find to total boot time. To test its capacity it can also be stress tested with a very large image file to see if the application will still responds with reasonable response time under intense stress. To test the applications ability to scale images a very small image can be uploaded and see if it scales it down correctly. Large images can be used to see if it can scale the photo in a reasonable amount of time. To test the scaling algorithms accuracy images with gradients can be put into the application to test if they scale properly. Black and white images can be used to see if the algorithm will still work in a limited color scale. Images

that are entirely one color can be uploaded to see how the program will respond under these circumstances.

6 Test Plan Schedule

The updated Gantt Chart can be found in the git repository.