**Detailed Design & Implementation**

**Course:** CS 474 Object-Oriented Design

**Project Name:** RADIS

**Team Members**

Andrew McClellan

Austin Traverse

Kelsey Hilton

**Professor:** Rob Byrd, iSchool, Abilene Christian University

**Last updated:** 23 April 2012

**Document Revision Number:** 1

|  |  |
| --- | --- |
| **Detailed Design and Implementation** | |
| **UML** | **% Done** |
| 1. Final UML .pngs |  |
|  |  |
| **Source Code** | |
| 1. Header Files |  |
| 1. Main |  |
| **Test Data** | |
| 1. Explanation |  |
| 1. Images |  |
| 1. Patient (user input) |  |
| **User’s Guide** | |
| 1. Installation Guide |  |
| 1. Tutorial on how to implement the API |  |
| 1. Sample Application |  |
| 1. Doxygen Documentation |  |
| **Appendices** | |
| * 1. dA Function explainations/declarations/examples |  |
| * 1. Functions Inherited from CVIP |  |
| * 1. Functions Inherited from Qt |  |

**UNIFIED MODELING LANGUAGE**

1. **Final UML .pngs**

**SOURCE CODE**

1. **Header Files**
2. **Main**

**TEST DATA**

The main image processing technique for this project was comparing histogram features. A histogram is the plot of the grey values versus the number of pixels in an image.

1. **Images**
2. **Patient: Example User Input**

**USER’S GUIDE**

1. **Installation Guide**

It is assumed that you are working on a Windows platform.

1. If you don’t already have Microsoft Visual Studio Professional Edition 2010 you will need to either, purchase it, or download a trial edition from [www.microsoft.com/visualstudio](http://www.microsoft.com/visualstudio).
2. You will also need the Qt Visual Studio Add-In and Qt libraries 4.8.1 for Windows which can be downloaded from <http://qt.nokia.com/downloads>

NOTE: does not work with the Express Edition of Visual Studio

1. Open
2. To run the application, open the CVIPlab.vs file. This should open Visual Studio. You will be able to see the source code for both the application and the underlying API.
3. **Tutorial**

Prerequisites:

Microsoft 2010 Visual Studio Professional Edition

Qt add-in for Visual Studio

Qt 4.8.1

If you do not meet the prerequisites see the Installation Guide.

1. **Sample Application**
2. **Doxygen Documentation**

**APPENDIX A**

**dA Function Definitions**

**APPENDIX B**

**Functions Inherited from CVIP**

The following CVIP functions were used in the development of our dA API:

void **display\_image**(const char\*image\_name, IMAGE\_FORMAT format)

<image\_name> name of the image file

<format> the image format

void **view\_image**(Image \*inputImage, char \*imageName)

<inputImage> pointer to the input image

<imageName> character string as the image name in the display window

double **\*hist\_feature**(image \*originalImage, \*labeledImage, int r, int c)

<originalImage> pointer to the original image

<labeledImage> pointer to the labeled image

<r> row coordinate of a point on the labeled image

<c> column coordinate of a point on the labeled image

Returns 5 histogram feautres: mean, standard deviation, skew, energy and entropy

Image **\*read\_Image**(char \*filename, int showmessages)

<filename> - pointer to an character string containing the file name

<showmessages> - show messages

unsigned **GetNoOfCols\_Image**(Image\* image)

<image> pointer to an image

unsigned **GetNoOfRows\_Image**(Image\* image)

<image> pointer to an image

Image \***duplicate\_Image(**const Image \*a)

<a> pointer to an image

Image **\* label**( const Image \*imageP )

<imageP> - pointer to an Image

Image **\*threshold\_segment**(Image \*inputImage, unsigned int

threshval, CVIP\_BOOLEAN thresh\_inbyte)

<inputImage> - pointer to Image structure

<threshval> - threshold value

<thresh\_inbyte>

- CVIP\_NO apply threshval directly to image data;

- CVIP\_YES threshval is CVIP\_BYTE range; remap to

image data range before thresholding.

**APPENDIX C**

**Functions Inherited from Qt**