Laplacian Coordinates based Seeded Image Segmentation

Gourab Ghosh Roy, Raabid Hussain and Mohammad Koujan

January 11, 2016

Outline

- Introduction
- Algorithm
- Implementation
- Results
- Conclusion

Seeded Segmentation

- Segmentation
  - Dividing an image into meaningful segments
- Seeded Segmentation
  - Seeds (labels) for some pixels known
- Existing methods mostly involve heavy computation.

Laplacian Coordinates for Seeded Segmentation

- Proposed by Casaca et al.
- Minimization of quadratic Energy functional
- Similar pixels grouped together, region boundaries are highlighted

Algorithm

- Pixel weights defined as

$$w_{ij} = e^{(-\beta \|(I_i - I_j)\|_\infty^2 / \sigma)} \tag{1}$$

- Graph Laplacian L

$$L = D - W \tag{2}$$

- Energy functional to be minimized

$$E(x) = x^t (I_s + L^2) x - 2x^t b + c \tag{3}$$

- Solution to the linear system

$$(I_s + L^2) x \;=\; b \tag{4}$$

Properties

- $I_s + L^2$ is symmetric, positive definite and sparse
- Unique solution guaranteed
- Solving linear system of equations easy for implementation

Implementation

- $C++$ Programming Language in Qt
- Use of OpenCV and Eigen Libraries

Use of Eigen Library

- OpenCV Mat objects did not work
- SparseMatrix class in Eigen Library
- Use of triplets
- SimplicialLDLT linear solver

Object Oriented Programming

- Initial procedural way optimized through use of OOP. Make the code modular and reusable.
- Class SeededImgSeg is defined.
- Private data members of SeededImgSeg class are parameters used in solving the linear equation system. Getter and setter methods are defined.

Object Oriented Programming

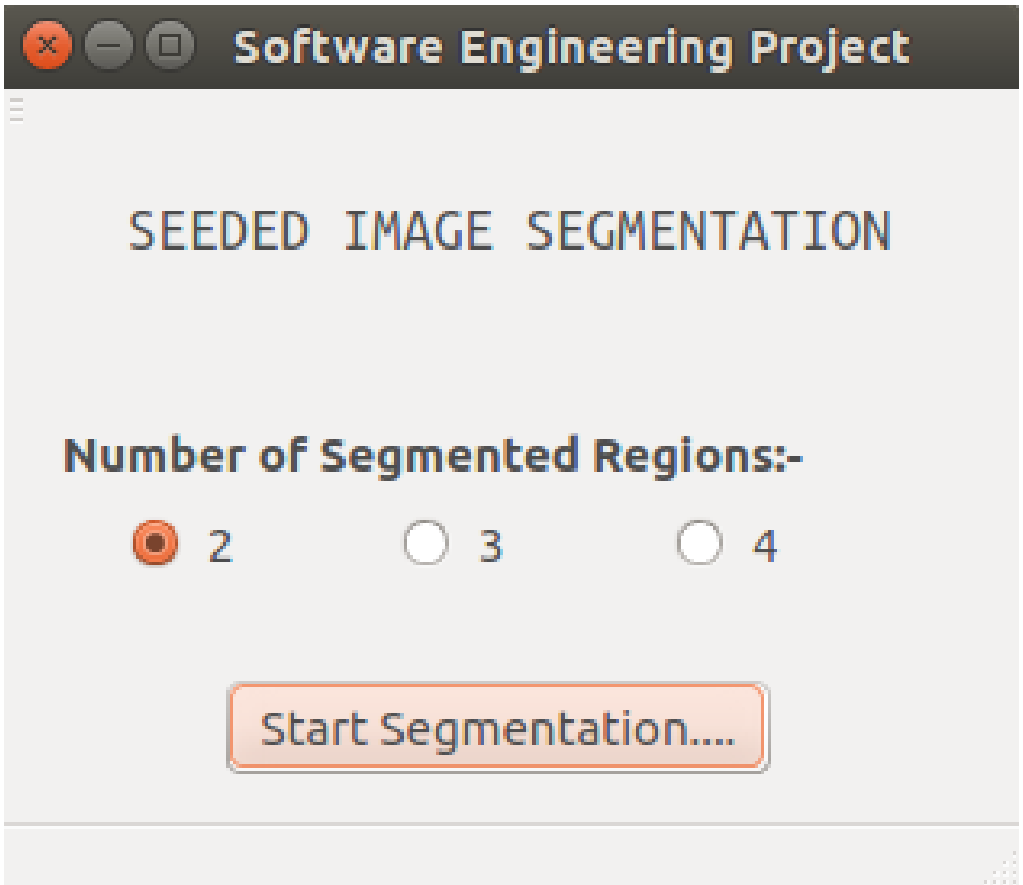
- SeededImgSeg class has methods for -
  - Computing seed independent variables
  - Computing seed dependent variables
  - Solve linear system of equations for segmentation
- It also has constructors and destructors. Assignment operator overloaded.

Object Oriented Programming

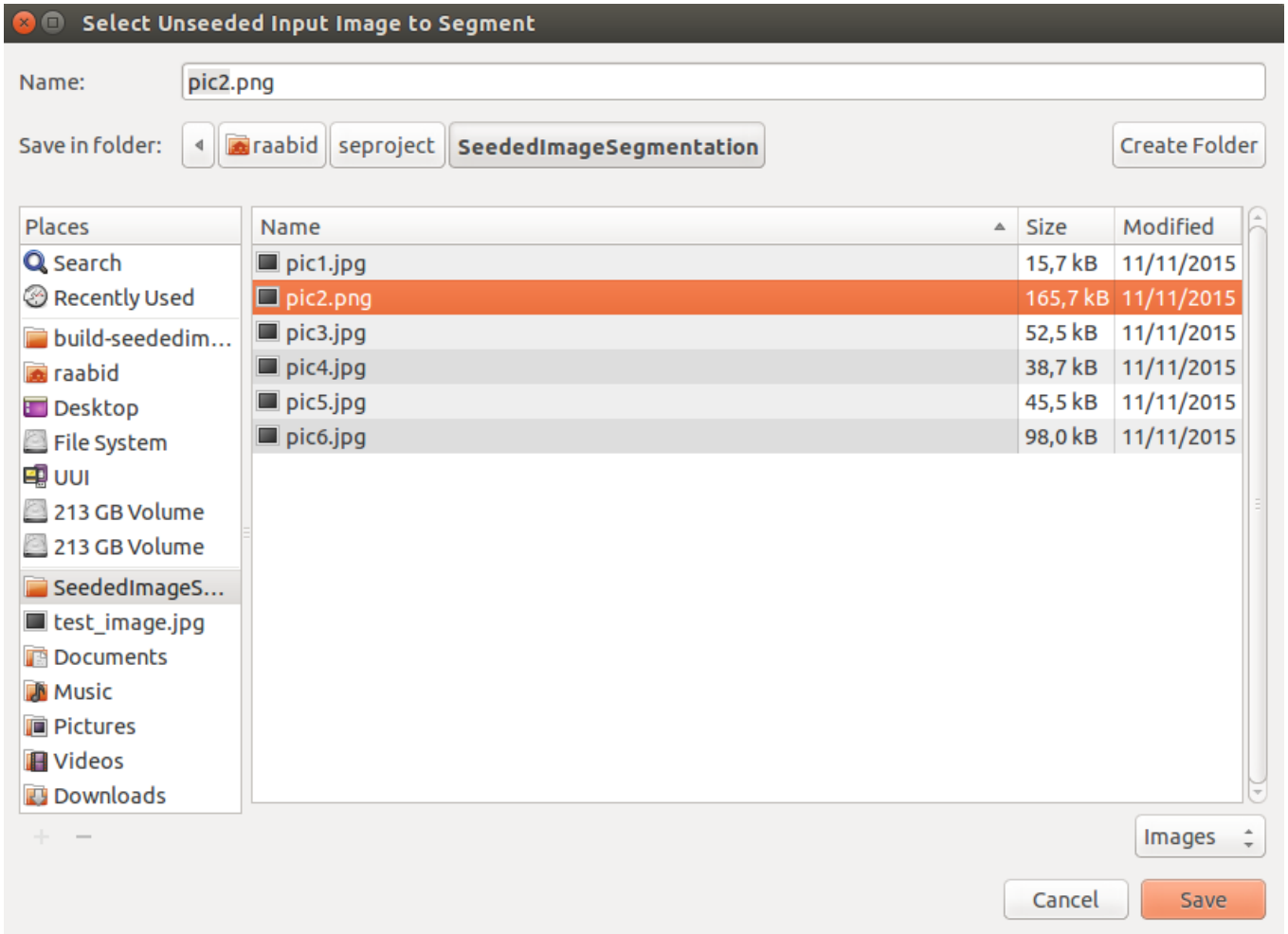
Benefits

- Code modular and reusable.
- Implementing Multi Region Segmentation was just modifying some methods.
- Linking with GUI was easier.

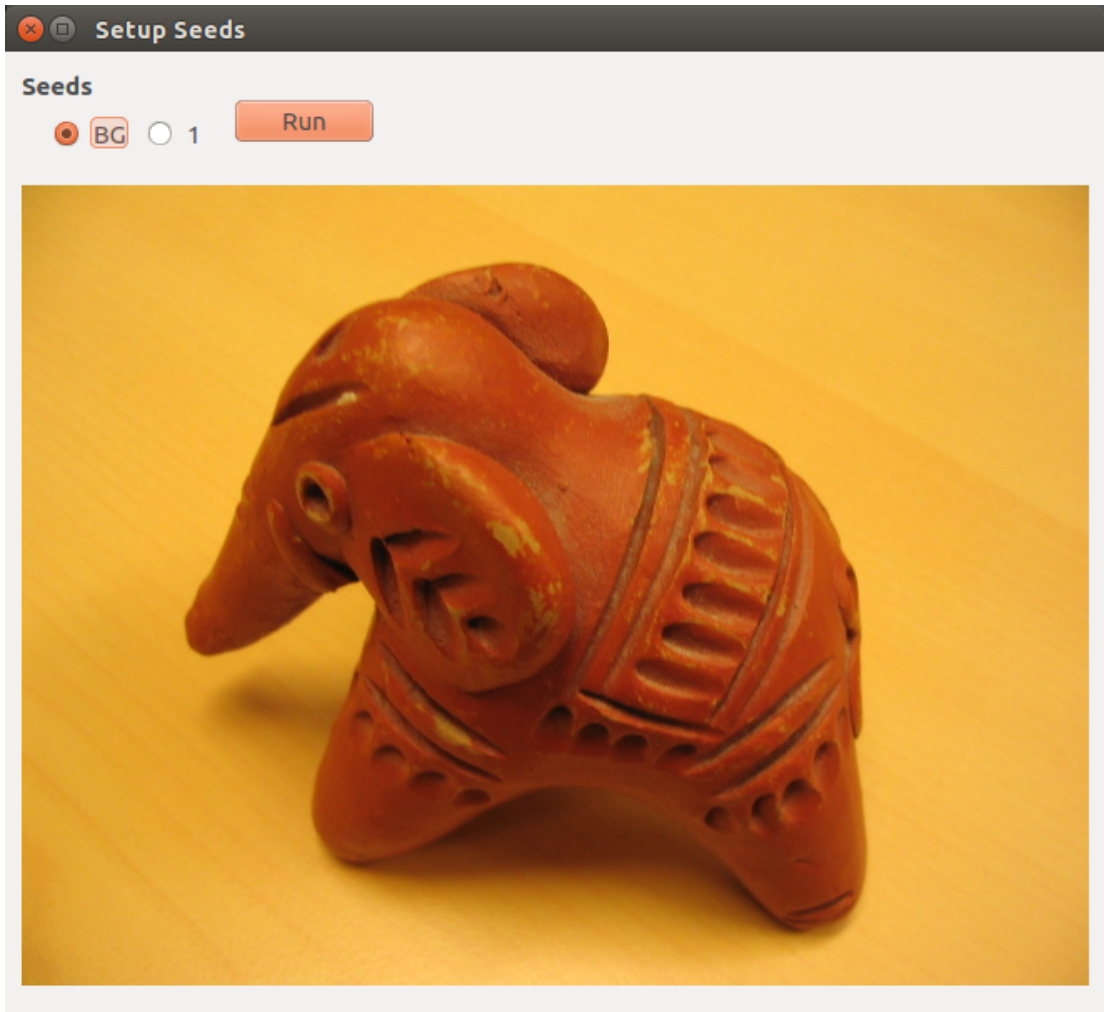
Graphical User Interface



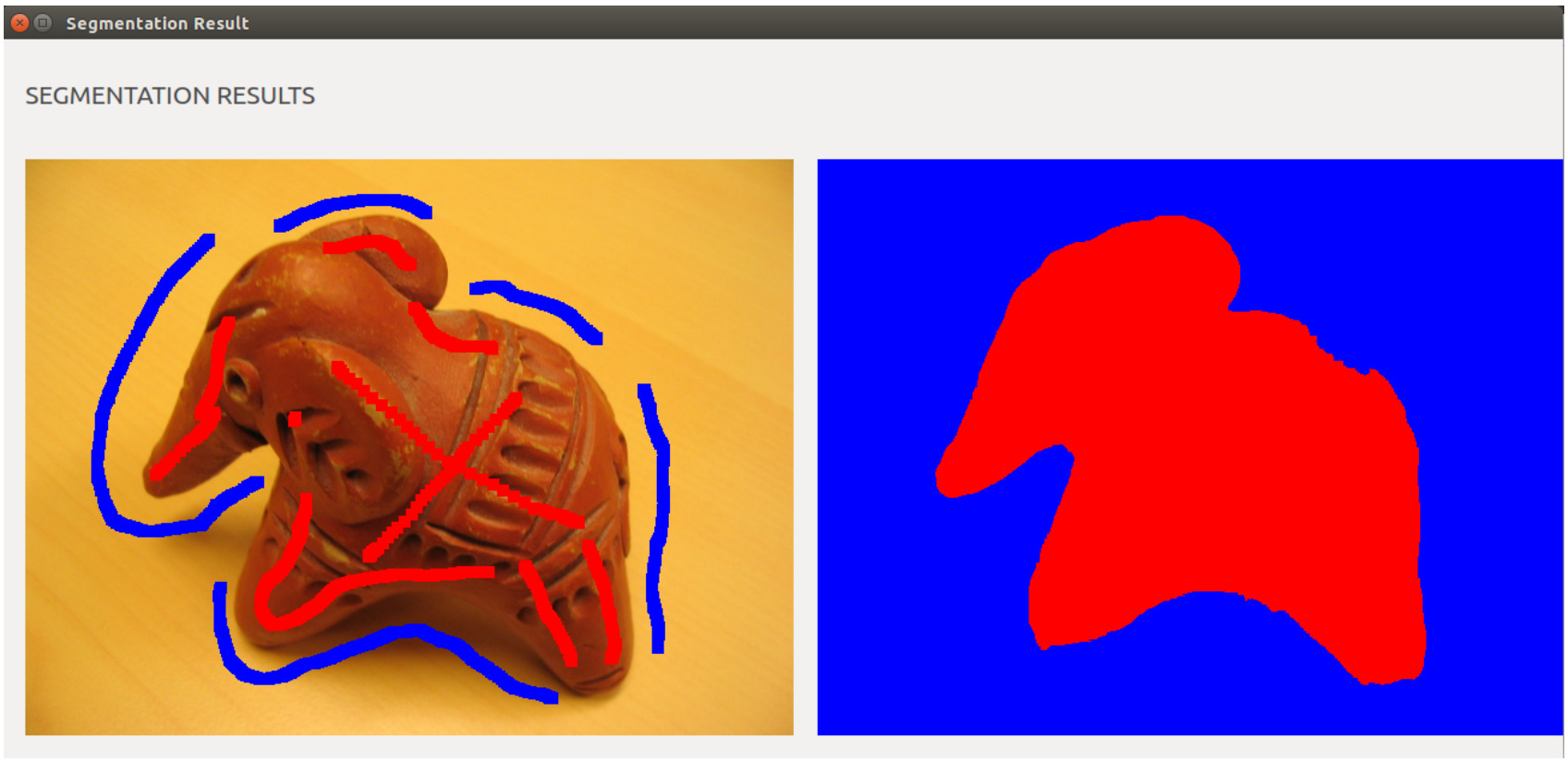
Graphical User Interface



Graphical User Interface



Graphical User Interface



Use of cmake

- Build Portability
- Link Dependencies
- Unit Test

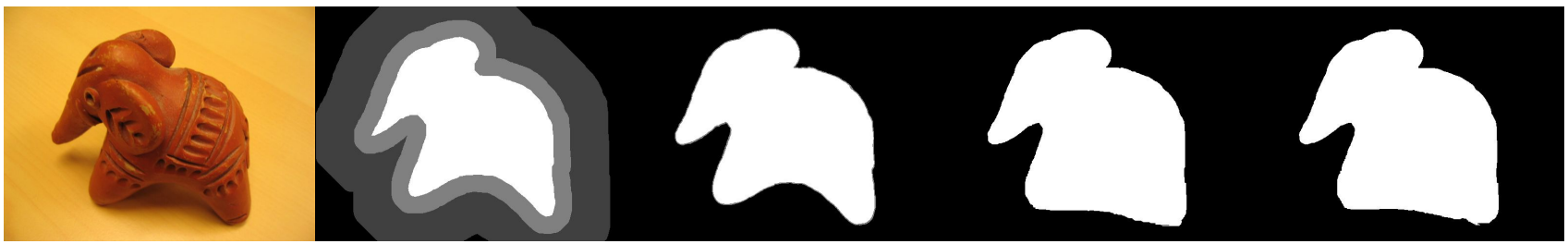
Results

- Two-region Segmentation
- Multi-region Segmentation

Comparison

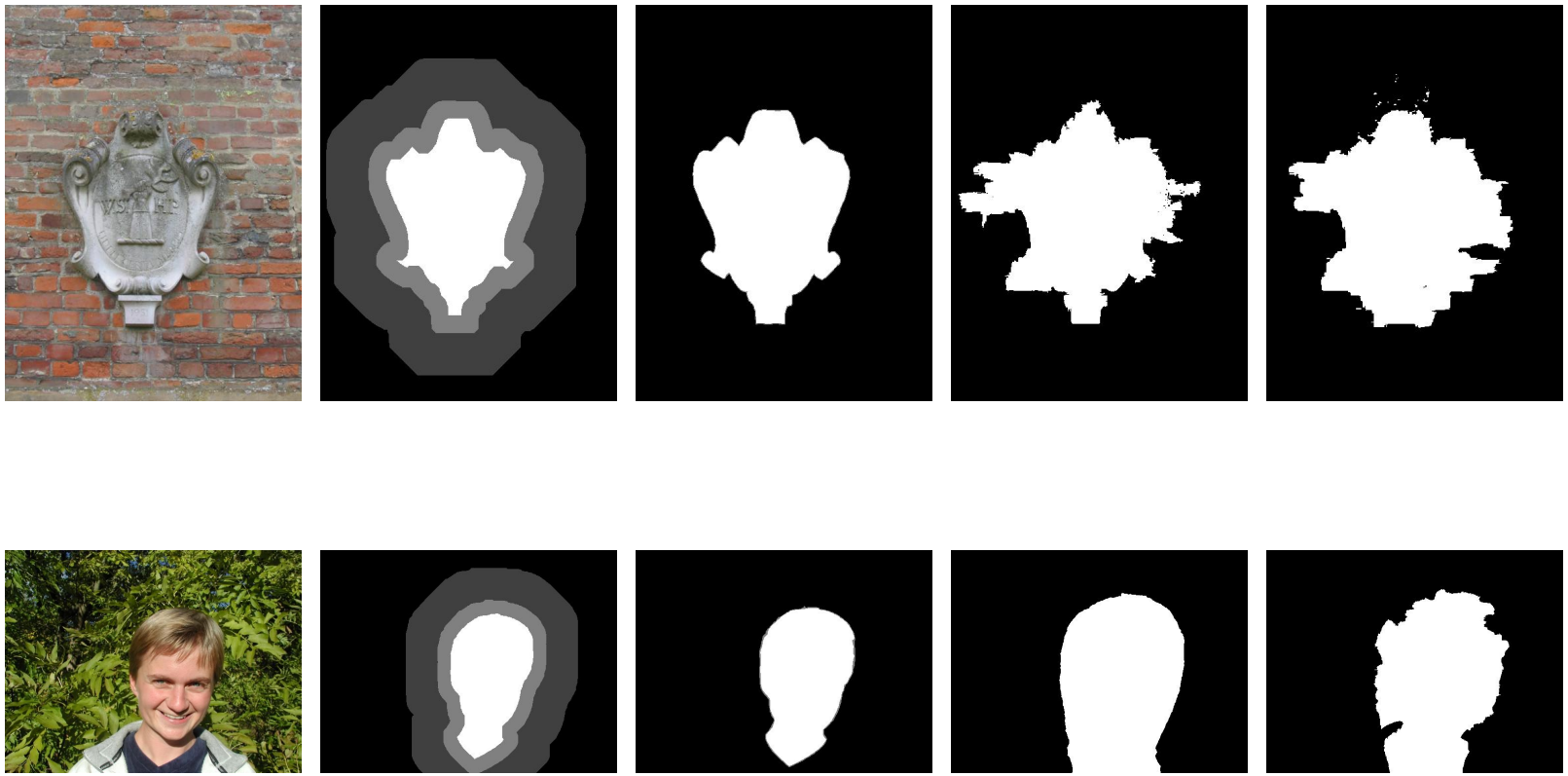
- A subset of "Grabcut" Dataset from Microsoft Cambridge.
- Compared with Random Walker method proposed by Grady.

Qualitative Comparison



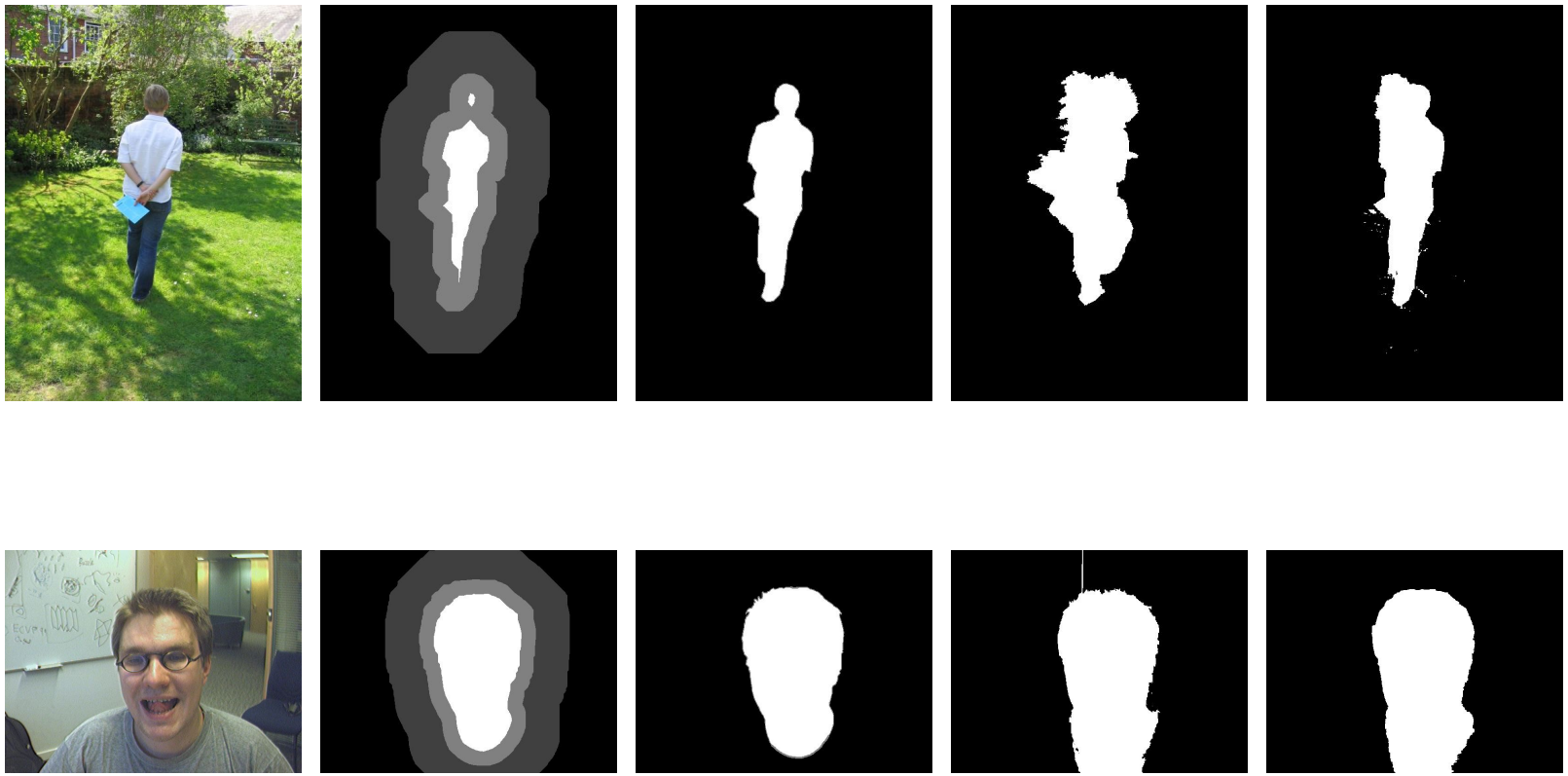
From left to right : Original Image, Given Seed Map, Ground Truth, RW Output, LC Output

Qualitative Comparison



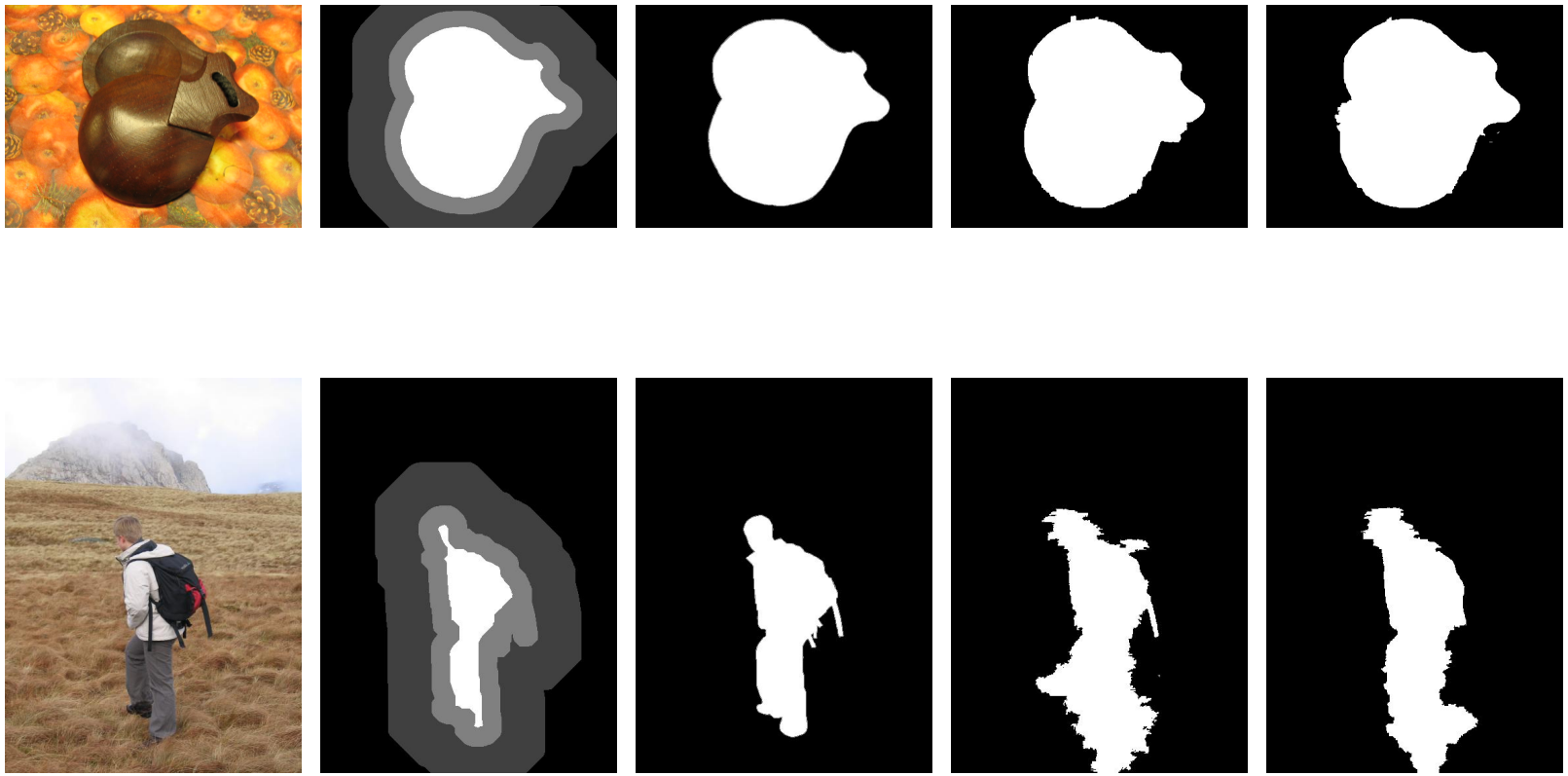
From left to right : Original Image, Given Seed Map, Ground Truth, RW Output, LC Output

Qualitative Comparison



From left to right : Original Image, Given Seed Map, Ground Truth, RW Output, LC Output

Qualitative Comparison



From left to right : Original Image, Given Seed Map, Ground Truth, RW Output, LC Output

Quantitative Comparison

- Rand Index (RI) values are compared.

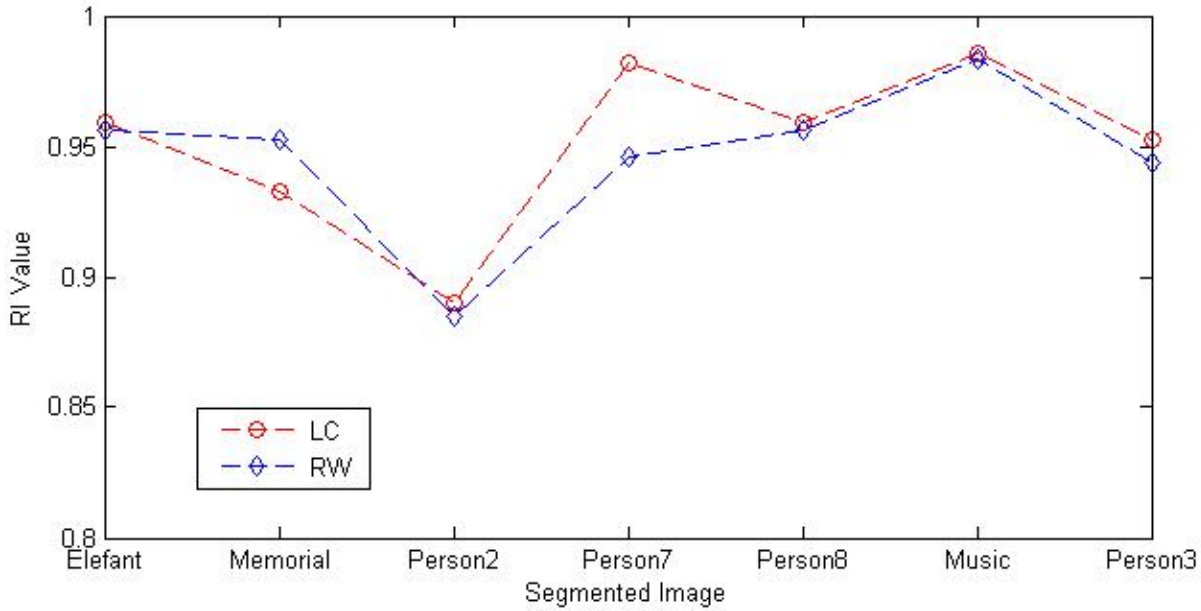
  RI = (Number of pixels in output with same label as ground truth image) / (Total Number of pixels)

- Higher RI values are better.

Quantitative Comparison

Image	RI Value for RW	RI Value for LC
Elefant	0.9565	**0.9589**
Memorial	**0.9525**	0.9324
Person2	0.8848	**0.8900**
Person7	0.9459	**0.9817**
Person8	0.9565	**0.9589**
Music	0.9830	**0.9859**
Person3	0.9436	**0.9527**
Average	0.9462	**0.9516**

Quantitative Comparison

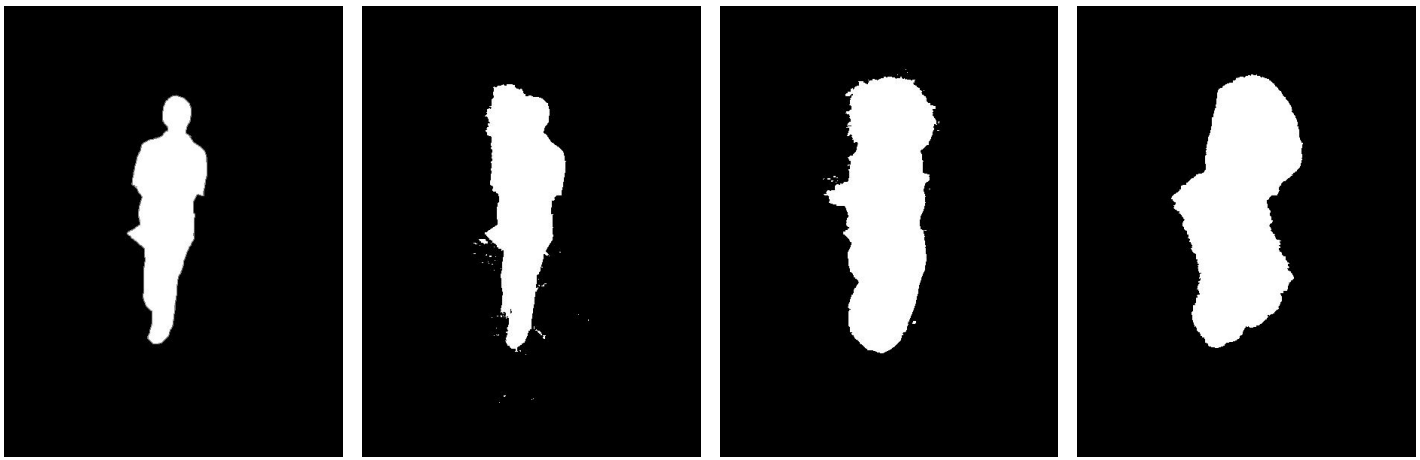


Comparison of RI Values for LC and RW methods

Effect of Parameters

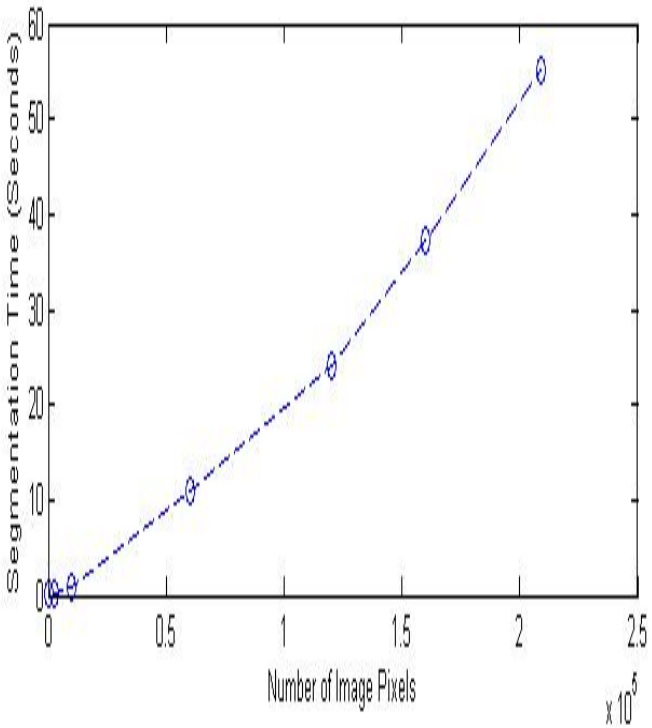
- Size of pixel neighborhood
- σ in pixelwise weight calculation
- β in pixelwise weight calculation

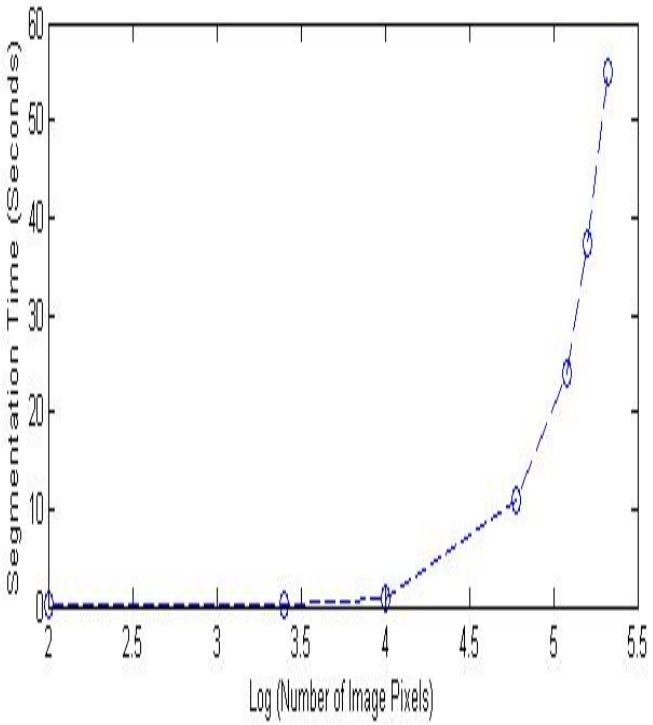
Effect of β



From left to right - Ground Truth, LC Segmentation with $\beta = 0.0025$ (RI $= 0.9817$), $\beta = 0.00025$ (RI $= 0.9386$), $\beta = 0.025$ (RI $= 0.9230$)

Execution Time





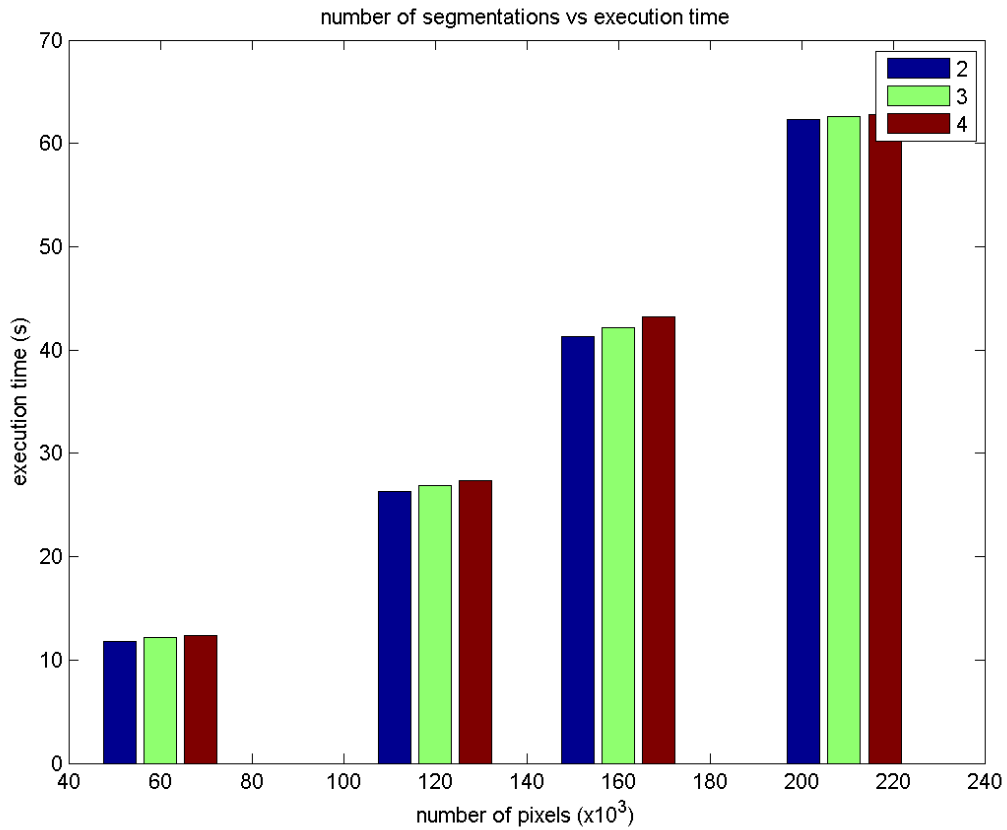
Execution Time versus number of image pixels

Multi-Region Segmentation

- Consider all seeds as foregrounds
- Best weighted result
- Repetion of code
- Minimizing

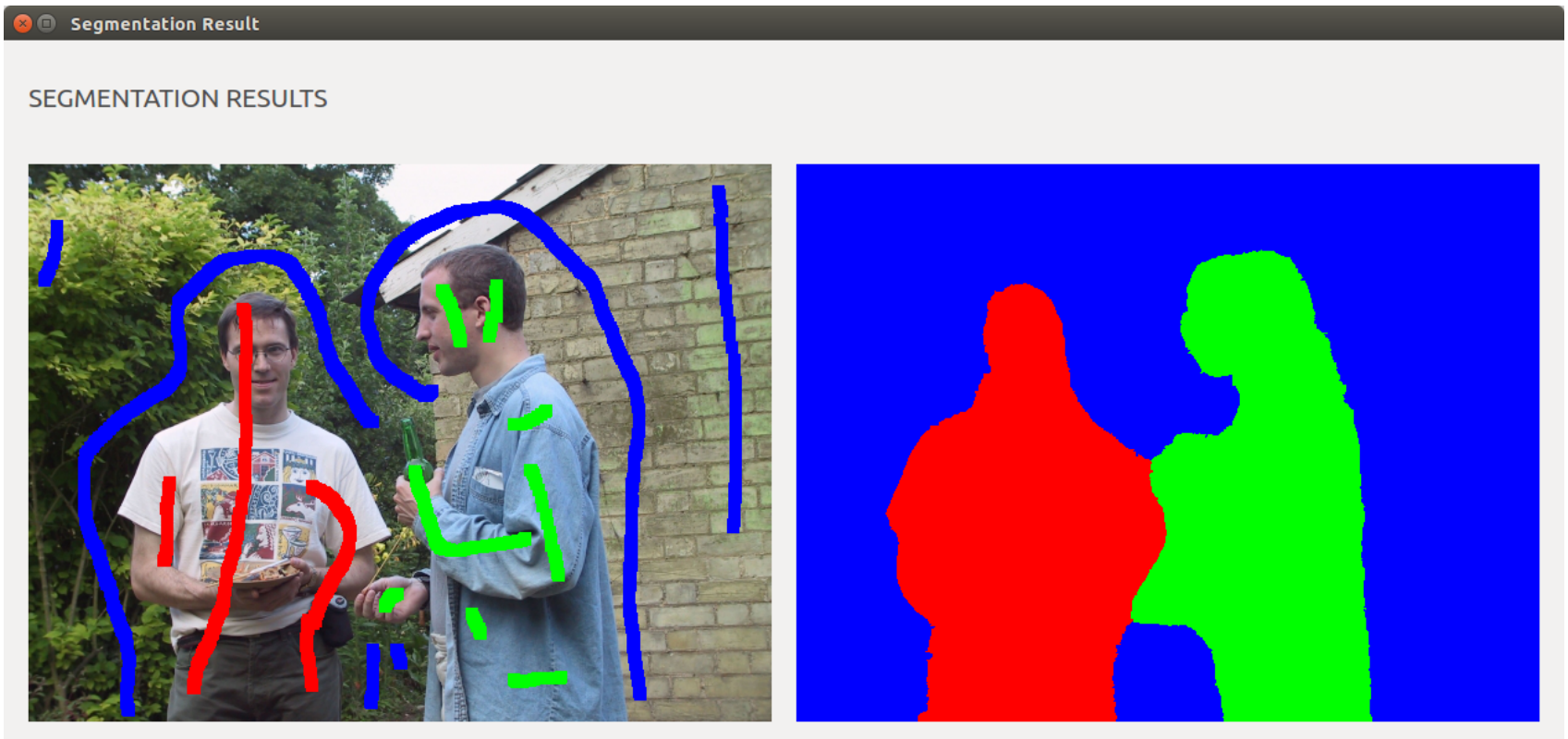
$$(I_s + L^2)x = b \tag{5}$$

Multi-Region Execution Time



Bar chart showing Execution Time changes with Number of Segmentations

Results



Project Management

- Github
- Microsoft Project

Timeline

Task Name	Start Date	End Date
Reading the research paper	27/09/2015	08/10/2015
Writing first version	10/10/2015	30/10/2015
Modifications and Optimizations	01/11/2015	12/11/2015
Testing	15/11/2015	18/11/2015
GUI Implementation	20/11/2015	03/12/2015
OOP implementation	22/11/2015	30/11/2015
GUI Integration	01/12/2015	03/12/2015
Multi-region Segmentation	05/12/2015	14/12/2015
Further Optimization	15/12/2015	21/12/2015
Testing and Comparisons	28/12/2015	30/12/2015
CMake Test	06/01/2015	08/01/2016
Writing the Report	15/12/2015	08/01/2016
Writing the Presentation	08/01/2016	10/01/2015

Table : Duration of Project tasks

Conclusions

- Algorithm based on Linear Algebra
- Implementation of the technique
- Optimization of code
- CMake and Unit test
- Comparison with state of the art methods
- Multi-region Segmentaion

References

- Casaca, W.,Nonato, L. G.,and Taubin, G. (2014, June). *Laplacian Coordinates for Seeded Image Segmentation.* In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on (pp. 384-391). IEEE.
- L. Grady. *Random walks for image segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(11):1768-1783, 2006.
- R. Unnikrishnan, C. Pantofaru, and M. Hebert. *Toward objective evaluation of image segmentation algorithms.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):929-944, 2007
- V. Jankovic. *Quadratic functions in several variables.* The Teaching of Mathematics, VIII:53-60, 2005

References

- http://opencv.org/
- http://eigen.tuxfamily.org/index.php?title=Main_Page
- http://eigen.tuxfamily.org/dox/group__TutorialSparse.html
- http://eigen.tuxfamily.org/dox/group__TutorialSparse.html
- http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm
- http://scikit-image.org/docs/0.10.x/auto_examples/plot_random_walker_segmentation.html

Thank You