

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# Automatic Alignment and Comparison of Petri Dish Images Containing Cell Colonies

by

Safar A. Alqahtani

Submitted to the Institute of Information & Mathematical Sciences  
in partial fulfillment of the requirements for the degree of  
Master of Information Science: Software Engineering

at

MASSEY UNIVERSITY

February 2016

© Massey University 2016. All rights reserved.



# Automatic Alignment and Comparison of Petri Dish Images Containing Cell Colonies

by

Safar A. Alqahtani

Submitted to the Institute of Information & Mathematical Sciences  
on Feb 22, 2016, in partial fulfillment of the  
requirements for the degree of  
Master of Information Science: Software Engineering

## Abstract

This work proposes a novel approach comprising of a chain of algorithms, for comparing, matching and aligning pairs of cell colony images taken at different stages on a Petri dish. The objective is to provide an assistive tool for microbiologists to quantify the loss or growth of cell colonies on two Petri dishes, by mapping cell colonies between a pair of images. This problem is highly non-trivial, as the shape, size and position of the corresponding colonies vary randomly. In addition, the cell colony images for comparison were taken at different times and from slightly different perspectives (i.e. effects of shearing); therefore, amplifying the complexity of the problem. Preliminary studies show that approaches purely based on SIFT or SURF, as well as algorithms used in astronomy, do not perform well on the problem domain. We therefore introduce a new approach to addressing these problems. A novel iterative technique that combines triangulation algorithms with the RANSAC alignment algorithm and AdaBoost classifier for alignment validation is proposed. Using 60 pairs of images of Petri dishes containing real biological cell colonies, we demonstrate the efficacy of the new algorithm in comparison to existing ones found in the literature. Empirical results show that the new proposed algorithm, we call K-NT for cell colonies matching, performed 4 times more accurate than other existing triangulation-based pattern matching algorithms. In the last stage of processing, we were able to generate an AdaBoost classifier with an accuracy of 98.5% that helps validate if an image was successfully aligned or not.

Thesis Supervisor: Dr. Andre L.C. Barczak  
Title: Senior Lecturer

Thesis Supervisor: Dr. Napoleon H. Reyes  
Title: Senior Lecturer



## Acknowledgments

I would like to thank my supervisors Dr.Andre & Dr.Napoleon for their tremendous support before, during and after the period of this thesis. Also a special thanks to Dr Austen Ganley for providing the Petri dishes images and thanks to Dr Teo Susnjak for his great insight in Software Architecture . I would also like to thank the open source communities for providing a wide range of application and tools that have helped me in research and made my life much easier. These tools include, but not limited to, OpenCV, KDE, Linux, GNU, Qt, and L<sup>A</sup>T<sub>E</sub>Xthe application that made this document.



# Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Significance of the Research . . . . .	16
1.2	Scope and Limitations . . . . .	17
1.2.1	Scope . . . . .	17
1.2.2	Limitations . . . . .	18
<b>2</b>	<b>Literature Review</b>	<b>19</b>
2.1	Introduction . . . . .	19
2.2	OpenCV . . . . .	19
2.2.1	Image Segmentation . . . . .	20
2.3	Feature Descriptor . . . . .	23
2.3.1	Scale-Invariant Feature Transform (SIFT) . . . . .	23
2.3.2	Speeded Up Robust Features (SURF) . . . . .	26
2.4	Matching Algorithms . . . . .	28
2.4.1	Fast Approximate Nearest Neighbour (FANN) . . . . .	28
2.5	Triangulation Methods . . . . .	30
2.5.0.1	Groth: Pattern Matching Algorithm . . . . .	31
2.5.0.2	Valdes: Pattern Matching Algorithm . . . . .	33
2.6	AdaBoost . . . . .	35
2.7	OpenMP . . . . .	37
2.8	Qt Application Framework . . . . .	38



<b>3</b>	<b>Methodology</b>	<b>41</b>
3.1	Criteria for Image Selection . . . . .	41
3.2	Segmentation . . . . .	42
3.3	Cell Detection . . . . .	44
3.4	Cell Colonies Matching . . . . .	44
3.4.1	Preliminary Experiments . . . . .	44
3.4.1.1	Delaunay Triangulation . . . . .	45
3.4.1.2	Circle Intersection Approach . . . . .	47
3.4.2	K-NT Algorithm . . . . .	50
3.4.3	An Example of Cell Colonies Matching . . . . .	53
3.5	Image Alignment . . . . .	54
3.5.1	OpenCV . . . . .	54
3.5.2	Automatic Alignment Evaluation . . . . .	55
<b>4</b>	<b>Results and Discussion</b>	<b>59</b>
4.1	Matching Algorithms . . . . .	59
4.2	The Proposed AdaBoost Classifier . . . . .	62
4.2.1	Further Optimizing the AdaBoost Results . . . . .	63
4.2.2	Weak Learners . . . . .	64
<b>5</b>	<b>The Complete System: Software Architecture &amp; Usage</b>	<b>69</b>
5.1	MainWindow . . . . .	70
5.2	ImageHolder . . . . .	70
5.3	DialogExtract . . . . .	71
5.4	FiltersCV . . . . .	72
5.5	DialogCompare . . . . .	73
5.6	System Walk-through . . . . .	73
<b>6</b>	<b>Conclusion and Future Work</b>	<b>79</b>
6.0.1	Summary of Contributions . . . . .	79
6.0.2	Future Work . . . . .	80

<b>A Tables</b>	<b>87</b>
<b>B Figures</b>	<b>95</b>
<b>C Code</b>	<b>99</b>



# List of Figures

1-1	Example of matching problem. top row: Petri dish images. Bottom row: segmentation and blob counter. Note that the blobs are different in each segmentation, and that the original images are noisy. . . . .	16
1-2	All Tested Algorithms Combinations. . . . .	18
2-1	OpenCV Modules . . . . .	20
2-2	Number of points to form the shape is $k = 3$ . . . . .	30
2-3	Groth's algorithm. . . . .	32
2-4	Valdes Triangle (VT) . . . . .	34
2-5	Triangle Space . . . . .	34
2-6	Qt Framework GUI Example . . . . .	39
3-1	Hough Circle Transform . . . . .	42
3-2	A pair of cell colonies in Manual + SURF modes . . . . .	43
3-3	Corresponding Points Example . . . . .	45
3-4	Example Images for Preliminary Experiment . . . . .	46
3-5	Applying Delaunay Method . . . . .	46
3-6	Applying Voronoi Diagram Method . . . . .	47
3-7	Applying Delaunay with Circle Intersection Method . . . . .	49
3-8	Applying Voronoi Diagram with Circle Intersection Method . . . . .	49
3-9	Nearest Neighbour N . . . . .	50
3-10	The top 10 matching triangles using Valdes's algorithm. . . . .	53
3-11	The top 10 matching triangles using K-NT n=5. . . . .	53
3-12	Triangle Points . . . . .	54

3-13	The alignment after applying RANSAC to (a) K-NT n=5 triangle and (b) Groth's matching triangles. . . . .	55
3-14	Flow chart of the current alignment rules . . . . .	56
4-1	One of the pairs of images used in the experiments. Refer to Table 4.1 for SURF and SIFT settings. . . . .	60
4-2	Classification using Red & Yello Ratio Values + Green & Yellow Ratio Values . . . . .	65
4-3	Classification using Red & Green & Yellow Ratio Values . . . . .	66
5-1	Software Architecture Overview . . . . .	69
5-2	MainWindow Class GUI: The numbers represent which class (Figure 5-1) handles the specific area. . . . .	70
5-3	ImageHolder Class GUI . . . . .	70
5-4	DialogExtract Class GUI . . . . .	71
5-5	FiltersCV Class GUI . . . . .	72
5-6	DialogCompare Class GUI . . . . .	73
5-7	The Main GUI . . . . .	74
5-8	Load Image Dialog GUI . . . . .	74
5-9	Image Cropping Dialog GUI . . . . .	75
5-10	Filtering Toolbox GUI . . . . .	75
5-11	Blobs Detection GUI . . . . .	76
5-12	K-NT Matching GUI Top . . . . .	76
5-13	K-NT Matching GUI Reverse . . . . .	77
B-1	Image t0.100.2 . . . . .	96
B-2	Image t0.200.1 . . . . .	97
B-3	Image t1.200.3 . . . . .	98

# List of Tables

4.1	Best parameters for SIFT and SURF. . . . .	61
4.2	SIFT & SURF performance example . . . . .	61
4.3	60 pairs results . . . . .	62
4.4	AdaBoost Accuracy Details . . . . .	64
4.5	Confusion Matrix . . . . .	64
5.1	RANSAC Matching GUI Aligned . . . . .	77
5.2	RANSAC Matching GUI Misaligned . . . . .	78
A.1	Triangulation Based Matching Algorithms Performance Summery . .	88
A.2	Figure B-1 Results . . . . .	89
A.3	Figure B-2 Results . . . . .	90
A.4	Figure B-3 Results . . . . .	91
A.5	Original Attributes . . . . .	92
A.6	Normalised Attributes . . . . .	93