SOFTWARE REQUIREMENTS SPECIFICATION

for

Edge Detection using ACO Algorithm

Version 1.0 approved

Prepared by- Somnath Paul (19MCMI09)
Kirti Sharma (19MCMI25)

School of Computer and Information Sciences

November 6, 2019

Contents

1 Introduction

1.1 Purpose

In this project a new algorithm for edge detection using ant colony search is proposed. The problem is represented by a directed graph in which nodes are the pixel of an image. To adapt the problem, some modifications on original ant colony search algorithm (ACSA) should be applied. Several experiments should be done, to found relationship between size of the image to be analysed and algorithm parameters. Several experiments should be made the results suggest the effectiveness of this algorithm.

1.2 Intended Audience and Reading Suggestions

This project will be built under guidance of University of Hyderabad professors. This project result will be accessible to all the university student for future use of their project or to built any tool. To use this product a user should have basic knowledge of computer programming language Java and Image Processing.

1.3 Project Scope

This project can be used for different projects of Image processing, Computer Vision, Machine Learning, Machine vision.

1.4 References

•Image Edge Detection Using Ant Colony : Anna Veronica Baterinaand Carlos Oppus: January 15, 2010.

•Ant System: Optimization by a Colony of Cooperating : Marco Dorigo, Member, ZEEE, Vittorio Maniezzo, and Albert0 Colorni : December 28, 1994

•Edge detection using ant algorithms: Hossein Nezamabadi-pour · SaeidSaryazdiEsmatRashedi: Published online: 1 August 2005 © Springer-Verlag 2005

•Edge Detection of an Image based on Ant Colony Optimization Technique , CharuGupta, Sunanda Gupta : 6 June, 2013

2 Overall Description

2.1 Product Perspective

It enables us to detect edges in an Image using optimal resources (CPU, MEMORY, TIME). Output of this product can be used in different aspects of image processing, computer vision, machine vision.

2.2 Product Functions

2.2.1 Edge Detection

This product should detect edge optimally.

2.2.2 Input

This product should take any image file.

2.2.3 Output

This product should give output in readable image file format.

2.3 Operating Environment

- Operating System: Windows 10 Home
- Compiler: JAVA

2.4 Design and Implementation Constraints

The major design constraint is the trade-off between parameters such as speed, area, and power consumption on one hand, and image quality on the other hand.

3 External Interface Requirements

3.1 System Interfaces

- Keyboard
  - Mouse

3.2 Hardware Interfaces

- System Model: Acer Nitro 5
  - Processor: Intel Core i5-8300H CPU @ 2.30Ghz 2.30Ghz
  - Memory: 8.00GB
  - Hard Disk: 1TB

3.3 Software Interfaces

- Operating System: Windows 10 Home
  - Compiler: JAVA

4 System Features

4.1 Description and Priority

In this project a new algorithm for edge detection using ant colony search is proposed. The problem is represented by a directed graph in which nodes are the pixels of an image. This project is of moderate priority as it is used in various image detection and recognition algorithms.

4.2 Stimulus/Response Sequences

- It should take a image file in .jpeg , .png format.
- It should give an output in readable image file format.

4.3 Functional Requirements

4.3.1 Edge detection using Ant Colony Optimization Algorithm

It should use only ant colony optimization algorithm.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

It should be fast enough to produce output.
This project should be completed within the stimulated time period.
The cost involved in marketing the project should be less.

5.2 Safety Requirements

This system must be highly robust. If there is extensive damage due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the image that was backed up to archival storage (typically tape) and reconstructs a more current state from the backed up log, up to the time of failure.

5.3 Security Requirements

This requirement is present, as this system will interact with the user.

5.4 Software Quality Attributes

5.4.1 AVAILABILITY:

This edge detection should work for every image file, regardless of their size and quality.

5.4.2 CORRECTNESS:

This edge detection software should give correct result for every input image, whether it's a color image or black and white image.

5.4.3 MAINTAINABILITY:

The algorithm or program should be changeable in future.

5.4.4 USABILITY:

This program should be used in other image processing related works.