Edge Detection Using Ant Colony Optimization Algorithm

Software Requirement Specification

A Software Requirement Specification (SRS) is a requirements specification for a software system that is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements that impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

An SRS describes the functionality the product needs to fulfil all stakeholders (business, users) needs.

A typical SRS includes:

- A purpose
- An overall description
- Specific requirements

The best SRS documents define how the software will interact when embedded in hardware — or when connected to other software. Good SRS documents also account for real-life users.

Characteristics of Good SRS:

An SRS should be:

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

Edge Detection Using Ant Colony Optimization Algorithm

Software Requirement Specification

1. INTRODUCTION & PURPOSE:

1.1. What is EDGE Detection means? What it does?

Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.

Common edge detection algorithms include Sobel, Canny, Prewitt, Roberts, and fuzzy logic methods.

1.2. What is Ant Colony Optimization Algorithm?

Real ants follow their own agenda of tasks independent from each other, however, when act as a community, they are capable to solve their daily complex problems, which require sophisticated planning, such as selecting and picking up materials, or, finding and storing foods. However, there is no kind of supervising or controlling. Finding the shortest route between the colony and a food source, is done by an exchange of information about the path should be followed. Ants communicate with each other by means of pheromone trail. They mark the path by leaving a certain amount of pheromone. Ants probabilistically prefer to follow a direction, proportional to the amount of pheromone on it. The more ants follow a trail, the more attractive this trail becomes to be followed by other ants. This process could be considered as a positive feedback loop.

An artificial ant colony system is an agent-based algorithm, which simulates the behaviour of real ants. Artificial

ants are like real ants with some major differences:

- Artificial ants have memory;
- They are not completely blind;
- They live in a discrete time environment.

However, they have some adopted characteristics from real ants:

- Artificial ants probabilistically prefer paths with a larger amounts of pheromone.
- Shorter paths have larger rate of growth in the amount of pheromone.
- The ants communicate to each other by means of the amount of pheromone laid on each path.

ACS is an iterative algorithm. At each iteration, it performs a loop containing two basic operations:

(1) construction or modification of solutions of the problem,

(2) updating the pheromone trails

1.3. What is the purpose of the project?

Edge Detection Using Ant Colony Optimization Algorithm

Software Requirement Specification

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.

Audience and Intended Use:

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.

Product Scope:

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

2. THE 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.

2.1.1. System Interface:

- Keyboard
- Mouse

2.1.2. Hardware Interface:

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

2.1.3. Software Interface:

- Operating System: Windows 10 Home
- Compiler: JAVA

2.1.4. Operations:

- It should take a image file in .jpeg , .png format

- It should give an output in readable image file format

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. User Characteristics:

- Knowledgeable user

3. SPECIFIC REQUIREMENTS:

3.1. External Interfaces:

- Command Line Interface

3.2. Performance Requirements:

- Optimal Edge Detection
- Should work for every image file
- Should give a readable image output file
- Should work on any environment

3.3. Functional Requirements:

In software engineering, a functional requirement defines a function of a software-system or component. A function is described as a set of inputs, the behaviour and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that show how a use case to be fulfilled. Typically, a requirements analyst generates functional requirements after building use cases. However, this may have exceptions since software

development is an iterative process and sometime certain requirements are conceived prior to the definition of the use case. Both artefacts (use cases documents and requirements documents) complement each other in a bidirectional process. A typical functional requirement will contain a unique name and number, a brief summary, and a rationale. This information is used to help the reader understand why the requirement is needed, and to track the requirement through the development of the system. The core of the requirement is the description of the

required behaviour, which must be a clear and readable description of the required behaviour. This behaviour may come from organizational or business rule, or it may be discovered through elicitation sessions with users, stakeholders and other experts within the organization. Software requirements must be clear, correct unambiguous, specific and verifiable.

3.3.1. Edge detection using Ant Colony Optimization Algorithm

It should use only ant colony optimization algorithm.

3.4. NON-FUNCTIONAL REQUIREMENTS:

In systems engineering and requirements engineering, non-functional requirements are requirements that specify criteria that can be used to judge the operation of system, rather than specific behaviours. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are “constraints”,” quality attributes”, “quality goals” and “quality of service requirements”. Qualities, i.e. non-functional requirements can be divided into 2 main categories:

1. Execution qualities such as security and usability are observable at run time.
2. Evolution qualifies, such as extensibility and scalability embody in the static structure of the software system.

The Non-Functional requirements of our project are:

- Time:
  This project should be completed within the stimulated time period.
- Cost:
  The cost involved in marketing the project should be less.
- Usability:
  This requirement is present, as this system will interact with the user.
- Reliability:
  This system must be highly robust.
- Performance:
  It should be fast enough to produce output.