Software Requirements Specification

CS684 Projects

Vector Image extraction and Autonomous Duplication by Firebird V

Group 5

Abhirup Ghosh	09305052
Anirban Basumallik	09305008
Gourab Roy	09305037

Table of Contents

1 Introduction

The project is aimed at implementing a system which can extract a vector image from a raster image and draw the same autonomously on a chart paper. The whole project will be open-source and the code is being implemented using coding standards and standard naming conventions. The code of the Visual Studio will be reusable, readable and portable. The AVR Studio code will be readable and reusable but not portable to other Microcontrollers. The hardware requirements are strict. The software requirements for the Visual Studio are strict. But for Microcontroller code, the IDE is not mandatory but the specific compilers and programmer is required.

1.1 Definitions, Acronyms and Abbreviations

- MSDN – Microsoft Developer Network
- OpenCV – Open Computer Vision
- ERTS – Embedded and Real-Time Systems

1.2 References

1. *A thinning algorithm by contour generation*, Paul. C. K. Kwok, Communications of the ACM, November 1988
2. *Raster-to-vector Conversion by Line Fitting Based on Contours and Skeletons*, Osamu Hori and Satohide Tanigawa, Research and Development Center, Toshiba Corporation1, Komukai Toshibic-cho, Saiwai-ku, Kawasaki 210, JAPAN
3. http://opencv.willowgarage.com/documentation/cpp
4. MSDN
5. Firebird V Hardware and Software manuals

2 Overall Description

Product Perspective: Automatic map follower.

Product Functions: The project is aimed at implementing a system which can extract a vector image from a raster image and draw the same autonomously on a chart paper.

User Characteristics: Academic purposes only

Constraints: Input needs to be a non-intersecting continuous image.

Assumptions and Dependencies: Illumination of the room is high and the friction of the tip of the pen attached to the robot is less.

Requirements Subsets: ZigBee modules, Firebird V and Visual Studio 2008.

3 Details

3.1 Functionality

- An image will be captured from a vertical source like wall, whiteboard, or projector screen using a webcam on a PC, where it will be processed.
- The image will be smoothened to prevent broken lines and remove noise.
- A vector image in the form of a continuous non-intersecting line will be extracted from the captured image.
- A path will be identified and a set of instructions to reproduce the path will be generated.
- The set of instructions describing the path will be communicated to the Firebird V robot using wireless communication.
- Then the shape will be drawn on a chart paper using a pen attached to the Firebird V robot.
- The communication will have a signature in the header that will prevent the robot from receiving signals from other wireless sources other than the designated PC.

3.2 Supportability

- Our code will be readable, reusable and portable (The PC side algorithm).
- Our code will pass the Splint Static analyzer.
- It will follow the naming convention mentioned at http://www.psgd.org/paul/docs/cstyle/cstyle.htm
- Our code will follow the indentation pattern inherent to Microsoft Visual Studio.
- Our code will follow the documentation standards of Microsoft Visual Studio.
- The AVR Studio code will follow the Doxygen standards.
- Our project will also follow the firebird specific guidelines and the project submission format as specified in the ERTS laboratory Coding Guidelines.

3.3 Design Constraints

- The lines of the image should be continuous and non-intersecting.
- The lines of the image should be straight lines.
- The pen used for drawing should have minimal friction.
- There should be adequate lighting during the capture of image.

- The board should be white enough to increase the contrast of the image.

3.4 On-line User Documentation and Help System Requirements

Microsoft Visual Studio 2008 standards documentation will be followed by using the inherent XML documentation generation facilities in the MS Visual Studio.

For the AVR Code, we will be using Doxygen.

3.5 Interfaces

3.5.1 User Interfaces

A menu-driven control system

1. Capture the image
2. View the thresholded image
3. View the thinned image
4. View the sequence of vectors extracted
5. View the sequence of movements
6. Signal the robot to trace the generated sequence of movements

3.5.2 Hardware Interfaces

Webcam:

- Capable of capturing RGB image.
- Capable of capturing images of resolution 640x480.

ZigBee module:

- Connected to ATmega 2560 through RxD0/PE0 (pin 2) and TxD0/PE1 (pin 3)
- Connected to the serial port of the computer
- AVR ISP mkII for programming the microcontroller

3.5.3 Software Interfaces

Microsoft Visual Studio 2008:

- Documentation for Visual Studio 2008:

http://msdn.microsoft.com/en-us/library/aa187919.aspx

- Documentation for VC++:

http://msdn.microsoft.com/en-us/library/60k1461a%28v=VS.90%29.aspx

OpenCV:

Documentation for Open Computer Vision for C++:

http://opencv.willowgarage.com/documentation/cpp

AVR Studio 4:

IDE for programming AVR micro-controllers

3.5.4 Communications Interfaces

The system has no external communication interface. It only has an internal communication interface which uses the wireless ZigBee modules.