DIGITAL IMAGE PROCESSING
FOR A MODEL RAILROAD

GRADUATE PROJECT

Submitted to the
Department of Computing and Mathematical Sciences
TEXAS A&M UNIVERSITY – CORPUS CHRISTI

BY
PAMELA CASTRO
SPRING 2001

COMMITTEE MEMBERS

R. Stephen Dannelly, Ph.D.
Chairperson

Holly Patterson-McNeill, Ph.D.
Member

David Thomas, Ph.D.
Member
Abstract

This project was the development of software used for image-processing that locates five model trains in the Real-Time Computing Laboratory at Texas A&M University - Corpus Christi. A server program was created to allow students access to the image-processing software located on the model train system control server, conductor.tamucc.edu. The image-processing software analyzes in real time the image files obtained from two ZoomCam webcams in a fixed position above the train track. Through image analysis, the software translates image data into track coordinates. The final product allows students to receive the location of the corresponding track numbers for each model train that is detected in the captured image.
# Table of Contents

Abstract ............................................................................................................. i  
Table of Contents .......................................................................................... ii  
List of Figures ................................................................................................. iv  
List of Tables .................................................................................................. vi  
I. Introduction and Background .................................................................... 1  
II. Train System Overview ............................................................................ 4  
    2.1 Related Projects .............................................................................. 4  
    2.2 Image-Processing Project .............................................................. 4  
    2.3 User Interface .................................................................................. 5  
III. Environment ............................................................................................ 6  
IV. Chronological Order of Events .............................................................. 7  
    4.1 Hardware and Operating System Configuration ......................... 7  
    4.2 Pure Images ................................................................................... 12  
    4.3 BGR Values for Actual Image ...................................................... 16  
    4.4 Algorithms Attempted ................................................................. 16  
V. Software Overview .................................................................................. 18  
    5.1 Algorithms ..................................................................................... 18  
    5.2 Design of Image-Processing Project .......................................... 25  
    5.3 Bitmap File Format ....................................................................... 46  
    5.4 Sample Images ............................................................................... 48  
VI. Module Details ....................................................................................... 51  
    6.1 Get Data Module ............................................................................ 51
6.2 Find Trains Module ........................................... 68
6.3 Output Data Module ........................................... 102
6.4 Misc Module ..................................................... 112
6.5 Configuration ................................................... 117
6.6 Server Program .................................................. 118
6.7 Makefile ........................................................ 118

VII. Conclusion and Future Work .................................. 120

7.1 Modifications .................................................... 120
7.2 Statistics ........................................................ 121
7.3 Future Work ..................................................... 122

References .......................................................... 125

Appendix A Color Table ........................................... 127

Appendix B Source Code on CD .................................. 130
List of Figures

Figure 1 - Modules Listing Including USB ....................................................... 8
Figure 2 - Modules Listing Including CPIA ....................................................... 9
Figure 3 - ./caminfo ....................................................................................... 10
Figure 4 - ./camstream .................................................................................. 10
Figure 5a and 5b - Image Settings ................................................................. 11
Figure 6 - Crontab Entries .............................................................................. 12
Figure 7 - Solid Box ....................................................................................... 13
Figure 8 - Single Box ..................................................................................... 14
Figure 9 - Multiple Box .................................................................................. 14
Figure 10 - Multiple Colors .......................................................................... 15
Figure 11 - Image Processing Software - Context Level DFD ....................... 26
Figure 12 - Image Processing Software - Level 1 DFD .................................. 28
Figure 13a - Image Processing Software - Level 2 - GetData Module DFD .... 30
Figure 13b - Image Processing Software - Level 2 - FindTrains Module DFD ... 32
Figure 13c - Image Processing Software - Level 2 - OutputData Module DFD ... 34
Figure 14 - Image Processing Software - Level 3 - readFile DFD ................. 36
Figure 15 - Image Processing Software - Level 3 - FindColors DFD ............. 37
Figure 16 - Image Processing Software - Level 3 - FindArea DFD ................. 38
Figure 17 - Image Processing Software - Level 3 - FindCertainty DFD .......... 39
Figure 18 - Image Processing Software - Level 4 - searchColors subroutine DFD ... 40
Figure 19 - Image Processing Software - Level 4 - mark subroutine DFD ....... 41
Figure 20 - Image Processing Software - Context Flow Diagram ................. 42
Figure 21 – Flowchart of Server Program ......................................................44
Figure 22 – Correlation between Server Program and Library (Client)Function......45
Figure 23 – Five Model Trains .........................................................................48
Figure 24 – Outpud for Five Model Trains..........................................................49
Figure 25 – Three Model Trains .........................................................................49
Figure 26 – Output for Three Model Trains ..........................................................50
Figure 27 – Module Design Modification ..............................................................120
List of Tables

Table 1 – Camstream Installation.................................................................9
Table 2 – BGR calculations for BLUE.........................................................16
Table 3 – Bitmap File Header.................................................................46
Table 4 – Bitmap Information Header...................................................47
Table 5 – Bitmap Color Table...............................................................48
Table 6 – GetData Subroutines...............................................................55
Table 7 – FindTrains Sub-Modules and Subroutines..............................69
Table 8 – FindColors Sub-Module..........................................................73
Table 9 – FindArea Sub-Module.............................................................84
Table 10 – FindCertainty Sub-Module....................................................96
Table 11 – OutputData Subroutines.......................................................103
Table 12 – misc Subroutines...............................................................113
Table 13 – Total Lines of Code..............................................................122
I Introduction and Background

The Real-Time Computing Laboratory, housed in the Computing and
Mathematical Sciences Department at Texas A&M University – Corpus Christi, was
created to serve as an active learning environment for computer science students.
Professors can create new and innovative laboratory assignments rather than using
traditional teaching practices for a broad range of computer science concepts. Students
have the opportunity to interact with Pioneer Mobile Robots, Lego Mindstorm Robots,
and model trains to give them a more “hands-on” experience.

The main goals of the Real-Time Computing Laboratory were to have the
facilities to make available new computer science concepts, to allow students to remotely
access a model train system, and to complete assignments from anywhere – not just from
inside the laboratory. Students will be able to remotely program the model trains, but
need a way of detecting the locations of the model trains if not physically in the
laboratory. Although block detection sensors are currently being added to the model train
tracks, there is no way to determine which train is occupying a block. It could be any of
the five model trains or even something that was accidentally placed on the track.

One solution was to mount two ZoomCam webcams (webcams) above the model
train system and create an image-processing software to detect the model trains’ locations
in the images. The webcams captured images in specified time intervals and displayed
them on a World Wide Web (Web) page. Students will access this Web page when they
need to view the model train system.

This image-processing project has allowed students to create a client program that
connects to a server process which launches the image-processing software that is located
on the server machine conductor.tamucc.edu. The locations of the model trains are
determined by finding the corresponding track numbers they reside on at time the image
was captured. Thus, students can use this software whenever the locations of the model
trains are needed. Students can concentrate on the specific purpose of their program and
not get side-tracked with the details that occur while the image-processing is
accomplished.

This image-processing project creates a new tool that can be used in computer
science courses in existence and those that will be developed in the future from the new
facilities housed in the Real-Time Computing Laboratory. It will allow professors more
diversity in the programs assigned to the students. Courses such as Data Structures, X-
windows, Design and Analysis of Algorithms, Computer Graphics, System Simulation
and Modeling, Image Processing, Computer Vision, and Systems Programming can take
advantage of this image-processing software to create new and interesting laboratory
assignments and lectures for students.

There is a significant difference in what each course will use the image-
processing software for. Some courses can use the image-processing software as an
indirect tool needed to complete assigned programs. Others may look at the image-
processing software and focus solely on how it operates. The X-windows, Computer
Graphics, and System Simulation and Modeling courses can use it to assign graphics
programs related to the model trains. For example, a program can ask for the trains’
locations several times to get the track numbers of the model trains for each image
desired. Using these track numbers, the students can simulate the motion or route the
model trains underwent during their travel. Courses such as Data Structures and Design
and Analysis of Algorithms can use the image-processing software directly, since it allows students to analyze the chosen data structures and algorithms applied to the function to successfully complete analysis on the image. Image Processing and Computer Vision can use the image-processing software directly by allowing students to use it as a reference on image analysis to create other image-processing programs for the Real-Time Computing Laboratory. System Programming can analyze the client-server applications used to reduce time spent processing images, improving the efficiency of the software.

The images captured by the webcams are digital representations of the actual model railroad system. A digital image is discrete and has a limited number of colors compared to actually seeing the image with your eyes. Your eye vision is continuous, (i.e., has no gaps in the color spectrum) and interprets images three dimensionally. The digital image is two-dimensional and lacks depth perception. Interpreting distance is suggested from the size of the objects in the image. Also, when an image is digitized, the data is compressed allowing for significant loss in data.

Image-processing converts digital images from their pictorial representation into their data representation in order to analyze the image. The data representation is an array of pixels that create the image. Image Analysis attempts to calculate information needed for the image using only those pixels. It groups pixels that have a common characteristic and categorizes them as belonging to the same object. Characteristics that can be recognized include shapes, colors, size and brightness.