I. Mid-Course Report

[1] Naqeeb Ameeri and Ali Seraj, entitled - ***Motion Detection and Object Tracking in Real Time.*** The Technical Faculty Advisor for this project was Professor Bir Bhanu.

This is an excellent, short and informative mid-course report that was submitted at the beginning of the second quarter for the work performed primarily up to the end of the first quarter.

Midcourse Report of Senior Design Project
Winter/Spring 2001
Professor Mansour Eslami

The University of California at Riverside
College of Engineering
Department of Electrical Engineering

Motion Detection and Object Tracking in Real Time

Prepared by: Naqeeb Ameeri & Ali Seraj

Technical Faculty Advisor: Professor Bir Bhanu

This project is supported by the Visualization and Intelligent Systems Laboratory at the University of California Riverside

1. Introduction

Background Information

Motion detection has been a key issue for researchers in the field of computer vision and image processing for many years [7]. In particular, computer image processing involving large images of XGA (extended Graphics Array, 1024X768) resolutions have not been possible until recently. Images are stored as large matrices in a computer system's memory. These matrices hold the pixel location, brightness and color value at that location. Images containing 256 different color levels at XGA resolution can become very large, anywhere from 100 kilobytes to 6,291 kilobytes (6.2MB) [6]. Current computer technologies, including memory and processor speed, are now powerful enough to handle complex computation involving matrices. Modern computer systems also have the ability to perform image processing in real time. The phase "real time" has several interpretations. For our purposes, this term is used as a performance characteristic describing the speed at which the images are processed. Using very simple motion extraction techniques we are able to determine the presence of motion. This brings us to our next topic: motion tracking.

1.1 Problem Statement

A Historical Perspective

Once motion has been detected, how can we determine its direction and track its movement? There are several methods that can be utilized. Everyone wants to use the simplest approach, which requires the least resources in terms of time and computing power. The approach should not require the construction of complex electrical hardware and should use existing technology. Preferably, this method should use digital image processing as the means of detection, and software as the means of tracking the object. The simplest approach should also utilize a low-level block operation.

Computers vision tasks break down into a hierarchical of low, medium, and high-level block operations. Low-level operations perform tasks such as edge detection. Edge detection is a common approach for detecting meaningful discontinuities in gray level [3]. Thus, an edge corresponds to a sharp change in brightness of an image. This sharp change in brightness usually corresponds to boundaries between objects. Our project relies on continuously detecting the edge of an object in motion. Thus, the algorithms being used are all low-level operations. There are many edge detection algorithms available but the chosen one for this project is thresholding. On top of that, low-level operations include pixel-by-pixel differencing and mask filtering. These methods are typically the fastest since they involve simple arithmetic operations on matrices.

The next operation in the hierarchical machine vision world is medium-level processing. In this level, edges need to be refined into useful subsets. These subsets can later be grouped together for different purposes such as outlining objects. Our project is not dependent upon subsets and thus, does not utilize medium level processing.

The last of the machine level operations is high level processing. High-level processing involves finding regions within boundaries. At this level of processing, some specific knowledge of the application is often needed. Thus, prior information about an image is needed to recognize objects of interest [1]. Once again, this level of operation is well beyond simple motion detecting and thus, is not utilized. Based on these historical considerations and applications, we offer our problem statement as follows.

Problem: Given the presence of motion of an object, what is the best way to determine its direction of movement and track the object?

1.2 Problem Solutions

Applicable Solutions

1. Sensor based systems using micro-controllers.
    - Applications: One possible solution to the given problem is the use of motion sensors and micro-controllers. A micro-controller is an integrated circuit special purpose chip used to control other devices. It contains an ALU, memory, and I/O circuits. It is a special-purpose device suitable for a variety of applications. Typically, a micro-controller must respond to inputs within a short period of time to produce outputs [4.1]. The inputs correspond to data coming in from the sensors. The outputs correspond to signals directing the motors to move in the appropriate direction. The sensors could be of various types. One type of sensor is a motion sensor. If 5 motion sensors are laid out to cover a range of 180 degrees for example, each sensor corresponds to an area of 30 degrees. The sensors collectively identify the location of the object [5]. Thus, as an object moves, each corresponding sensor is tripped and this data feeds into the micro-controller, which directs the motors to move a camera to the appropriate position. The advantages and drawbacks are as follows:

    - Drawbacks:
        - Limited coverage area of sensors.
        - In-depth hardware and software interaction, which can be difficult to troubleshoot.
        - Cannot perform image analysis.

    - Advantages:
        - Often inexpensive to design and build.
        - Highly portable.

2. Transmitter/Receiver based system.
    - Applications: A second possible solution to the given problem is the use of a transmitter and receiver based system. In this system, the object would carry a transmitter, which would transmit a signal to 3 receivers. Through simple geometry and mathematics, the objects position can be triangulated [3]. The advantages and disadvantages are as follows:

    - Drawbacks:
        - Object that is being tracked must carry a transmitter.
        - Real World objects cannot be tracked using this approach.
        - Cannot perform image analysis.
        - Receiver range can be limited.

    - Advantages:
        - Object can be tracked regardless of surrounding environment.
        - Does not require complex computer systems to implement.

3. Image processor based system.
    - Applications: A third possible solution to the to the given problem is the use of an image processor. An image processor is a special type of computer dedicated for special image processing functions. The purpose behind this special piece of hardware is that computations can be done a lot faster and often-in real time over a general central processing unit. Algorithms are written for the image processor and are inputted via a serial port on the Sun microcomputer. Once the image processor is done computing, the output is feed back into the Sun through the serial port once again. During this time, the Sun is able to perform other processing tasks. Thus, the use of the image processor helps implement many time dependent functions.

    - Drawbacks:
        - Systems can be very expensive depending on application.
        - Requires complex and powerful software and computer systems.
        - Not easily portable.
        - Complex algorithms must be written in order to extract object movement from surroundings for certain applications.
    - Advantages:
        - Can perform image analysis and extract movement based upon models.
        - Adaptable to most real world situations when motion detection and tracking are required.
        - Software programmable image processor makes changing applications relatively simple.

- These systems are often extremely fast allowing real time processing.

2. Selected Approach

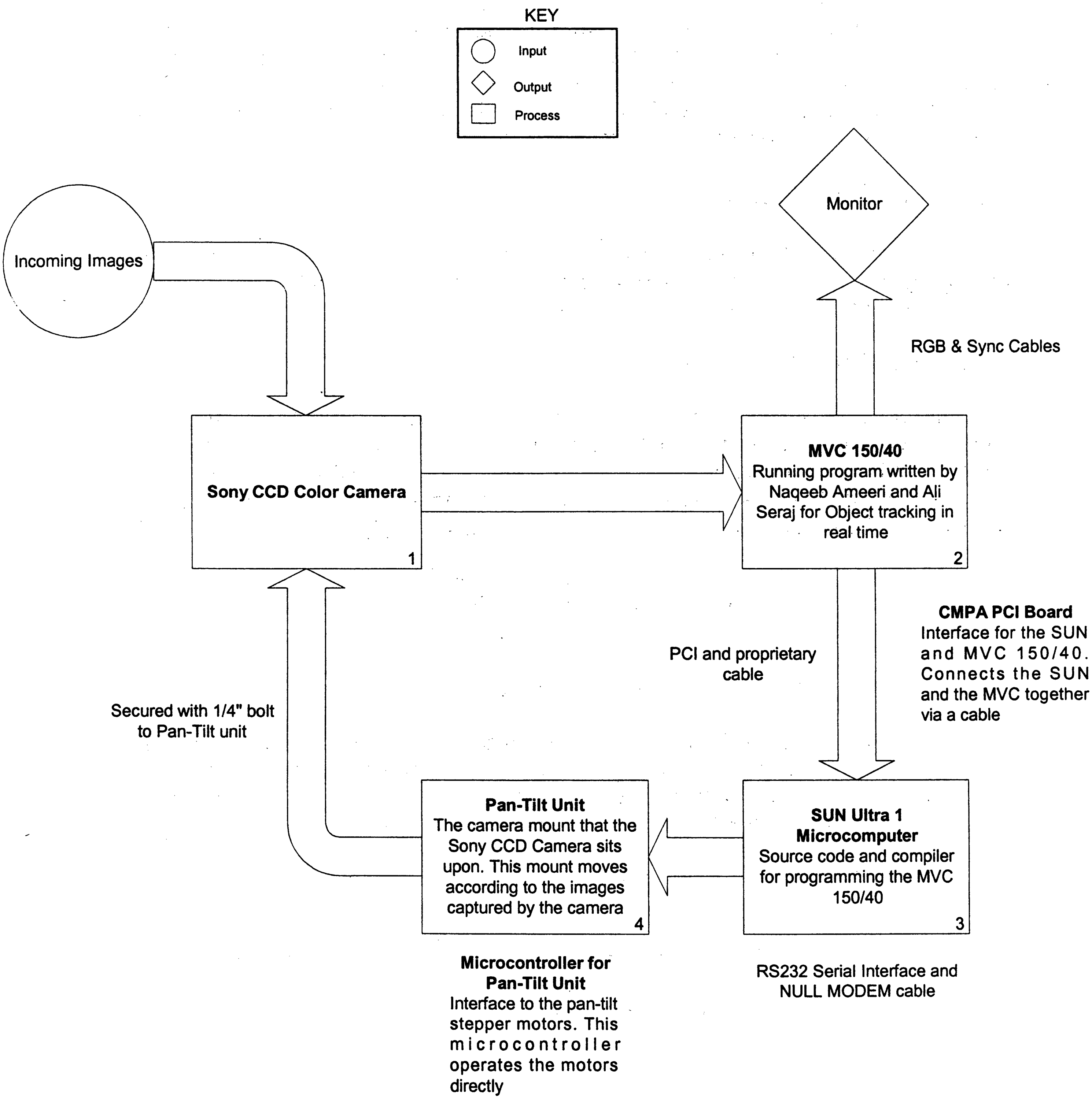
Our selected approach utilizes the image processor based system. We decided to use this system due to the advantages listed above. Also, an image processor is readily available in the Visualization and Intelligent Systems Laboratory at UCR.

Our proposed senior design project involves the use of the MVC 150/40 image-processing computer from Imaging Technologies. The MVC image processor can process images in real time, making it a powerful tool for motion detection, object tracking and various other tasks that most personal computers have difficulty doing in real time. Our project makes use of the MVC 150/40 by detecting and tracking a moving object and by the intervention of command signals to the pan-tilt controller. The pan-tilt controller operates two stepper motors, each responsible for moving the camera mount pan and tilt planes. The controller can interface with a computer through a standard RS-232 serial connection. Commands can be issued through the serial interface and the pan-tilt mount responds accordingly. We use this serial port to issue commands based on the processed images and move the mount accordingly. Thus, the basis for this project involves coding the interface to the pan-tilt unit using the ITEX programming library, as described next.

The ITEX programming library was designed especially for the MVC 150/40. It contains all of the routines and functions that are necessary to use the computer and control the various hardware features exclusive to the MVC 150/40. This includes functions for the onboard serial interface as well as all of the image capture and processing functions. Our design will add to the code, which was designed by previous students. The goal of this project is to develop a suitable program and algorithm that can keep an object centered in the camera's image plane at all times. All of the code that has been previously written for the computer involves motion capture and object tracking only. As of yet, no attempts were made to create an interface for the pan-tilt camera mount. This project establishes a foundation for an automated vehicle control system using the UC Rover.

3. Technical

System Block Diagram:



3.1 Analytical Discussions

A. Difference Equation for determining motion:

One of the simplest approaches for detecting changes between two image frames $P_1(i, j)$ and $P_2(i, j)$, is to compare the two images pixel by pixel [4]. If the region of interest allows frames to be captured before and after some time frame, then the pixel-by-pixel subtraction of the two images will yield a picture of what is changed [1]. The subtraction of the two images is performed straightforwardly in a single pass. The output pixels are given by the Difference Equation:

$$Q(i,j) = P_1(i,j) - P_2(i,j)$$

Thus for motion detection, stationary objects cancel each other out while moving objects are highlighted when two images of the same dynamic scene, which have been taken at slightly different times P1 and P2, are subtracted.

B. Lowpass Filtering for noise reduction:

Lowpass filters are used for noise reduction. This is primarily done through blurring. Blurring removes small details from an image such as noise. Pixels that are out of place in a connected cluster are due to noise and must be filtered out [3]. The pixels that belong to an 8-connected component are retained for further analysis. The following 3x3 kernel is used for the non-linear Lowpass filter [4].

LPF = 1/16 *

1	2	1
2	4	2
1	2	1

C. Centroid Detection for a two dimensional image:

After the Difference Equation and Lowpass filter have been applied, the center of the image must be determined in order for a coordinate system representation to be established. The Centroid Detection Algorithm from J. Nguyen and C. Home [4] is used to find the center of an object and is shown here due to its importance.

Centroid Detection Algorithm

1. Initialize variables X, Y, and N to zero
2. Scan the image from left to right and from top to bottom.
3. If the pixel is 1 (black) then,
   3.1 Increment N by 1.

3.2 Increment X by x coordinate value + 1.
3.3 Increment Y by y coordinate value + 1.
4. If there are more pixels to consider, then go to step 3.
5. Calculate Centroid: {Xcent,Ycent) = (X/N-1, Y/n-1)

Equipment List:

- **Sun Ultra Sparc 1 Microcomputer**: The Sun Ultra 1 is where the program source code is written and compiled. The Sun Ultra 1 microcomputer is the interface between the programmer and the Modular Vision Computer.
- **Modular Vision Computer** (MVC 150/40): The MVC 150/40 is the bridge between the camera, pan-tilt unit, and the Sun computer. The MVC 150/40 performs all of the real time object tracking and image processing and sends control commands to the pan-tilt unit.
- **Pan-Tilt Unit**: The pan-tilt unit is a 2 axis, 2 degrees of freedom camera mount. The mount is stationary and can pan around 1 axis, and tilt along a second axis. The unit contains two stepper motors each responsible for the pan angle and the tilt angle respectively. Both motors are controlled by an outboard microcontroller.
- **Sony CCD Color Camera**: The CCD camera is the source of images that the MVC 150/40 performs calculations upon.

Software:

ITEX programming library, GNU gcc ANSI C compiler, GNU Emacs Text editor

Special Resources:

Visualization and Intelligence Systems Laboratory
(Marlan and Rosemary Bourns College of Engineering – University of California, Riverside)

4.0 References

[1] R. Boyle, R. Thomas, *Computer Vision: A First Course.* Great Britain: Blackwell Scientific Publications, 1988.

[2] W. Burger, B. Bhanu, *Qualitative Motion Understanding.* Norwell, Massachusetts: Kluwer Academic Publishers, 1992.

[3] R. Gonzalez, R. Woods, *Digital Image Processing.* Reading, Massachusetts: Addison-Wesley Publishing Company, 1993.

[4] J. Nguyen and C. Horne, *Tracking moving objects in Real-Time*, Senior Design Report, Electrical Engineering Dept., University of California Riverside, Riverside, CA, 1999

[5] P. Spasov, *Micro-controller Technology.* Upper Saddle River, New Jersey: Prentice-Hall, Inc.
1992

[6] S. Tzafestas, *Microprocessors in Robotic and Manufacturing Systems.* The Netherlands: Kluwer Academic Publishers, 1991.

[6] D. Vernon, *Machine Vision, Automated Visual Inspection and Robot Vision.* London: Prentice Hall, 1991.

[7] M. Vincze, G. Hager, *Robust Vision for Vision-Based Control of Motion.* Bellingham, Washington: SPIE Optical Engineering Press, 1995.