Autonomous Video Feedback Controlled Surveillance Using an Embedded DSP System

Patrick Flaherty and Edward Chung
Department of Electrical Engineering
Rochester Institute of Technology
79 Lomb Memorial Drive, Building 9, Rochester, NY 14623
www.ee.rit.edu

***Abstract* – Transmission of video data over long distances is unreliable and bandwidth intensive. In the application of surveillance systems, video signals may need to be transmitted many floors around a building to a central monitoring location where an operator is responsible for dozens of monitors to visually track. It is desirable to reduce the amount of full-frame video data to only crucial data regarding intrusion. The results are better bandwidth utilization and higher reliability. The design described in this paper is an embedded DSP-based imaging surveillance system that detects the presence of a target, determines its location and tracks its movement. In the case of a target moving through a protected area, the DSP tracks the target and controls the position of the camera to keep the target in the frame of view. This system eliminates unnecessary data transmission by using a difference frame comparison algorithm. It is also more reliable because only pertinent data regarding intrusion is monitored and transmitted. Furthermore, the paper describes the experience learned from implementing the system in a TI-C54x DSP. The implementation takes advantage of several special features of the TI-C54x that include low power mode for battery operations, high speed performance for video frame processing and the extendibility to control system applications directly from the embedded DSP.**

I. Introduction

It is common in surveillance applications to monitor multiple locations with high bandwidth sensors and cameras in dynamic environments. Surveillance requires constant human monitoring that induces operator fatigue. Naturally, the operator's experience and attention span play critical roles in the overall system reliability. Current video-based surveillance systems are also limited by their infrastructure and bandwidth requirements. Full frame video transmission in NTSC format requires a bandwidth of more than 5MHz. This is reasonable in controlled environments where the necessary hardware infrastructure (e.g. high-bandwidth coaxial cables, video switches) is in place. However, in situations where low-bandwidth non-twisted copper wiring is used or a wireless solution is needed, the bandwidth requirement becomes a major burden on the surveillance system. The purpose of the design presented in this paper is to address the bandwidth issue and to support the human operator to increase the overall efficiency of the surveillance system. By employing a DSP solution (using a DSP chip such as a member of the TI-C54x family) our design is also able to provide important features including low cost, low chip count, low power requirements and flexible programmability.

Our prototype system consists of three main modules: an inexpensive NTSC camera, a DSP-based video and motion processing (DVMP) module and a PC-based monitoring station (see Figure 1 for the system block diagram).

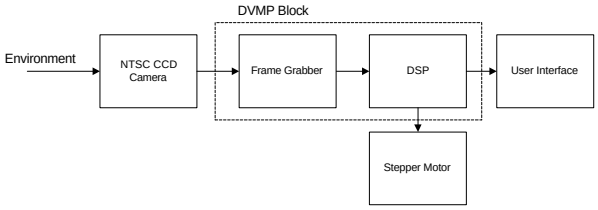

Figure 1: System Block Diagram

The system reduces bandwidth requirements by performing signal processing on the NTSC video at the DVMP module prior to presenting data to a human operator. The DVMP module makes decisions as to the presence or absence of an intruder by employing an absolute value difference frame algorithm. When an intruder is detected, the DVMP module transmits only relevant data to the PC-based monitoring station so that the location of the intruder in the frame of view is displayed if there is one present. Since the frame of view of the camera is limited by the curvature of its lens (typically about 50 degrees) the DVMP must also track movements of the intruder by panning the camera. Figure 2 depicts the case when an intruder moves beyond the frame of view of the camera.

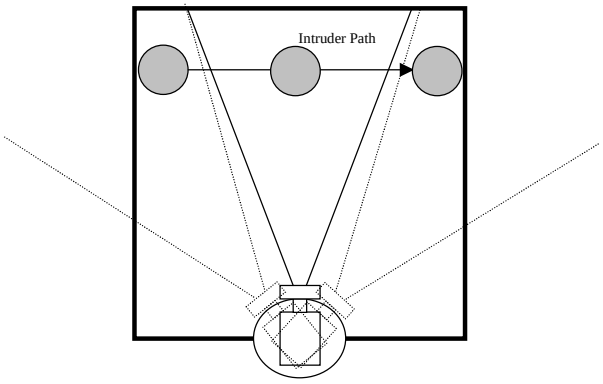

Figure 2: Camera panning to follow intruder

By transmitting only the data relevant to the detection and the location of an intruder, the system bandwidth requirements decrease dramatically. Furthermore, by designing the system to detect the presence or absence of an intruder and to track movements once detected, the human

operator is liberated from the mundane task of watching many screens or browsing hours of video for clues after an intrusion has taken place.

II. System Design Overview

The DVMP module is the heart of our system. It is actually composed of two sub-modules: a PC-based frame grabber and a DSP board (see Figure 3 for details of the hardware connections in the DVMP). It is feasible to replace the PC-based frame grabber by another dedicated board. However, due to time limitations of the project, an off-the-shelf PC-based frame grabber solution is adopted.

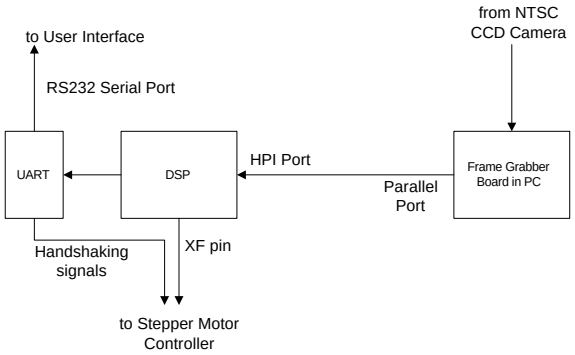

Figure 3: Detail View of DVMP Hardware Connections

Since the overall system is designed to operate independent of user control, the major role of the DVMP is to determine, in the presence of noise, whether an intruder is in the frame. The intruder now becomes the target that the DVMP must detect, track and report. In capturing digital video via a CCD camera, a 3-dimensional world is projected to 2-dimensions. The loss of depth information is inevitable. This becomes a problem in the target recognition application due to the fact that a target may be very close to the camera or very far away. The raw size measure of a target may be misleading or prone to errors. For this reason, small targets are eliminated as noise in the system. A side effect of this is the avoidance of false-detection when small house pets roams around the area under surveillance.

The major responsibility of the DSP in the DVMP module is to provide the computational engine to process video frames, decide if a target is present and determine its location. In order to accurately locate a target with the minimum amount of data, a resolution of 128x128 pixels is chosen. The DSP uses a down-sampled difference frame to locate the target in the frame. If the target is near the edge of a frame, the DSP tracks the moving target by generating control signals directly to a stepper motor controller to pan the camera accordingly. Meanwhile the location of the target is transmitted to the PC-based monitoring system to alert a human operator. If the DSP determines that there is movement within the frame, but the movement is confined to the center of the frame, motor control is not necessary. However, the coordinates of the location of the target are still transmitted for operator viewing.

III. Frame Processing

The benefits of choosing the TI-320C54x processor for this application were its high clock speed and its low power consumption. From a product development perspective, this system should eventually become self-contained and the major design decisions should enable future size and cost reduction. The drawback of the TI-C54x compared to the TI-C8x series processor, which is more commonly used for video applications, is the comparatively small core memory.

In our first attempt, the design used an NTSC CCD camera to capture a 512x512 8-bit grayscale frame of the scene directly to the DSP. This frame was used as the background frame and stored for later comparison. While operating in the surveillance mode, a “target frame” was captured and compared to the background frame in memory. If an empirically determined number of pixels were sufficiently different from the background frame (an absolute value difference frame) then a target was detected and tracked.

The drawbacks of this system are the memory requirement to store many frames of the surrounding environment for comparison and the necessity to capture many background frames before the system can be used for target tracking. In addition, the PC we used to power the frame grabber board has a 66MHz DX2 processor with only 8MB of RAM. As a result, the transfer of the high-resolution video frame through the parallel port is a major burden on the PC processor and a performance bottleneck of the overall system.

In order to eliminate or reduce these problems, the advantages of the TI-C54x DSP were employed to repeatedly capture two frames in rapid succession for difference comparison. Generally, when the two frames are subtracted, the areas of occlusion due to motion are enhanced and generate an artificial edge surrounding the body in motion. Taking advantage of partial scene occlusion due to motion, the DSP can approximate the center of the body in motion by finding the average location of all the edge pixels. Assuming that occlusion errors occur on both sides of the body in motion, the average will yield a good approximation to the actual location of the intruder.

Figure 4 shows the differencing algorithm. Two frames closely separated in time are captured and subtracted. An absolute value is taken to eliminate negative pixel values. The final image below is used to calculate the location of the moving target. What is being show are the edge pixels that result from partial scene occlusion. The algorithm checks for the number of pixels in the row that surpass a threshold. If there are more than 20 pixels in the row that surpass the threshold, then the row is considered to have a significant edge. When all of the rows are processed, the algorithm counts the number of rows that have a significant edge. If there are more than 30 rows with a significant edge, then a target is considered detected.

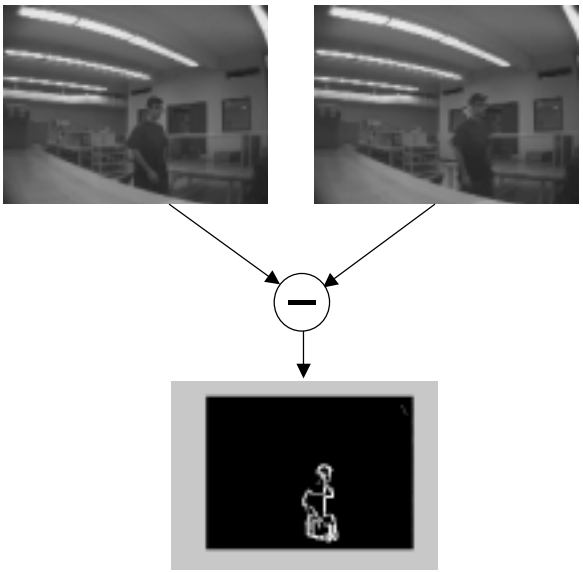

Figure 4: Illustration of Differencing Algorithm

The PC frame grabber board we used has enough on-board memory to store the two frames. One memory frame corresponds to image ***t1*** and the second memory frame to store image ***t2*** where ***t1*** is captured first and ***t2*** is captured soon thereafter.

Due to the bandwidth limitations of the parallel port, the difference and absolute value is taken in the PC. A threshold is applied and compressed row is transmitted to the DSP for processing. Eight pixels are compressed to one 8-bit transmission over the parallel port. Figure 5 shows a flowchart diagram of the preprocessing done in the PC frame grabber.

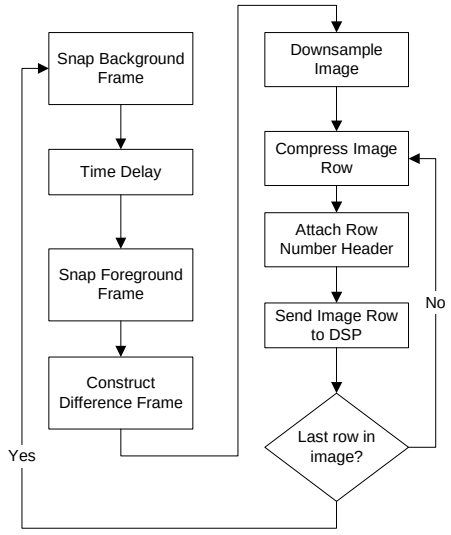

Figure 5: PC Frame Grabber Preprocessing Algorithm

To further reduce the bottleneck imposed by the parallel port, it is desirable to stream data to the DSP. Streaming the data to the DSP by rows speeds processing of the frame by allowing the image processing to start before the entire frame has arrived and has been stored. Additionally, the system can be connected and disconnected with ease for a camera replacement or if multiple camera data is being processed by one DSP.

A 2-byte header containing the row number being transmitted is added to the beginning of each transmitted row, effectively making the communications between the PC and DSP unidirectional. The PC writes directly to the DSP memory through the HPI port. The DSP keeps track of what row it expects to see as they stream in from the PC. If the 2-byte header matches the row number the DSP expects, the row will be processed. Otherwise a mismatch will result in the discarding of the row being inspected. The major time constraint here is in the processing of the streaming rows. Since the PC is writing directly to the DSP memory, the DSP must complete the decompression and processing of the previous row before it is overwritten by the next incoming row.

Figure 6 depicts a simplified flowchart of the compression scheme. A hard threshold is applied to each pixel in the row. Then a value of 1 or 0 is assigned to it for final transmission. By doing so, an 8:1 compression ratio is achieved. This compression preprocessing is done to mitigate some of the bandwidth issues related to the parallel port and the DX2 processor.

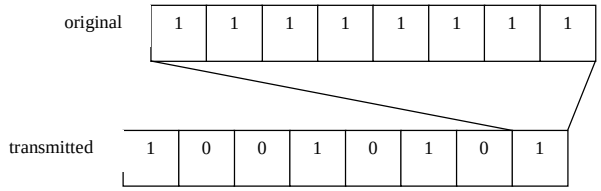

Figure 6: Compression Diagram

The DSP decompresses the row and sets values of 1 back to 255 so that integer arithmetic can take place. See figure 7 for a flowchart of the DSP processing algorithm. If it is determined that there are edge pixels in the row, the mean of the row is taken and stored in one element of a vector (the mean vector) the same length as the height of the image. At the end of a frame, the number of rows with edge pixels is counted. If the number of rows with edge pixels is large enough, it is determined that a target is in the frame and the mean of the mean vector is computed to find its location. This information is used to generate commands to pan the camera if the location of the target is sufficiently close to the edge of the frame.

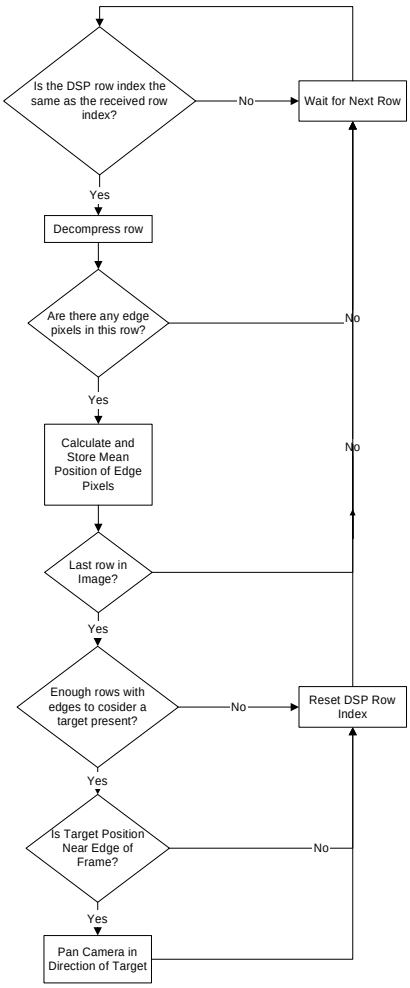

Figure 7: DSP Processing Algorithm

IV. Embedded Motor Controls

Typically a DSP has a limited number of general purpose I/O pins. Designers find themselves lacking the I/O resources for implementing data communications and motor control signals. To solve this problem, a microcontroller is added to the design just to handle the motor control signals. The drawbacks of this implementation include complicated handshaking, additional hardware cost and wasted board space.

To eliminate the need for an additional microcontroller, our system implements the motor control functions in the DSP. The difficulties of implementing the motor control functions were circumvented in this design by using the UART that is already available on the DVMP module for communication with the user interface. Two handshaking signals on the UART are used to control the direction and resetting of the stepper motor. The XF pin on the DSP is used to control the pulse input to the motor control circuitry to step the motor (refer to Figure 3). The motor is only stepped far enough so that there is some overlap between scenes as depicted in Figure 2.

V. Conclusion and Future Development

The image processing technique employed in our project is dependent upon a stable scene background that does not change much without a target present. This technique is more suited for an office environment at night rather than a crowded city street. More complex motion estimation and robust techniques can be substituted in this system to allow for more complicated scenery.

It is also worthy to note that the frame processing and motion control algorithm described here is interrupt driven. As a result the processor may go into an idle mode when not processing. The idle mode takes further advantage of the TI-320C54x low power operation and allows for further integration to a stand-alone system. Compared to video processing ASIC chips, the features of the TI-320C54x processor enables us to integrate low power operations and motor control into a single design platform.

Future expansions of this system include storing a frame of video for later transmission in a serial manner. Another option is the control of multiple systems operating in parallel with inter-processor communication. For example, if one camera system detects a target, other nearby camera systems may pan in the direction of the intrusion. More complex and robust motion estimation algorithm can be employed to enhance the system reliability in a wider variety of environments.

VI. References

Oh Hong Lye, “The TMS320C54x DSP HPI and PC Parallel Port Interface (TI spra151),” March 1997.

“TMS320C54x DSP Reference Set: Volume 1 CPU and Peripherals (TI spru131F),” April 1999.

“TMS320C54x, TMS320LC54x, TMS320VC54x
Fixed –Point Digital Signal Processors (TI sprs039B),” February 1998.