Development of Embedded Real-Time and High-Speed Vision Platform

Zhenxing Ouyang, Yimin Dong, and Hua Yang
State Key Laboratory of Digital Manufacturing Equipment and Technology
Huazhong University of Science and Technology
Wuhan, China 430074
Email: huayang@hust.edu.cn

ABSTRACT

Currently, high-speed vision platforms are widely used in many applications, such as robotics and automation industry. However, a personal computer (PC) whose over-large size is not suitable and applicable in compact systems is an indispensable component for human-computer interaction in traditional high-speed vision platforms. Therefore, this paper develops an embedded real-time and high-speed vision platform, ER-HVP Vision which is able to work completely out of PC. In this new platform, an embedded CPU-based board is designed as substitution for PC and a DSP and FPGA board is developed for implementing image parallel algorithms in FPGA and image sequential algorithms in DSP. Hence, the capability of ER-HVP Vision with size of 320mm x 250mm x 87mm can be presented in more compact condition. Experimental results are also given to indicate that the real-time detection and counting of the moving target at a frame rate of 200 fps at 512 x 512 pixels under the operation of this newly developed vision platform are feasible.

Keywords: High-speed Vision, Moving Target Detection, Object Counting.

1. INTRODUCTION

With the development of the high-speed vision systems, there is a growing demand for high-speed vision in many application fields, such as high-speed target tracking in games and competitions, robotics[1,2] and automation industry[3]. Compared with the high-speed vision system, the traditional vision systems that are usually operated at lower frame rates (e.g., PAL 25fps)[4] can only capture low-speed targets, similar to human eyes. Therefore, the development of high-speed vision in software and hardware becomes more and more important. Traditional high vision systems are usually realized by PC in which the image acquisition card captures images and the CPU processes the image data[5]. It has been proven that the PC-based vision system can realize complex image processing algorithms, but the processing speed of CPU in this kind of system is not fast enough to satisfy high-speed application fields requirements. Hence, various embedded high-speed vision systems[6,7] which have better performance and are more compact have been developed with the research advancement in embedded technology over the past years. For example, Idaku Ishii and Taku Taniguchi[8] developed a FPGA-based high-speed vision platform, which was able to simultaneously process a 1024 x 1024 pixel image at 1000 fps and a 256 x 256 pixel image at 10000 fps. Wenhao He and Kui Yuan[9] designed a FPGA and DSP based robot vision where the 800 x 600 pixel images were processed at 200fps. A high-speed vision system is introduced which can execute real-time image processing and high frame rate video recording simultaneously in [10]. A FPGA-based high-speed vision platform that has the performance of the real-time implementation is sufficient to deal with a video frame rate of 350 fps at 256 x 256 pixels resolution is introduced in [11].

In the high-speed vision systems mentioned above, PC is still an essential component for image processing or human-computer interaction. The over-large size of PC is not suitable and applicable in compact systems. Unfortunately, there are no high-speed vision systems that can work completely without PC in the early work. Therefore, this paper proposes a newly embedded real-time and high-speed vision platform called ER-HVP Vision which features the further volume reduction of the system. In the new platform, an image processing board and a CPU-based board instead of PC are used to implement the image algorithm and realize human-computer interaction. Besides, the effectiveness of this new platform is verified with examples of target tracking and objects counting in this paper.

The rest of this paper is organized as follows. In Section 2, ER-HVP Vision is proposed and its components and working procedure are described. Two demonstrations are illustrated in Section 3 and Section 4 for the verification of the
new system effectiveness. Finally, conclusions are drawn in Section 5.

2. THE EMBEDDED REAL-TIME AND HIGH-SPEED, ER-HVP VISION

2.1 Outline of System

ER-HVP Vision is an embedded real-time and high-speed vision platform operated under high frame rate in which various types of image algorithms can be implemented and verified, and the images and processing results can also be recorded and stored. Fig. 1 shows the architecture of ER-HVP Vision with size of 320mm x 250mm x 87mm.

Two cameras are able to be connected to ER-HVP Vision, but only one of them is needed in this platform. Therefore this platform consists of a high-speed camera, a FPGA and DSP based image processing board, a CPU-based board with a touch screen and a portable power. Images captured by the camera are transferred to the image processing board where the image algorithms designed by the user is implemented and then the results are sent to the CPU-based board that records the results. Finally, the touch screen displays the results.

![Figure 1. Architecture of ER-HVP Vision](image)

2.2 Components

The high-speed camera used in ER-HVP Vision with a frame rate of 200 fps at 512x512 pixels is developed by our team members. The camera appearance is in Fig.2 (a).

The FPGA and DSP image processing board which has high performance and flexibility is developed for real-time and high-speed image processing. The Xilinx XC6VLX240T chip has been chosen as FPGA processor for its high performance. The FPGA subsystem is designed to include two Cameralink Base ports for connecting to two cameras, an image preprocessing module, a JTAG serial port for testing, a serial RapidIO (SRIO) IP-core for communicating with DSP and memories (including DDR3 SDRAM and FLASH). The TI TMS320C6678 chip with eight 1GHz frequency cores is suitable for the embedded high-speed vision system, thus it has been chosen as DSP processor. Peripherals of the DSP consist of a 10/100/1000M Ethernet module for communication with the CPU-based board, a SRIO module, a JTAG/UART serial port for testing and memories (including 1GB DDR3 SDRAM and FLASH). Fig.2 (b) shows the appearance of the board. Fig.3 is the functional block diagram.
The main chips of image processing board are the FPGA for image preprocessing and the DSP for complex image processing. The working procedure of the board and the function of the peripherals are elaborated as follows.

1) Power-On sequence of operation. Specific power supply and clock timing sequences necessary for the DSP normal operation are realized by FPGA.

2) The initialization of FPGA and DSP. The FPGA and DSP initialize all units of the image processing board respectively. After the completed initialization, the FPGA is waiting for the image data from the camera.

3) Capturing images. The image capturing module managed by FPGA receives 8-bit parallel data (8 bits/pixel) via Cameralink Base port and stores data in FIFO or DDR3 memory temporarily.

4) Preprocessing images. The FPGA reads image data from FIFO or DDR3 memory, then processes the data with user-implemented image preprocessing algorithm (such as filter, edge detection etc.) in the preprocess module.

5) Sending data to DSP via SRIO port. The preprocessed images of (4) are transmitted to DDR3 memory of DSP via SRIO port. FPGA sends a flag to DSP via General Purpose Input Output (GPIO) port as soon as the transmission of an image is accomplished.

6) Processing images. The images obtained by DSP are processed in the image processing module where user-defined image processing in DSP is executed.

7) Transferring processing results to the CPU-based board. The image sending module in DSP transfers the processing results of (6) to the CPU-based board via 1000M Ethernet port. User Datagram Protocol (UDP) is chosen as the transmission protocol for the purpose of high-speed transmission.
The CPU-based (Intel-i7) board receives network packets from the image processing board, decodes the packets and extracts out the image data and the processing results. Since the main tasks (acquisition and processing) of ER-HVP Vision have been completed in the image processing board, the CPU-based board does not implement image processing algorithms. The requirements for the CPU-based board are image saving, simple operations (such as curve fitting) and results display in the touch screen. Therefore, the requirement for the performance of the board is not necessarily high, but a Network Interface Card (NIC) is indispensable for the board to communicate with the image processing board. Specifications of the board used in the study are showed in Table 1, and the appearance is showed in Fig. 2 (c).

<table>
<thead>
<tr>
<th>Components</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>4GB(DDR3 4GB X 1)</td>
</tr>
<tr>
<td>OS</td>
<td>Windows 7</td>
</tr>
<tr>
<td>Interfaces</td>
<td>1000M Ethernet port X 2</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Mobile i7-3612 2.1GHz</td>
</tr>
</tbody>
</table>

Power for ER-HVP Vision whose power dissipation is about 130W are two lithium batteries of which capacities are 11000mAh and 18000mAh, which makes the vision system is able to work properly about two and half hours.

3. MOVING TARGET DETECTION

3.1 Outline of Algorithm

The algorithm for the moving target detection is designed to satisfy the real-time processing at the speed of 200fps. Since the system contains two processing units (FPGA and DSP) that can realize parallel processing, the DSP and FPGA has 5ms redundancy to implement the algorithm respectively.

In this paper, noise filtering which cannot be executed easily in real time by software is implemented firstly in FPGA by hardware. After the preprocessing in FPGA, the image processing algorithm in DSP can be simplified. In DSP the image processing module receives the images and implements the moving target detection algorithm, thus the coordinate of moving target is obtained. So far, three existing methods are used widely in moving target detection [12]: frame difference, optical flow and background subtraction. Frame difference method [13] realizes the moving target detection by using the pixel wise difference between consecutive image frames. This method has great adaptability under dynamic environment, but generally cannot extract the moving target completely. For example, the extracted target may have a cavity. Optical flow method [14] can be used in the unstable background, but its algorithm is complicated and time-consuming. Compared to the two methods mentioned above, background subtraction method [15] is easier and more widely used. Therefore, the background subtraction method is chosen as the algorithm of moving target detection in this paper.

However, background subtraction method is sensitive to the change of background introducing noise inevitably. This paper proposes a simple, effective and fast algorithm which can obtain the real-time coordinate of moving target under a relatively stable environment. Firstly, Gaussian filter is implemented in FPGA to smooth images. Secondly, background subtraction method is used to detect the moving area. Thirdly, mathematical morphology is utilized to eliminate noise and repair shape of the moving target. Finally, the coordinate of moving target can be acquired by connected component analysis. The detail algorithm flow is showed in Fig.4.
Figure 4. Algorithm flow for moving target detection

1) Gaussian filter. Vision system will inevitably be disturbed by inherent system noise which will influence the quality of subsequent processing. Hence, the noise filter is necessary in image preprocessing. This paper uses Gaussian filter implemented by FPGA to eliminate the noise [11].

2) Background subtraction. The basic steps of the background subtraction method are as follows: the current image subtracts the background image, the results of which are processed by the binaryzation.

\[
D_{t}(x, y) = \begin{cases} 
1 & \|I_{t}(x, y) - B_{t}(x, y)\| > T \\
0 & \|I_{t}(x, y) - B_{t}(x, y)\| \leq T
\end{cases}
\]

(1)

Where \((x, y)\) is the pixel coordinate, \(I_{t}\) is the current image, \(B_{t}\) is the image of background model, \(T\) threshold.

If \(D_{t}\) is one, the pixel belongs to foreground region, otherwise, it is background.

The image without moving object is chosen as the initial background model. A dynamic background updating method is used to adapt for changes in the environment. A common background updating method that only updates pixels of background has the form of

\[
B_{t+1}(x, y) = \alpha B_{t}(x, y) + (1 - \alpha)I_{t}(x, y)
\]

(2)

Where \(\alpha\) is the weight of background updating. The extracted foreground may contain noises that should be eliminated by using the mathematical morphology method.

3) Mathematical morphology. After background subtraction algorithm is implemented, images are converted to binary images which can be processed by binary morphology method. Binary morphology handles binary images as sets and probes them with a structure element. Dilation, along with erosion, is an elementary morphological operation. Dilation expands binary objects in an image and is commonly used to connect neighboring objects. Comparatively, erosion shrinks binary objects in an image and is commonly used to remove noise before further analysis. Other morphological operations, such as opening and closing, can be built by dilation and erosion. In this paper, an opening operation is used to eliminate noises.

4) Connected component analysis. The connected components labeling algorithm examines the binary image, groups foreground pixels that have other foreground pixels as 4- or 8-connected neighbors, and labels discrete groupings as components. Once accomplished, component properties can be measured and used to extract foreground information.

The two-pass method [16] is chosen to label the component which is the foreground area in this paper. Then the coordinates of the foreground can be acquired. Finally the coordinate of spherical center which is approximately equivalent to the average value of the coordinates of the foreground can be obtained.
3.2 Implementation and Operation

A high-speed target detection system, as the configuration is showed in Fig. 5, has been developed. In this detection system, a single camera was used to capture the image (512 x 512 pixels) at the speed of 200fps, and then the image data were sent to the image processing board. After the image processing algorithm implementation, the detection of real-time coordinate of moving target was realized.

![High-speed target detection system](image)

Figure 5. High-speed target detection system

Ping Pong was chosen as the moving target of this experiment. In the experiment, the indoor lights were turned off, and then the server hit the ball with a racket. The camera captures images which are showed in Fig.6, then the images were transferred to the image processing board in which the program of moving target detection algorithm had been downloaded. After the CPU-based board received the results implemented by the algorithm, it would execute the curve fitting module and display the final results on the touch screen. The results of each step and curve fitting are shown in Fig.4. Fig.7 shows the $x - y$ trajectories and velocity distribution of Ping Pong. Some abrupt points in Fig.7 were caused by the error of connected component analysis. The time consumed in each step of the algorithm is shown in Table 2. As it is shown in Table 2, the consumed time in FPGA and DSP is less than 5 ms respectively, which has fulfilled the requirement of the 200fps real-time image processing.

The experimental results indicate that the system is able to detect the real-time coordinate of Ping Pong in relatively stable background. More importantly, the system has the advantage of smaller size which is more suitable for compact condition and more convenient for setting up the experimental platform.

![The captured images of Ping Pong](image)

Figure 6. The captured images of Ping Pong
OBJECTS COUNTING

4.1 Algorithm of Objects Counting

The algorithm of objects counting contains two major sections, the moving target detection and the objects counting. Details of the algorithm are elaborated as follows.

1) Moving target detection. In this application, the simplest condition that the background is relatively stable is considered. Therefore, the algorithm of moving target detection in Section 2 can be adopted. The differences from the algorithm in Section 3 is that the coordinates of incomplete objects are ignored when the number of objects is counted and the moving target is multi-target.

2) Objects counting. In the previous step, the centroid coordinates of objects are obtained. In this step, a special area is chosen. The counter adds the number of objects whose centroid coordinates fall into this special area.
Now we consider the case of single object. When the object is moving, the camera, continuously capturing the same
moving object, will obtain a series of centroid coordinates. If the counter adds one as soon as a coordinate is
acquired, the number of objects will be counted repeatedly. For the purpose of avoiding repeated counting, a special
area is chosen which must satisfy the condition that there is one and only one centroid coordinate of the same
moving object in this area.

The movement of objects in the pipeline is approximate to uniform linear motion. When \( Y \) direction is ignored, the
object's centroid coordinate \( X_n \) in the consequent images can be expressed as:

\[
X_n = An + X_0
\]  

(3)

Where \( A \) is the displacement between two consecutive images, \( X_0 \) is the initial valid centroid coordinate, \( n \)
represents the \( n \)-th image. Suppose that the chosen area is \([M, M+K]\). The meaning of parameters is also illustrated
in Fig.8.

Therefore, the problem that how to choose the area is transformed into that \( n \) has the unique solution in equation (3)
when \( M < X_n \leq M + K \).

\[
\frac{M - X_0}{A} < n \leq \frac{M - X_0}{A} + \frac{K}{A}
\]  

(4)

Where \( n \) is integer and the unique solution of equation (4), so \( K = A \). The error in \( Y \) direction is taken into account.
Then the chosen area is the black area showed in Fig.8. When detected objects are multiple, the chosen area is the green
area showed in Fig.8.

4.2 Implementation and Operation

A real-time objects counting system illustrated in Fig. 9, was developed based on ER-HVP Vision which was able to
count products and parts in industrial pipelining. A linear motor was used to simulate the pipelining and several coins
were treated as the objects.
As it is shown in Fig. 9, the high-speed camera captures the images when the motor was moving back and forth. The detail program flow is showed in Fig. 10. Firstly, the $K$ ($K = 24$ in this experiment) was obtained by calculating the centroid coordinate of two consecutive images. Then the images captured by the camera were processed in ER-HVP Vision, and the centroid coordinates were obtained. If the coordinates were in the special area, the counter added the number of objects whose centroid coordinates fell into the chosen area, which is illustrated in Fig. 11. The consumed time of the algorithm was approximate to the time of moving target detection, which indicated that the system was able to count objects in high speed.
4. CONCLUSION

This paper develops an embedded real-time and high-speed vision platform, ER-HVP Vision with the capability of processing 512x512 pixel images at 200fps in real time. Compared with traditional high-speed vision platform, this newly high-speed platform features in smaller size and is more suitable for compact conditions, such as mobile robots, unmanned aircrafts and instruments.

5. ACKNOWLEDGMENT

The work described in this paper is partially supported by National Natural Science Foundation of China under grant Nos. 51327801, 51475193. The authors would like to thank the reviewers for their constructive comments that improved the presentation of the paper.

REFERENCES