Low-Cost Hardware-Based Marker Tracking

Travis Deyle, Volker Roth

German Abstract

Eine Reihe interessanter Anwendungen können mit Hilfe von Kameras gestaltet werden. Ein Bereich ist die Verfolgung von Objekten und die Bestimmung von deren Position im Raum z.B. zur Analyse von Crash Tests oder des Aufpralls eines Golfschlägers auf den Golfball. Dies kann beispielsweise durch das automatische Erkennen von speziellen Markierungen erfolgen, welche an den zu verfolgenden Objekten angebracht werden. Viele der »interessanten« Vorgänge finden jedoch innerhalb von Millisekunden statt und erfordern teure Hochgeschwindigkeitskameras. Ein maßgeblicher Kostenfaktor solcher Kameras ist der Speicher, in dem die Bilder zwischengespeichert werden müssen. Wir forschen an einer Alternative, welche die Implementierung der Erkennung von Markern in kostengünstiger Kamerahardware erlauben soll, ohne dass die komplette Aufnahme gespeichert werden muß. Dies hätte den Vorteil, daß hohe lokale Bandbreite zur Bildauswertung genutzt werden kann und nur noch Positionsdaten der Markierungen an den Rechner übermittelt werden müssen.

Introduction

Digital video cameras have a sea of applications ranging from film making to video chats to industrial applications that involve the tracking of objects in space. One means of tracking objects is the use of markers which can be detected easily and robustly within images. The markers are fixed to prominent parts of the object to be tracked, and their positions are registered. A software then analyses images received by the camera for the presence and position of these markers and uses that information to calculate the position of the object relative to the camera position and field of view.

However, in many interesting cases, the motions or deformations to be analyzed happen within fractions of a second. Examples would be a car crashing or a golf club hitting a golf ball - an event which has a duration on the order of 0.4 milliseconds. Measuring such an event requires expensive high-speed cameras which easily cost between fifty and a hundred thousand dollars per unit. A significant cost factor of such a camera is the amount of fast memory that is available to cache the recorded imagery before it is transmitted to a host computer for analysis. The cache size also limits the maximum recording time of such a camera. For instance, at 50,000 frames per second, a current high-speed camera would deliver at most a few seconds worth of 10 bit gray scale imagery at a resolution of 128x128 pixels. This amounts to generating and storing one Gigabyte of data every second.

We are interested in investigating the possibility of implementing the marker tracking in hardware such that only the positions of markers are stored and perhaps streamed to a receiving host computer. This would reduce the memory requirements and data rate transmission requirements tremendously and thereby offer opportunities to track markers continuously at very high speeds at substantially reduced costs. A disadvantage of this approach is that the imagery would not be available. However, this disadvantage can perhaps be alleviated by taking a few single frame snapshots if a preconfigured number of markers is detected in an image.

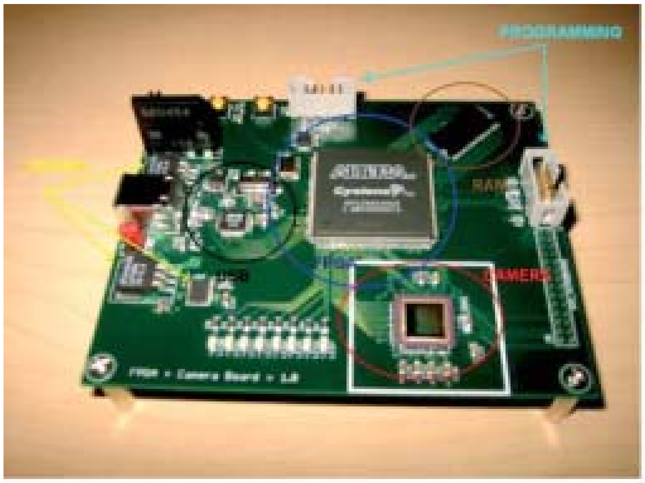

Figure 1: The produced PCB with FPGA, USB, memory and image sensor.

As a first step, we designed and developed a hardware board with USB connectivity that has a Field Programmable Gate Array (»FPGA«) chip and a low-cost image sensor on it. The FPGA is reprogrammable and interfaces directly with the image sensor. Below, we report on the design of the board and some of the lessons we learned.

Image Processing Board

Figure 1 shows a photograph of the prototype image processing Printed Circuit Board (»PCB«) that we built. The PCB is powered by either a DC jack or by the USB connector (selectable by a jumper) and requires three different voltages, which are generated from the main power source: 5V for the USB chip, 3.3V for the debugging LEDs, FPGA I/O, RAM and image sensor, and 1.5V for the internal core and PLLs of the FPGA. Power is routed via two independent supply layers, one for ground and one for positive voltages. The ground is a solid plane of copper on layer 2 of the PCB. The three supply voltages are delivered through copper polygons on layer 3. This approach greatly simplified the power distribution during the physical layout.

The PCB can be connected to a host computer through a USB port. The USB functionality is provided by the FT2232 dual USB UART/FIFO chip by FTDI. Using FTDI's drivers, the PCB can be addressed as a virtual COM port. This makes communication with the PCB relatively straightforward.

As the image sensor, we used the Kodak KAC-9648, which produces color images with SXGA resolution (1288x1032) at 18 frames per second. The sensor is connected with the FPGA through a 10-bit wide data bus and an I2C control bus. We decided to operate the chip in the master mode, which means that it generates its own pixel clock and synchronization signals.

The on-board memory consists of a 128 Mbit SDRAM from Micron Technologies and can comfortably store five images and leave some working memory.

The heart of the PCB is the FPGA. We selected the EP1C6Q240C6. This FPGA has over 92 Kbit of RAM, almost 6000 logic elements and 185 I/O pins. It is one of the densest FPGA chips available that can still be hand-soldered to a PCB, which is what we did. Its two internal phase-locked loops (»PLLs«) can operate in the hundreds of megahertz. We connected the FPGA to a 50 MHz external clock. While we wrote a custom code to test the PCB, numerous »soft-cores« exist for such FPGAs, which implement for example, microcontrollers and memory controllers.

Development Process

The development process proved to be challenging. More than 1000 pages of documentation were collected and digested in order to design the hardware. The choice of CAD software to be used for capturing schematics and for laying out the PCB was involved as well, given that we had tight budget constraints. We finally settled on the commercial version of the Eagle software by CadSoft. The schematics and footprints of several parts we needed were not readily available, thus several libraries had to be custom-made. Fortunately, Eagle comes with extensive documentation, which we appreciated very much during the process.

The manufacture of the PCB was done through PCBexpress, a manufacturing service company. PCBexpress also provided documentation and scripts that guided us through the process of generating the necessary input files from the Eagle board file. This included the gerber layer, outline, solder mask, silkscreen, aperture, NC drill and drill tool list files.

Once we had the printed board in hand (actually two boards, we anticipated that we would need a backup and that turned out to be true), we soldered down the surface mount chips. Soldering the half-millimeter spaced pins of the FPGA was a demanding task that had to be accomplished without special or expensive equipment. Subsequently, we verified that all components were connected and operated correctly by programming the FPGA with our custom testing and debugging code. For this task, we used the Altera Quartus II Web Edition software.

Preliminary Conclusions

The development of custom hardware boards is an expensive and time-consuming process with many risks, particularly due to the incremental dependency of the development steps. A mistake that is made early in the process is hard to detect until much later in the process, and corrections can be costly, as new boards may have to be produced. Fortunately, we managed to steer clear of most of the pitfalls and had fall backs in place for those cases in which we made more critical errors. Overall, we spent more than $2000 on the hardware and software and, being new to the process, burned more than 300 hours of intense work on the project. On the other hand, the project has been extremely rewarding - we learned invaluable lessons on the intricacies of hardware design. The image board is ready to be taken to the next phase, which consists of algorithm development, implementation and experimentation. We believe that, if produced in a larger batch, the costs of an individual board can be brought down to a range that is affordable to the common desktop computing user, ultimatively approximating the costs of a contemporary web cam. This raises the interesting question about what types of useful home or office applications could be implemented based on our board design.

Acknowledgements

This project was generously funded by PGA Pro Golfer Tom Sieckmann, who is an active research sponsor and investigator of technology that helps to improve the understanding of the game of golf.

Point of Contact

Dr.-Ing. Volker Roth
OGM Laboratory LLC, Omaha,
Nebraska, USA
E-mail: vroth@ogmlabs.com,
volker.roth@acm.org