1996

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

HARDWARE/SOFTWARE ISSUES FOR VIDEO GUIDANCE SYSTEMS: THE CORECO FRAME GRABBER

Prepared by:	John W. Bales, Ph.D.
Academic Rank:	Associate Professor
Institution and Department:	Tuskegee University Department of Mathematics
NASA/MSFC:	
Laboratory:	Astrionics
Division:	Avionics Simulation
Team:	Orbital Systems and Robotics
MSFC Colleague:	Fred Roe, Jr.

S2-35

THE CORECO OCULUS F64 FRAME GRABBER

The F64 frame grabber is a high performance video image acquisition and processing board utilizing the TMS320C40 and TMS34020 processors. The hardware is designed for the ISA 16 bit bus and supports multiple digital or analog cameras. It has an acquisition rate of 40 million pixels per second, with a variable sampling frequency of 510 kHz to M0 MHz. The board has a 4MB frame buffer memory expandable to 32 MB, and has a simultaneous acquisition and processing capability. It supports both VGA and RGB displays, and accepts all analog and digital video input standards.

CAMERA CHARACTERISTICS

The PULNiX TM9701 is a progressive scanning full frame shutter camera. It is capable of both digital (EIA-422) and analog output. Shutter speed can be set manually or by an external double-pulse signal. Output can be in either interlace or non-interlaced mode. Whereas other CCD cameras scan only 350 lines per screen, the PULNiX scans 484. In addition, the PULNiX is available in a flight qualifiable package.

SOFTWARE INTERFACES

All commands to the F64 frame grabber are mediated by the software driver. The driver resides in the host 486 machine as a terminate and stay resident (TSR) program which is loaded on system power-up. Coreco provides several methods of accessing the driver.

For the Windows operating environment, there is the F64Pro program. This is not for develping user applications, but solely for the purpose of demonstrating the features of the frame grabber. There is also a program, WConfig, which allows user configuration of the frame grabber from within the Windows environment.

For the DOS operating environment, Coreco provides several software alternatives. First, is the Oculus Driver Command Interface (ODCI). In its menu driven mode, ODCI is just another platform for demonstrating the functions of the frame grabber. But ODCI also operates in "shell" mode. One enters the OCDI shell mode by pressing <ESC> when in menu mode. Coreco provides a simple programming language for it's ODCI shell mode. One may use this to program simple macros utilizing most of the features of the frame grabber. In addition to ODCI, Coreco provides modules for building either assembler language or C language applications. The C language modules and resulting applications may be compiled using MicroSoft C/C++ version 7.00.

OCULUS DRIVER

The Oculus driver (ODX) is implemented as a TSR program. User applications communicate with ODX using a message passing interface. Only one command is processed at a time. Messages sent while ODX is busy are queued and processed on a first in first out (FIFO) basis. The state of ODX and the frame grabber are specified by the 115 Oculus Parameter Registers (OPR). User applications can communicate with ODX and the frame grabber by manipulating the contents of the OPR directly, or indirectly by using the 83 Coreco supplied functions comprising the Application Programming Interface (API). Both the OPR and the API functions may be classified into five types according to their purpose: Management, Memory, Processing, Acquisition and Display.

Programmers may learn the state of the various OPR using the 'opr_inq' function, and may change the state of a proper subset of the OPR using the 'opr_set' function. It is possible to directly access the contents of the frame buffer (which resides on the frame grabber card) by mapping the frame grabber memory to the host 486 bus. This is accomplished using the API functions 'memopen' and 'memclose'. To grab an image one may use the functions 'fbgrab' or 'fbgrabalu'. One may check to see whether a frame grab request has been completed by checking the status of the FGON register using 'opr_inq'. One may extract the contour of a binary object in the grabbed image using the API functions 'cntrfirst' and 'cntr'. One may perform various arithmetic logic unit (ALU) operations "on the fly" using the API functions 'fbgrabalu' and 'fbalu'. For example, one by automatically subtract the current frame from the previous frame and display the difference. Using the API function 'event' one may use the vertical or horizontal sync signals to trigger the execution of a user function.

EXAMPLE USER FUNCTION

Figure 1 show an example user function using the ODX API functions. This particular applications grabs one frame for reference, then grabs a second frame and subtracts it from the first, displaying the result. The API function 'odxbind' determines the number of ODX drivers installed on the host machine, and binds to those drivers. The API function 'setdmajor' selects one of the installed drivers as the major driver. First, the application determines the index of the display frame buffer (DFB) using the 'opr_inq' function. Next, the same index is assigned to the input frame buffer (FFB) using the 'opr_set' function. This has the effect of displaying the currently grabbed image on the display monitor. Then, using the 'fbgrab' function, a single frame is grabbed. Using the 'opr_inq' function, one waits until the FGON register has a value of zero, indicating that the grab has been completed. Then one uses the 'fbgrabalu' function to grab a second frame, while simultaneously subtracting it from the first frame. Figure 2 shows the result of running this sample application while waving at the camera. Because the slight displacement of the hand between the two successive grabs, the two images were not identical. Thus subtracting the second image from the first resulted in an outline of the hand.

APPLICATION TO VIDEO GUIDANCE SYSTEMS

Video guidance systems employ video images of moving targets to determine the range, orientation and velocity of the targets. Such systems must use the changes in successive images to extract range, orientation and velocity information in real time. The Orbital Systems and Robotics team of the Avionics Simulation Division of the Astrionics

Laboratory of Marshall Space Flight Center has an ongoing effort to develop a suitable Video Guidance System for Automated Rendezvous and Docking of space craft. The Coreco F64 frame grabber and the Pulnix TM9701 camera are potential components of such a system.

```
#include <odx.h>
#include <opr.h>

extern int  FAR odxbind(void);  /* Bind to ODX drivers installed      */
extern void FAR setdmajor(int); /* Set working major device           */

int main( int argc, char **argv)
{
 int nodx;                      /* Number of installed drivers        */
 int fb;                        /* Frame buffer index                 */
 int key;                       /* Keyboard response                  */

 nodx = odxbind();              /* Bind to installed drivers          */
 setdmajor(0);                  /* Select driver # 0                  */

 printf("Press any key to take snapshot.\n<RETURN> to quit.\n");

 while ( (key = getch()) != '\r')  /* Do until <RETURN> pressed       */
  {
   fb = opr_inq(DFB);           /* Get index of display buffer        */
   opr_set(FFB,fb);                     /* Place input into display            */
   fbgrab(1);                   /* Grab one frame                     */
   while( opr_inq( FGON));      /* Wait until grab finished           */
   fbgrabalu(2,fb,1);           /* Subtract a second grab from first  */
  }
 return( 0);
}
```

Figure 1: A demonstration user application

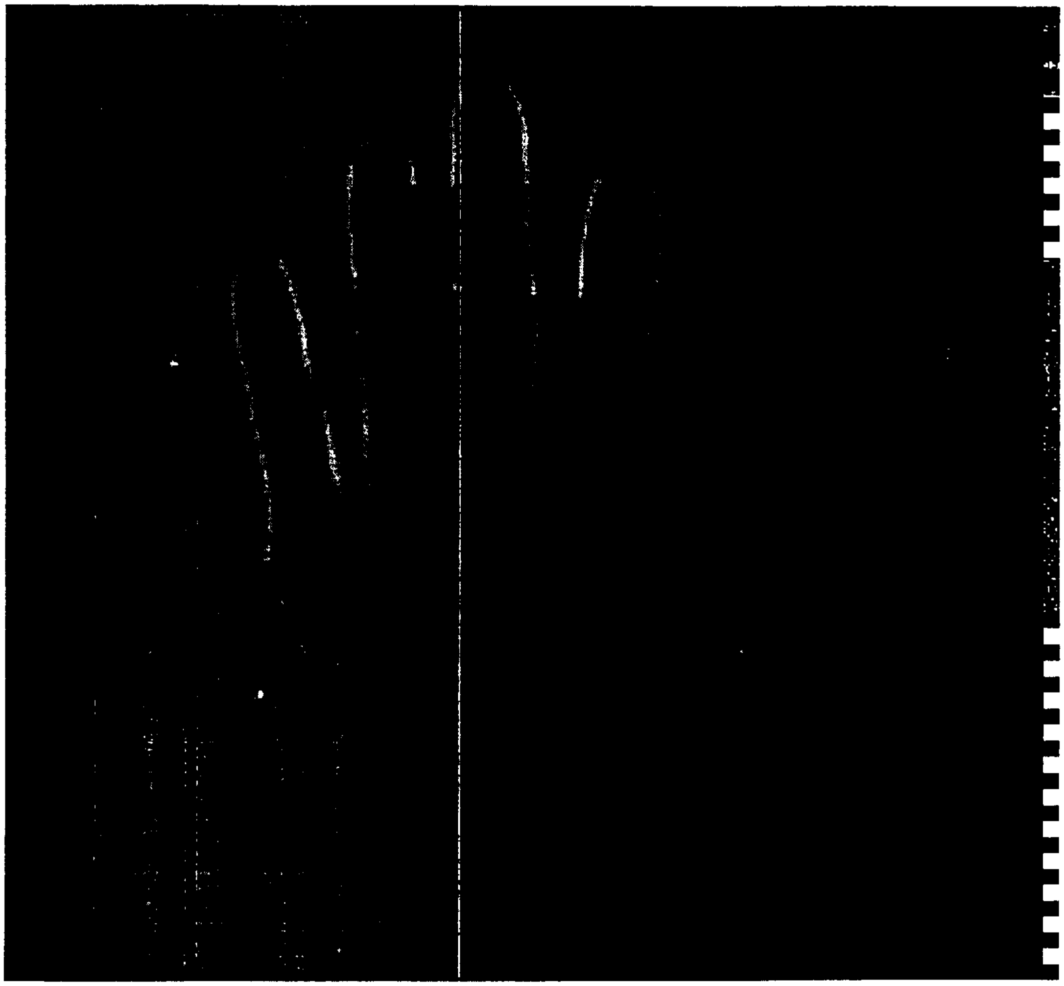

Figure 2: Output of demonstration user application