Tcl Extension for IEEE1394(Firewire) Video Cameras and Raw-to-Color Image Conversion

Akifumi Kobashi
Henry M. Gunn High School
Palo Alto, California
U.S.A.

Abstract

We present a Tcl/Tk extension that is designed to enable Tcl users to easily build Tcl applications for using IIDC-compliant IEEE1394(Firewire) video cameras. It is made up of two sets of procedures: the video camera controllers and the "raw" digital image processors. The camera controlling commands can control every IIDC IEEE1394(Firewire) compliant camera regardless of its manufacturer. The raw-to-color conversion commands can be used not just for these video cameras but also for any digital camera, still or video, that can generate Bayer-filter-based raw image data. Users can customize the built-in raw-to-color converters as well as create new procedures in Tcl implementing their own algorithms to enhance the image quality.

1 Introduction

The IIDC compliant IEEE1394(Firewire) video cameras are widely used in the scientific and industrial video applications as well as consumer products for various objectives such as surveillance, video conferencing, and robotics. In the consumer market, Apple Computer's iSight is a well-known example that uses this protocol. These cameras can be controlled by the same application regardless of manufacturer or model, as long as they are IIDC IEEE1394 compliant unlike the cameras based on USB. Therefore, the users of our extension can control all the IIDC compliant IEEE1394(Firewire) cameras with a single Tcl script without modification.

The raw-to-color converters incorporated in this package are necessary to recover the real color images from the image data transmitted from the IIDC IEEE1394 cameras that use a color filter array such as Bayer filter (most cameras fall in this category), since the original data coming from these cameras is a sequence of "raw" images which are not color images. We enhanced the capability of the converters so that they can be used to satisfy the complex requirements by even the demanding digital still camera users.

The video camera controller part of the extension is based on a set of well-known open-source libraries for Linux: libraw1394 and libdc1394. We created a set of Tcl commands that are capable of controlling and querying all of the camera features individually. There are about 17 controllable features including focus, zoom, aperture, frame rate, saturation, and gamma. In addition our package provides useful image analysis tools that can be called repeatedly within C for processing each frame. The reason for not making them independent Tcl commands is the speed needed in video processing, since systems are expected to process up to 30 frames per second. The user can also apply any user-created Tcl procedure on any frame of the video stream.

The raw-to-color converters provide most of the well-known conversion algorithms with plenty of adjustable parameters. Moreover, our extension provides procedures that can be used as components for building new converters based on new algorithms that a user may design. The commands for these component procedures enable the customization to a greater degree than any existing converter can provide.

2 IIDC IEEE1394 Video Camera Controller

2.1 IEEE1394 and IIDC(DCAM)

IEEE1394 protocol is also known as firewire (Apple's naming) and i.Link (Sony's naming). Although it is not as widely used as USB, IEEE1394 excels USB in various respects, especially for the purpose of video transmission. One of IEEE1394's strengths is it has a mode that guarantees the bandwidth of transmission unlike USB. Therefore, it is ideal for the video and other applications that require guaranteed bandwidth.

IIDC is a specification created by The 1394 Trade Association for the IEEE1394 cameras [2]. IIDC is also called DCAM. Most consumer grade video cameras that have recording media, tape or DVD, transmit the video data in a compressed form. They are not IIDC compliant. The IIDC compliant cameras do not compress the image data. Therefore, IIDC compliant cameras are more suited for real-time video processing, as there is no need to uncompress the image data at the receiving ends.

Any camera that is compliant to the IIDC and IEEE1394 specifications can be controlled by a single application that is based upon these protocol specifications. For these characteristics, IIDC compliant IEEE1394 video cameras have been widely used for the uses such as robotics, machine vision for inspections, surveillance, and video conferencing.

2.2 Related Work and Merit of The Tcl Extension

To write this Tcl extension, we have used two open-source libraries: "libraw1394" [1] and "libdc1394" [5]. "libdc1394" gives users full and complete control over IIDC IEEE1394 compliant video cameras. This library also serves as the basis for "Coriander" [4], a graphic user interface to the "libraw1394" and "libdc1394" libraries. "Coriander" is a well featured program, giving users easy access to camera controls. It also has provisions for saving the video data in various ways. However, it is just a single application, and its use is limited to viewing and saving video.

With an ever-increasing rise in video camera usage, flexibility and extensibility are very important in the development of applications. The demands on video camera controlling applications cover such a wide variety of fields that no single application can fulfill all needs. To face the problem, we have developed a Tcl extension to harness the simplicity of the Tcl/Tk command set, and the power of the ”libdc1394” camera control library.

2.3 Tcl Commands

To put our extension in perspective, Figure 1 illustrates the flow of data between camera and computer. Our extension handles the jobs in the two boxes drawn in bold blue. The camera control part of the extension allows almost complete control over each IIDC IEEE1394 compliant camera. On the other hand, the Raw-to-Color conversion part is supplied to convert the raw images from cameras into the color images. For ease of use, we have split our command set into two namespaces – one for camera control and another for raw conversion.

The camera control commands can be split into two groups – initialization, and control. When cameras are connected to the system, the dc1394 library recognizes their existence, but does not take any actions. Initialization commands can also query the camera for its capabilities, such as resolutions and frame-rates that the camera can provide. Upon initiation, the camera becomes ready to start sending video data, which the user can use in various ways. All the features of the camera become available upon initialization. The available features include white balance, gamma, exposure, and iris size.

The following code illustrates the initialization process. It assumes that the first camera, when multiple cameras are connected, can be started with the preset specifications for format, mode, frame rate as shown in the code.

```
namespace eval ::dc1394 {
    set format 0
    set mode 320x240_YUV422
    set framerate 30

    set camera_list [list_cameras]
    ;# Assume at least one camera exists
    set camera [lindex $camera_list 0]

    initialize $camera $format $mode $framerate
}
```

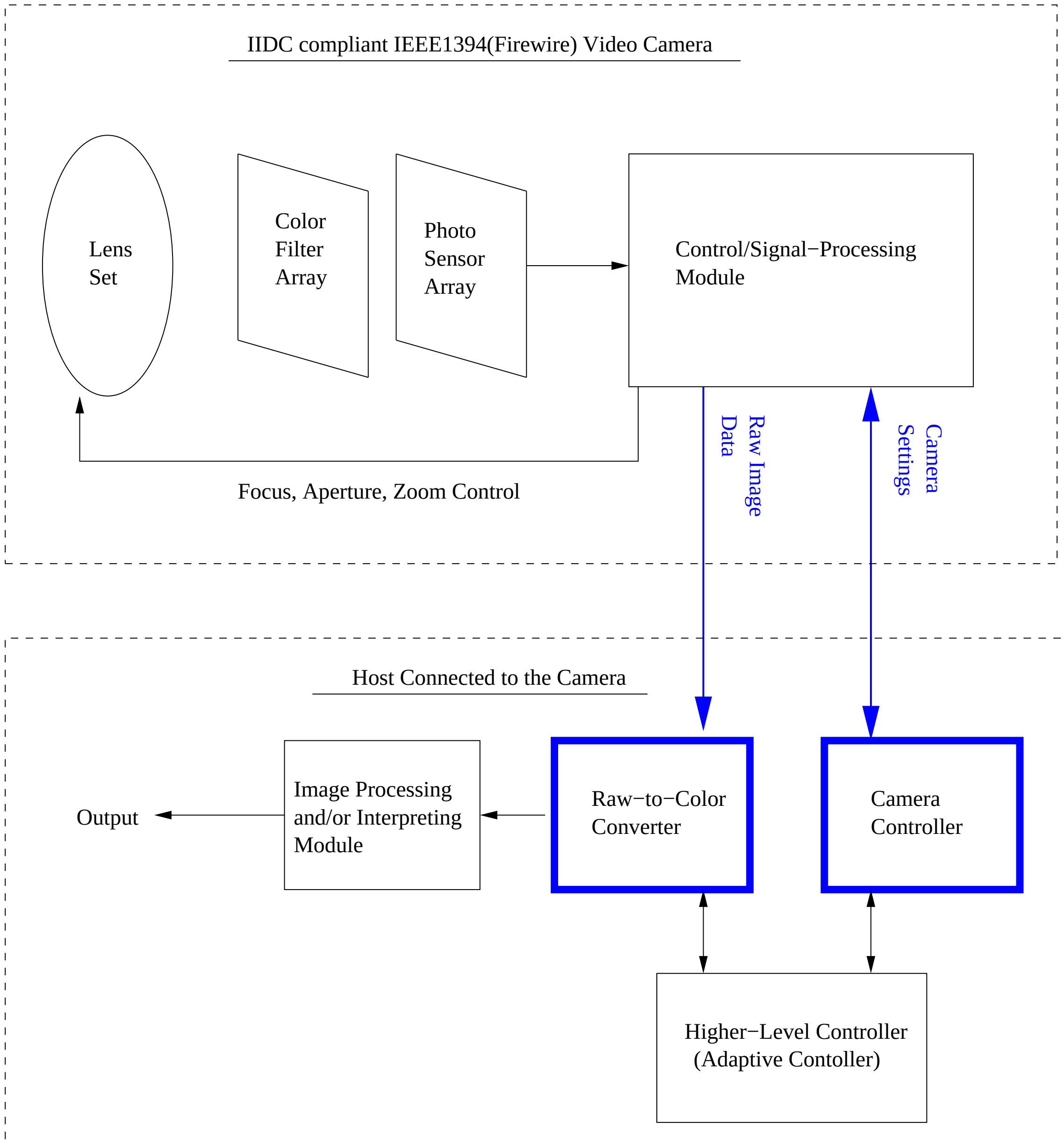

Figure 1: Data/Control-Signal Flow of IIDC IEEE1394 Camera and Computer

3 Raw-to-Color Converter

3.1 Raw Image and Color Filter Array

Most digital cameras today have what is known as a color filter array on top of their image-sensor. For each element in the sensor array the color filter limits the kind of light that can pass through the filter and reaches the element. In the case of the RGB Bayer filter, which is widely used in many digital cameras, it has three types of filter elements as shown in Figure 2: R (red), G (green), and B (blue). The R elements pass only red light, the G elements only green light, and the B elements only blue light. These three colors are the primary colors because every color can be created by a combination of them. This mechanism is necessary, since very few image sensors today can capture three colors simultaneously at every sensor array element. The raw image is the uncompressed data acquired by the sensor array elements in this manner.

However, the raw image does not make a color image in the usual sense, since each sensor array element can provide the intensity of only one of the three primary colors. In order to convert it to a true color image, the intensity of all three primary colors must be known at each sensor array element. Thus, various algorithms have been invented to guess the intensity of missing colors at each sensor. Raw-to-color converters do this guess work.

Unfortunately, there is no perfect algorithm for the conversion. Performance of the converter depends on various factors such as type of camera, type of target scene, and even temperature of the sensor array. There is also a trade-off between speed of the algorithm and the quality of the resulting color images.

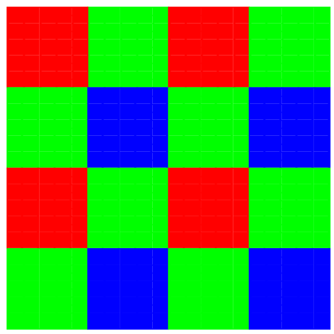

Figure 2: Part of Bayer Color Filter

3.2 Related Work and Merits of Tcl Implementation

Currently there is a very popular raw image converter, DCRaw [3], available in the free software arena. This software is updated very frequently to be compatible with new cameras and new raw file formats. However, it is a very minimalistic piece of software, and can only convert to PPM format. There are quite a few derivatives based on DCRaw. UFRaw [6] is one of them. It can convert raw images to just about any image format. It also has batch-processing capabilities, as well as a nice GIMP plug-in.

While these applications have many attractive features, their capability is limited by their limited customizability. As stated above, there is no perfect algorithm in raw-to-color conversion. Appropriateness of the conversion algorithm depends upon various factors as mentioned above. In such situations, customizability plays an important role to make an ideal converter for each particular problem. If Tcl could be used to control the component algorithms of those converters, it would greatly facilitate the customization of the conversion work. Unfortunately, these converters are not designed to work as Tcl components.

Therefore, we have implemented the various interpolation algorithms as Tcl procedures. We have also provided methods for users to implement their own algorithms by providing fundamental components that can be used for building new algorithms. By harnessing the power of Tcl, users can now select from a variety of algorithms depending on the circumstance.

3.3 Tcl Implementation

We are still in the process of augmenting the conversion capability. Currently our extension can perform four of the most commonly used conversion algorithms: nearest neighbor replication, bilinear interpolation, cubic convolution interpolation, and smooth hue transition interpolation. The implementation of each algorithm provides broad customizability in terms of the number of adjustable parameters.

However, the most important property of our implementation is that it provides various component procedures (which can be used as components of customized new algorithms) as Tcl commands. For example, its command set includes procedures for gradient computation, Gaussian convolution, Laplacian of Gaussian convolution, etc. This scheme enables great customizability that no existing monolithic converter can achieve. By using these component procedures, users can implement their own algorithms by taking advantage of the rapid and easily programmable nature of the Tcl language.

4 Conclusion

We have been developing a unique Tcl extension that can facilitate the use of IIDC compliant IEEE1394(Firewire) video cameras and raw-to-color digital image conversion. Although there are other applications serving these objectives, they are implemented in a compiled language and do not provide a facility for building customized applications easily in a scripting language such as Tcl. Their degree of customizability is limited to changing only a few parameters.

In both the video camera control and the raw-to-color conversion, there is a great demand for customization beyond the capabilities of the existing applications. This can be ascribed to the wide range of objectives and environments in which video cameras are used. Our Tcl implementation makes an ideal solution to achieve the goal of building customized applications easily for controlling IIDC IEEE1394 video cameras and converting raw image data to true color images.

Appendix

Commands for Controlling the IIDC IEEE1394 Video Cameras

- List all cameras connected on all available cards

  ::dc1394::list_cameras
  (Example) puts [::dc1394::list_cameras] → cam0 cam1 cam2

- Initialize a camera specifying format, mode, and frame rate

  ::dc1394::intialize <cam> <format> <mode> <frame rate>

- Retrieves the camera's available formats, modes, and frame rate

  ::dc1394::caminfo <info type> <other arg> ?...?

Info Type Arg	Following Args
format	<cam>
modes	<cam> <format>
framerate	<cam> <format> <mode>

  Note: Modes for the corresponding formats can be found on page 13 of the IEEE1394 specs

- Save a frame to image object or file

  ::dc1394::save image <cam> ?<imgname>?
  ::dc1394::save file <cam> <filename>

  Not specifying <imgname> creates a default tcl image name (i.e. image1). The format used when saving to a file is ppm.

- Start processing frames indefinitely using callbacks (see "callback" command below.)

  ::dc1394::startprocessing

- Stop the processing started by the above startprocessing command

  ::dc1394::haltprocessing

- Get, set, and query feature values

  ::dc1394::feature get <cam> <feature>
  ::dc1394::feature set <cam> <feature> <value>
  ::dc1394::feature available <cam> <feature>

  The last command (feature available) returns 1 when true, 0 when false.

(Available Feature Args)	
brightness	iris
autoexposure	sharpness
whitebal	hue
saturation	gamma
shutter	gain
focus	temperature
whiteshd	framerate
tilt	pan
zoom	

- Set or remove a callback function associated with an event

::dc1394::callback <cam> <eventname> ?<callback>?

If <callback> is empty, the current callback is removed. <eventname> includes ”newframe”.

References

[1] libraw1394. *http://www.linux1394.org*.

[2] *IIDC 1394-based Digital Camera Specification Ver. 1.31*. The 1394 Trade Association, 2004.

[3] D. Coffin. Dcraw. *http://cybercom.net/ dcoffin/dcraw*.

[4] D. Douxchamps. Coriander. *http://damien.douxchamps.net*.

[5] D. Douxchamps. libdc1394. *http://damien.douxchamps.net/ieee1394/libdc1394/*.

[6] U. Fuchs. Ufraw. *http://ufraw.sourceforge.net*.