Enhanced Video Capture Support in OpenCV under Linux

Csaba, Kertesz

University of Szeged, Faculty of Science

Abstract

The wide-ranging multimedia support is very important in a desktop operating system if somebody would like to use the PC for entertainment or research projects (e.g. image processing, robot vision applications etc.). It is unavoidable to acquire sound or image sequences with different multimedia devices. The Intel’s OpenCV (Open Source Computer Vision Library) library can be used for image processing and computer vision solutions. This library is available under different operating systems such as Linux and grown to a standard in the image processing applications last years. The program is constantly under development thus the author of this paper also significantly improved a few video capture features of this library. These results are summarized in the following.

Subjects: OpenCV, Video4Linux, Video4Linux2, video capture, webcam

1. Introduction

Linux produced a noticeable advance last years, also in the area of driver programming. A lot of programmers recognized the possibility to make drivers for various multimedia devices and allowed of better work with Linux day by day. However the support for multimedia hardware is still far behind that for other operating systems, sometimes due to the lack of support and documentation by hardware manufacturers, sometimes due to licensing problems. Another major reason for this was the lack of a standardized programming interface.

A device driver is not usable for everybody if the author of driver writes it without collaborating with others using a common API. This purpose is served by V4L1 and V4L2 specification2. Video for Linux is the first generation of standards for using video devices under Linux. The V4L API was firstly introduced in Linux 2.1 to unify and replace various TV and radio device related interfaces, developed independently by driver writers in prior years.

After a few years Video for Linux Two was a very good replacement for the V4L API that comes with the official newest Linux kernels. Starting with Linux 2.5 the much improved V4L2 API replaces the V4L API, although existing drivers continued to support V4L, either directly or through the V4L2 compatibility layer3.

OpenCV maintained by Intel is an image processing library which is available under Windows and Linux too. The OpenCV implements a wide variety of tools for image interpretation. In spite of primitives such as filtering, image statistics, pyramids, OpenCV is mostly a high-level library implementing algorithms for calibration techniques, feature detection and tracking etc.5

OpenCV has a special sublibrary which does not contain image processing methods but it equips with wide-ranging facilities for handling image data like loading/saving images, video capture/storage functions. One of these features is video capture which has different support under Windows (DirectShow) and Unix-like systems (Videoodev module, FireWire).

More improvements were written in video capture capabilities under Linux because there was only poor support in OpenCV regarding all prospects of V4L/V4L2 API. The following chapters describe the main features of V4L/V4L2, the old status of the video
capture support in OpenCV and the new improvements of the author in this part.

2. V4L/V4L2 Technical Background

There are two different parts in the Linux kernel regarding this multimedia programmable interface:

- API (Application Programming Interface),
- V4l2 module.

The API of V4L is defined in videodev.h, Videodev module in videodev.c. The API of V4L2 in videodev2.h and module in videodev2_c respectively (files are parts of official kernels)7. The API is responsible for defining specific and Videodev module for handling several video drivers (registration and administration).

The user reaches the interface through /dev logical directory under /dev/videoX nodes (X means an integer value). If a new device is plugged Linux will try to use an appropriate driver with Videodev module. The driver of device will register by Videodev module and V4L/V4L2 will create a special node (for example /dev/video0) which is a connection point between the user and the kernel driver of device. Abstraction layers are demonstrated in Figure 2 after References section. Nevertheless Videodev module only takes care that the right driver is reached with ioctl calls.

V4L/V4L2 supports following features with their API:

- Get/set capture main capabilities,
- Get/set image properties,
- Switch capture memory buffer modes,
- Synchronization of capture,
- Set audio mode etc.

The present situation is that many drivers of capture devices have got only Video for Linux support available. The V4L is the older specification and drivers are more stable in this mode thus V4L and V4L2 are living parallel and providing standard API layer for all video drivers.

3. Old V4L/V4L2 support in OpenCV

OpenCV did not support V4L2 till this summer. It handled only Video for Linux with following features:

- Automatic video source detection (it was ready for V4L2 also),
- Automatic palette detection,
- Memory allocation with mmap mode,
- Synchronization of capture,

- Ability to change capture resolution during capture process.

The biggest lack was no possibility to change image capture capabilities and adjust image properties (e.g. contrast, brightness) in OpenCV. This is very important in several applications to improve image quality with changing hardware capture control parameters which are useful many times instead of standard software algorithms because of e.g. decreasing CPU load or better image quality.

4. The developed novel image properties for OpenCV

Firstly the image properties for V4L and V4L2 devices were changed and the API of OpenCV (in concert with OpenCV developers) was modified with a few new constants in order to handle new properties:

- Brightness,
- Contrast,
- Saturation (it has a different name "Colour" in V4L),
- Hue,
- Gain (this feature in V4L is not supported).

These five image properties cover all features of the capture devices. Naturally, support of the several elements depends on implementing of the device drivers and the hardware design. If a driver or device does not support a property it generates an error message and functions of OpenCV return with value -1.

Nevertheless, there are differences between handling image properties in V4L and V4L2. Properties have a constant range 0-65535 in V4L but it changes in different driver implementations. Many times the limits of this range must be surveyed with manual modification of properties because of the lack of automatic detection of these values in V4L. On the other hand Video4Linux2 supports to get minimum and maximum limits of capture image properties that the new patches taking it into account.

The user scope is that the image features have float ranges 0-1 specified by the official API of OpenCV. The distinct value ranges in V4L/V4L2 solutions were handled and normalized in the new codes. But OpenCV detects range limits of properties only for V4L2 devices since the achievement would be very difficult in V4L.

5. New V4L2 support in OpenCV

Only another possibility was for the users if they did not have a V4L compatible device under Linux Cam-
era with FireWire connection could be used instead of a V4L device but it was a partial solution because V4L2 provides also a very good multimedia support.

The first step of V4L2 development was the buying of a webcam (Genius VideoCam NB Security) which possesses a real stable V4L2 compatible device driver. After originating of hardware basis of development, initialization process of capture devices in OpenCV was rewritten and latter separation of V4L/V4L2 codes was designed. If user installs a new device in OpenCV the process is following:

1. Detection as V4L compatible device
2. Detection as V4L2 compatible device

Note that the API of V4L2 is more advanced than the API of V4L however many times device drivers are more stable in V4L mode therefore OpenCV tries to use every capture device as V4L at first.

A new palette (codec) detection for V4L2 was written. It supports following codecs/colorspaces:
- RGB (24 bit, uncompressed),
- YUV 4:2:0,
- YUV 4:1:1,
- SBGGR8 codec,
- SN9C10x codec.

First three standard codecs were implemented earlier in OpenCV for V4L devices and they are used by many webcams and capture cards. Last two codecs are specific by Genius VideoCams and they were introduced by the author in OpenCV however original SBGGR8 codec is part of a SN9C10x webcam application⁸ and SN9C10X codec was written by Takaumi Mizuno⁹.

The developed patches use mmap memory allocation function in image buffer. In that way the capture process will be faster as the user mapped function (latter provides only one frame in each capture period after clearing out image buffer). Another implemented feature is the synchronization of the capture process which is important to capture each frames after the others.

6. Test hardware/distributions

The developed patches for OpenCV were tested on several hardware platforms and distributions with various webcams (Genius NB Security, Creative Vista and some Logitech). That shows Figure 3 and Figure 4 after References section. Basically, test hardwares were i386 architectures although new 64 bit systems were presented also in tests (both AMD and Intel).

It laid special emphasis on testing more versions of software environments:
- Different branches of Linux kernel (2.4.x and 2.6.x),
- Latest official release and CVS version of OpenCV,
- Debian (Stable and Testing) and Suse distributions.

The development in testing status was not met with big difficulties, there were only a few problems with compiling OpenCV and CamView (the later discussed sample application) caused by distinct ways of Linux distributions (e.g. library locations in filesystem). That was not needed to make a lot of changes in new patches because Linux systems have only different versions of V4L/V4L2 layer is provided by kernel and VideoCodec module which can be handled simply. Eventually all tests were successful without any problems.

![Figure 1: Screenshot from CamView program](image)
7. Sample application

A simple viewer application called CamView\(^6\) was developed in GTK/GNOME graphical environment for webcams and other V4L and V4L2 compatible devices. It uses standard capture procedures of OpenCV and a few calibration functions in order to test easily effects of changing capture control parameters (Figure 1). The program can detect a chessboard on an image by help of OpenCV and show internal corner points detected by subpixel accuracy.

The window of CamView application was divided into four parts. The first part is the captured image of V4L/V4L2 device, in the second there is the main controls of capture features (device index /dev/videoX, fps and resolution), the third shows the properties of the captured image (contrast, brightness, saturation, hue, gain) and the task of the last part is the settings and to recognize and show the calibration chessboard.

Usage of program is very simple. After application was run the user can change the calibration control and the options of image properties and commit with Configure button. If an image property is not supported by capture device or driver the horizontal slider belongs to it will jump back to value zero. Program halts and prints error messages by strong situation (e.g. unsupported capture resolution).

The user can test the capture device in a few minutes with CamView and experience which capture capabilities are adjustable on it.

8. Summary

The paper have described the new API changes of OpenCV and the patch design with collaboration of two maintainers of Intel’s library (Olivier Bonet and Vadim Pisarevsky). Four more important additions of OpenCV codes were created and tested on more types of computers and Linux distributions with several webcams. Test results were good and big differences can not be found between different computer platforms. This is a main advantage of the common standards and APIs in Open Source Community under Linux. New modifications are available on CVS page of OpenCV\(^4\).

Current status of the V4L/V4L2 support patched by the author with maintainers of OpenCV was discussed and conclusion is the following: the new additions are stable and will take care of only future modifications of V4L/V4L2 kernel components or other standards.

This project confirmed V4L/V4L2 solutions have a very pleasant, good purposeful API for wide-range of the multimedia applications which can be implemented quickly and efficient. Eventually an open source project with good planned API and program architecture (e.g. OpenCV) is very useful and it can be improved by a programmer swimmingly.

References

1. Video4Linux resources: specifications, drivers for many devices etc.
   URL: http://www.exploits.org/v4l/

2. Video4Linux2 main homepage: Specifications, drivers, programs etc.
   URL: http://linux-hatesex.org/v4l2/
   URL: http://linuxtv.org/v4lwiki/index.php


4. OpenCV library homepages: General informations on Intel’s pages and newest releases, CVS version of code on Sourceforge.net project page.
   URL: http://www.intel.com/technology/computing/opencv/
   URL: http://sourceforge.net/projects/opencvlibrary/

   URL: http://developer.intel.com

6. CamView program. Part of Cognitive Vision project.
   URL: http://www.sourceforge.net/projects/aibo/

7. Official kernel distributions of Linux systems.
   URL: http://www.kernel.org

8. SN9C10x official driver and webcam application.
**Figure 2:** Abstraction layers to handle video devices in Linux.

<table>
<thead>
<tr>
<th>Name</th>
<th>Processor</th>
<th>Kernel</th>
<th>Distribution</th>
<th>OpenCV</th>
<th>Tester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toshiba Tecra 8200 Notebook</td>
<td>P3</td>
<td>2.6.12.2</td>
<td>Debian Stable</td>
<td>0.9.7</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>PC</td>
<td>AMD Athlon XP</td>
<td>2.6.12.2</td>
<td>Debian Stable</td>
<td>0.9.7</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>PC</td>
<td>Intel P4</td>
<td>2.6.10</td>
<td>Debian Stable</td>
<td>0.9.7</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>PC</td>
<td>Intel P3</td>
<td>2.4.21-99-default</td>
<td>Suse 9.0</td>
<td>0.9.7</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>PC</td>
<td>Intel P4</td>
<td>2.6.11.5</td>
<td>Debian Testing</td>
<td>CVS</td>
<td>Olivier Bomet</td>
</tr>
<tr>
<td>PC</td>
<td>P4 bit Pentium</td>
<td>2.6.12</td>
<td>Debian Testing</td>
<td>CVS</td>
<td>Olivier Bomet</td>
</tr>
<tr>
<td>PC</td>
<td>P4 bit AMD</td>
<td>2.6.12</td>
<td>Debian Testing</td>
<td>CVS</td>
<td>Olivier Bomet</td>
</tr>
</tbody>
</table>

**Figure 3:** Features of tested platforms

<table>
<thead>
<tr>
<th>Name</th>
<th>Driver name</th>
<th>Driver version</th>
<th>Driver compliance</th>
<th>Tester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genius NB Security</td>
<td>smac10x</td>
<td>1.24</td>
<td>v4l2</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>Creative Vista</td>
<td>spca5xx</td>
<td>0.57</td>
<td>v4l</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>Logitech Quickcam Express</td>
<td>spca5xx</td>
<td>0.57</td>
<td>v4l</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>Logitech Messenger</td>
<td>qc-usb-messenger</td>
<td>0.8</td>
<td>v4l</td>
<td>Csaba Kertesz</td>
</tr>
<tr>
<td>Logitech Quickcam</td>
<td>pwc</td>
<td>10.0.7-2</td>
<td>v4l/v4l2 unstable</td>
<td>Olivier Bomet</td>
</tr>
</tbody>
</table>

**Figure 4:** Tested webcams.