Second Cut

a software to record and replay fencing videos

General software architecture

Second Cut can be seen as a wrapper that manages an interaction logic between elementary programs. Indeed, it uses *mplayer* to play the video, it uses *ffmpeg* to record the camera live feed. After the recording process has been stopped, *mencoder* will rebuild the index of the movie, so that it can be correctly played frame by frame.
Since a device (like the /dev/video0 that represents a camera) in linux can only be opened and read by a single software, Second Cut uses the solution consisting in redirecting the live feed from the original camera device to a virtual one called *v4l2loopback* (created by the *v4l2loopback* module loaded in the linux kernel). The loopback has the nice properties of being fast (because powered at the kernel level) and readable by as many softwares as needed, in this case mplayer and ffmpeg. That redirection is made by *gstreamer*. To grasp information about the cameras, it uses *v4l-info.*

Installation steps

The following instructions apply to install correctly the software an a machine running Ubuntu Oneiric. On older and younger versions of ubuntu, similar instructions may apply, as well as on Debian-based distributions. Their similarity with Ubuntu Oneiric (actually ubuntu is based on Debian) supports that there is a good chance Second Cut will correctly run, however there is no strict guaranty, since the versions of the dependent softwares might be different and contain diverse features/bugs.
note : the following format :

- `echo ok`

indicates the exact string (command line) to type into a terminal. in this example type echo ok then hit ENTER : it will write ok on the terminal screen.

First, open a terminal or a console.
The assumption is made that the user is able to execute sudo commands, i.e. to make administration of the computer or have superuser power.

Installing Gambas IDE and runtime (needed even if no source is modified, needed to run the executable, since gambas is an interpreted language)

- `sudo add-apt-repository ppa:nemh/gambas3`
- `sudo apt-get update`
- `sudo apt-get dist-upgrade`
- `sudo apt-get install gambas3`

install mplayer and mencoder :

- `sudo apt-get install mplayer mencoder`
- `mencoder -v`

MEncoder svn r34540 (Ubuntu), built with gcc-4.6 (C) 2000-2012 MPlayer Team

install gstreamer:

- `sudo apt-get install gstreamer-tools`
- `gst-launch --version`

gst-launch-0.10 version 0.10.36
GStreamer 0.10.36

install modprobe and lsmod

- `sudo apt-get install module-init-tools`
- `modprobe -V`

module-init-tools version 3.16

check grep, cut, dmesg are installed

- `sudo apt-get install grep coreutils util-linux`

check the v4l2loopback is loaded

- `lsmod | grep v4l2loopback && echo ok`

if nothing appears (not ok) then load the module

- `sudo modprobe -v v4l2loopback  && dmesg | tail`

if it says FATAL: Module v4l2loopback not found. then install the module

- `sudo apt-get install v4l2loopback-source module-assistant`
- `sudo module-assistant auto-install v4l2loopback-source`
- `sudo modprobe v4l2loopback && dmesg | tail`

the last line output by the last command should say :
v4l2loopack driver version 0.4.1 loaded

install the v4l2loopback module everytime linux starts :
edit the /etc/rc.local file as super user :

- `sudo nano /etc/rc.local`

add the following lines (BEFORE the last line exit 0, insert :)

```
##############
apt-get install -f module-assistant v4l2loopback-source  > /tmp/rc.local.out
module-assistant auto-install v4l2loopback-source >> /tmp/rc.local.out
# MAKE 3 loopback devices
modprobe v4l2loopback devices=3 >> /tmp/rc.local.out
ls -l /dev/video* >> /tmp/rc.local.out
##############
```

press CTRL and o to save the file (then Enter)
press CTRL and x to exit the editor

install v4l-info from the v4l-conf package:

- `sudo apt-get install v4l-conf`

test if the v4l2loopback gstreamer plugin is available

- `gst-inspect v4l2loopback`

if it drops one line saying no element or plugin «v4l2loopback» then install the compilation tools

- `sudo apt-get install git make gcc autoconf automake autotools-dev m4 libltdl-dev libtool`
- `sudo apt-get install libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev`

(this installs also : gir1.2-gst-plugins-base-0.10 gir1.2-gstreamer-0.10 libglib2.0-bin libglib2.0-data libglib2.0-dev libgstreamer-plugins-base0.10-dev libgstreamer0.10-dev libxml2-dev zlib1g-dev)

and compile and install the gstreamer plugin **for each linux user** that will use Second-Cut on the machine :

- `cd /tmp && git clone https://github.com/umlaeute/gst-v4l2loopback.git`
- `cd gst-v4l2loopback`
- `./autogen.sh && make && make install && echo ok`
- `gst-inspect v4l2loopback`

install ffmpeg (the latest version not the default ubuntu one)

- `sudo add-apt-repository ppa:jon-severinsson/ffmpeg`
- `sudo apt-get update`
- `sudo apt-get dist-upgrade`
- `sudo apt-get install ffmpeg`
- `ffmpeg -version`

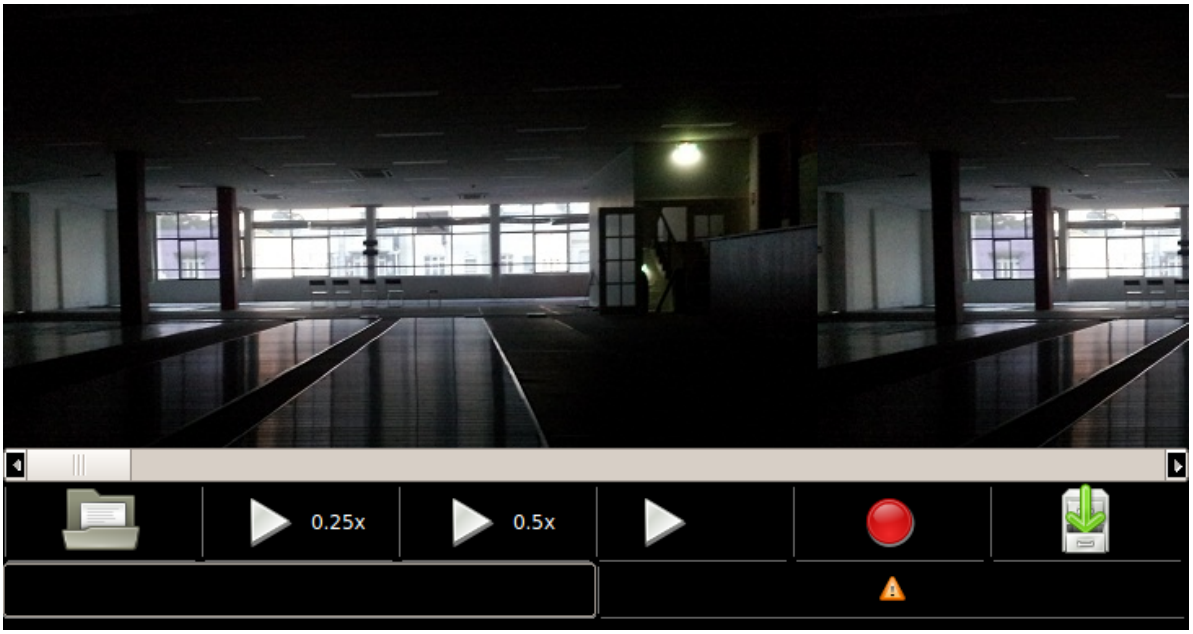
ffmpeg version 0.10.6-6:0.10.6-0ubuntu0jon1~oneiric1

finally, download the Second-Cut software : source and binary

- `cd ~ ; git clone https://github.com/sebkst/Second-Cut.git`

it should download a *Second-Cut* directory with a *bin* subdirectory containing the binary executables.
To run SecondCut3.4.gambas : write its FULL path name in the terminal, or as a shortcut destination (edit the file second-cut.desktop to correct the paths, and put it inside your desktop directory)

Usage



The main GUI window (just above) is composed of an empty space to show the movie, and 3 rows of elements that help to control it.
On the first row there is just the slidebar representing the total length of the movie. The cursor position shows the actual time position playing,
On the second row, 6 buttons : open file, play at quarter, half, normal speed, record, configure.
On the third row, there is a text field on the right that shows the current time position in hour minutes seconds, while recording and playing. On the right there is a «panic button» that stops the recording and the playing processes in case the logic of the other buttons gets stuck. This button should not normally be pressed, there is a risk the video recording does not end correctly and the video might not be usable, this button is only useful when trying to modify/ test new parameters or cameras; As a general rule, **Press the record button again to stop a recording instead of using this panic button.**

Slidebar
The slidebar is only active/enabled for interaction when playing a movie. The cursor position shows the actual time position playing, If the user wants to go to another position, just click on that position in the sidebar and it will seek in to the stream to fetch that position.
This move in the chronology of a movie is activated on 2 different events : mouse button pressed and mouse button released. So if you drag the mouse pressing the button at position A, the moving towards different position B and releasing the button at position B, the movie will be forced to goto position A first, then forced to go to position B

Logic of pressing a button:
When playing a movie at speed X, pressing a button to play at a *different* speed Y will change the speed and let the movie play.
When playing a movie at speed X, pressing a button to play at the *same* speed X will make the process pause.
When the movie is paused, or finished, pressing any of the play buttons will make it play/ resume at the specific speed of the button.

You cannot change the configuration, the target directory for saving the movie,... while recording.The configuration button has to be pressed before the recording button.
When the recording button is pressed, its background becomes gray during the recording time. To stop the recording : press the same record button again, or a play button.
If there is an error (of configuration, of space, of availability of camera ...) a message box will appear. an the recording process should stop.

Keyboard shortcuts : (small or capital letters)
R : press the record button
S: pause/resume the movie playing. do not change the speed. While recording: no effect
SPACEBAR : like S
P: Replay Last video at normal speed (like pressing play at normal speed button)
H: Replay Last video at half speed (like pressing play at half speed button)
Q: Replay Last video at quarter speed (like pressing play at quarter speed button)

F: Replay Last video, frame - by - frame mode : it synchronizes (and pauses) the video stream on the next available frame. letting F pressed acts as repeating pressing F (according to the normal keyboard repeat rate)

D : seek backwards to the last X seconds, depending on the value o the «time to seek backwards parameter». by default : go back one second before the current position.

UP arrow : seek to the start of the video and pause there
DOWN arrow : seek to the end of the video and pause there

LEFT arrow : like D : seek backwards to the last X seconds, depending on the value o the «time to seek backwards parameter». by default : go back one second before the current position.

RIGHT arrow : like F : Replay Last video, frame - by - frame mode : it synchronizes (and pauses) the video stream on the next available frame. letting the right arrow pressed acts as repeating pressing the arrow (according to the normal keyboard repeat rate)

View the movie in frame by frame replay, with dynamic adjustment of the speed: dragging inside the video frame while playing a movie:

An other mode than repeating pressing the F key: to view frame by frame, clic with left button inside the video image.
Keep the left button pressed will activate a frame by frame automatic replay. (or seek function backwards)
When you release the mouse button, the movie will be paused (auto refresh stops).
The frame refresh frequency is determined by the horizontal distance between the mouse cursor and the center of the video frame.
Meaning if you press in the center of the frame, the frames will refresh very slowly.
If you press near the right border then the frame will refresh very quickly, close to normal speed.
If you press on the *left half* side of the frame, it will go backwards the the fixed number of seconds according to the configuration, repeatedly.
To modify the speed, just drag the mouse cursor closer to the center or to the edges (move while keeping the left button pressed).

Configuration file

Not al the parameters are configurable graphically, because they do not need to be customized on everyday use. (see FConfiguration below)
The software relies on a file placed in the home directory of the current user : «.secondcut»
Notice the initial dot that makes that file hidden by graphical filemanagers.
If the file is not found, then it creates a new one (when saving, i.e. when the ok button from the configuration window is pressed, or when you close the program)
If the program is run on command line as

- ./SecondCut.gambas -config /path/to/myconfigurationfile

Then the file /path/to/myconfigurationfile is read at first when the program starts. After that, any other modification /or closing of the program will write the new configuration to the usual «.secondcut» default file.

The name of the parameters are prefixed with the class name of the running instance made by the Main module (see Source Code Architecture), in this case FRecffmpeg.

Parameters:

FRecffmpeg.videodevice is the path of the device created by the linux kernel for the camera, usually /dev/video0 if there is a built-in camera. Default value is "/dev/video1" since the genius widecam is a usb camera connected to a pc with already a built-in camera.

FRecffmpeg.saveto is the path of the directory where to save the recorded videos. Default is the home folder of the current user.

FRecffmpeg.filenameskeleton is the Gambas code to format the name of the video recorded. It produces a unique name for each files. Default is :
"Format(Now, \"yyyy-mmm-dd_hh-nn-ss\") & \".mpg\""
If you want to add a prefix to the name, say "sydney_" then change the default for :
"\"sydney_\" & Format(Now, \"yyyy-mmm-dd_hh-nn-ss\") & \".mpg\""

FRecffmpeg.recordcommand is the command line used to record the live video from the device (actually the v4l2loopback device). it also specifies to include the sound stream recorded on the default alsa peripheral, which is the built-in microphone. Default value is
"ffmpeg -shortest -f alsa -i default -itsoffset 00:00:00 -ab 128k -f v4l2 -i $(VIDEODEVICE) -c:v mpeg2video -pix_fmt yuv420p -bf 2 -b:v 2500k -bt 300k -q:v 30 -bufsize 64M -y $(FILE)"

FRecffmpeg.recordformat is the «caps» passed to gstreamer to redirect the live feed from the camera device to the v4l2loopback. It specifies an height for the frame size, and gst-launch will negotiate the framerate and the frame width with the driver. Default value (records hd720 format) is "video/x-raw-yuv,height=720,interlaced=false,pixel-aspect-ratio=1/1"

FRecffmpeg.lastplayed is the path of the last video that has been recorded (even if not played) , or the name of the last file that was chosen using the «open file ...» button. Default value is: ""

FRecffmpeg.geometry is the dimension and position fo the main GUI window taken at the time the configuration file is saved (usually, the gometry of the program when you close it. No default. Typical value is *WidthxHeight+Xcoordinate+Ycoordinate*

FRecffmpeg.playcommand is the command line used to replay a video file.(not the same to show the live feed). Default value is :
"mplayer -vo xv -ac ffmp2, -demuxer lavf -noframedrop -identify -osdlevel 0 -wid $(WINDOWID) -speed $(SPEED) -input file=$(INPUTFILE) -input conf=/dev/null -nolirc -nojoystick -noar $(FILE)"

FRecffmpeg.recordsize is the size of the video frames to be recorded. It is strongly linked with the value of FRecffmpeg.recordformat. Default value is 1280x720
(which might not be standard for any camera in general)

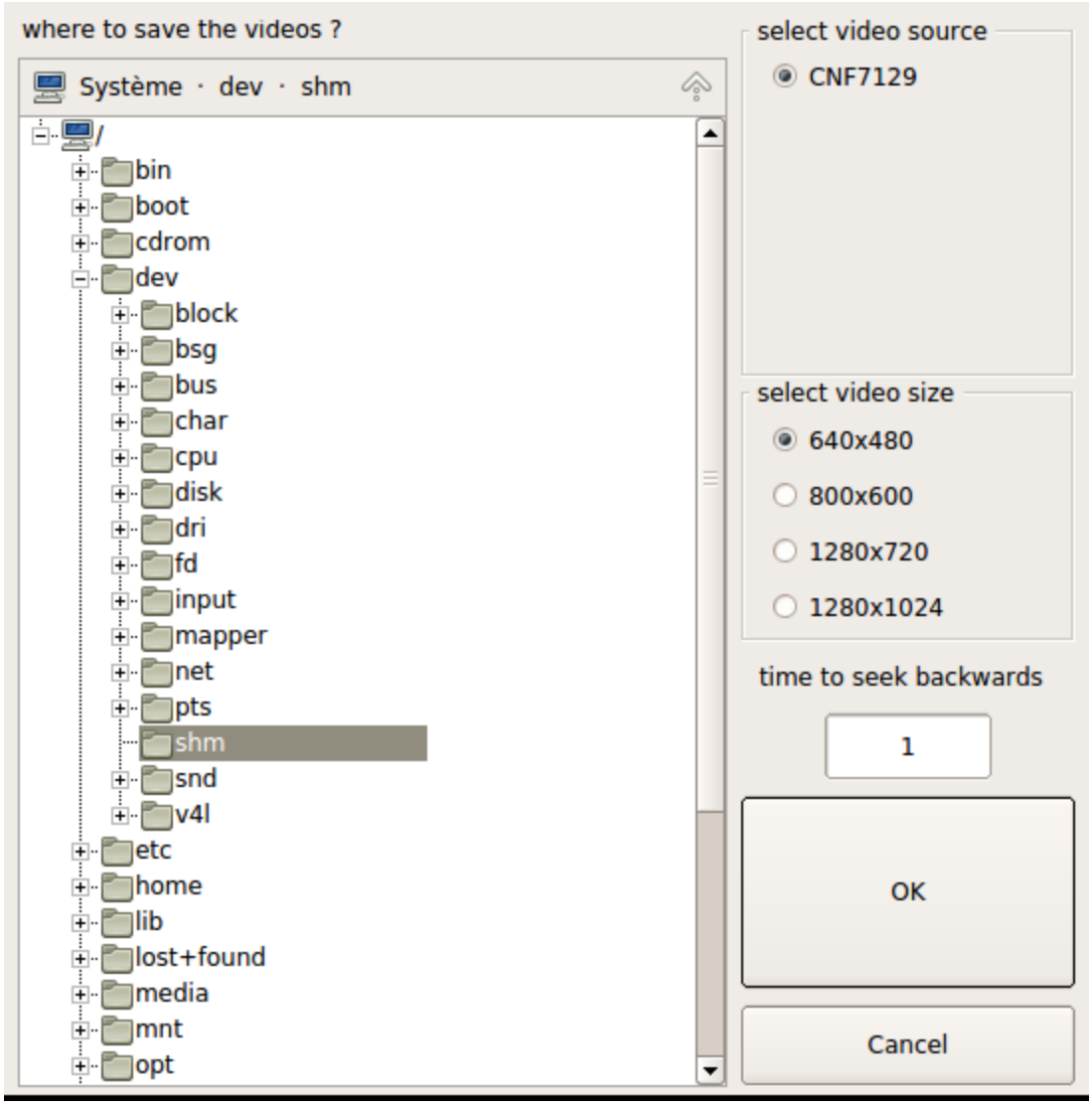
FRecffmpeg.seekbackwards is the number of seconds that the movie layer will seek backwards when pressed the LEFT arrow or the D key. The same number of seconds is repeatedly used when dragging the mouse inside the video frame while playing. This value is a float number. too small values may lead to desynchronisation of audio/video streams. Default value is 1

Source Code Architecture

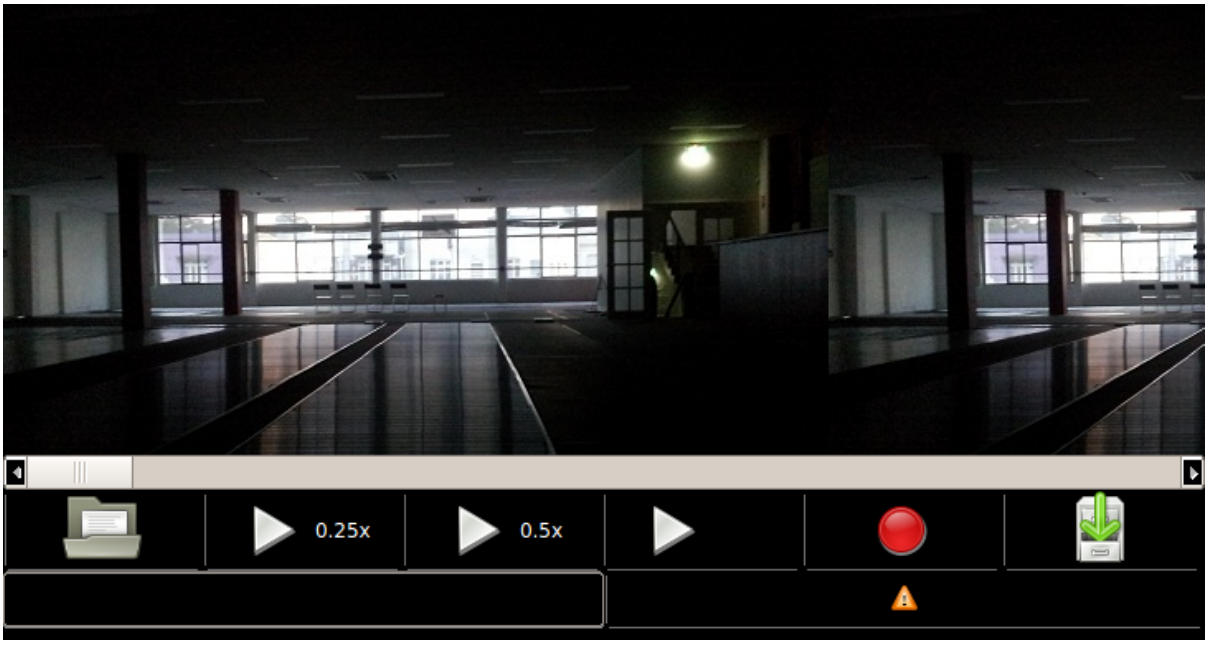
Second Cut is a project named SCMoviePlayer in the Gambas IDE(Integrated Developpment Environment) (v3.3.4). it is composed by modules, Forms and classes.

Main is the launcher module. It creates a new instance of the class FRecffmpeg.

MConfiguration is a module that retrieves/saves the parameters from/to a configuration file, namely «.secondcut» located in the user's home directory.



FConfiguration is the form that shows the main parameters to the user. It allows at run time to choose the directory where to save the recorded videos, choose the camera, choose the video size, and the number of seconds (float) used when seeking backwards.
The cameras are instantly listed by looking at the devices named /dev/video*, and their name is grabbed by the v4l-info program. The devices created by v4l2loopback is intentionally not listed. The list of formats (video size) is static: it might be that with some cameras, higher formats will not work because of the driver specifications. Lower formats lead to higher framerate, but less details. When OK is pressed, the parameters are saved to the configuration file.



FMoviePlayer is a form that draws the main GUI (Graphical User Interface) window and owns the logic of playing videos.
E.g. : logic of pressing a button :
If no movie is playing, it plays the movie at the button' speed (normal, half or quarter).
If a movie is playing but at a different speed, then it changes the speed.
If a movie is playing at the very same speed of the button, then it will make it pause.

FRecffmpeg is a class. It directly inherits of FMoviePlayer, so it has the exact same GUI look, and the same logic behind the buttons, but has additional features, namely all the logic for recording and showing the live video.

Building a new version of the software

open the gambas3 IDE. (start menu, programming, gambas3)
open the project (with File menu, open project ... on the left
  and navigate to choose the SCMoviePlayer directory

you can open the forms (GUI) or modules or classes by double clicking on their name in the «Sources» directory on the left side of the IDE.
The class (=module attached to the GUI form) is opened when you click on the first button «code» on the left in the taskbar of the central part showing the GUI

to rebuild an executable
menu project -> make -> executable