Multi-format Video Frame Grabber ImageJ Plugin - MVFG

Isabel Laranjo*, Joel Braga and Victor Alves

University of Minho, School of Engineering, Informatics Department, 4710-057, Braga, Portugal

ABSTRACT

Currently ImageJ presents itself as a good working tool for medical imaging research. Image processing and analysis using specific ImageJ plugins is a simple process, mainly due to its natively support for a wide range of image formats. However, there are medical imaging modalities (e.g. endoscopy, ultrasound), with video output formats that are still not supported by ImageJ (e.g. VOB, MPG). With the base package of ImageJ it is possible to open videos in AVI format[1] and with additional plugins (e.g., *QuickTime Movie Writer*[2]*, JMF Movie Reader*[3]*, QuickTime Opener*[4]) videos in *Quicktime Movie format* (i.e. MOV and PIC) are also supported. For other video formats it is necessary to convert the video to a supported format using a third party video converter application.

A video is basically a temporal sequence of images. An average endoscopy produces a five to ten minute video with a frame rate of 50 fps. Of these tens of thousands of images only a few are relevant, thus it is necessary to select and extract the relevant images to be analyzed / processed in ImageJ.

In this paper, we present an ImageJ plugin developed to select and extract frames from video in diverse formats (e.g. MOV, AVI, VOB, MPG, MP4, FLV, MKV). Its user interface allows for video viewing and browsing. During video visualization the user can grab frames and export them as images or an Image Stack to ImageJ. This plugin, entitled "*Multi-format Video Frame Grabber - MVFG*" was developed using the vlcj framework (a Java framework for the vlc media player)[5].

Keywords: ImageJ Plugin, Frame Grabbing, Multi Format Video

1. INTRODUCTION

ImageJ is a free and powerful software tool for image processing and analysis. With this software you can open, edit, analyze, process, save and print 8-bit RGB and grayscale, 16-bit integer and 32-bit floating-point images. This tool works with many image formats (e.g. BMP, DICOM, GIF, JPEG, TIFF), however when dealing with video, it becomes limited in terms of supported formats[1].

In the medical imaging field there are modalities/techniques with image output (e.g. *Computed Tomography* - CT, *Magnetic Resonance Imaging* - MRI) and others with video output (e.g. ultrasound, endoscopy). With the base package of ImageJ it is possible to open videos in AVI format[1]. It is also possible to work with videos in *Quicktime Movie format* (i.e. MOV and PIC) installing additional plugins (e.g., *QuickTime Movie Writer*[2], *JMF Movie Reader*[3], *QuickTime Opener*[4]). For any other video formats it is necessary to convert the video to a supported format using a third party video converter application (e.g. Format Factory for Windows[6], Any Video Converter for Windows[7] or Mac OS X[8],).

The number of frames that a particular device or optical electronic processes displays per time unit is called *Frame Rate*. This value is expressed in frames per second (fps) and the main systems deal with frame rates between 30 and 50fps. An average endoscopy video of 10 minutes with 50fps has about 30000 frames. It is not practical to work with a stack of so many images using ImageJ.

The plugin we propose tackles these two problems, working with most of the common used video formats and providing an user interface for frame grabbing.

* Further author information: (Send correspondence to I.L.)
I.L.: E-mail: isabel.laranjo88@gmail.com, Telephone:
J.B.: E-mail: joeltelesbraga@gmail.com, Telephone:
V.A.: E-mail: valves@di.uminho.pt, Telephone:

In the next section we will present a description of the developed plugin, used tools, requirements and functional limitations. In section three, we will expose functional details and examples and in the last section some conclusions will be presented.

2. MVFG PLUGIN

The "*Multi-format Video Frame Grabber - MVFG*" plugin was developed in order to solve the ImageJ lack of support for various video formats and provide users a new tool to help capturing images during video visualization.

This plugin was developed using the *vlcj library,* a library for the vlc media player, written in Java and used to implement several features, which allow for video manipulation. Using this library's resources, the developed plugin is able to read most of the common used video formats. Being both ImageJ and vlcj Java based frameworks, all programing tasks involved in the MVFG plugin development were greatly simplified (Figure 1). The existing ImageJ plugins are capable of converting all the frames of a video in AVI[9], MOV and PIC format into ImageStacks. One of the MVFG Plugin goals is to allow the dynamic addiction of selected frames to an ImageStack. These MVFG's features become important in video processing, not only by allowing to process videos in the most varied formats, but also by enabling video visualization, navigation and frame capture. The user can reduce a long duration video to a simple set of selected frames containing the main regions of interest, thereby shortening the video length.

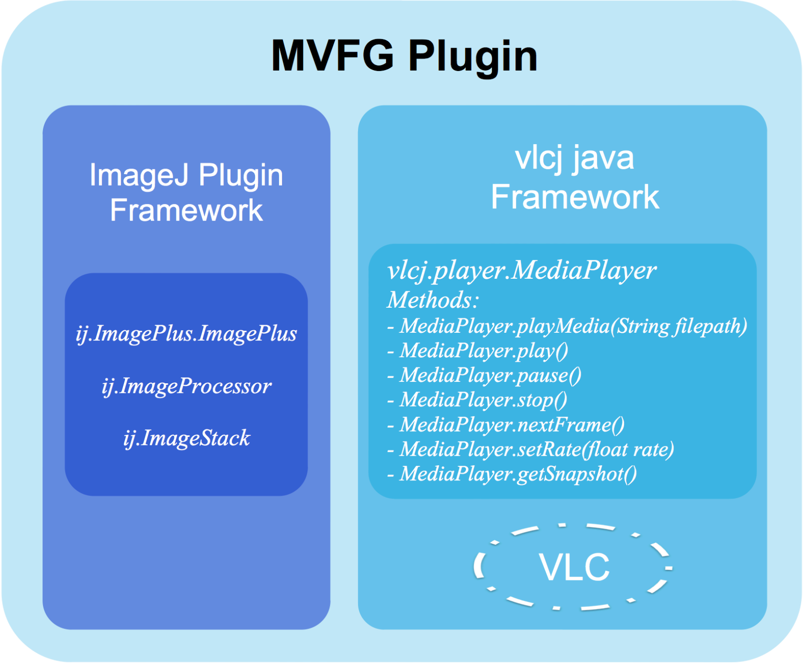


Figure 1 – MVFG architecture.

The implementation of the MVFG plugin was quite straightforward. Its main class extends the *PluginInFrame*, which is a subclass of an *AWT frame* that implements the *PluginIn* interface. This implementation detail was necessary since the vlcj framework requires an *AWT.Panel* to place the video using the *vlcj.component.EmbeddedMediaPlayerComponent* component. With this embedded component it is possible to get the component vlcj.player.MediaPlayer which provides for actions on the video. For example, to start the video playback it is necessary to use the *MediaPlayer.playMedia(String filepath)* method, whereas the video manipulation operations can be performed with the *MediaPlayer.play(), MediaPlayer.pause(), MediaPlayer.stop(), MediaPlayer.nextFrame(), MediaPlayer.setRate(float rate)* methods. The *MediaPlayer.getSnapshot()* method captures the video current frame when it is invoked. This corresponds to an *AWT.image.BufferedImage* which is passed as a Image parameter, creating a new instance of the *ij.ImagePlus.ImagePlus(String title, Image img)* object. Once created the ImagePlus instance, the *ij.ImageProcessor* can be obtained and therefore the *ij.ImageStack*. These components are then easily used in ImageJ.

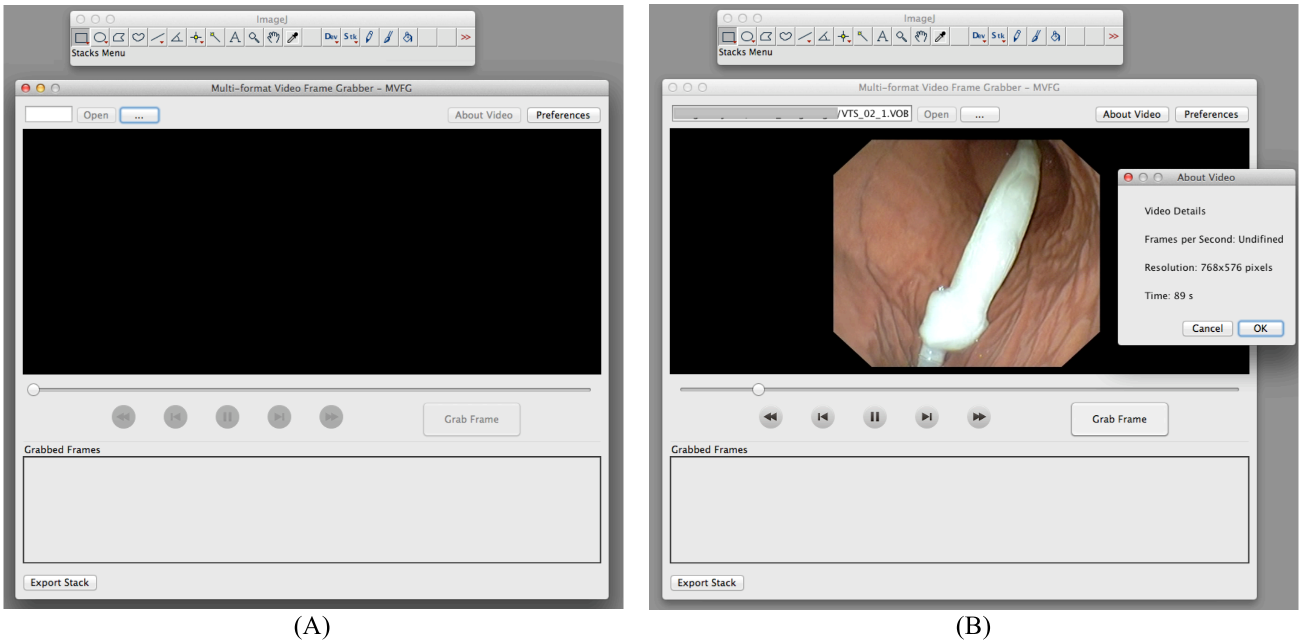
There are other vlcj Framework methods used in our plugin, e.g. *MediaPlayer.getFps()* to get the frames per second, *MediaPlayer.getTime()* to get the total time of the video playing, which in conjunction with the *MediaPlayer.setTime()* allows for video navigation.

When using MVFG the only requirements are an operating system that supports Java and ImageJ. The only known limitation that we encounter was the mandatory use of a 64-bit version of ImageJ on Mac OS.

The plugin has been tested in MAC OS 10.6 or later and Windows 7 operating systems, using the following video formats: VOB, MKV, MPG, AVI, MP4. With the tested formats, the expected results were obtained (i.e. video visualization, frame capture, export to stack). However, using the VOB video format it was not possible to obtain the video's resolution using the "*About video*" option. For all other formats there were no errors or problems detected.

3. USER INTERFACE

The developed plugin's user interface is quite intuitive. Through the start menu it is possible to open a file by searching the hard disk or by entering the path to the file in the text box. The panel for video visualization and its controls as well as the buttons "*Grab Frame*" and "*Export Stack*" are located in the center of the window (Figure 2 (A)). After opening the video file several actions are enabled. In the upper right corner, the user can request additional information about the loaded video (e.g., *FrameRate, Total time of video, Resolution*) using the "*About Video*" button (Figure 2 (B)). Using the "*Preferences*" option it is possible to modify the size of the plugin´s window.



Video Details

Frames per Second: Undifined

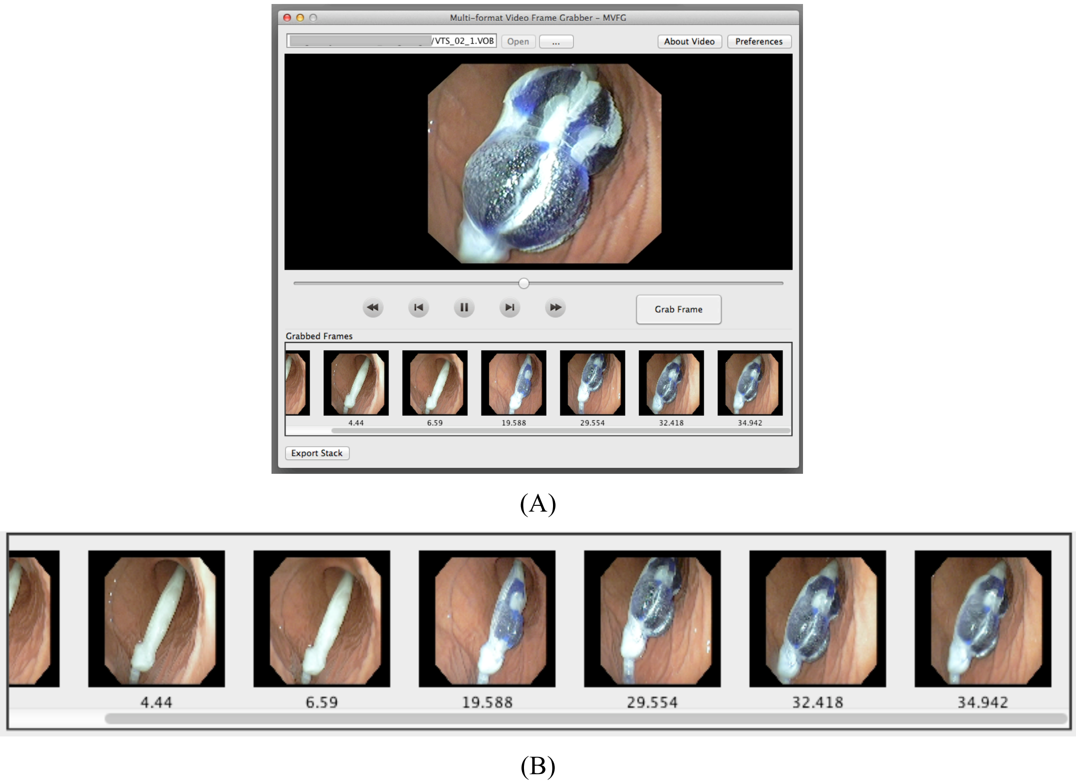
Resolution: 768x576 pixels

Time: 89 s

(A) (B)

Figure 2 – MVFG Plugin. (A) Plugin Interface. (B) Additional information about the loaded video.

When playing the video, button "*Grab Frame*" allows for frame capture in real time, appearing in the lower box small thumbnails of each frame grabbed (Figure 3 (A)). This capture is sorted by time when frame appears in the video instead the moment that the user captures that frame (Figure 3 (B)).

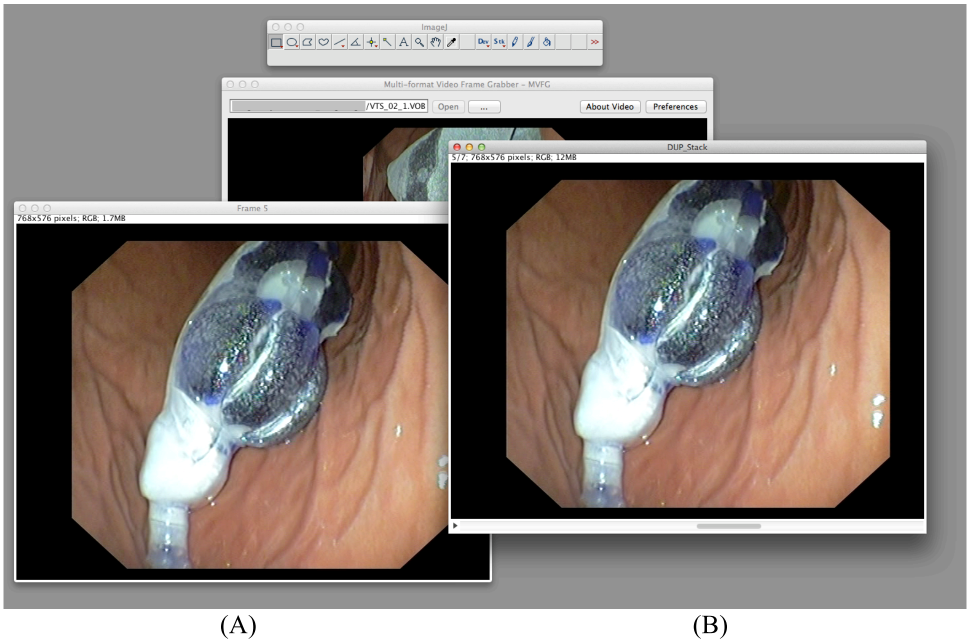


(A)

(B)

Figure 3 - MVFG Plugin. (A)Frame capture, (B) Zoom of lower part where the grabbed frames are sorted

The export of individual frames to ImageJ is accomplished by clicking on the image reference of the desired frame, (Figure 4(A)). Using the button "*Export Stack*" is also possible to export a "*stack*" with all grabbed frames (Figure 4(B)).


ImageJ
Multi-format Video Frame Grabber – MVFG
/VTS_02_1.VOB
Open
...
About Video
Preferences
DUP_Stack
5/7; 768x576 pixels; RGB; 12MB
Frame 5
768x576 pixels; RGB; 1.7MB

(A)

(B)

Figure 4 – MVFG Plugin. (A) Individual Frame. (B) ImageStack.

4. SUMMARY AND FUTURE WORK

We have been using MVFG to grab images from endoscopy videos for further processing and research. It has served its purposes allowing to use ImageJ to work on images grabbed from different video formats, depending on the endoscope manufacture, that till now had to be converted. MVFG has proven to be easy to use even with very long HD videos.

The developed plugin allows to perform a number of different tasks such as viewing, browsing and reading the various video formats, frame capture, export the frames as individual frame or as a ImageStack.

Some of the features implemented seek to address the lack of ImageJ plugins to work/manipulate videos. Without this plugin, tasks such as selecting different frames and capturing the same video of 2 hours would not be possible with such versatility and efficiency.

We intend to develop in the near future same new functionalities such as, automatic detection of regions of interest in endoscopy videos and intelligent compression for endoscopy videos.

REFERENCES

[1] T. Ferreira, and W. Rasband, "*The ImageJ User Guide - Image Types and Image Formats*", pp 8 – 11 (2011).

[2] W. Rasband, "*QuickTime Movie Writer*," ImageJ Plugin, 2000, <http://rsb.info.nih.gov/ij/plugins/movie-writer.html> (29 May 2012).

[3] W. Rasband, "*JMF Movie Reader,*" ImageJ Plugin, 2001, <http://rsb.info.nih.gov/ij/plugins/jmf.html> (29 May 2012).

[4] W. Rasband, R. Dougherty, J. Hardin, and J. Woodward, "*QuickTime Opener,*" ImageJ Plugin, 2000, <http://rsb.info.nih.gov/ij/plugins/movie-opener.html> (29 May 2012).

[5] M. Lee, "*vlcj - Java bindings for the vlc media player*," 2012, <http://www.capricasoftware.co.uk/vlcj/> (8 May 2012).

[6] Free Time, "*Format Factory Home Page - Free media file format converter*," <http://www.formatoz.com/> (3 August 2012).

[7] AnvSoft Inc., "*Any Video Converter Freeware*," <http://www.any-video-converter.com/products/for_video_free/> (3 August 2012).

[8] Anvsoft Inc., "*Any Video Converter Lite,*" <http://itunes.apple.com/ca/app/any-video-converter-lite/id479472944?mt=12> (3 August 2012).

[9] M. Schmid, and D. Marsh, "*AVI Reader,*" ImageJ Plugin, 2002, <http://rsb.info.nih.gov/ij/plugins/avi-reader.html> (29 May 2012).