gVision

Open Source Video & Image Acquisition for Scientific Applications

Streaming Video Acquisition
AVI, TIFF, FMF File Streams w/ Compression
Split Live Video Streams into Multiple Sub-ROI File Streams
Supports: 1394 DCAM, DirectShow, Camera Link, GigE Vision Cameras, and Several Others
Custom Rig Integration via gVision Plugins
Custom Low Level C/C++ Camera Interface Adaptor Authoring

User's Manual & API Documentation v1.1

© 2010 Gus K Lott III, PhD

Neurobiological Instrumentation Engineer
Howard Hughes Medical Institute
19700 Helix Dr., Ashburn, VA 20147
lottg@janelia.hhmi.org

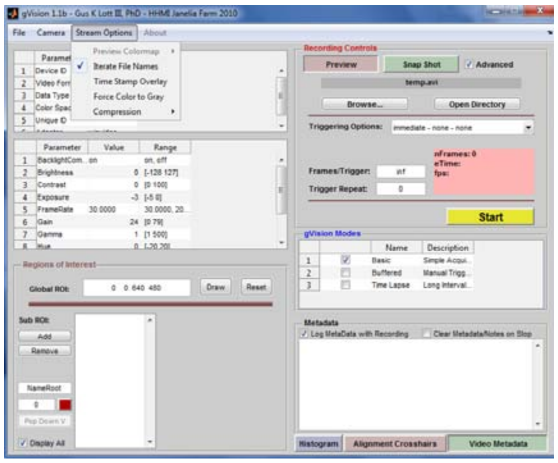
HHMI janelia farm research campus

September 15, 2010

Introduction

gVision is a suite of tools for real-time streaming from streaming from video cameras in scientific applications. gVision was primarily developed for behavioral monitoring in systems neuroscience as a byproduct of the Fly Olympiad project at HHMI’s Janelia Farm Research Campus between 2007 and 2010.

gVision is written in Matlab®, and supports a variety of common camera systems and can be configured to capture from essentially any data stream. gVision derives much of its basic hardware interface support from the Matlab Image Acquisition Toolbox®. As of September 2010, the toolbox supports **DCAM 1394 Cameras, Camera Link, GigE Vision, DirectShow, and several others.**



	Parameter	Value	Range
1	BacklightCom...	on	on, off
2	Brightness	0	[-128 127]
3	Contrast	0	[0 100]
4	Exposure	-3	[-5 0]
5	FrameRate	30.0000	30.0000, 20...
6	Gain	24	[0 79]
7	Gamma	1	[1 500]
8	Hue	0	[-20 20]

temp.avi

Triggering Options: immediate - none - none

Frames/Trigger: inf

Trigger Repeat: 0

nFrames: 0
eTime:
fps:

Global ROI: 0 0 640 480

		Name	Description
1	✓	Basic	Simple Acqui...
2		Buffered	Manual Trigg...
3		Time Lapse	Long Interval...

gVision’s Advanced Camera Control Interface

gVision provides an interface allowing the user to configure their camera properties (i.e. shutter, gain, brightness, and global region of interest) while providing image preview and histogram feedback on image quality. gVision will stream to video files of several types including Color/Grayscale AVI v2.0 (uncompressed or Motion JPEG), 8/16-bit Color/grayscale Tiff stacks via LibTiff (with a variety of compression options), and grayscale to v1.0 of the Dickinson Lab “Fly Movie Format” (FMF) files.

gVision may also split an incoming video stream into several sub-regions in software. This mode allows for easy logging of individual chambers of a tiled/parallel assay without post-processing to split the video stream. Each software selected video stream may have a user defined name, and associated text Metadata.

As an open source Matlab application, gVision is designed to be expanded to fit your custom Image Acquisition application. From custom C/C++ adaptor development to Matlab Based “gVision Plugin” development, the tool provides a powerful framework for Machine Vision applications in science and education.

Features:

- Image Streaming of Full Frame or Multiple Sub-Region Slices to AVI, TIFF, FMF Files
- Several Real-Time Compression Options, Flatten Color to Gray, Time Stamp on Video
- Support DCAM 1394, GigE Vision, Camera Link, DirectShow (i.e. Webcams), and Other Cameras
- Camera Property control, Histogram, Line Scan for rig alignment, Text Based Metadata
- Hardware Trigger Control for Supported Hardware
- Pre-Buffered Event Capture & Time Lapse Logging Modes
- Customize your experiment via a custom gVision Plug-in developed in Matlab
- Develop high performance backend code in C/C++ via the Matlab IMAQ Toolbox Adaptor Kit

Contents

Installation & Supported Hardware

Install from Matlab Source:

gVision may be run directly from an active Matlab license (version 2010b and above) containing the Image Acquisition & Image Processing toolboxes. Other toolboxes may be required for custom adaptor development.

To install gVision from source, copy the gVision directory structure to your current Matlab directory and run “gVision.m”. The install directory should contain a “+plugins” and “doc” directory.

The current code snapshot can be acquired from the sourceforge subversion server:

https://gvision-hhmi.svn.sourceforge.net/svnroot/gvision-hhmi/gVision

Installing the Compiled Version:

A stand-alone Windows ”compiled” version of gVision is also provided for users who do not have access to a Matlab license. This version is full featured and free, but cannot be customized beyond what was included at compile time. All basic modes (Basic, Buffered, & Time Lapse) are supported in this installation.

32-bit Windows Compiled installation files may be downloaded from:
http://sourceforge.net/projects/gvision-hhmi/files/

To install the compiled version of the code:

1. Download MCRInstaller_2010b.exe and gVision.exe
2. Run MCRInstaller_2010b.exe to install Matlab Runtime Components
3. Run gVision.exe to launch the program

The first time running gVision may take a while to launch. This is due to Matlab’s initial configuration of a new program. Subsequent launches should be considerably faster.

Compiled Version Conflicting with an Existing Matlab Install

If you have a separate Matlab install on the PC and find a conflict (i.e. missing DLL errors), you will need to explicitly set the program path to the Runtime Components.

Create a blank text file named “gVision.bat” in the directory with gVision.exe. This file should contain two lines of text:

```
PATH=C:\Program Files\MATLAB\MATLAB Compiler Runtime\v714\bin\win32
gvision.exe
```

Run gVision.bat to prevent path conflicts between an existing Matlab install and the Compiled code.

Multi-Camera Support from Multiple gVision Instances

In its current form, gVision only supports a single camera stream. One way to use gVision to support multiple camera streams is to launch the program several times. Each instance can be connected to a different camera installed on your system. You can either hardware synchronize the cameras with an external trigger, or manually click the start buttons in rapid succession. A software socket based synchronization of multiple gVision instances may appear in a future version of the tool.

Connecting to Hardware & Camera Formats

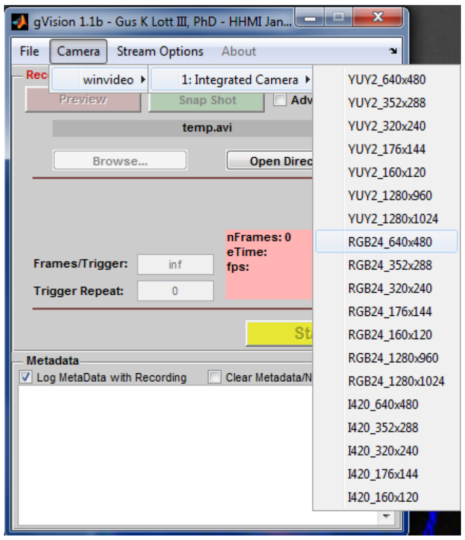
Selecting a Camera Format & Connecting to a Camera

When gVision launches, it scans for installed hardware. If you plug in a new camera to the system while gVision is open, you must restart the program to detect the change.

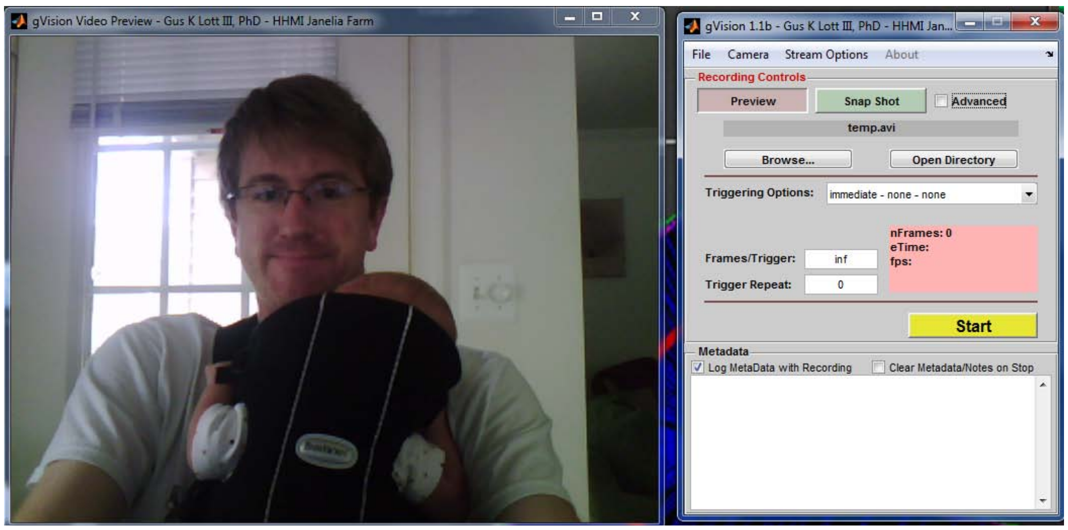
In order to connect to a camera, you will need to select an adaptor, a specific camera, and a camera format (figure to right). Available formats for a given camera are available under the "Camera" menu in the main gVision window.

Once a camera is selected, a preview window will appear displaying the camera stream (below). The user interface will activate to allow for configuration and control of the video stream.

gVision Advanced Mode provides access to properties and mode control for the program and may be accessed via the check box in the upper right of the main gVision control window.



Selecting a Camera Mode & Connecting



Camera Formats & DCAM Format 7

DCAM Format 7

Many cameras present fixed frame rate options to gVision. These modes typically present a fixed pixel size (i.e. Y8_640x480). One format worth mentioning is the Format 7 (F7) mode available on most DCAM cameras. This mode is typically reserved for streaming without a fixed frame rate. In DCAM F7 mode, the frame interval is limited by either the exposure time or the firewire data rate. In this mode, you can reduce the frame interval by reducing the hardware global region of interest size. By reducing the number of pixels per frame, the frame rate will scale based on data rate limitations.

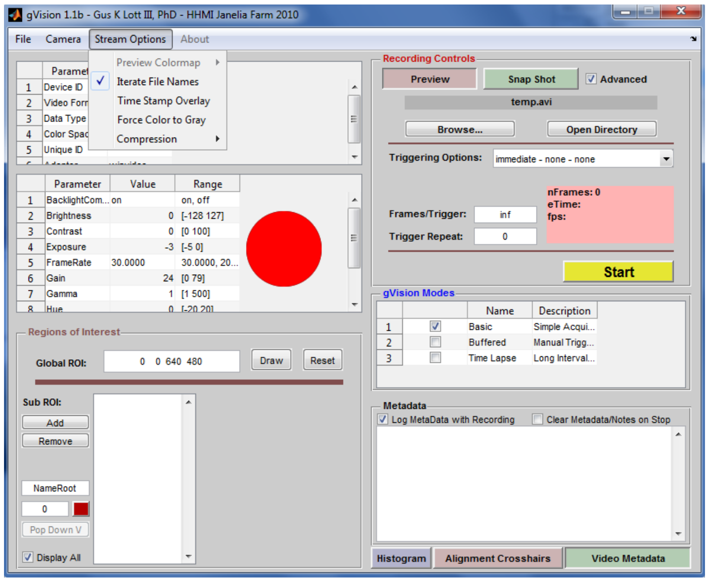
Many commercially available cameras will use a similar mode to achieve very high frame rates. For example, at 100x100 pixels, the Basler A602fm Monochrome CMOS Camera can achieve approximately 635Hz frame rate shutter permitting.

In these modes, there is no "Frame rate" property exposed in gVision advanced mode. Instead, there is a frame rate box that the user must edit to set the frame rate for playback within the video file logged from the variable speed camera stream.

Setting & Reading Camera Properties

In the advanced mode in gVision, the user may configure any number of exposed parameters in the hardware. The properties, with their allowed ranges, may be edited in the property list indicated in the figure at the right. Examples of these properties include

- Frame Rate
- Brightness
- Contrast
- Shutter
- AutoExposure



	Parameter	Value	Range
1	BacklightCom...	on	on, off
2	Brightness	0	[-128 127]
3	Contrast	0	[0 100]
4	Exposure	-3	[-5 0]
5	FrameRate	30.0000	30.0000, 20...
6	Gain	24	[0 79]
7	Gamma	1	[1 500]
8	Hue	0	[-20 20]

Global ROI: 0 0 640 480

[✓ Advanced]

temp.avi

Triggering Options: immediate - none - none

Frames/Trigger: inf

Trigger Repeat: 0

nFrames: 0
eTime:
fps:

		Name	Description
1	✓	Basic	Simple Acqui...
2	☐	Buffered	Manual Trigg...
3	☐	Time Lapse	Long Interval...

[✓ Log MetaData with Recording] ☐ Clear Metadata/Notes on Stop

Recording & File Types & File Names

Selecting a File Name & Type

In the recording control panel, the "Browse…" button will provide a standard file selection user interface. Here, the user may select a file name and type.

- Grayscale streams: FMF, AVI, & TIFF
- Color streams: AVI & TIFF

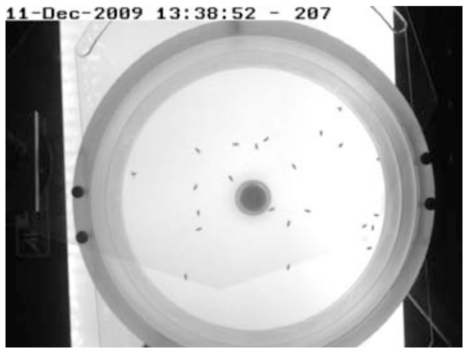
- 16-bit streams: TIFF

When the file selection user interface is launched, the name is automatically populated with a string containing the current date and time. If the user selects a file name that already exists, and the "Iterate file names" option is selected from the "stream options" menu, the file name will be modified with an iterating integer in order to prevent the existing file from being overwritten. Every time that recording is initiated, the file name will be incremented.

Other Stream Options

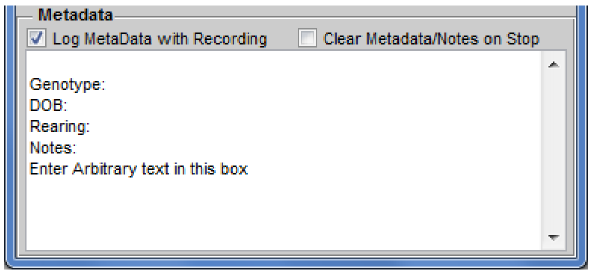
From the "Stream Options" menu, the user may select a variety of options to apply to the video file during recording:

- *Force Color Stream to Gray*
  - Will flatten a color image using a simple average of the red, green, and blue image planes when writing to disk
- *Time Stamp Overlay on Video*
  - Draw the date and time and frame index as text on the video stream (see right)
- *Compression*
  - By default, no compression is applied to the video streams created by gVision. The compression option provides MJPEG support for AVI files as well as several options for TIFF compression if these file formats are selected.



A Time Stamp Overlay

Logging Text Metadata with Video

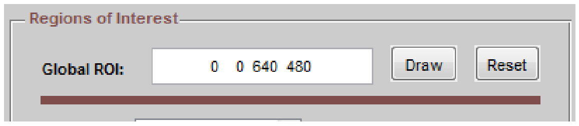


Genotype:
DOB:
Rearing:
Notes:
Enter Arbitrary text in this box

gVision provides a raw text metadata field in the bottom right of the main control window. When this field has contents, and the "Log Metadata with Recording" button is checked, gVision will log the text contents of the metadata field to a text (*.txt) file with the same root name as your video file.

If the "clear metadata/notes on stop" button is checked, the text field will automatically be cleared when a recording session stops.

Regions of Interest (ROI)



Global ROI: 0 0 640 480

Hardware Global Regions of Interest

A region of interest defines a subset of pixels that will stream from the camera to the selected file. In most cases, the actual camera streams only the subset of data to the PC. As mentioned above in the section on DCAM F7 mode, this can change the effective frame rate of the camera.

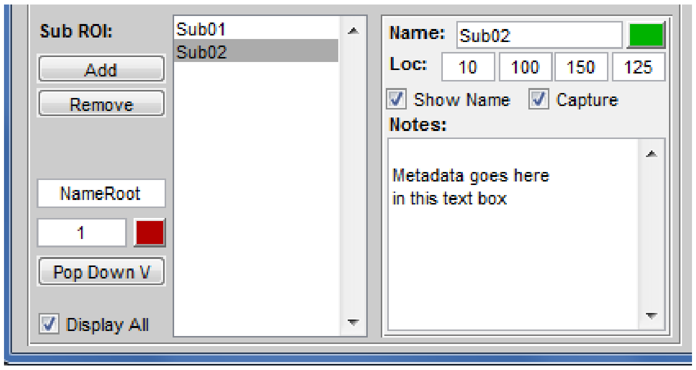
To define a global region of interest in gVision, the user may click the "Draw" button next to the global ROI input box in the bottom left of the advanced menu. This will activate a red bounding box in the preview window that may be resized and dragged to graphically specify the subset of pixels to be included in the global ROI. Hitting the "Draw" button a second time will commit the specified global ROI.

The Global ROI definition contains four numbers. In order:

1. Pixels in from the left
2. Pixels in from the top
3. Width of image
4. Height of image

When defining a global region of interest, the sum of the first and third numbers may not exceed the maximum width of the image frame and the second and fourth numbers may not exceed the maximum height.

Software Defined Sub-Regions of Interest



Sub01
Sub02

NameRoot
1

Name: Sub02
Loc: 10 100 150 125
✓ Show Name ✓ Capture
✓ Display All

Notes:
Metadata goes here
in this text box

Software Sub-ROIs are an important feature in gVision. They allow the user to create set of sub-streams of videos from a global frame. This may be important for a behavioral assay involving many tiled sub-assays in a high throughput screen. Another application might involve small files captured from specific anatomical regions in a tethered animal when the entire frame of video contains relatively less significant information.

Sub-ROIs may be created in the Sub-ROI control panel in the bottom left of the gVision Advanced Mode window (image right).

- Each sub-ROI appears in the preview window and may be dragged and scaled to a target location in the global frame.
- Each Sub-ROI must have a unique name identifying it
- sub-ROIs may overlap if desired.
- Each sub-ROI has a position defined as described above for the global ROI. These coordinates are with reference to the current global region of interest setting.

- Each sub-ROI may have a custom color (click the colored button in the upper right of the sub-ROI detail panel).
- The capture checkbox will determine if the ROI’s contents will be streamed to file when the record start button is pressed.
- Each sub-ROI may have custom associated metadata defined in the Notes text field
- When logging, sub-ROI filenames have “BASENAME_SUBROINAME.avi” format
- Metadata files have the same root file name, but a .txt extension

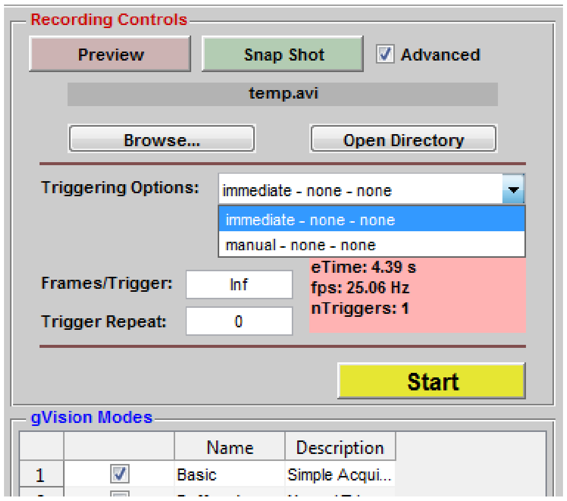
A feature is included to auto populate Sub-ROI names with a new root and index. The “NameRoot” edit box in the sub-ROI panel accepts an arbitrary name root (i.e. genotype) to rapidly rename sub-ROIs between experiments. The “Pop Down” button will implement the new name root and index and step to the next sub-ROI for rapid name population.

Triggering

Immediate, Manual, & Hardware Trigger Modes

In the Basic gVision capture mode, the “Record Controls” panel contains a “Triggering Options” drop down menu. For all cameras, this menu will present immediate and manual trigger modes. These are software trigger modes that allow the user to either immediately begin streaming a number of frames to disk or to manually trigger a capture via a software signal.

This drop down menu will also be populated with supported hardware trigger modes that your camera may offer. When recording is started in these modes, frames will not be acquired until an external hardware signal initiates acquisition.



☑ Advanced

temp.avi

Triggering Options: immediate - none - none

Frames/Trigger: Inf

Trigger Repeat: 0

eTime: 4.39 s
fps: 25.06 Hz
nTriggers: 1

		Name	Description
1	✓	Basic	Simple Acqui...

Parameters: Frames/Trigger & Trigger Repeat

This important pair of parameters in the “Recording Controls” panel provide automated control of frame acquisition during any acquisition session. Frames per trigger define how many frames will be acquired after a hardware or software trigger is presented and trigger repeat defines how many times the system will acquire triggered bunches of frames before stopping the recording session. The string “Inf” (stands for infinite) may be entered into these boxes to indicate that you wish to record continuously.

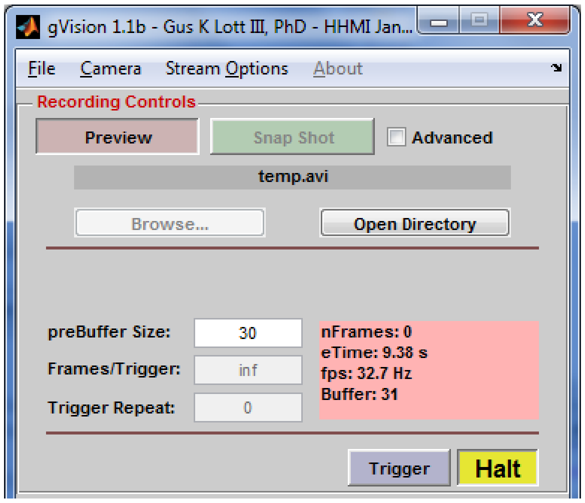
One Frame/Trigger in Hardware Triggering Modes

A common image acquisition task may involve the need to trigger every frame of a video stream with a hardware signal. In order to accomplish this, the frames per trigger parameter should be set to 1 and the trigger repeat should be set to one less than the total number of frames you wish to acquire.

Alternate Built-In Modes

Pre-Buffered Manual Triggered Acquisition

In this mode, gVision allows the user to deliver a software trigger that begins capture of a specified number of frames in history followed by an arbitrary number of frames in the future. This mode is useful for the capture of events that happen rapidly to which the user might not be able to rapidly respond to in a standard no-history capture mode.



temp.avi

preBuffer Size: 30
Frames/Trigger: inf
Trigger Repeat: 0

nFrames: 0
eTime: 9.38 s
fps: 32.7 Hz
Buffer: 31

All sub-ROI and file name conventions in this mode are identical to the basic mode. The user must specify a number of frames to store (this is the depth of the history buffer). When the user hits the trigger button in this mode, gVision streams the historical frames to file and continues streaming until the trigger button is depressed. The "Halt" button cancels the buffering mode and closes the stream file(s).

Time Lapse Mode

This mode captures frames with arbitrary intervals (typically longer than 1 second). The user specifies the capture interval and the number of frames to capture per trigger (typically 1). The "Frame Rate" field appears in this mode to define the frame rate at which the captured frames will be played back in the associated AVI file.

API for Development of gVision Mode Plug-ins

Included in the source release of gVision is a +plugins package directory in which definitions for the gVision Basic, Pre-buffered, and Time Lapse modes exist. Along with these mode definition classes is provided an abstract class "gvPlugin" and a stand-alone tool pulled from the Mathworks File Exchange "Text2Im.m" which creates the timestamp overlay option for video streams in gVision.

A user may wish to create a custom gVision Mode that allows them to configure and control several other components of their assay. A gVision mode can leverage the camera configuration and file logging features of gVision while allowing arbitrary extensibility for control of other components of an experimental apparatus.

To create a gVision mode, a user simply adds a matlab class file to the +plugins package directory. The class file will be automatically parsed upon launching the application. This class file should inherit from gvPlugin (a handle class) and must implement several standard methods:

- A static method "gvPlugName" that returns the file name, the name of the mode, and a text based description of the mode
- A constructor that adds to or reconfigures the UI in whatever fashion you need. This constructor can also launch connections to other hardware, etc.
- A callback for the "start" button in the record panel.

The gvBasic.m class file contains an example implementation of a gVision mode and should be used as a guide for development of custom modes.

A figure handle is passed to the constructor of your gVision module. This figure handle may be referenced at any location in your code. The userdata of this figure contains an up to date structure of handles to default elements of the gVision GUI as defined in the maingui function of gvision.m. It is necessary for the user store this figure handle as a property of mode class named 'fig'.

Inherited Methods from gvPlugin

gvPlugin offers several useful inherited functions for use in your application.

- logdata = openVidFiles(plug)
    - This file extracts the file name from the gVision GUI and stores state information about the file stream in the logdata struct.
    - The logdata struct must be stored in the userdata of the active video input object
- writeFullROI(plug,frame,time,metadata)
    - Accepts the output of the getdata command to write a frame to the file stream created in "openVidFiles."
- writeSubROI(plug,frame,time,metadata)
    - Accepts the output of the getdata command to write subframes (as defined in the gVision GUI) to the separate file streams as defined by the user
- wrapupVid(plug)
    - Flushes the remaining frames from the video object to the file stream and closes video files.
- deactivateGvUI(plug), activateGvUI(plug)
    - Disables and enables components of the gVision GUI during logging to prevent the user from changing parameters during logging.
- File = checkFileName(File)
    - Method that determines the existence of a file with the same name as that selected in the gVision UI and appends/iterates an integer to prevent overwriting. Called by "openVidFiles" if the "iterate file names" option is selected.

API for Development of Custom C/C++ Backends & gDCAM

Matlab offers a C++ Interface to the Image Acquisition Toolbox via the IMAQ Adaptor Kit. This kit is thoroughly documented and provided by the mathworks:
http://www.mathworks.com/help/pdf_doc/imaq/adaptorkit.pdf

The documentation is geared for developers using Microsoft Visual Studio, and describes how to create an arbitrary multi-threaded data/frame pump into the image acquisition toolbox. An adaptor created using this toolkit should automatically plug into gVision once registered with your Matlab instance.

gDCAM is an open source adaptor designed to interface the CMU DCAM driver library for 1394 cameras. This is the same driver that the built-in IMAQ “dcam” adaptor interfaces. The gDCAM code may act as a starting point for a high performance logging or real-time vision application based in gVision, or otherwise in Matlab.