

Camera IO

Version 2.1.1

The DIVA3D Camera IO module allows the user to process video taken from a USB camera in two ways:

- either by recording the movie, saving it to the hard disk and then processing it offline, just like any other video file.
- by associating the capture window of the camera with a video stream buffer and processing the grabbed video frames grabbed “on the fly”.

In order to install the DIVA3D Camera IO module, the user has to copy the camera.dll file to DIVA 3D working directory. By doing so, the entry ‘Camera IO’ appears under the DIVA3D *Modules* menu.

The *streamio.dll* module must exist in DIVA3D working directory, so that the Camera IO module works properly.

Before using the Camera module for the first time, the user has to set the capturing device. This is done from Modules->Camera IO->Select Driver.... The following message box appears and the user can select the driver he wishes for capturing.

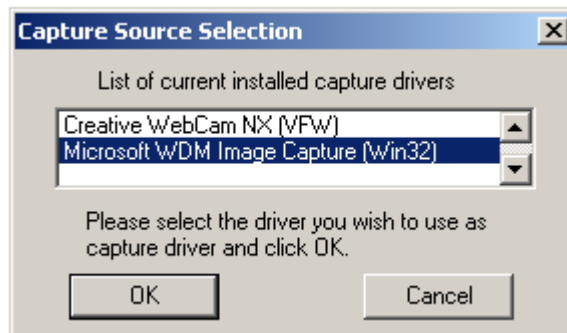


Figure 1: Capture source driver selection.

Tip: Microsoft WDM Image Capture driver is the appropriate one to be used for USB cameras, unless the camera comes with its own drivers.

After selecting the appropriate driver, the camera capturing previewing process can start from Modules -> Camera IO ->Start.... The following window appears:

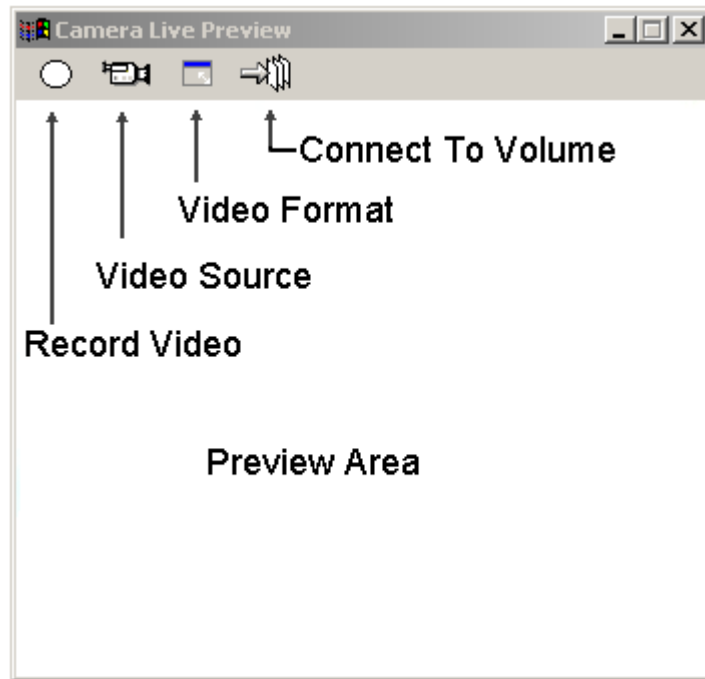



Figure 2: Live camera preview window.

This window has the following buttons:

-  Record: It gives the user the ability to record a video and store it to the hard disk as an *.avi file. By pressing this button, the following dialog box appears:

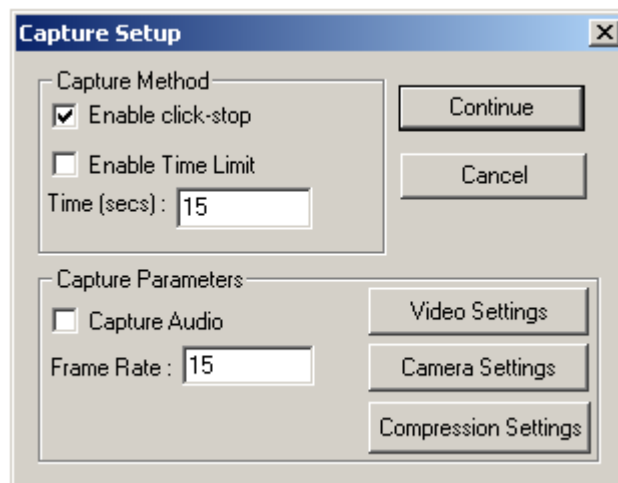



Figure 3: Capture setup dialog box.

This window is composed by two sections:

- *Capture Method:* There are two ways to set the recording duration: Time limit and Click Stop. By choosing *Enable click-stop*, the recording will start immediately after pressing the *Continue* button and will stop when the user presses the right mouse button inside the preview window area. By choosing *Enable Time Limit*, the recording will start immediately after pressing the *Continue* button and will stop after the time that was defined by the user in the *Time (secs)* edit box has elapsed.

○ *Capture Parameters*: In this section, the user can define if he wants also to capture audio in his/her output *.avi file, the desired *Frame Rate* and *Compression Settings* for the output avi file. For the buttons *Video Settings* and *Camera Settings* see Video Source and Video Format.

●  *Video Source*: It displays a dialog box to set some of the camera acquisition parameters: *Contrast*, *Brightness*, *Frame rate*, *Saturation* and much more. This dialog box is not standard and each USB camera comes with its own one. For example, this dialog box for the Philips PVC-740K camera is the following one:

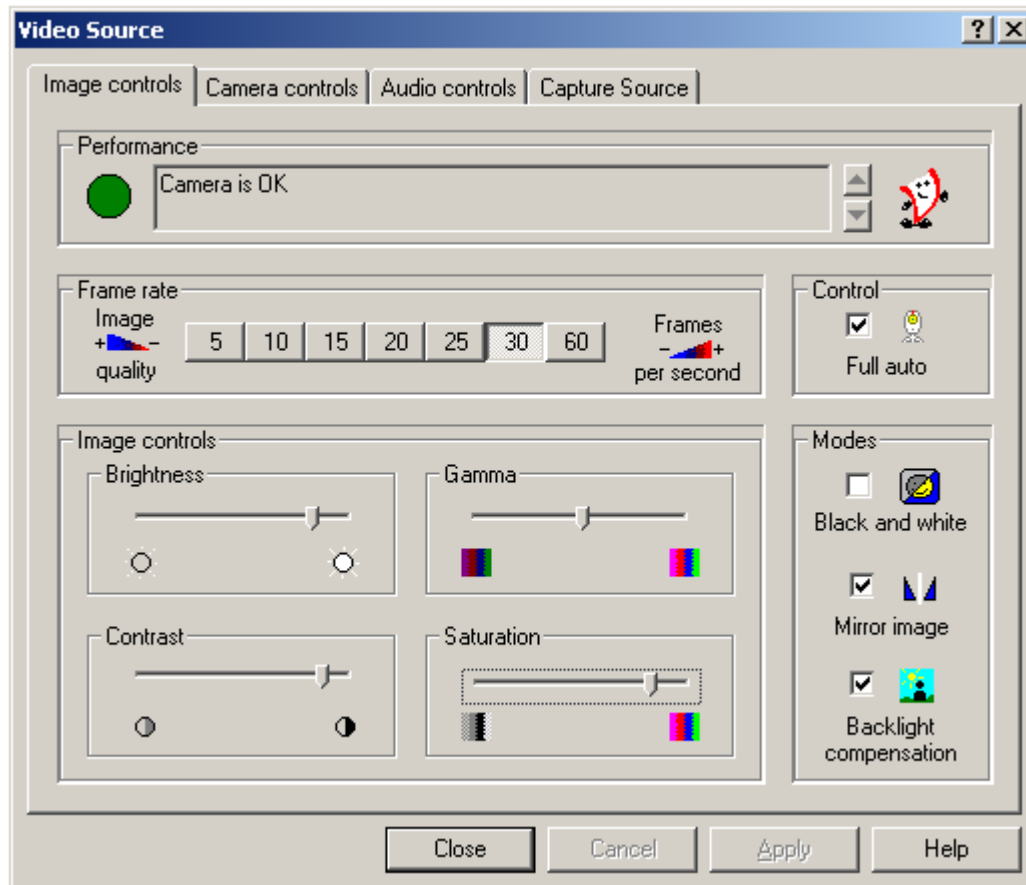



Figure 4: Example of Video Source dialog box.

●  *Video Format*: It displays the following dialog box, where the user can select the spatial *Resolution* (in pixels), *Pixel Depth* (in bits) of the video stream and the compression used during video acquisition and storage.

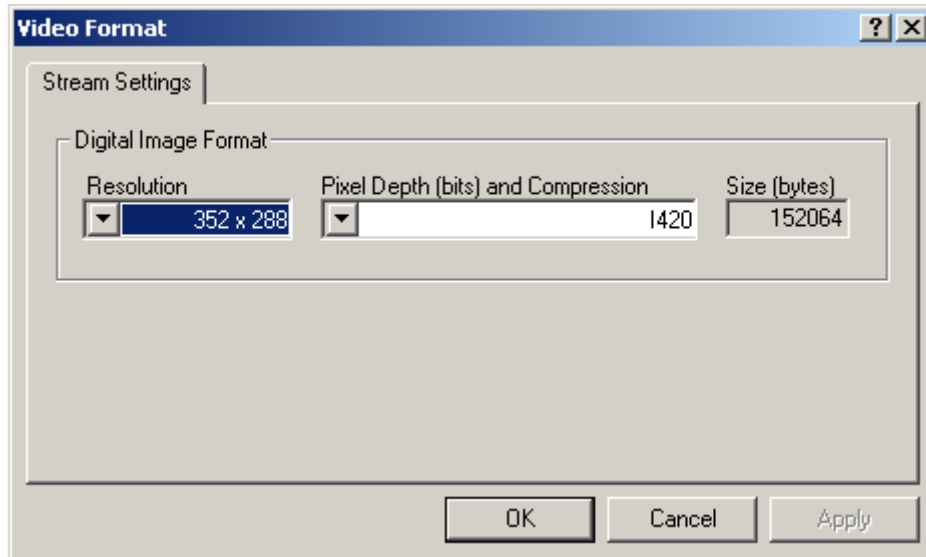



Figure 5: Video format dialog box.

- 
Connect To Volume: This button creates a video stream buffer named “USB Real Time Camera Input” and associates the camera with that video buffer. If the association is completed with no errors, a message box informs the user that the association is completed successfully. From this point onwards, the user can select the newly created video stream buffer as an input to any other DIVA3D video processing and analysis module.

Tip 1: Once the capture window is associated with the video buffer, the Video Format button is disabled. In order to change the dimensions of the capture window one has to destroy the video buffer associated with the camera, change the dimensions of the capture window and reconnect the camera to a new video buffer.

Tip 2: When the user closes the Live Preview window, the association (if one exists) between the camera and the video buffer is automatically destroyed. This is done because, by closing the Live Preview window, the capture window (which is child to the Live Preview window) is destroyed and, hence, the association between camera and video buffer ceases to exist.