

EyeTalkTrak User Guide

September 30, 2009

EyeTalk Systems, Inc.

113 Executive Drive, Suite 125, Sterling, VA 20166

Tele: 703.437.6002 Fax: 703.689.0297

Email: info@eyetalksystems.com

Web: www.eyetalksystems.com

PixelPusher and EyeTalk are registered trademarks of EyeTalk Systems, Inc.
VISCA is a registered trademark of Sony Corporation.
Windows, Windows XP, Windows Vista, and Windows 2000 are registered trademarks of Microsoft Corporation.

Copyright (c) 2007-2009 EyeTalk Systems, Inc.

Table Of Contents

Overview	1
Features	1
System Requirements.....	3
Quick Installation	4
Video Sources.....	7
Physical Connections.....	7
Configure A Video Source	8
Set Video Capture & Snapshot Properties	9
Video and Image Processing Display	11
Where Do I Go From Here?	14
Feature Walkthrough	15
Application Development.....	19
Overview of Alternatives	19
Working with the Sample Application Solutions	20
Application Deployment	21
Menu Descriptions.....	22
Acquire Menu.....	22
Camera Menu	25
Video Display Menu	27
Image Processing Menu	29
VISCA® Menu	31
View Menu (MFC Only).....	33
Help Menu.....	34
Technical Support	36
Known Problems	36
Troubleshooting	38
Technical Support	42
FAQs	43
Glossary	46
Index	47

Overview

Features



The EyeTalkTrak software and PixelPusher® video capture hardware have been designed to provide an economical platform for developing and deploying applications where image processing is an essential component of the application. This platform is intended for use by programmers proficient in any of the following Microsoft languages: .NET languages including C# and Visual Basic, as well as Microsoft Foundation Classes (MFC).



PixelPusher® is a standalone frame grabber that connects to a computer through a USB 2.0 port. It converts RS-170 (monochrome) video input to true VGA at 30 frames per second (fps) through a USB cable connection. It converts color composite video input to true VGA at 22.5 fps. There is no need to open your computer case to install an adapter card. PixelPusher® displays on the computer monitor the source video stream as well as the image processing results applied to the source video (if any). PixelPusher® also records both of these video streams to disk. It has the following features:

- USB 2.0 bus interface
- Standard analog scan: NTSC/RS-170 and PAL color and monochrome video
- Accepts input from up to 6 CVBS, or 4 CVBS and 1 S-Video input sources
- Video jacks: 1 RCA jack (CVBS), 1 S-Video jack, and one DB9 jack to connect to an optional video breakout cable
- 640 x 480 and 704 x 480 pixel resolutions in NTSC/RS-170
- 704x 576 pixel resolution in PAL
- 320 x 240 and 352 x240 pixel resolution in quarter screen NTSC/RS-170
- 352 x 288 pixel resolution in quarter screen PAL
- Sony VISCA® camera control interface
- Brightness and contrast control from the host PC
- Windows® Hardware Quality Lab (WHQL) signed drivers for Windows® 2000 and XP, as well as Vista® drivers
- Case dimensions: 6-7/8 x 5-1/2 x 1-7/16 H" (17.4 x 13.9 x 3.63 H cm)
- Case material: polystyrene, UL94-HB flammability rating
- Power supply: +5V DC input
- Complies with FCC and CE test requirements for an information technology, Class B device.

The EyeTalkTrak software development kit (SDK) consists of the following elements:

- PixPushR.dll, a dynamic link library (DLL) implemented as a regular DLL to access the video capture functionality of the PixelPusher® hardware. Client applications that can link to general Microsoft DLLs should be able to link to PixPushR.dll
- PixPushMFC.dll, a dynamic link library (DLL) implemented as a regular DLL to access the video capture functionality of the PixelPusher® hardware. MFC client applications that can link to general Microsoft DLLs should be able to link to PixPushMFC.dll
- EyeTalkTrak.exe, an MFC compiled/linked executable that can be used right out of the box to view analog video sources on the screen and write out .avi video files. No programming is required to use this application
- EyeTalkTrakNET.exe, a C# .NET compiled/linked executable that can be used right out of the box to view analog video sources on the screen and write out .avi video files. No programming is required to use this application
- EyeTalkTrakVB.exe, a Visual Basic .NET compiled/linked executable that can be used right out of the box to view analog video sources on the screen and write out .avi video files. No programming is required to use this application.
- Each of these projects in the SDK provides a functional client program that can be used as is, or extended to provide additional features that the programmer requires. Collectively, the projects give the programmer a head start in building an application from the language of his/her choice. Each project provides the core functionality needed to grab a source video frame, perform some image processing operations on that frame, display on the monitor both source and processed video frames, and write source and processed video streams to disk file.
- The SDK does not provide a comprehensive image processing library. However, programming support is provided that allows a developer to access the freely available open source [OpenCV](#) image understanding library for application development. This support makes it easy to call one or more of the 500+ OpenCV functions so that the programmer has a rich image processing development environment at his disposal.
- Documentation includes:
 - *EyeTalkTrak User Guide (this document)*
 - *EyeTalkTrak Reference Manual and Application Programming Interface (API) for the PixelPusher® Framegrabber*

System Requirements



- Windows XP® with Service Pack 2 or later, Windows 2000 with Service Pack 4, or Windows Vista®
- CDROM drive for software installation only (this may be located on the host PC or on a network drive)
- Internet Explorer, Version 4.0 or later for on line HTML Help
- Intel Pentium III or later processor, 1.8 MHz or more (recommended)
- USB 2.0 port
- Video display adapter with at least 16 bit color
- To use the EyeTalkTrak SDK to build your own applications, Microsoft® Visual Studio 2008 or Visual Studio 2005 with Service Pack 1 or later. IMPORTANT: you must have at least Visual Studio 2005, SP 1 installed in order to link to the framegrabber DLL.

Quick Installation



In the product box, you should find the following items:

- These Quick Installation instructions
- The PixelPusher® hardware
- A +5V DC power supply, with an IEC 320 power cord appropriate for your locality
- USB 2.0 cable
- (Optional) DB9 video breakout cable (if you ordered this)
- (Optional) MiniDIN9 camera control cable (if you ordered this)
- CDROM with driver, EyeTalkTrak software, and documentation.

The hardware and software should be installed in the specific sequence of steps listed. To avoid driver installation problems, **the hardware should always be installed first**. Go to the section for the operating system on which you wish to install PixelPusher®.

Windows XP® and Windows 2000®

Install the hardware first by following these steps:

- Connect the power supply input into the power jack located on the back of the hardware. Connect the power supply cable into a wall outlet and the other end into the power supply itself. **IMPORTANT:** Some USB devices are powered through the USB cable. The PixelPusher® hardware, however, requires an attached +5V power supply. It will not work using only the USB bus power supply
- Connect the white USB cable (Type B) to the back of the hardware, and run the other end to the host PC (Type A) jack. **NOTE:** the host PC must support USB 2.0
- Supply a video input signal to a suitable video port (see [Video Sources](#)) by connecting an appropriate cable to the PixelPusher® hardware
- Insert the installation CDROM into the CDROM player
- Press the power switch on the front of the hardware
- Normally, this is all that is required to install the driver. In some circumstances, the host PC may raise a "Files Needed" dialog. If so, browse to the Driver folder, e.g., d:\Driver on the installation CDROM, select the pixpush.sys file, click **Open**, and then click **OK** on the "Files Needed" dialog
- If the host PC does not recognize the new hardware, please see the applicable [Troubleshooting Instructions](#).

Install the software second by performing these steps:

- **IMPORTANT:** You must have Administrator privileges to install this software
- Browse the CDROM drive and double click the file **EyeTalkTrakSetup.exe**
- **IMPORTANT:** it may take as much as 45-60 seconds for the installation wizard dialog to appear as Windows copies files to your hard drive, so please be patient
- The install program will first load the Visual C++ 2005 SP distributables to your hard disk. When prompted, allow these to be installed by clicking **Yes**. Follow the instructions in the installation wizard, including accepting the license agreement and selecting an install directory
- When the installation has finished, click **Finish**
- To save live video to .avi files, you will need a suitable CODEC installed on your host that supports the FOURCC "FFDS" encoder. If you do not have an acceptable CODEC, you can run **ffdshow** from the installation CD to install Microsoft DirectShow:
 1. Installing **ffdshow** and accepting all the defaults should provide a configuration that supports the sample applications in the SDK distribution
 2. In the event that you already have **ffdshow** installed (by default located in \Program Files\ffdshow\ffdshow.ax), but you cannot output .avi files from a sample SDK application, perform the following steps. From the Windows Start menu, select **ffdshow** and then select the "Video decoder configuration" menu item. This raises the "ffds video

decoder configuration" dialog. Under the "Format" column, find the "FVFW,FFDS" row, and make sure that libavcodec is selected in the "Decoder" column by clicking on the drop down menu in the corresponding cell in the "Decoder" column. This selection supports the FOURCC "MJPG" encoder which is the default encoding in the SDK's sample applications. Choose the "FFDS" FOURCC code from the list of CODECS.

Windows Vista®

Install the hardware first by following these steps:

- Connect the power supply input into the power jack located on the back of the hardware. Connect the power supply cable into a wall outlet and the other end into the power supply itself. **IMPORTANT:** Some USB devices are powered through the USB cable. The PixelPusher® hardware, however, requires an attached +5V power supply. It will not work using only the USB bus power supply
- Connect the white USB cable (Type B) to the back of the hardware, and run the other end to the host PC (Type A) jack. **NOTE:** the host PC must support USB 2.0
- Supply a video input signal to a suitable video port (see [Video Sources](#)) by connecting an appropriate cable to the PixelPusher® hardware
- Insert the installation CDROM into the CDROM player
- Press the power switch on the front of the hardware
- The host PC will start the Found New Hardware wizard. When prompted, click **Locate and install driver software (recommended)**
- Click the **Continue** button when the User Account Control dialog comes up. If you have set up Vista to always check online for hardware updates, Windows will take 15 to 30 seconds or more while it attempts to locate the driver software online in Windows Update. If your PC is connected to the Internet, it will find the driver software in Windows Update, but will not be able to install it because the PixelPusher driver has only been WHQL tested for Windows XP and Windows 2000, but not Windows Vista. Click **Close**
- To install the driver, go to the Device Manager: right click **Computer**, select **Properties**, select the **Device Manager** from the Tasks list, and click **Continue** when the User Account Control dialog appears. There should be an "Unknown" or "PixelPusher" device with an exclamation point through it shown in the **Other devices** node
- Right click on this device and select **Update Driver Software...** Click on **Browse my computer for driver software**. Browse to the CDROM Drive and check the "Include subfolders" checkbox. Click **Next**
- When Windows is finished installing the driver, you should see a dialog with the message "Windows has successfully updated your driver software." Click **Close**
- If the host PC does not recognize the new hardware, please see the applicable [Troubleshooting instructions](#).

Install the software second by performing these steps:

- **IMPORTANT:** You must have Administrator privileges to install this software
- Browse the CDROM drive and double click the file **EyeTalkTrakSetup.exe**
- **IMPORTANT:** it may take as much as 45-60 seconds for the installation wizard dialog to appear as Windows copies files to your hard drive, so please be patient
- Click the **Continue** button when the User Account Control dialog comes up
- The install program will first load the Visual C++ 2005 SP1 distributables to your hard disk. When prompted, allow these to be installed by clicking **Yes**. Follow the instructions in the installation wizard, including accepting the license agreement and selecting an install directory
- When the installation has finished, click **Finish**
- To save live video to .avi files, you will need a suitable CODEC installed on your host that supports the FOURCC "FFDS" encoder. If you do not have an acceptable CODEC, you can run **ffdshow** from the installation CD to install Microsoft DirectShow:
 1. Installing **ffdshow** and accepting all the defaults should provide a configuration that supports the sample applications in the SDK distribution
 2. In the event that you already have **ffdshow** installed (by default located in \Program Files\ffdshow\ffdshow.ax), but you cannot output .avi files from a sample SDK application, perform the following steps. From the Windows Start menu, select **ffdshow** and then select the "Video decoder configuration" menu item. This raises the "ffds video decoder configuration" dialog. Under the "Format" column, find the "FVFW,FFDS" row,

and make sure that libavcodec is selected in the "Decoder" column by clicking on the drop down menu in the corresponding cell in the "Decoder" column. This selection supports the FOURCC "MJPG" encoder which is the default encoding in the SDK's sample applications. Choose the "FFDS" FOURCC code from the list of CODECS.

To test the basic operation of the system:

- Double click the EyeTalkTrak icon that was placed on your desktop during the software installation step, or select from the Programs menu the EyeTalkTrak program (click Start|Programs|EyeTalkTrak|EyeTalkTrak)
- Select the appropriate camera port (i.e., one you attached a cable to) from the **Camera** menu
- If video is not already being displayed, select Show Video from the **Acquire** menu.

If the system fails to show video, follow the steps in [Troubleshooting](#).

For a quick introduction to the software, see the [Feature Walkthrough](#). This will familiarize you with the key features of the software.

Video Sources

Physical Connections



PixelPusher® accepts video from any device that outputs an NTSC or PAL video signal. This includes CCD and CMOS cameras, VCRs, CD and DVD players, camcorders, and other analog video devices.

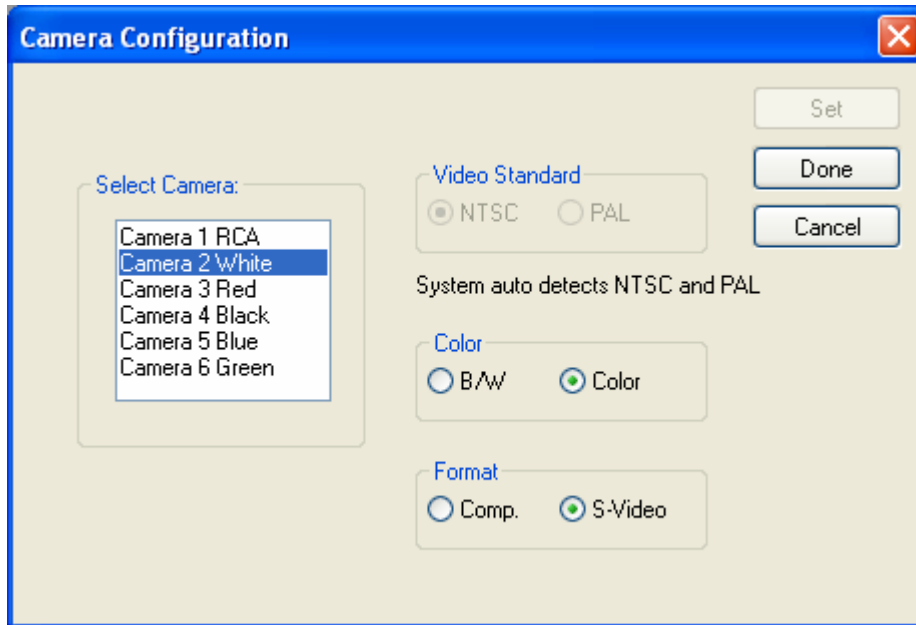
Video sources can be connected to PixelPusher® through three different types of jacks located on the back of the device case:

- The RCA jack allows input of a single composite NTSC video source. This is selected by choosing "Camera 1 RCA" input from the [Camera Menu](#)
- The S-Video jack allows input of a single S-Video source. This is selected by choosing "Camera S-Vid or White 2" input from the [Camera Menu](#). Note that when an S-Video source is connected to and operating at the S-Video port, the white and black camera leads (for Cameras 2 and 4) in the optional video breakout cable should not be connected to active video sources
- The DB9 connector allows input of up to 5 composite video sources, identified as Camera 2 through Camera 6. In the Camera Menu, the camera numbers are associated with the color of the individual camera cable lead in the video breakout cable.

Configure A Video Source



The Camera Configuration dialog can be accessed from the Camera Menu (and the Camera Select toolbar for the MFC client application build). In the dialog shown below, you select a camera from the Select Camera list on the left, and on the right, you check the various radio buttons that characterize the type of video source for the selected camera.



As soon as you click on one of the radio buttons to change a setting for the selected camera, the Set button will be enabled. **IMPORTANT:** when you have configured the selected camera, you **MUST** click on the Set button to save your changes. When you are done setting all the cameras you wish to configure (don't forget to click on the Set button each time), click on the Done button.

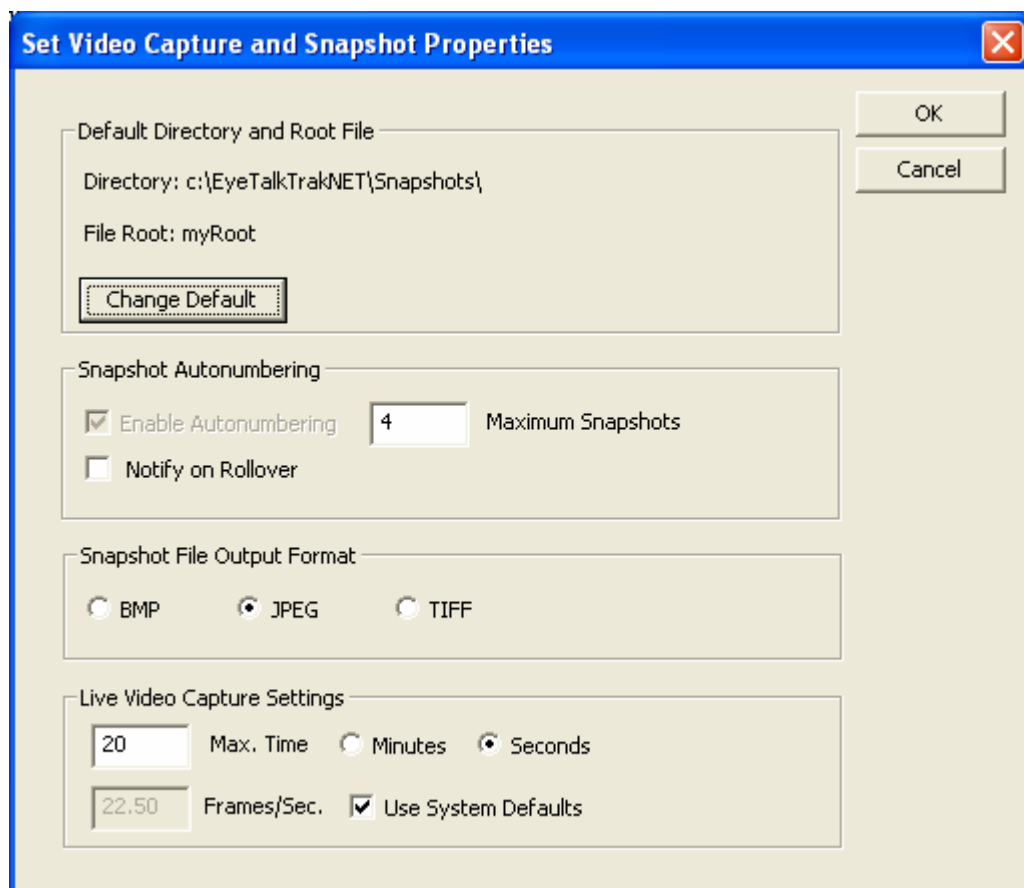
In this example,

- There is no need to specify NTSC or PAL as PixelPusher automatically detects the video standard, so these buttons are grayed out
- The user is configuring Camera 2 White (the color of the line on the video breakout cable), as an S-Video color source. So input will be taken from the S-Video port rather than the DB9 connector.
- You can only assign an S-Video source to Camera 2. If Camera 2 is selected, the S-Video button is enabled; for all other camera numbers, the S-Video button is disabled. **IMPORTANT:** If you supply video input to the S-Video port, then no active camera should be attached to the White or Black leads of the video breakout cable (since the S-Video port shares these leads).
- If you have made changes to any camera, when you click on the Done button, you will be prompted to effect the changes you have made now. If you answer No, the changes will not go into effect until the program is restarted. If you answer Yes, the program will restart automatically only if the changes you made apply to the currently selected camera (as indicated on the Camera menu). Changes that you made to another camera will not be effected until you switch to that camera during the current session, or until you restart the program.

Set Video Capture & Snapshot Properties

The Set Video Capture & Snapshot Properties menu item is listed in the [Acquire Menu](#).

The settings specified in the Snapshot Properties dialog determine how snapshots are saved and how live video is captured in a .avi file.



- Select a directory (folder in which you wish to save snapshots. In Vista in particular, make sure you specify a directory in which you have write privileges; otherwise, your snapshots will not be written
- Enable Aut numbering of snapshots. This is particularly useful if you want to take multiple snapshots without pausing to name each one. NOTE: This is always enabled; you cannot disable aut numbering in this release. This means that after taking a number of snapshots, up to the number specified in the Maximum Snapshots edit box, you will need to navigate to the directory in which snapshots are saved and rename the individual files if you wish to keep them; otherwise, they will be written over after you have exceeded the Maximum Snapshots number
- Specify the Maximum Snapshots before the naming scheme rolls over. This is to prevent you from using excessive file space to save your snapshots
- If you wish to be notified when the Maximum Snapshots has been reached, click on the Notify on Rollover box
- In Windows XP®, the default viewer image viewer is Windows® Picture and Fax Viewer. When you select the menu item View Snapshot, Picture and Fax Viewer will be started and you can page through all the images stored in the directory (folder) you specified

- In Windows 2000®, the default viewer image viewer is Microsoft Paint. When you select the menu item View Snapshot, Microsoft Paint will be started with the most recent snapshot displayed
- In Windows Vista®, the default viewer image viewer is Windows® Photo Gallery. When you select the menu item View Snapshot, Photo Gallery will be started. When you specify a new directory (folder) and click on View Snapshot for the first time, Photo Gallery will come up displaying a default directory. You must click on the File button, and select Add Folder to Gallery. Then navigate to the new directory you specified, select it, and click OK. On subsequent invocations of View Snapshot, Photo Gallery will be started with the new directory you have specified
- You can specify a different image viewer by modifying the registry key SnapshotViewer to reflect the full path to the display program (requires Administrative privileges)
- You can specify the format for snapshots as BMP, JPEG or TIFF. The JPEG is compressed while BMP and TIFF are not compressed.
- You can specify the frame rate at which video is recorded. The value that you enter is only a target rate, as the actual rate will depend on a number of factors, including the resolution of the video frame, the computational requirements of the image processing algorithms you are using (if any), whether color or grayscale images are being captured, and how many video sources you are recording, i.e., whether you are recording source only, image processing only, or both. A good rule of thumb to determine the best recording rate is to start your application with the source and image processing video running (assuming you plan to record image processing results), and observe the frame rate that the client application, e.g., EyeTalkTrakNET reports. Then, specify a capture rate equal to or less than the reported rate. If you select the Use System Defaults checkbox, the software will enter a recording rate that is generally appropriate for the resolution and color mode and how many sources will be recorded, but the default rate does not take into account how intensive the image processing operations are.
- The maximum capture time is intended to be used to prevent large video files from being created if the application is left unattended. The idea is that when the elapsed time exceeds the value entered, any video captured will automatically be stopped and the file closed. You can specify time in units of seconds or minutes. **IMPORTANT:** The value entered maximum capture time only updates the registry key CaptureMaxDuration. If you want to use this value to time-out your video recordings, you must implement a timer in your client application. Example code to implement a suitable timer is included in each of the sample application solutions, i.e., EyeTalkTrak, EyeTalkTrakNET, and EyeTalkTrakVB. Note that the sample applications provided will NOT start a timer if you have entered a value of 0 for the maximum capture time; this is an effective way to disable the timer.

Video and Image Processing Display



The SDK does not provide a comprehensive image processing library. However, programming support is provided in the SDK through the **CImgPro** class that allows a developer to access the freely available open source [OpenCV](#) image understanding library for application development. This support makes it easy to call one or more of the 500+ OpenCV functions so that the programmer has a rich image processing development environment at his disposal. Alternatively, the programmer can insert his own image processing functions into the image processing pipeline that is implemented in the CImgPro class; these might be functions developed under a previous project, or functions that he/she wants to create from scratch.

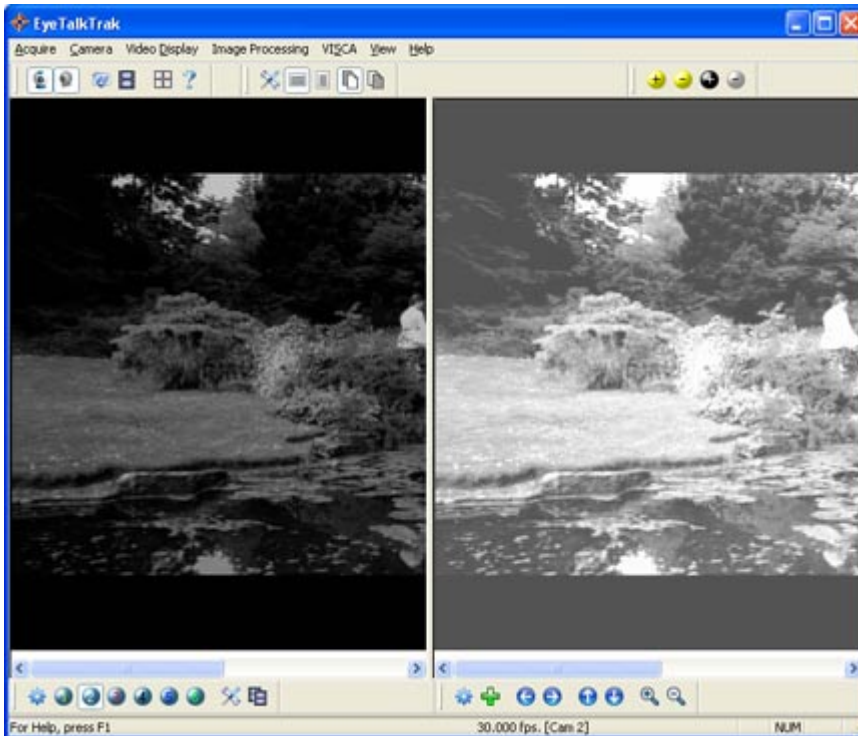
The model for incorporating your own image processing functions into the EyeTalkTrak framework is based on the following principal definitions and assumptions:

- An image **array** refers to the data comprising a single video frame
- An image processing **operation** is defined as some modification to an image array. That is, an operation maps the array I_n to the array I_{n+1}
- An image processing **function** implements in code one or more image processing operations. An image processing engineer writes an image processing function
- An image processing **stage** refers to any of the $(n + 1)$ image arrays beginning with the original source video array I_0 and continuing to the final array I_n resulting from application of all the image processing operations
- An image processing **pipeline** consists of the sum total of all image processing operations, i.e., the pipeline represents the sum total of all the stages.

In the MFC sample application, the main window contains a splitter window, which has two panes that can be resized by the user. One of these panes contains the active video input view and the other pane contains the results of any image processing operations that are applied to the active video input.

In either of the .NET sample applications (i.e., EyeTalkTrakNET.exe or EyeTalkTrakVB.exe), the active video and the image processing views are organized under tabs. The tabs and image frames are sized to fit the current resolution, and cannot be adjusted by the user as in the MFC sample application.

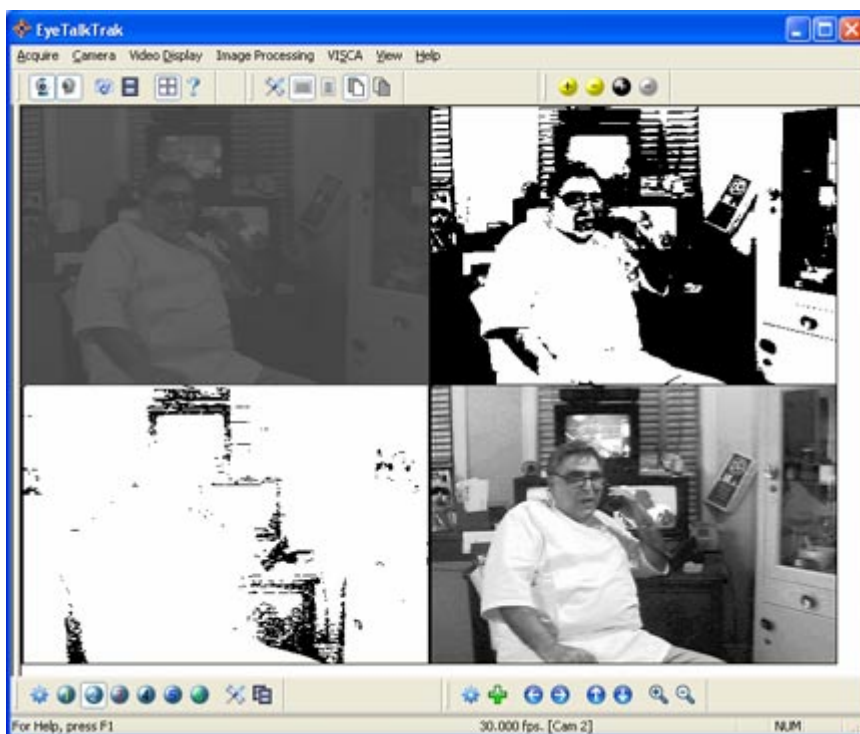
In the following screen shot taken from the MFC sample application, the window on the left contains the original image, while the window on the right contains the result of applying a histogram equalization function to the original image. By selecting and moving the splitter window bar, the user can eliminate entirely the display of either view. For example, if the user only wants to see the source video, he/she can drag the splitter window bar to the right, leaving only the source video view exposed.



The above screen shot is an example of the display operating in full frame (or full screen) mode. The user can select the particular stage in the image processing pipeline that he wishes to display (see [Image Processing Menu](#)).

In quarter screen mode, it is possible to display up to four (4) panes containing the source video and image processing pipeline stages. This is done by assigning the results of an image processing stage to a particular quadrant (see Configure Quarter Screen).

In the screen shot below, the original video is shown in quadrant 1 (top left); it has low contrast and is underexposed. Quadrant 2 (top right) shows the image after a simple thresholding operation has been performed; note that while the contrast has been improved, much of the detail has been lost because there are no mid-tones. Quadrant 3 (lower right) shows the image after applying a histogram equalization function to the original image. The image in quadrant 4 (lower left) shows application of a thresholding operation to the image shown in Quadrant 3.



Thus, it is relatively easy to see how the result of an individual image processing step contributes to converting the image to the desired form, whatever that may be for the application at hand.

If the source video is dynamic, particularly in terms of illumination, it may be helpful to record the image processing video stream to see how a sequence of one or more image processing operations affects the desired result. PixelPusher® and the EyeTalkTrak sample applications support writing both the source video and the image processing results to an .avi file.

Where Do I Go From Here?



1. You can use any of the three EyeTalkTrak sample applications and the PixelPusher® video capture hardware right out of the box. No programming is required. If you haven't installed the hardware and software yet, just follow the [Quick Installation](#) instructions
2. Once you have installed the hardware and software, you should go through the 5 minute [Feature Walkthrough](#) to gain an understanding of the basic features of EyeTalkTrak
3. You can extend and modify any of the three sample application projects (EyeTalkTrak.sln, EyeTalkTrakNET.sln, or EyeTalkTrakVB.sln) to meet your specific requirements (see [Application Development](#)). This makes maximum use of the existing EyeTalkTrak user interface functionality
4. You can review the API (by referring to the *EyeTalkTrak Reference Manual and Application Programming Interface (API) for the PixelPusher® FrameGrabber*), embed calls to the API in your own client program, and link your program to the PixPushR.dll (see [Application Development](#)). This approach makes no use of the user interface source code in any of the EyeTalkTrak sample applications. However, you may wish to study the code in one of the sample applications which contain working examples of how to use the APIs.

Feature Walkthrough



Once you have completed the [basic installation test](#), you should follow the steps listed below to gain a better understanding of the features offered by EyeTalkTrak.

To follow these steps, set up your hardware as follows:

1. Connect a DVD player to PixelPusher®, attaching both an RCA cable (for Camera 1) and an S-Video cable (for Camera 2). Insert a DVD and hit Play, OR
2. Connect two other video sources to any of the camera ports. In the walkthrough, you will be instructed to configure a camera source for color. Even if you do not have a color video source, you should configure it as color in order to see how the program handles color changes; the video will just appear as black & white
3. Start PixelPusher.

We assume that you have not made any changes to the default settings for the program made at install time. If you have, there may be some minor differences between what you will see and what we indicate you will see during this walkthrough.

In the instructions that follow, we use the term "EyeTalkTrak" to refer to any of the three sample application builds, i.e., EyeTalkTrak.exe, EyeTalkTrakNET.exe, or EyeTalkTrakVB.exe. If you start the MFC build (EyeTalkTrak.exe), the application will contain a splitter window. If you start one of the .NET builds (EyeTalkTrakNET.exe or EyeTalkTrakVB.exe), the video and image processing of that video will appear in separate tabs.

Feature: Start EyeTalkTrak

1. From the PC Start menu, select **Programs, EyeTalkTrak** and **EyeTalkTrak** again. This starts the EyeTalkTrak program.

Feature: Switch Between Black & White and Color Video

1. From the **Camera** menu, select **Camera 1 RCA**
2. From the **Camera** menu, select **Configure...** Select **Camera 2** and click the **Color** and **S-Video** radio buttons. Click **Set** and then **Done**
3. From the **Acquire** menu, select **Show Video**. Black & white video from the DVD player should appear under the Source Image tab (or the left pane of the splitter window for the MFC build)
4. From the **Acquire** menu, select **Show Image Processing**. The Image Processing tab (or the right pane of the splitter window for the MFC build) shows a "smoothed" or blurred image of the source video in the left pane. The smoothed image is produced by calling the image processing function "CvOpenSmooth," which appears in the source code files: ImgPro.css, ImgPro.vb, and ImgPro.cpp
5. From the **Camera** menu, select **Camera 2** (it is labeled "S-Video or White 2," so named because it handles S-Video or input from the white lead of the video breakout cable, but not at the same time). A dialog appears informing you that a color mode change is required (EyeTalkTrak must restart to accommodate a color mode change). Click **OK**. After a few seconds, color video from Camera 2 should appear under the Source Video tab (or in the left pane of the splitter window for a MFC build).

Feature: Switch Between Full Screen and Quarter Screen for a Color Camera Source

1. From the **Video Display** menu, select **Quarter Screen**. The image is now reduced size
2. From the **Video Display** menu, select **Quarter Screen again**. The image is now restored to full size.

Feature: Switch Between Full Screen and Quarter Screen for a Black & White Camera Source

1. From the **Camera** menu, select **Camera 1 RCA**. A dialog appears informing you that a color mode change is required (this time it must change from color back to black & white). Click **OK**. After a few seconds, black & white video from Camera 1 should appear under the Source Video tab (or in the left pane of the splitter window for an MFC build)
2. From the **Acquire** menu, select **Show Image Processing**. The image processing results reappear as before under the Image Processing tab (or in the right pane of the splitter window for an MFC build)
3. From the **Video Display** menu, select **Quarter Screen**. This time, a dialog appears informing you that the program needs to restart to accommodate a change in screen size, i.e., from full to quarter screen. While a restart is not required for a color video source, it is required for a black & white source (click [here](#) to find out why).

Feature: View Image Processing Results at Quarter Screen Resolution

1. (MFC Only) If you are running EyeTalkTrak.exe, drag the vertical window splitter handle all the way to the left. You should now see four (4) quarter screen panes. If you do not, you are not running in quarter screen mode, or Show Image Processing has not been enabled from the Acquire Menu
2. (.NET Only) If you are running EyeTalkTrakNET.exe or EyeTalkTrakVB.exe, you need only select the Image Processing tab. You should now see four (4) quarter screen panes. If you do not, you are not running in quarter screen mode, or Show Image Processing has not been enabled from the Acquire Menu
3. Going clockwise from the top left, Pane 1 is the original source image; Pane 2 reflects a smoothed (blurred) version of the source (by applying the function CvOpenSmooth); Pane 3 is the result of applying a Canny edge detector to the source video (through the function CvOpenCanny); and Pane 4 applies an adaptive thresholding technique to the source video (through the function CvOpenAdaptiveThresh).

Feature: Adjust the Brightness and Contrast of the Source Video Image

1. (MFC Only) Find the Image Control toolbar, which should be at the top of the window; it has two yellow buttons with + and - signs. To increase the image **brightness** of the source video, click the yellow + button 2 or 3 times and observe the results of the images in Panes 3 and 4. Which of the images shown in Panes 3 and 4 is more robust to changes in brightness, i.e., you can still tell what objects are in the image? Press the yellow - button 2 or 3 times to return to normal brightness
2. (.NET Only) To increase the image **brightness** of the source video, select **Brighter** from the **Camera** menu 2 or 3 times and observe the results of the images in Panes 3 and 4. Which of the images shown in Panes 3 and 4 is more robust to changes in brightness, i.e., you can still tell what objects are in the image? Select **Darker** from the **Camera** menu 2 or 3 times to return to normal brightness
3. (MFC Only) To increase the image **contrast** of the source video, click the black + button 2 or 3 times and observe the results of the images in Panes 3 and 4. Which of the images shown in Panes 3 and 4 is more robust to changes in contrast? Press the gray - button 2 or 3 times to return to normal contrast
4. (.NET Only) To increase the image **contrast** of the source video, select **More Contrast** 2 or 3 times from the **Camera** menu and observe the results of the images in Panes 3 and 4. Which of the images shown in Panes 3 and 4 is more robust to changes in contrast? Select **Less Contrast** from the **Camera** menu to return to normal contrast
5. Note that if you want to return to the original brightness and contrast settings, but have forgotten how many times you pressed the brightness and contrast buttons or selected the corresponding action from the **Camera** menu, you need to close the program, power down the hardware and restart.

Feature: Switch From 640x480 to 704x480 Base Resolution

1. From the **Video Display** menu, select **704x480**. This will display full screen video at 704x480 pixels and the quarter screen format at 352x240. A dialog appears informing you that a restart is required to accommodate the resolution change. Click **OK**. After a few seconds, the same video should appear, but now each quarter screen panel measures 352x240, instead of 320x240. **IMPORTANT:** there is a problem with quarter screen display of 704*480 base resolution in this

release. The panes may appear with a vertical bar inside each pane; the video is wrapping around. This problem is discussed [here](#). Usually this can be corrected by toggling video display off and on a few times (**Acquire | Show Video**), or by exiting the program and restarting.

Feature: (MFC Only) More Details on Color Mode, Screen Resolution and Screen Format (Quarter or Full Screen) when Using the Splitter Window

1. From the **Camera** menu, select **Camera 2**. As before, a dialog appears informing you that a color mode change is required. Click **OK**
2. But remember, we pulled the splitter window handle all the way to the left, thus covering up the source video. Pull the splitter window handle to the right. The handle is parallel to the left edge of the window and the cursor will change its icon when you move over the handle. Now do you see the color video image? It should still be at quarter screen resolution
3. From the **Video Display** menu, select **Quarter Screen**, which will cause the video format to change to full screen. Pull the splitter window handle more to the right to reveal the full screen image. Leave a little white space on the right. You may have to resize your whole window larger to see the full extent of the image
4. From the **Video Display** menu, select **640x480**. This will return the full screen display to 640x480 pixels. As before, a dialog appears informing you that a restart is required to accommodate the resolution change. Click **OK**. After the program has restarted, notice that the white space has grown larger as you would expect since we reduced the horizontal dimension of the resolution (from 704 to 640 pixels).

Feature: Take and View a Snapshot

1. Make sure that active video is playing in the Source Image tab (or leftmost window in MFC). If not, select **Show Video** from the **Acquire** Menu
2. From the **Acquire** menu, select **Take Source Snapshot** or press **F3**. This takes a snapshot of the source video stream from the currently selected camera and saves it to a directory of your choice. Since you have not yet selected a directory, the snapshot is saved to the default directory. You should hear a soft beep each time you take a snapshot. If you receive a program error at this point, it is probably because you have not installed **ffdshow** as described in the [Quick Installation instructions](#)
3. Hit **F3** two more times to take additional snapshots
4. Make sure that image processing results are playing in the Image Processing tab (or rightmost window in MFC). If not, select **Show Image Processing** from the **Acquire** Menu
5. From the **Acquire** menu, select **Take Image Processing Snapshot** or press **F4**. This takes a snapshot of the image processing results video stream and saves it to a directory of your choice. Since you have not yet selected a directory, the snapshot is saved to the default directory. You should hear a soft beep each time you take a snapshot
6. Hit **F4** two more times to take additional snapshots
7. From the **Acquire** menu, select **View Snapshot**. This raises an image viewer which varies depending on the operating system. To see which viewer is selected for your operating system, click [here](#).
8. Close the image viewer when you are done viewing your snapshots.

Feature: Record Source and Image Processing Video Streams

1. Make sure that active video is playing in the Source Image tab (or leftmost window in MFC). If not, select **Show Video** from the **Acquire** Menu
2. Press **F7**. Note on the status bar the message: "Recording Source Video". You have started recording of the source video stream
3. Let the video run for a few seconds and press **F7** again. Note that the "Recording Source Video" message disappears. You have stopped recording of the source video stream
4. Make sure that image processing results are playing in the Image Processing tab (or rightmost window in MFC). If not, select **Show Image Processing** from the **Acquire** Menu
5. Press **F8**. Note on the status bar the message: "Recording IP Video". You have started recording of the image processing video stream
6. Let the video run for a few seconds and press **F8** again. Note that the "Recording IP Video" message disappears. You have stopped recording of the image processing video stream
7. From the **Acquire** Menu, select **Capture Source Video -> Close Source Recording**
8. From the **Acquire** Menu, select **Capture IP Video -> Close IP Recording**

9. Now browse to the Snapshots directory, which by default is installed directly under the application directory. For example, if you are running the MFC version, the Snapshots directory would be located in \EyeTalkTrak\Snapshots. If you are running the .NET C# version, the Snapshots directory would be located in \EyeTalkTrakNET\Snapshots. If you are running the .NET VB version, the Snapshots directory would be located in \EyeTalkTrakVB\Snapshots
10. Start each .avi file and observe the recorded results.

Feature: Test Individual Application Programming Interfaces (APIs)

The EyeTalkTrak software and the PixelPusher® hardware working together provide an environment to develop image processing applications. To facilitate this development, there is an extensive set of application programming interfaces (APIs) available to the developer. To familiarize the developer, and to provide snippets of code that use a particular API, there is a feature available to test these APIs from the EyeTalkTrak user interface. Once the developer sees what the API does, he/she can go directly to the code in the applicable source file (i.e., EyeTalkTrakNET.cs, EyeTalkTrakVB.vb or EyeTalkTrakDoc.cpp) and copy the code for use in his/her own application.

1. From the **Camera** menu, select **Camera 2**. The image should be black & white.
2. From the **Help** menu, select **Exercise API**
3. In the top scrolling list box, select the **Change Cam Color Mode** API. Notice that there is a "Description" field for the API and a "Requirements" (prerequisites) field. There is also a field called "API Identifier," which provides a convenient way to find the code snippet in the applicable source code file
4. Click the **Test** button. If you have properly configured the video sources, i.e., Camera 2 is configured for color and you are currently viewing a camera (other than Camera 2) configured for black & white, you should see the program restart and display color video from Camera 2
5. From the **Help** menu, again select **Exercise API**.
6. In the top scrolling list box, select the API **Get Cam Info**.
7. Click the **Test** button. This raises a message box with camera number and frame rate
8. You can test other APIs in the same way
9. When you are done testing APIs, click the **Close** button.

Feature: Stop EyeTalkTrak

1. From the Acquire menu, select **Exit**, or simply close the window by clicking the window Close button

Application Development

Overview of Alternatives



The .NET version of the EyeTalkTrak software is designed around the Windows® Form. Each version (EyeTalkTrakNET and EyeTalkTrakVB) uses a main form, supplemented with an additional form to implement the API test feature. The source video and the image processing of that video are displayed in the main form under separate tabs.

The MFC version of the EyeTalkTrak software is designed as a single document interface (SDI) application. The software consists of a set of MFC classes, notably CEyeTalkTrak, CEyeTalkTrakDoc, CEyeTalkTrakView, and CMainFrm (derived from CWinApp, CDocument, CScrollView and CFrameWnd respectively) that implement and extend the basic SDI functionality.

In addition to these classes, each sample application includes a **CImgPro** class that controls the application of image processing steps to the source video; it is intended to be extended by the application developer to include all image processing operations (or stages) that the application developer wishes to apply to the source video.

Applications can be developed by writing image processing functions that perform the desired transformations on the source video input. In the SDK, the class CImgPro has been defined to encapsulate these functions. The software developer merely adds functions to this class and sequences their invocation in the function CImgPro.GetIPResults() (CImgPro::GetIPResults() in MFC).

To assist in image processing algorithm development, the original source video and intermediate image processing stages, as well as the final result can be displayed in four (4) quarter screen quadrants. This functionality can be easily extended by the programmer to meet his/her development needs.

If the application has user interaction requirements that are not satisfied by the built-in functionality of the EyeTalkTrak software, the programmer can add other user interface elements to meet the particular system requirements. This is accomplished by adding additional classes, or methods to existing EyeTalkTrak classes. Go to [Working with the Sample Application Solutions](#) if you wish to extend the applications that have already been developed.

The *EyeTalkTrak Reference Manual and Application Programming Interface (API) for the PixelPusher® Framegrabber* documents the API. This document is located in the Doc sub-directory of your installation directory.

Working with the Sample Application Solutions



1. During the installation process, three (3) Visual Studio® solutions are installed in the default directory \Program Files\EyeTalk Systems\EyeTalkTrak\SDK\. They are:
 - EyeTalkTrak, containing the header and source files to build EyeTalkTrak.exe, which produces a MFC build
 - EyeTalkTrakNET, containing the source files to build EyeTalkTrakNET.exe, which produces a C# build
 - EyeTalkTrakVB, containing the source files to build EyeTalkTrakVB.exe, which produces a Visual Basic .NET build.

To use any of these solutions, the following steps are required and/or recommended:

1. **Recommended.** Copy the solution directory contained in the default directory SDK to another location on the hard drive, treating the copy of the directory in the SDK directory as an archive. Make any changes to the copied solution to meet your application needs
2. **Required for MFC builds.** If you did not accept the default installation directory during the install process, you must change the path to PixpushR.lib in order to build your solution. To do this, open the solution (EyeTalkTrak.sln) in Visual Studio® and select Project|Properties (Alt-F7) and make the following change: under Linker|Input|Additional Dependencies, enter the path to the PixPushMFC.lib file. **IMPORTANT:** in Visual Studio®, you must put path names that have spaces in quotes. For example, if you want to link to PixPushMFC.lib located in the default software installation, you would enter the following: "\Program Files\EyeTalk Systems\EyeTalkTrak\Lib\PixPushMFC.lib"
3. **Recommended for MFC builds.** The MFC executable (EyeTalkTrak.exe) needs to restart when changing color mode, resolution and screen format. When it restarts, it looks up the current working directory and restarts the application there. To facilitate this when running the application from Visual Studio®, you need to make the following change: select Project|Properties (Alt-F7), select Debug from the Configuration drop down list, expand Configuration Properties, select Debugging, and navigate to the directory containing the .exe file. Next, select Project|Properties (Alt-F7), select Release from the Configuration drop down list, expand Configuration Properties, select Debugging, and navigate to the directory containing the .exe file.

Application Deployment



Once the image processing algorithm development has been completed, the application can be deployed on the same hardware and with the same EyeTalkTrak interface. Alternatively, the programmer can develop his own application interface and link to the PixPushR.dll (or PixPushMFC.dll for MFC clients).

Menu Descriptions

Acquire Menu



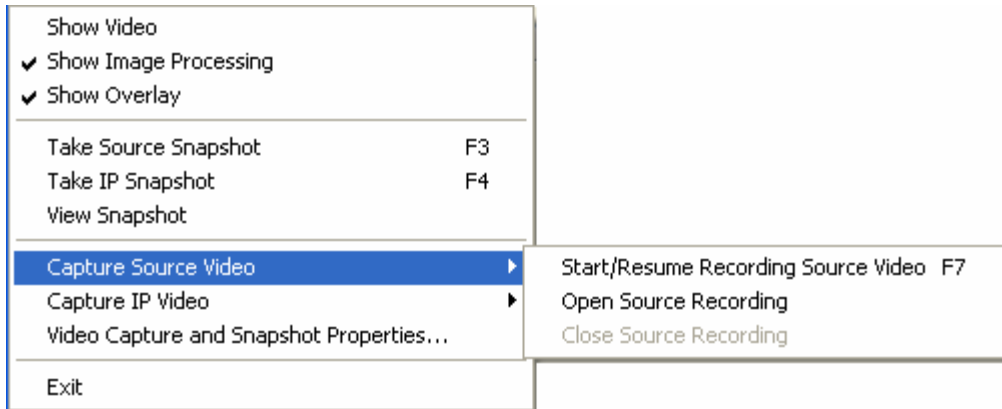
Show Video	
✓ Show Image Processing	
✓ Show Overlay	
<hr/>	
Take Source Snapshot	F3
Take IP Snapshot	F4
View Snapshot	
<hr/>	
Capture Source Video	▶
Capture IP Video	▶
Video Capture and Snapshot Properties...	
<hr/>	
Exit	

The operations available from the Acquire menu are:

- Start and stop video acquisition and display by selecting Show Video
- Start and stop image processing and its display by selecting Show Image Processing. Note that if Show Video is not activated, you will not see any image processing results as there are no source video frames captured
- Show an overlay with the captured video that indicates the camera source and the frame rate. The overlay is displayed in the top left corner of the display for the active video source. For the MFC sample application, this information is also shown on the status bar
- Take a snapshot of the source video stream, i.e., save out a single video frame. The shortcut key is F3
- Take a snapshot of the image processing video stream, i.e., save out a single video frame. The shortcut key is F4
- View a snapshot, which opens a still image viewer application to display all snapshots of type BMP, JPEG and TIFF
- Record the live source video to a .avi file
- Record the live image processing video to a .avi file
- Set [Video Capture & Snapshot Properties](#), which enables you to set parameters so that you can easily take a sequence of snapshots and save these as .jpg, .bmp or .tif files to a folder of your choice. You can also set various parameters for recording live video streams such as frame rate
- Exit the EyeTalkTrak program.

Capturing Live Source Video

You can capture live streaming video to a .avi file by selecting Capture Source Video from the Acquire menu, and then selecting Open Source Recording from the cascading menu. Once a file is open for writing, you may start and pause recording by toggling the F7 key (or selecting the appropriate menu item from the cascading menu). When you toggle video recording on, it will record until you toggle it off. Once you have captured all the live video, you should select the Close Source Recording menu item from the cascading menu. If you don't close the file, it will automatically be closed when you close the EyeTalkTrak application.

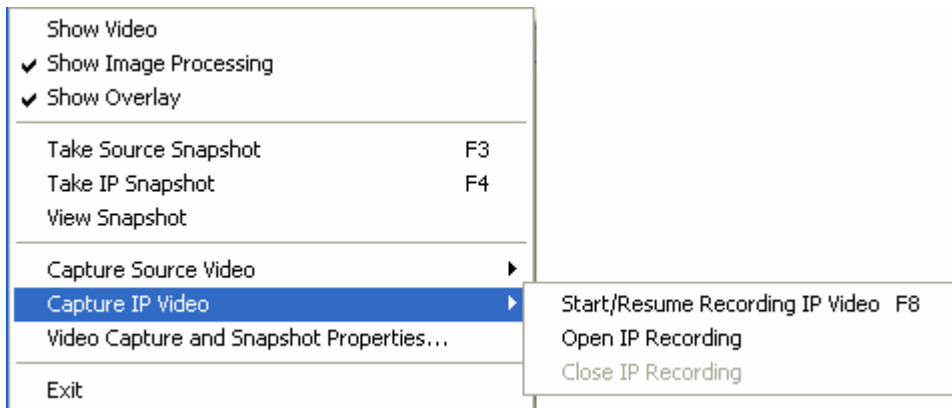


Instead of explicitly opening a file for recording, you may also select Start/Resume Recording Source Video (F7) and this will open a file automatically and begin recording video frames. To create two or more video files in a single session, you need to close the first one by selecting Close Source Recording, and open the next one either by selecting Open Source Recording again, or alternatively, by pressing F7 to open and start in one key action.

The status of recording source video is displayed on the status bar.

Capturing Live Image Processing (IP) Video

You can capture live streaming image processing video to a .avi file by selecting Capture IP Video from the Acquire menu, and then selecting Open IP Recording from the cascading menu. Once a file is open for writing, you may start and pause recording by toggling the F8 key (or selecting the appropriate menu item from the cascading menu). When you toggle IP video recording on, it will record until you toggle it off. Once you have captured all the live IP video, you should select the Close IP Recording menu item from the cascading menu. If you don't close the file, it will automatically be closed when you close the EyeTalkTrak application.



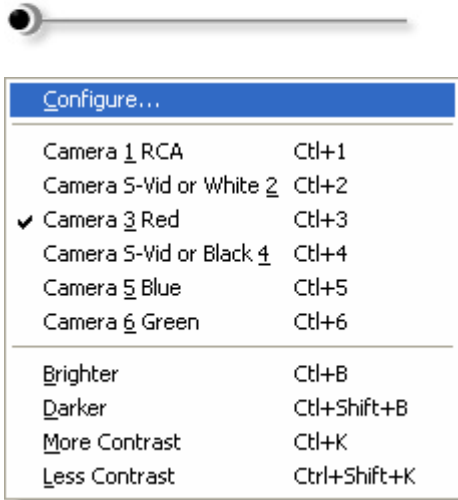
Instead of explicitly opening a file for recording, you may also select Start/Resume Recording IP Video (F8) and this will open a file automatically and begin recording IP video frames. To create two or more video files in a single session, you need to close the first one by selecting Close IP Recording, and open the next one either by selecting Open IP Recording again, or alternatively, by pressing F8 to open and start in one key action.

The status of recording image processing video is displayed on the status bar.

(MFC Only) Several of the menu items in the Acquire menu (Show Video, Show Image Processing, Take Source Snapshot, and View Snapshot) can be selected from the dockable General Controls toolbar if this toolbar is enabled from the [View Menu](#). In addition, the General Controls toolbar allows you to toggle between full screen and quarter screen display mode.



Camera Menu



You can perform the following operations from the Camera Menu:

- Configure or define the type of video source on a particular camera port, e.g., composite or S-Video, color or black and white
- Select a specific video source for display
- Control the brightness and contrast of the video image.

Cameras are specified on this menu using both a number and a color. The color associated with a particular camera number is the same color of the wire on the optional video breakout cable.

You can change the configuration of a camera by selecting the **Configure** menu item from the **Camera** Menu, or for MFC applications, the Camera Select toolbar (see [Configure a Video Source](#)). If you have re-configured a camera (video source), once you close the Camera Configuration dialog, you will be prompted to effect the changes now or defer them until later (see the message dialog below).



If you have changed the settings for the currently selected camera, e.g., changing from color to black & white, and want to see the changes now, you **MUST** select Yes. Otherwise, you can select No and your changes will be saved so that next time you select that camera, it will reflect the changes you made.

(MFC Only) Configuring and selecting a video source can also be done from the dockable Camera Select toolbar if this toolbar is enabled from the [View Menu](#).

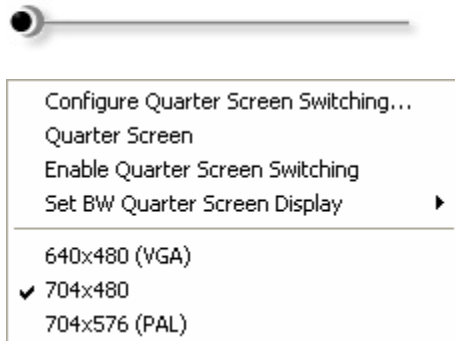


Image contrast and brightness controls increment/decrement these two video capture parameters by fixed increments on each click of the relevant button control (yellow for brightness, black/gray for contrast). When the limit is reached for a parameter, the setting "wraps around" to the smallest value of the parameter (if pressing an increment button) or the highest value of the parameter (if pressing a decrement button).

(MFC Only) Setting the image contrast and brightness can also be done from the dockable Image Control toolbar if this toolbar is enabled from the [View Menu](#). This is the preferred method of making these adjustments since you can view the results immediately without the distraction of bringing up a menu that overlays the video itself.

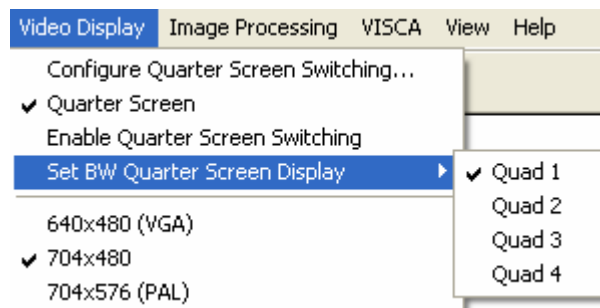


Video Display Menu



The operations available from the Display menu are:

- Select one of three full screen resolutions: 640 x 480 (VGA), 704 x 480, or 704 x 576 (PAL). After making a change, you will be advised that the software must be restarted to effect the change; you are given the option to cancel out of the change
- Configure Quarter Screen, which enables assignment of up to four (4) video sources to quarter screen image display areas. **IMPORTANT:** this function only applies to cameras that have been configured as black & white sources (of course, any color camera can be configured as a black & white source; see [Configure A Video Source](#))
- Quarter Screen, which toggles the display between full and quarter screen displays. This function behaves differently depending on whether the selected camera has been configured as color or black & white. If the currently selected camera is configured for color, the change does not require a restart and the results will be immediate. If the selected camera is configured for black & white, you will be advised that the software must be restarted to effect the change; you are given the option to cancel out of the change
- Enable Quarter Screen Switching, which causes the active video to be displayed in a particular quarter screen display area for a period of time specified by the user. This provides a means for viewing sequentially multiple camera sources
- Set BW Quarter Screen Display, which allows you to place the selected camera source (which must be configured as black & white) into a particular screen quadrant. **IMPORTANT:** to view the results of image processing in quarter screen mode, the source video of the selected camera source **MUST** be placed in quadrant 1.



A full VGA image measures 640 x 480 pixels. You may also display a 704 x 480 pixel image or a 704 x 576 (PAL) image.

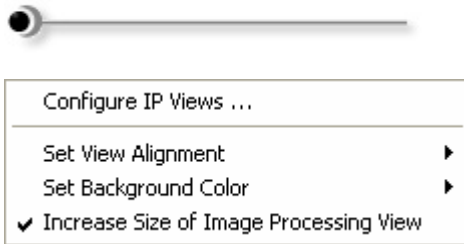
The Quarter Screen menu item allows you to configure a display to quarter screen resolution, which measures 320 x 240 pixels (when the primary resolution is set to 640 x 480), 352 x 240 pixels (when the primary resolution is set to 704 x 480), or 352 x 288 (when the primary resolution is set to 704 x 576). As many as four quarter screen windows of video source can be displayed at once on the screen. However, only one quarter screen panel is streaming active video at any given time. By using the built-in switching

function (see Configure Quarter Screen Switching), it is possible to update each panel in a round robin fashion.

(MFC Only) Two of the operations in the Video Display Menu can be accessed from the dockable Camera Select toolbar if this toolbar is enabled from the [View Menu](#). The two (2) rightmost buttons perform the Configure Quarter Screen and Enable Quarter Screen Switching functions respectively.



Image Processing Menu

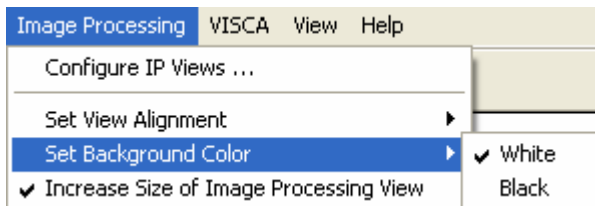


You can perform the following operations from the Image Processing menu:

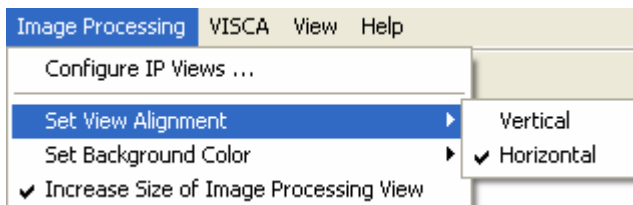
- Configure the image processing display to show as many as four (4) quarter screen views of the image processing pipeline. You can also show any single image processing stage in the image processing results view when operating in full screen mode. See [Configure Image Processing Views](#) for details
- (MFC Only) Set the view alignment so that the video display and image processing display views are arrayed horizontally or vertically
- Set the background color of the views
- (MFC Only) Toggle the display of the image processing results as small or large screen.

As many as four quarter screen panels, consisting of original video and image processing results, can be displayed at once on the screen. This is particularly useful when you wish to see the results of intermediate image processing steps. **IMPORTANT:** to view the results of image processing in quarter screen mode, the source video of the selected camera source **MUST** be placed in quadrant 1; this can be set from the [Video Display Menu](#).

You can set the background color of the splitter windows to either black or white by selecting an option from the cascading Set Background Color menu item as follows:



(MFC Only) The views containing the original video and the image processing results are implemented as two separate window "panes" of a splitter window. You can array these vertically (portrait style) or horizontally (landscape style) by selecting an option from the cascading Set View Alignment menu item as follows:



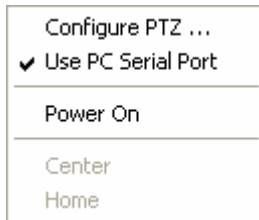
(MFC Only) You can also maximize or minimize the display of the Image Processing View by clicking on Increase Size of Image Processing View. This is a convenient method to switch back and forth between a screen that shows mostly the original video and a screen that divides its views roughly half between the source video and the image processing results of that video. The actual size of the views will depend on the overall size of the application window as you have sized it. This menu item simply makes it somewhat

easier to adjust the splitter window frames to the way you want to see the source video and image processing views.

(MFC Only) Most of the operations available from the Image Processing Menu can be accessed from the dockable Image Processing toolbar if this toolbar is enabled from the [View Menu](#).



VISCA® Menu



VISCA® is a camera control interface developed by Sony Corporation. This interface provides a serial communication protocol to control pan-tilt-zoom cameras. Several cameras support this interface, including the Sony EVI-D70 series. The menu above is taken from the MFC EyeTalkTrak sample application. .NET sample applications have a different menu, and do not implement the tilt and zoom control operations.

The PixelPusher® hardware supports a direct VISCA® interface through its 9 pin mini-DIN connector. In addition, communication with a VISCA® enabled camera is possible through the PC serial port. While not all operations defined in the VISCA® interface are supported in EyeTalkTrak, many commonly used ones are.

The following operations are available from the VISCA® menu:

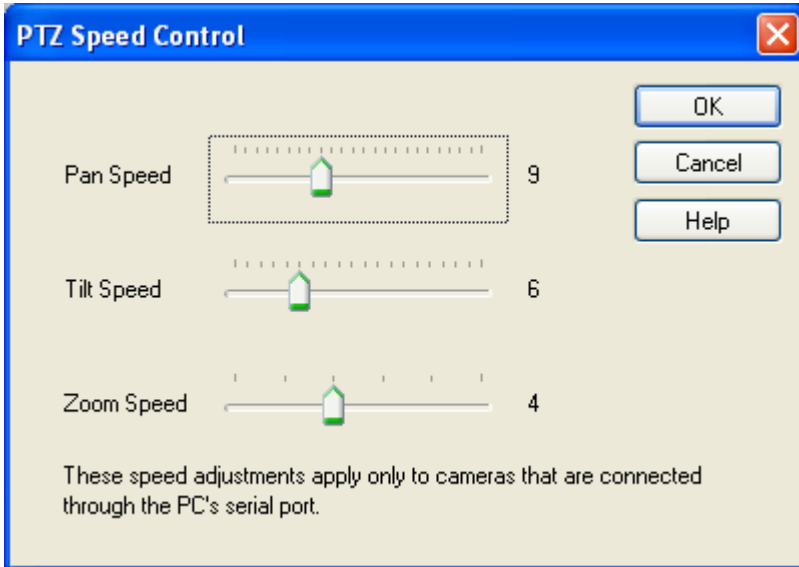
- Configure PTZ, which provides settings to control the speed of panning, tilting, and zooming operations (for cameras connected via the PC's serial port)
- Use PC Serial Port, which can be toggled on to enable control of a VISCA® enabled camera through the PC's serial port (versus through the 9 pin mini-DIN connector)
- Turn on/off camera power. IMPORTANT: this feature is only applicable when communication with the VISCA® enabled camera is through the PC's serial port. When the camera is connected through the 9 pin mini-DIN connector, it must be started from the camera's remote controller.

(MFC Only) The same operations can be accessed from the dockable Camera Control toolbar if this toolbar is enabled from the [View Menu](#). For MFC applications, pan, tilt and zoom operations are only available through the Camera Control toolbar pictured below. The buttons are the press and hold style. For example, the camera will continue to pan left as long as the user is holding down the left pointing arrow button. When it is released, panning will stop.



(.NET Only) Camera pan left, pan right and pan stop have been implemented in the .NET sample applications. These operations are available from the VISCA menu. Press and hold style buttons have not been implemented.

To configure the speed of camera motion in the pan, tilt and zoom directions, select the Configure PTZ menu item, which is also available on the Camera Control toolbar. Configure PTZ raises the following dialog:



Note that the ranges in the three parameters have different bounds, i.e., they are not all scaled between the same fixed range. This is not of practical importance. It is only important to know that the higher the number, the higher the speed of motion in any given direction.

The speed control settings only apply to cameras that are connected through the PC's serial port. Cameras connected through the 9 pin mini-DIN connector port can only move at a fixed rate in any direction, and will ignore any changes made through the dialog above. For this reason, it is recommended that the user make a connection to the camera through the PC's serial port to obtain the greatest flexibility in control of a VISCA enabled camera.

View Menu (MFC Only)

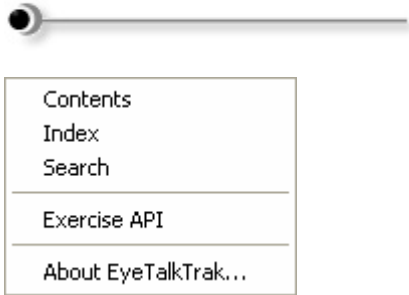


(MFC Only) The View Menu provides a means to display or turn off the various toolbars and the Status Bar. Display of the following toolbars can be accomplished through this menu:

<i>Toolbar</i>	<i>Purpose</i>
General Controls	Commonly used functions such as start and stop video capture and image processing display, take and view video snapshots, toggle between full and quarter screen display
Status Bar	Display source camera, video frame rate and default Windows messages
Image Control	Change the brightness and contrast of a displayed image
Camera Control	Configure and pan, tilt and zoom an attached camera
Camera Select	Configure and select a video source (camera) to display
Image Processing	Configure IP view panes, set background color, and view image processing results

All the toolbars are dockable, so that you can move them around to a configuration that best satisfies your needs.

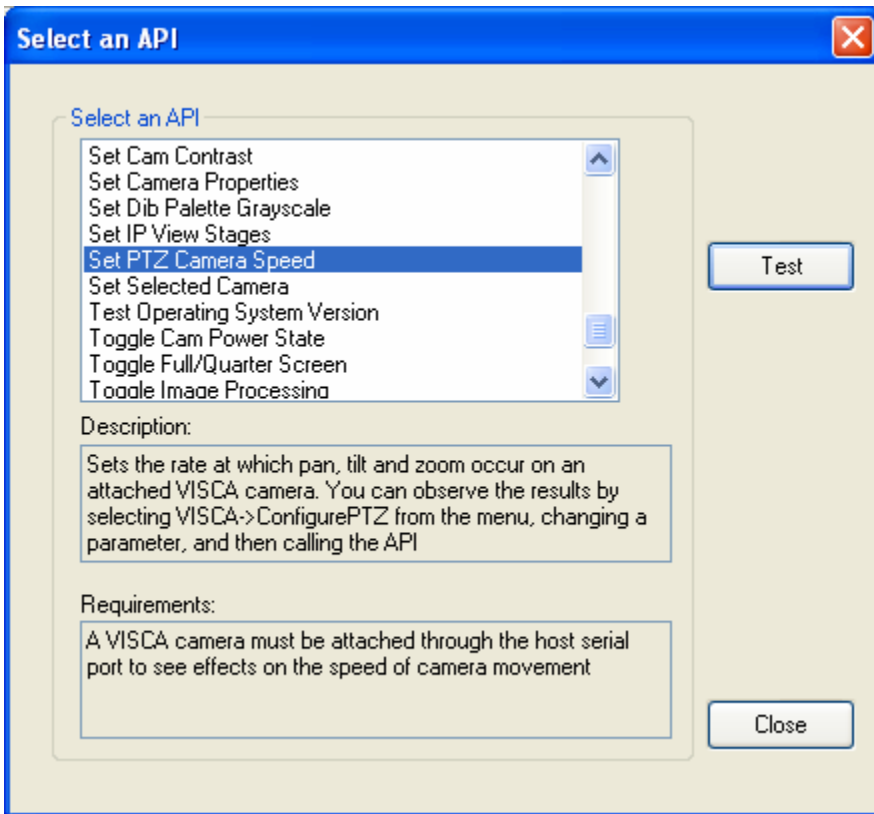
Help Menu



The following operations are available from the Help menu:

- Access to the online Help Contents tab
- Access to the online Help Index tab
- Access to the online Help Search tab
- Test an application programming interface (API) call
- Print out a diagnostic report that includes registry key values and other information.

The Exercise an API menu item raises a dialog that allows you to select an API and test it. Each API is shown with a description of what it does, i.e., what you should see when it is executed, and any special conditions that must be satisfied in order to demonstrate the API. This feature is intended for use by software developers that want to extend the EyeTalkTrak interface or build their own application that link through the PixPushR (or PixPushMFC) DLL to access the PixelPusher® hardware.



By selecting the About EyeTalkTrak menu item and clicking on the Write Diagnostics File button, you can print out a report (.txt) file that contains the registry key values, module version numbers, and other data that are useful in diagnosing problems that you might be having. If you contact our Technical Support, you may be asked to write out this file and send it to us.



Technical Support

Known Problems



Known problems in this release are as follows:

Problem: Quarter Screen Display at 704*480 Resolution Wraps Around

The quarter screen display at resolution of 704*480 sometimes comes up with a horizontal offset where video lines wrap around to the next line. The result is that the image appears split, with the rightmost part of the video appearing on the left side.

Workaround:

A normal display can usually be achieved by starting and stopping the video one or a few times, or by restarting the application

Problem: Display at 704*480 Resolution Is Tiled

At full screen size and at 704*480 resolution, the display is tiled into 8 separate video areas.

Solution:

This may occur if you have recently re-installed the driver software (and not the SDK) on your PC. Installing the driver only will place the file PixPush.spt dated 5/12/07 in your \windows\system32\drivers directory. You must replace this with the file PixPush.spt dated 6/22/07. This problem does not occur when you perform an initial install because the correct PixPush.spt (dated 6/22/07) is copied to the drivers directory. The new version is located in the Driver sub-directory of the install directory. If you accepted the default install location, this file is located at \Program Files\EyeTalk Systems\EyeTalkTrak\Driver. Copy this file to the \Windows\system32\drivers directory.

Problem: Camera Switching may Stop Unexpectedly

If you have started camera switching, whereby up to four (4) video sources are displayed at quarter screen resolution in a sequential manner, the display may stop unexpectedly, which can only be corrected by ending the task from the Windows Task Manager.

Workaround:

There is no known complete workaround. The problem seems to occur most often when running in debug mode, and when the switching interval is less than 1,000 msec. It may help to increase the switching interval to a larger value, especially when operating in debug mode.

Problem: Calling Some Combination of APIs Near Simultaneously May Halt Video

In addition to handling the video stream, PixelPusher® also polls a command stream from the user interface. During a single frame processing, PixelPusher® handles at most one command from the command stream. In practical use of EyeTalkTrak at the user interface, only a single command is presented at a given time (since it is essentially impossible for a human user to enter sequential commands quickly enough). However, in Debug mode, and using one or more API calls in a suitable context, it is possible to create a situation where PixelPusher® needs to handle multiple commands. In

this case, PixelPusher® may fail to handle the multiple requests and the video may freeze. In practical applications, this is not a problem.

Workaround:

This problem is probably restricted to Debug mode, so it may not ever be realized in practice with a Release build. To reproduce the problem, place a call to PP_SetCamContrast() in the EyeTalkTrak program at the beginning of the OnCameraCam1() handler, and create a Debug build. Run the program and select between say camera 1 and any other camera (configured for the same color mode) several times. This may cause the video to freeze. It may take several iterations to create the failure. The freeze results because at the same time PixelPusher® is handling the change to another camera, it receives a command to adjust the contrast.

Problem: EyeTalkTrak Fails to Start, or Video is a Greenish Monochrome Patchwork

If one camera is configured as black & white, and a second is configured as color, when you switch between the cameras, the EyeTalkTrak program automatically restarts itself in the alternate color mode. When this occurs, it is important that the registry keys for "Cam Current" and "Color Mode" be consistent. That is, if the camera configuration byte in "Cam Current" indicates a color camera (bit 2 = 1), then the "Color Mode" needs to be set to 1 also. Normally, EyeTalkTrak ensures consistency. However, if you use the APIs to build your own user interface, it is possible that these will get out of synch, in which case you may observe either of the following: (a) the application refuses to start, with a message that the device has lost power and/or may be disconnected; (b) you will see a greenish monochrome patchwork of video data on the screen. The former indicates that the "Color Mode" is set to 0 (for B&W) while the camera configuration byte in "Cam Current" indicates a color camera. The latter indicates that the "Color Mode" is set to 1 while the camera configuration byte in "Cam Current" indicates a black & white camera.

Quick Workaround:

For a quick fix, modify the registry settings to: "Color Mode" = 0; "Cam Current" = 0x20, indicating camera 1 is black & white; and "Cam 1" = 0x20. Then power cycle PixelPusher® to get the device to start correctly.

Solution:

To avoid this problem, modify your custom code to ensure that "Cam Current" and "Color Mode" are consistent when you are doing color mode changes. One way to ensure this is to call the PP_ChangeColorMode() which assigns consistent values to the "Cam Current" and "Color Mode" registry keys. See the code in any of the sample applications to see how to use PP_ChangeColorMode().

Problem: (MFC Only) Need to Restart to Allocate an Adequate Number of Image Processing Arrays

This is not really a problem, but part of the design of the software for the MFC sample application build. Nevertheless, it may trouble a new user, so an explanation is in order. When you first start a PixelPusher® enabled application after its installation, the program does not know, in general, how many image processing (IP) stages your IP functions will be using. It is not until you select [Show Image Processing from the Acquire Menu](#) that the software validates your IP function pointers to image arrays. Once IP processing has been invoked, the software performs the validation. If it finds that you have an insufficient number of image arrays allocated based on the parameters in your IP functions, it will inform you that it needs to restart to adjust the number of image arrays, and you must then **manually restart** the software. When it starts again, the correct number of arrays will have been created, and you will not see the message anymore.

Solution:

To avoid seeing this message at all, before you run the application for the **first time** after an install, you can manually change the registry key value in: HKEY_CURRENT_USER/Software/EyeTalk Systems/EyeTalkTrak/Version 1.0/Max Pipeline Stages to a large enough number to cover your anticipated image arrays. However, it is probably better to allow the software to discover the minimum essential number for you. That way, you are not wasting memory by creating too many arrays. After the application has been started, you can adjust the number of stages by selecting Configure from the Image Processing menu.

Troubleshooting



Installation Problems

Problem: I am running Windows **Vista**. I completed the install but when I tried to launch the program from the shortcut on the desktop, I receive the message: "The application has failed to start because its side-by-side configuration is incorrect. Please see the application event log for more detail."

Solution: You need current MFC and CRT redistributables, which can be installed on your computer by running the program `vcredist_x86.exe`, [located on your install disk](#). You can also download this file from Microsoft at [vcredist_x86.exe](#)

Problem: I am running Windows **XP**. I completed the install but when I tried to launch the program from the shortcut on the desktop, I receive the message: "Unable to start program 'c:\Program Files\EyeTalk Systems\EyeTalkTrak\EyeTalkTrake.exe'. This application has failed to start because the application configuration is incorrect. Review the manifest file for possible errors. Reinstalling the application may fix this problem. For more details, please see the application event log."

Solution: You need current MFC and CRT redistributables, which can be installed on your computer by running the program `vcredist_x86.exe`, [located on your install disk](#). You can also download this file from Microsoft at [vcredist_x86.exe](#)

Problem: I just installed the software on a **Windows 2000** host. I started video OK. However, when I switch from full to quarter screen or from a camera configured as black & white to one of color (or vice versa), I get an Unknown Device message.

Solution: This has been observed to occur on the Windows 2000 platform only. Usually, just restarting the computer after the initial install will prevent this from happening again. Instead of restarting, you can navigate to the directory you chose to install the program, e.g., `\Program Files\EyeTalk Systems\EyeTalkTrak\Driver` (if you accepted the default install location) and select the file that Windows is looking for, usually `PixPush.spt`. This should correct the problem. You can also perform a complete installation of the driver (see [driver reinstallation](#)).

Problem: I was able to install the software and can run the built executables for EyeTalkTrak and EyeTalkTrakLite that ship with the product. However, when I create a clean build and attempt to run it from Visual Studio, a message box pops up with following error message: "The application failed to initialize properly (0xc0150002). Click OK to terminate the application." Also, in the Visual Studio output window, its says "LDR: LdrpWalkImportDescriptor() failed to probe c:\EyeTalkTrak\debug\Pixpush.dll for its manifest, ntstatus 0xc0150002." What does this mean?

Solution: Your application must link to `PixPush.dll`, which was built with Visual Studio 2005 SP1 and which uses MFC and CRT side-by-side assemblies numbered 8.0.50727.762. This error only occurs when you run in debug mode. When you run in release mode, this problem does not occur because the EyeTalkTrak installation program automatically installs the correct **release** version of the MFC and CRT side-by-side assemblies. But for **debug** mode, the correct version of the side-by-side assemblies are only obtainable through the Visual Studio 2005 installation; these assemblies, unlike their release version counterparts, cannot be redistributed. To fix this, you need to upgrade your Visual Studio 2005 installation to service pack 1 by downloading and installing [Visual Studio 2005 SP1](#). If you are running Visual Studio 2005 on Windows Vista, after installing the SP1 update, you should also download and apply the special update for Vista, which is available from the link above.

Device Connection Problems

Problem: I get the message: "Unable to start the program. Check to be sure that PixelPusher® is turned on and that the USB cable is connected." What is causing this?

Solution: Check the following in the sequence given:

1. Is the PixelPusher® hardware plugged in and powered up? Note that the Off/On button will be lit when just the USB cable is connected. However, the device must also be powered from the +5V supply that is included with the hardware
2. Is the USB cable connected to both the computer and PixelPusher®? If not, the Off/On button will NOT be lit
3. Are you working with multiple computers in close proximity? Do you have the PixelPusher® hardware plugged into the same computer that you are trying to run the EyeTalkTrak software from?
4. Open Device Manager: right click My Computer, select Properties, select the Hardware tab, and click on the Device Manager button. Then expand the Sound Video & Game Controller device node. You should see "PixelPusher(R) PIXPUSH.INF - 1005" listed in the device list. (Another way to get to the Device Manager is to click on the Start menu, select Run, enter devmgmt.msc and click OK)
5. If the device "PixelPusher(R) PIXPUSH.INF - 1005" does not appear, but rather an "Unknown Device," try reinstalling the driver (see [driver re-installation](#)).

Re-installing the PixelPusher® Driver

Problem: The EyeTalkTrak software is already installed, but I need to reinstall the driver. How do I do this?

First, check to see if the host recognizes the driver. Open Device Manager: right click **My Computer**, select **Properties**, select the **Hardware** tab, and click on the **Device Manager** button. Then expand the **Sound Video & Game Controllers** device node. If the device is correctly installed, you should see "PixelPusher(R) PIXPUSH.INF - 1005" listed in the device list. (Another way to get to the Device Manager is to click on the **Start** menu, select **Run**, enter **devmgmt.msc** and click OK).

Solution: If the PixelPusher® hardware does not show up correctly, but there is a device called "Unknown Device" somewhere in the hardware device "tree," you need to reinstall or update the driver. In general, the source files you need may be obtained by accessing files from the **Driver** directory that was created when you ran the installation on your machine during a previous install, by accessing source files from the installation CD, or by accessing Windows Update. The solution also varies slightly across operating systems. Select your operating system for specific steps:

- [Windows 2000](#)
- [Windows XP](#)
- [Windows Vista](#)

Re-installing the Driver for Windows 2000

Re-install from the installation CDROM:

Open Device Manager: Right click on the "Unknown Device," select **Properties**, select the **Driver** tab, and click on the **Update Driver** button. Click **Next** on the "Welcome to the Upgrade Device Driver Wizard." Click on the **Display a list of the known devices ...** radio button, click **Next**, click on **Have Disk**, browse to the CDROM Drive subdirectory (e.g., d:\driver). Select Pixpush.inf, click the **Open** button, click **OK**, select "PIXPUSH.INF - 1005" from the driver list if you are given a choice, click **Next** twice, click **Finish**, and finally, click **Close**. IMPORTANT: after re-installing from the installation CD, you must copy the file pixpush.spt dated 6/22/07 from the Driver sub-directory of the install directory to the \Windows\system32\drivers directory.

Re-install from Windows Update (only if your computer has Internet access):

Make sure the installation disk is **NOT** in the CDROM drive. Open Device Manager: Right click on the "Unknown Device," select **Properties**, select the **Driver** tab, and click on the **Update Driver** button. Click **Next** on the "Welcome to the Upgrade Device Driver Wizard." Click on the **Search for a suitable driver for my device (recommended)** radio button, click **Next**, check the box **Microsoft Windows Update** and clear the other check boxes. Click **Next**. Windows update should find the correct driver. If it does not, you will be prompted to insert the "EyeTalk Systems PixelPusher(R) Installation Disk," which means that Windows Update did not find or was not able to install the appropriate driver software to your host. Click **Next**, click **Finish**, and finally, click **Close**. IMPORTANT: after re-installing from Windows Update, you must copy the file pixpush.spt dated 6/22/07 from the Driver sub-directory of the install directory to the \Windows\system32\drivers directory.

Re-installing the Driver for Windows XP

Re-install from the installation CDROM:

Open Device Manager: Right click on the "Unknown Device," select **Properties**, select the **Driver** tab, and click on the **Update Driver** button. Click **Next** on the "Welcome to the Hardware Update Wizard." Select the "No, not this time" radio button. Click **Next**. Click the "Search removable media" check box, and click **Next**. The host PC will prompt you to insert the "EyeTalk Systems PixelPusher(R) Installation Disk" installation into the CDROM drive. Insert the CDROM, and click **OK**. A "Files Needed" dialog will probably appear. If so, browse to the Driver folder, e.g., d:\Driver on the installation CDROM, select the pixpush.sys file, click **Open**, and then click **OK** on the "Files Needed" dialog. Click **Finish**. IMPORTANT: after re-installing from the installation CD, you must copy the file pixpush.spt dated 6/22/07 from the Driver sub-directory of the install directory to the \Windows\system32\drivers directory.

Re-install from Windows Update (only if your computer has Internet access):

Make sure the installation disk is **NOT** in the CDROM drive. Open Device Manager: Right click on the "Unknown Device," select **Properties**, select the **Driver** tab, and click on the **Update Driver** button. In the "Welcome to the Hardware Update Wizard" dialog, click on the "Yes, this time only (Recommended)" radio button and click **Next**. Click on the "Install the software automatically" radio button, and click **Next**. Windows Update should find one or more drivers and present them in a dialog. Click on the digitally signed driver from Windows Update and click **Next**. IMPORTANT: after re-installing from Windows Update, you must copy the file pixpush.spt dated 6/22/07 from the Driver sub-directory of the install directory to the \Windows\system32\drivers directory.

Re-installing the Driver for Windows Vista

Re-install from the installation CDROM:

Open Device Manager: Right click on the "Unknown Device," select **Update Driver Software...**, click on **Browse my computer for driver software**, insert the CDROM, browse to the CDROM drive, and click **Next**. IMPORTANT: after re-installing from the installation CD, you must copy the file pixpush.spt dated 6/22/07 from the Driver sub-directory of the install directory to the \Windows\system32\drivers directory.

Re-install from Windows Update (only if your computer has Internet access):

The PixelPusher driver has only been WHQL tested for Windows XP and Windows 2000, but not Windows Vista. You should therefore **NOT** attempt to re-install from the Windows Update at this time. An attempt to re-install this way will fail with the notification: "Windows found driver software for your device but encountered an error while attempting to install it."

Problems Related to Developing Your Own Client Application

Some users want to develop their own client applications and link to PixPushR.dll (.NET clients) or PixPushMFC.dll (MFC clients). The sample applications provide working code to show you how to compose a working application; these applications show what calls need to be made and in what order. In addition, in the TestAPISnippets() method (located in TestAPI.cs, TestAPI.vb, and EyeTalkTrakDoc.cpp source files), there is working code for every API exported from these DLLs.

If you want to add calls to PixPushR (or PixPushMFC) exported functions to an existing application that you have, you need to be aware of certain problems that may arise.

Problem: I have a .NET client application that I built that links to PixPushR.dll. However, I cannot get any source video or image processing results to display.

Solution: You need to ensure that the name of your application, as it appears in the main form's title bar is in agreement with the value you are passing in the call to PP_InitPixPush(). See section 4.3 in the [EyeTalkTrakReference Manual and Application Programming Interface \(API\) for the PixelPusher® Frame Grabber \(.NET Version 1.2\)](#), dated September 30, 2009 for details.

Note that when you use an application name different from one of the sample applications (i.e., EyeTalkTrak, EyeTalkTrakNET, or EyeTalkTrakVB), this creates a new registry key under HKEY_CURRENT_USER\Software\EyeTalk Systems\EyeTalkTrak. For example, if you call your application FOOBAR, a new registry key will be created as HKEY_CURRENT_USER\Software\EyeTalk Systems\EyeTalkTrak\FOOBAR. The first time you run your application, the program will assign default values to the registry keys under this registry path. With the default assignments, neither source video nor image processing video will display, so you must call the appropriate APIs to start video and image processing, i.e., PP_AcquireVideo() and PP_SetDoImageProcessing().

Other Problems

Problem: When running at 704*480 resolution, my screen shows eight 1/8 screen panels instead of a normal full screen image. The image at 640*480 is normal.

Solution: You may have an outdated pixpush.spt file in the Windows system directory. You need to replace the pixpush.spt file dated 5/12/07 with a newer version (dated 6/22/07 or later). The new version is located in the Driver sub-directory of the install directory. If you accepted the default install location, this file is located at \Program Files\EyeTalk Systems\EyeTalkTrak\Driver. Copy this file to the \Windows\system32\drivers directory and power cycle the hardware.

Technical Support



The most up-to-date information, including hardware drivers, release notes, and software releases can be obtained from our website at www.eyetalksystems.com.

FAQs



Question: Do I need to purchase the optional video breakout cable

Answer:

If you plan to only use the RCA and S-Video jacks on the PixelPusher® device, you do not need the breakout cable. However, if you plan to connect one or more cameras having BNC connectors, you will probably want to purchase the breakout cable. You can, of course, purchase an adapter from Radio Shack or other popular consumer electronics stores to convert a BNC connector to the RCA jack, but this limits you to attaching a single BNC device to PixelPusher®.

Question: Why does making a change in resolution require a restart?

Answer:

The design of PixelPusher® is such that when it is started and the video frame data structures are created, they are sized to work for a single resolution (e.g., 640*480=307,200 pixels). When the resolution changes, these data structures need to be re-created and the easiest way is to restart the application.

Question: Why does making a change in color mode require a restart?

Answer:

The design of PixelPusher® is such that when it is started and the video frame data structures are created, they are sized to work for a single resolution (e.g., 640*480=307,200 pixels). When the resolution changes, these data structures need to be re-created and the easiest way is to restart the application. A color video frame requires twice the pixel values as a black & white frame. Therefore, the video data structures are sized differently and a restart is required.

Question: Why does making a change in screen format (e.g., full to quarter screen) for a camera configured as black & white require a restart, whereas if the camera is configured for color, it does not require a restart?

Answer:

When PixelPusher® is set to run at quarter screen black & white, the downsampling of a full image (say from 640*480 to quarter screen 320*240) is accomplished on the hardware. Since it takes less time to pass a quarter screen image over the USB bus, this may improve the frame rate if the user is applying compute intensive image processing operations to the source video, i.e., less time will be required to handle the source video, allowing more time to be devoted to the image processing. The net result may be a better frame rate, particularly if the host PC processor is relatively slow. When switching between full and quarter screen black & white, a restart is required for the same reason that one is required when we do a resolution change, i.e., we need to re-create data structures that are sized to work for a single resolution.

In the case of processing video from a camera configured for color, the current hardware/firmware does not support downsampling of a full image on the hardware. To provide a quarter screen capability for color, we downsample the full screen image on the host PC. This is why, unlike with black & white, there is no potential frame rate advantage to viewing color at quarter screen.

Question: I was making code changes, restarted the application, and now my video is all scrambled

Answer:

When PixelPusher® starts up, it looks up the values stored in the registry keys "Img Res Width" and "Img Res Height" to establish the image resolution, i.e., 640*480, 704*480, or 768*480 (PAL). When the user chooses to switch to another resolution, the program writes the new values to the registry before closing. In that way, when the program is restarted (automatically), the intended resolution values are properly recorded, and the framegrabber is reset and capable of accepting new values.

If you have been making code changes, it is possible that you have (possibly inadvertently) changed the resolution values stored in the registry without having issued a reset to the framegrabber. Under these conditions, you may see scrambled video on the display. To fix this problem, simply power down the framegrabber and restart it. When it comes up, it should display the resolution recorded in the registry.

Question: I built one of the existing sample solutions in Visual Studio, but the program fails to restart correctly on color mode, screen format or resolution changes

Answer:

When you switch from a black & white camera to a color one, change screen format from full to quarter screen, or change screen resolutions, you receive a message like:

ERROR: CVideoDisplay::StartNewAppInstance: Unable to start a new application instance because the directory "" specified in the registry key "InstallDir" does not exist on this PC. Change the registry key value directly or from your code.

You need to make the Visual Studio project change described in [Working with the Sample Application Solutions](#) so that the location of the executable to restart is known to the application instance that is closing.

Question: I created two (or more) different PixelPusher® enabled applications. When I close one of the applications and start a different one, sometimes I get the message that it could not start the device

Answer:

In general, once you start running one PixelPusher® enabled application, you should power cycle the PixelPusher® hardware before attempting to start a different one. Sometimes switching without power cycling will work fine, but this is not guaranteed. The reason is that the two applications may have different starting conditions as defined in their respective registries. For example, if application 1 was last closed processing a black & white video source, but application 2 was last closed processing a color video source, and PixelPusher® is not restarted, it will attempt to launch the application with a color source, but the hardware (PixelPusher®) is still configured as a black & white source. The result will be the message notification that it is unable to start the EyeTalkTrak software.

Question: I created a clean build and attempted to run EyeTalkTrak from the Visual Studio interface but I get the error message: "The application failed to initialize properly (0xc0150002). Click OK to terminate the application." Also, in the Visual Studio output window, it says "LDR: LdrpWalkImportDescriptor() failed to probe c:\EyeTalkTrak\debug\Pixpush.dll for its manifest, ntstatus 0xc0150002." What does this mean?

Answer:

You need to upgrade your Visual Studio 2005 to service pack 1. Click [here](#) for more details, including how to get the service pack.

Question: I am not interested in applying image processing operations to the source video. Do I still need to supply a window handle for the image processing window in the call to PP_InitPixPush()?

Answer:

You can pass an IntPtr (int* in MFC) set to 0 for the parameter imgViewHandle.

Glossary

C

CVBS: Composite Video Broadcast Standard, i.e., composite video

D

DLL: Dynamic Link Library, a software module that exposes an interface (set of functions) which another software module may call

M

MFC: Microsoft Foundation Classes

N

NTSC: National Television Standards Committee, the technical committee that defines video broadcast standards in the United States. NTSC uses 525 lines per frame at 29.97 frames per second

O

OpenCV: OpenCV is a computer vision and image processing library originally developed by Intel. It is cross platform compatible and is freely downloadable for both commercial and research use at <http://sourceforge.net/projects/opencvlibrary/>. OpenCV has in excess of 500 array manipulation and image understanding/processing related functions and provides a solid foundation for creating application specific libraries.

P

PAL: Phase Alteration Line, the video standard used in Europe and other countries. PAL uses 625 lines per frame at 25 frames per second

Q

Quarter Screen: A video presentation that is one-quarter of the base resolution. For example, if the base resolution is 704 x 480, quarter screen would be 352 x 240

QVGA: Quarter VGA, which measures 320 by 240 pixels

S

S-Video: Video signal where the luminance and chrominance channels are separated into distinct components (vs. CVBS video where these two channels are "composited")

SDK: Software Development Kit

V

VGA: VGA or full VGA, which measures 640 x 480 pixels

Index

A	
Acquire Menu	22
Autonumbering	9
C	
Camcorders	7
Camera	7, 8, 22, 25, 33
Camera Configuration dialog	8
Camera Control	1, 31, 33
Camera leads	7
Camera Menu	7, 25
Camera Select	33
Capture Video	1, 22
CCD	7
CEyeTalkTrak	19
CEyeTalkTrakDoc	19
CEyeTalkTrakView	19
CImgPro	19
CImgProSpt	19
CMainFrm	19
CMOS cameras	7
Configure PTZ Camera	31
Configure Quarter Screen	27
D	
DB9	7, 8
Display Menu	27
DVD	7
E	
Enable Autonumbering	9
Enable Switching	27
F	
Fax Viewer	9
FCC	1
G	
General Controls	33
I	
Image Control	25, 33
image processing stage	11
Image Processing View	29
M	
Maximum Snapshots	9
MFC	1, 19
Microsoft Foundation Classes	1
Microsoft Paint	9
Microsoft® Visual Studio 2005	3
Mini-DIN	31
N	
Notify on Rollover	9
NTSC	1, 7, 8
O	
OpenCV	1, 11
P	
PAL	1, 7, 8, 27
PixelPusher®	1, 7, 19
Pixpush.dll	1, 21
Q	
Quarter Screen	11, 27
QVGA	27
R	
RCA	1, 7, 8
Resolution	
QVGA	1, 27
VGA	1, 27
Resolution	1, 27
RS-170	1, 8
S	
Select Camera list	8
Serial Port	31
Set Background Color menu item	29
Set Snapshot Properties	9
Set View Alignment menu item	29
Show Image Processing	22
Show Image Processing View	29
Show Overlay	22
Show Video	22
Snapshot Properties dialog	9
Snapshot Properties menu item	9
Sony Corporation	31
Sony EVI-D70	31
Sony VISCA®	1
Status Bar	33
S-Video	1, 7, 8, 25
System Requirements	3
T	
Take Snapshot	15
Technical Support	42
Test Individual Application Programming	
Interfaces	15
These Quick Installation	4
Toolbars	33
Turn on/off camera	31
U	
USB	1
USB 2.0	1, 3
V	
VCR	7
VGA	27
Video Display Menu	15, 27
video sources	7, 8, 25, 27, 33
View Menu	33
VISCA®	31
VISCA® Menu	31
W	
Walkthrough	15
Website	42