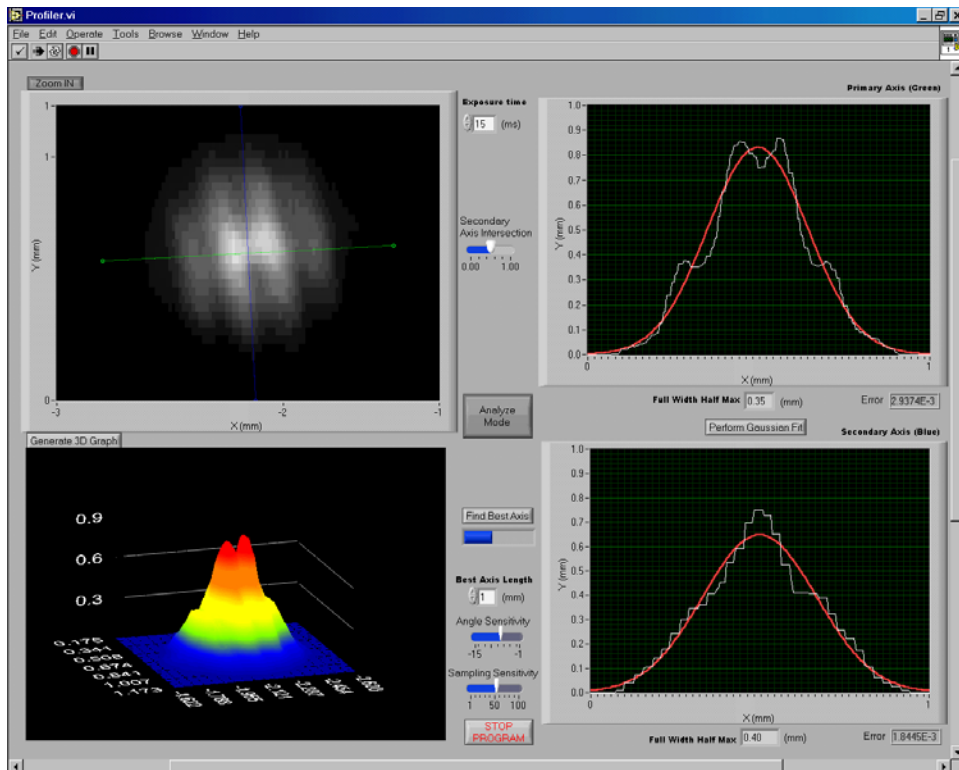


# User's Manual for the EDC-1000HR

## Beam Profiler



Jeffrey B. Cole  
University of California, Davis  
Department of Applied Science, LEORG  
Copyright © 2003



# Contents

<b>Chapter 1</b>	<b>General Information</b>	
	Overview .....	3
	Requirements .....	3
<b>Chapter 2</b>	<b>Setup</b>	
	Hardware .....	4
	Software .....	4
<b>Chapter 3</b>	<b>Using the Spatial Profiler</b>	
	Preview Mode .....	5
	Analyze Mode .....	6
	Generating Primary & Secondary Axis .....	7
	User defined axis generation .....	7
	Automatic axis generation .....	8
	Zoom In / Zoom Out .....	9
	Generate 3D Intensity Graph .....	9
	Best-Fit Gaussian .....	10
<b>Chapter 4</b>	<b>Coding and Algorithms Explanation</b>	
	Overview .....	11
	Preview Mode .....	12
	Analyze Mode .....	12
	Primary and Secondary Axis Cursors Concept .....	13
	Make Secondary Axis Perpendicular to Primary Axis.....	14
	Use the Four Endpoints to draw lines .....	14
	Generate Primary and Secondary Axis Cross-Sections...	15
	Find Best Axis Tool.....	16
	Generate Best-Fit Gaussian Tool.....	17
	Generate 3D Intensity Graph.....	18
	Zoom On Intensity Graph.....	19
<b>Chapter 5</b>	<b>EDC-1000HR Specifications</b> .....	20

# Chapter 1

## General Information

### Overview

This manual describes a laser beam spatial profiler software program written in LabVIEW. The program uses an EDC-1000HR camera to detect a two dimensional cross-section of a laser beam's intensity. The user is then able to perform the following calculations to analyze the spatial profile of the laser:

- Locate the primary and secondary axis
- Generate a 3D intensity graph
- Zoom in on the 2D intensity image
- Generate cross-sections of the intensity profile
- Generate best-fit Gaussian curves
- Find the full-width half-maximum (FWHM) of the beam

### Requirements

- EDC-1000HR camera
- LabVIEW
- A 15-pin pc cable
- PC card for the EDC-1000HR
- An open ISA slot for the PC card
- Profiler Software Package (written in LabVIEW)
- HRCAM32.DLL (Application file for camera)
- EDC360HR.VXD (Virtual Device Driver for camera)
- HRCAM32.LIB (Library file for camera)

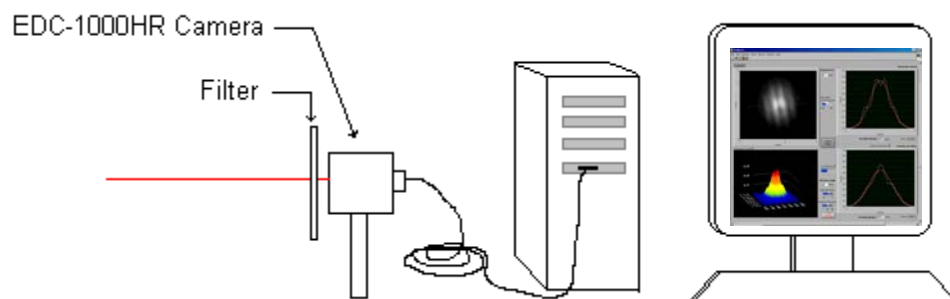
## Chapter 2 Setup

### Hardware Setup

1. Connect the EDC-1000HR to the computer by inserting the card into an open ISA slot in the PC.
2. Connect the camera to the card using a 15-pin cable.
3. Place an appropriate amount of density filters directly in front of the CCD detectors in the camera to attenuate the beam and protect the camera from over exposure.

### Software Setup

1. Place the driver file 'HRCAM32.VXD' into the Windows System folder.
2. Place the library file 'HRCAM32.LIB' into the Windows System folder.
3. Place the application file 'HRCAM32.DLL' into the Windows System folder.
4. Open the LabVIEW program titled 'BeamProfiler.llb'
5. Run the VI titled 'Profiler.vi'



# Chapter 3

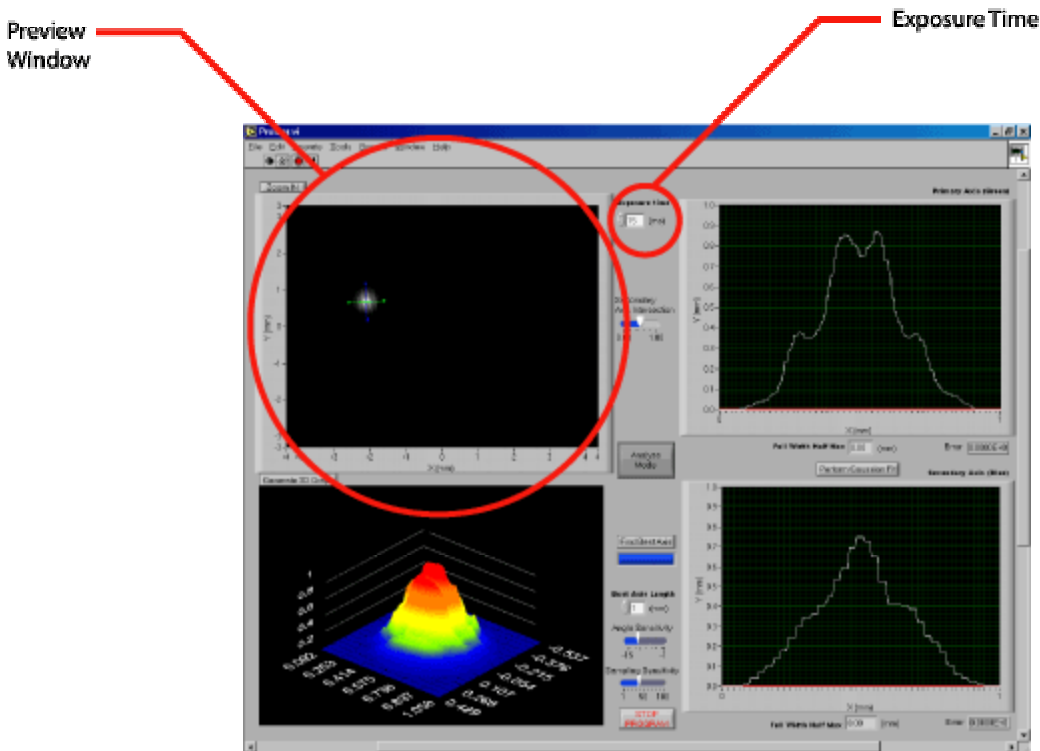
## Using the Spatial Profiler

### Preview Mode

When the program is first run it will be in Preview Mode. In Preview Mode the camera is essentially a video camera continually updating the intensity image in the top left corner of the screen.

This mode allows the user to align the laser beam and set the exposure time on the camera from anywhere between 1 millisecond to 10 seconds.

Once the image in the top left corner is acceptable and the image is ready for analysis, switch the program into Analyze Mode by pressing the 'Analyze Mode' button in the center of the screen.



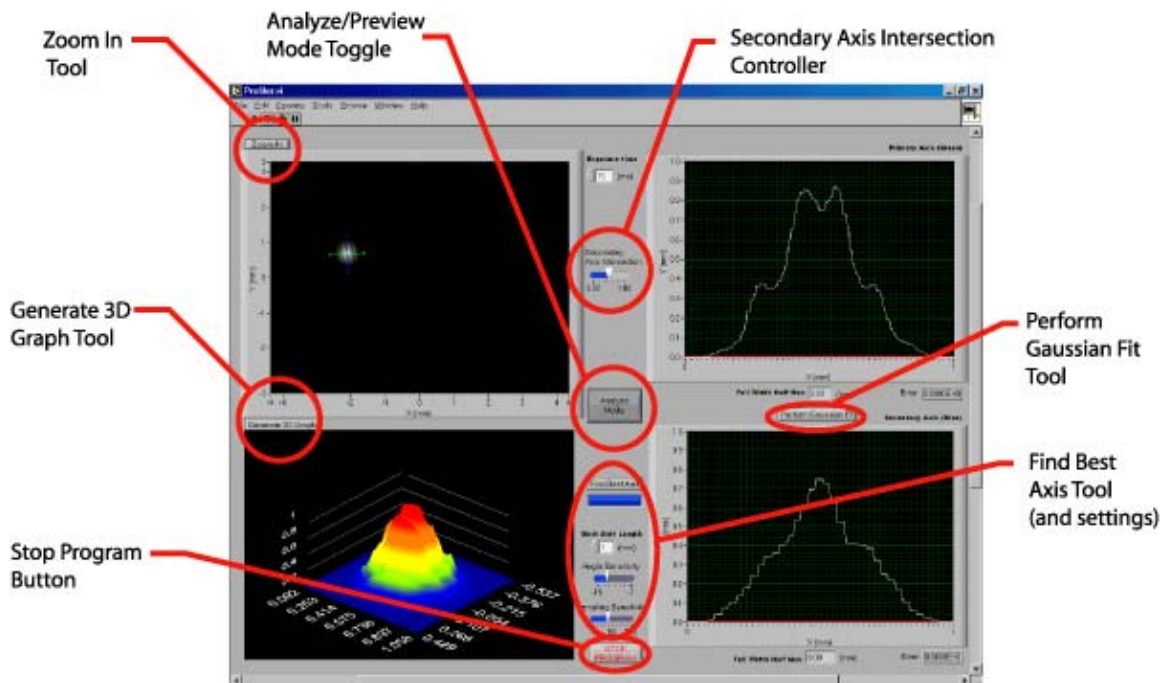
**None of the analysis functions will operate in Preview Mode. Preview Mode is meant only to allow the user to setup the laser alignment and camera exposure time.**

## Analyze Mode

Once in Analyze Mode, the program will stop receiving images from the EDC-1000HR and the user is then able to perform a number of numerical and graphical calculations to characterize the spatial profile of the laser beam.

In Analyze Mode the user can:

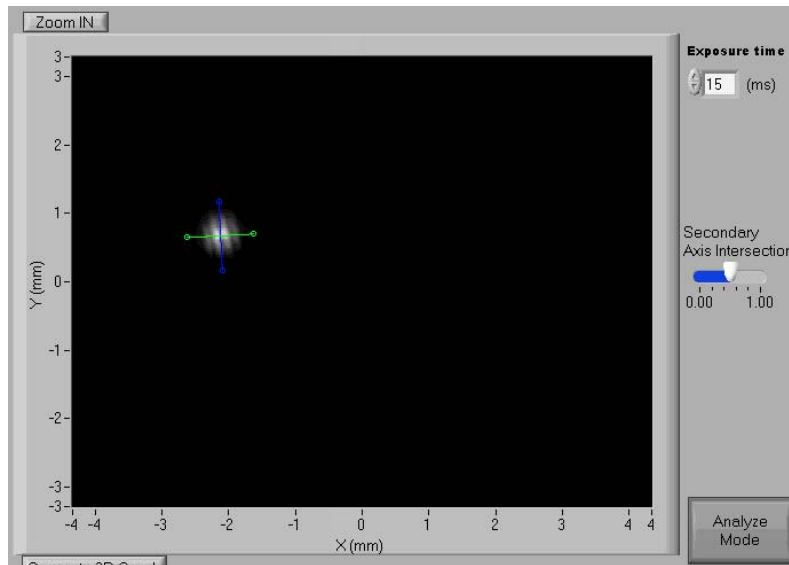
- Locate the primary and secondary axis
- Generate a 3D intensity graph
- Zoom in on the 2D intensity image
- Generate cross-sections of the intensity profile
- Generate a best-fit Gaussian curve
- Find the full-width half-maximum (FWHM) of the beam



**In Analyze Mode the cross-section graphs will be automatically updated any time that the primary and secondary axis are moved. However all other functions will only update when prompted by the user.**

## Generating Primary and Secondary Axis

All calculations in Analyze Mode are focused on an area of interest defined by the primary and secondary axis. So the first important step is to set the primary axis endpoints. The primary axis and secondary axis are green and blue respectively. The secondary axis is automatically generated perpendicular to the primary axis.



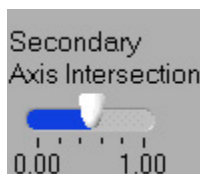
**Primary and Secondary Axis must be set first.**  
All analysis tools use the primary and secondary axis to define their areas of interest.

### User Defined Axis Generation

The simplest way to define the axis is to click and drag the endpoints of the primary axis (green) to the desired locations.

The program will then automatically generate the intensity cross-section between those two points. The program will also automatically generate a secondary axis of equal length perpendicular to the primary axis.

The secondary axis by default intersects the primary axis at its midpoint. However, the user can slide the intersection point along the primary axis using the 'Secondary Axis Intersection' controller.



## **Automatic Axis Generation** 'Find Best Axis' Tool

The program can automatically generate primary and secondary axis when 'Find Best Axis' is pressed.

To use this tool simply press the 'Find Best Axis' button. The progress bar below the button indicates the time remaining to finish the analysis.

The 'Find Best Axis' tool will first find the center of the beam using a 'center of mass' calculation. The program will then try a variable number of angles of rotation for the axis and set the primary axis at the rotation angle that generates the densest cross-section intensity.



There are three settings that the user should set before running Automatic Axis Generation. The user can specify the width of the desired axis (in millimeters), the number of rotation angles probed, and the sampling density of each test angle. The last two settings are controlled with sliders described below:

### **Angle Sensitivity Slide**

This slide control corresponds to the number of rotation angles that the program should probe. Moving the slider to the right will increase the sensitivity and vice versa.

$$\text{Number of angles probed} = \frac{-180}{\text{SensitivitySlider}}$$

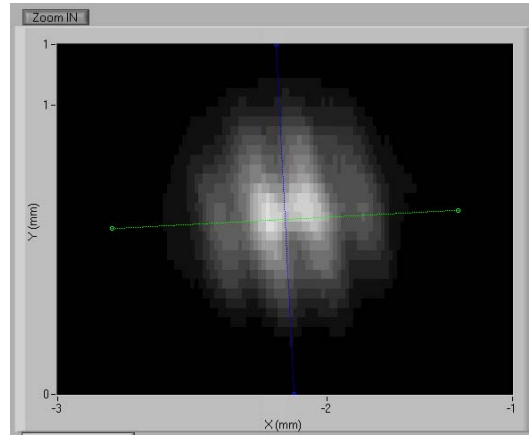
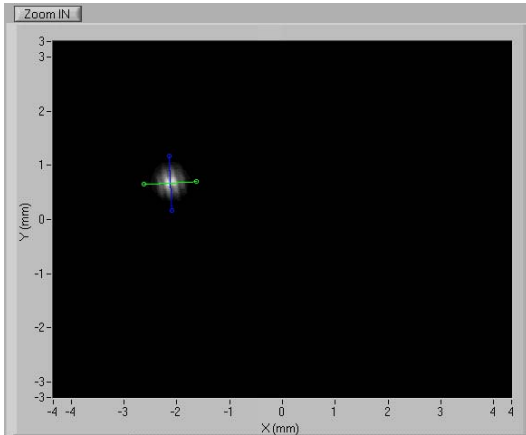
### **Sampling Sensitivity Slide**

The 'Sampling Sensitivity' slide indicates the number of points the program will sample on each angle probed. Increasing the number will increase the sensitivity and vice versa.



## Zoom In / Zoom Out

The 'Zoom In' tool is a toggle switch that, when pressed, zooms in on the area of interest (defined by the primary and secondary axis) and shows the camera's entire intensity array when un-pressed.

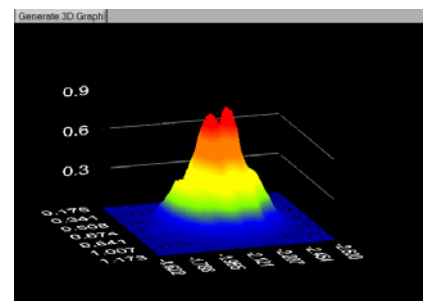
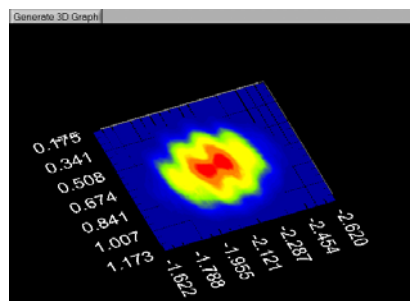
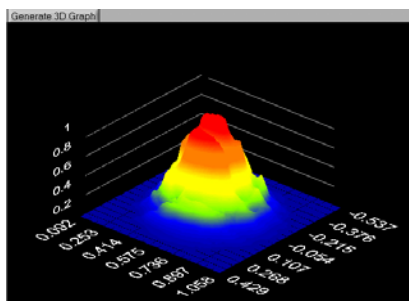


## Generate 3D Intensity Graph

The image on the bottom left corner of the program is an intensity graph representing the intensity of the beam as a function of its spatial coordinates (in millimeters). The 3D graph is automatically zoomed in on the area of interest specified by the primary and secondary axis.

The 3D graph will not automatically update when new primary and secondary axis are selected so the user must press the 'Generate 3D Graph' button whenever an update is desired.

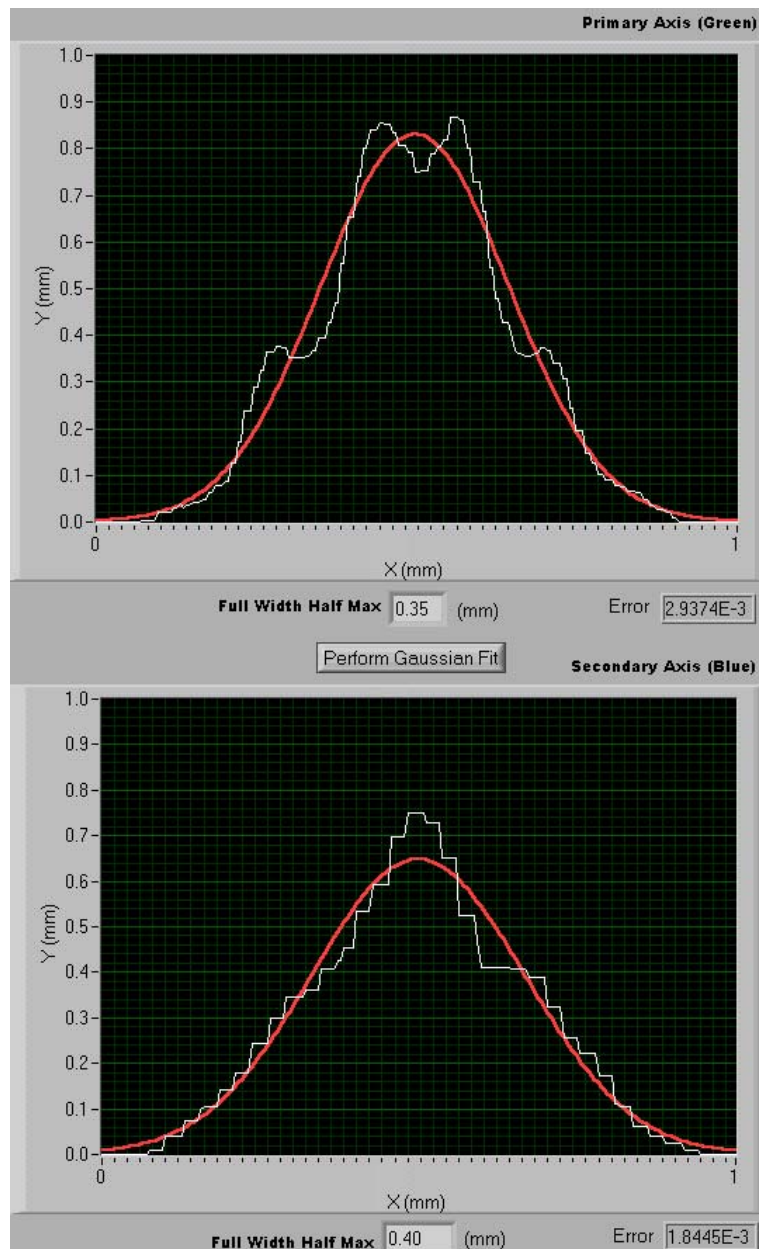
The 3D Graph can be swiveled in any direction by simply clicking and grabbing the 3D image with the cursor.



## Generate a Gaussian Best Fit

The 'Generate Gaussian Best Fit' tool will generate a Gaussian curve fit for both the primary and secondary axis, indicate the full width half maximum for each axis, and indicate the amount of error in the curve fits.

The Gaussian curves are drawn in red on top of the actually intensity cross-sections which are drawn in white. The Gaussian fits are not automatically updated so the user must press the 'Perform Gaussian Fit' button whenever an update is desired.



# Chapter 4

## Coding and Algorithms Explanation

### Overview

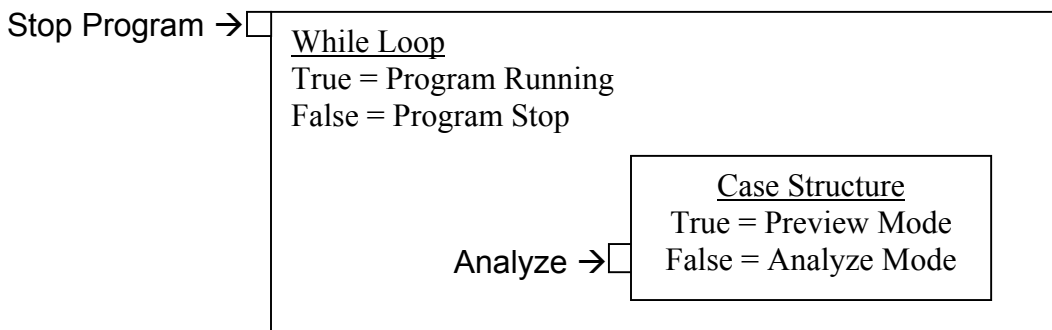
This program is written entirely in LabVIEW. The code uses some advanced analysis functions and requires that the user have the full version of LabVIEW to function properly.

The majority of the code is written inside an all encompassing while loop. The while loop is controlled by the 'Stop Program' button on the front panel. When 'Stop Program' is pressed, the while loop will terminate and the program will stop running.

Outside of the main while loop are a couple of variables that get initialized whenever the program starts. They are:

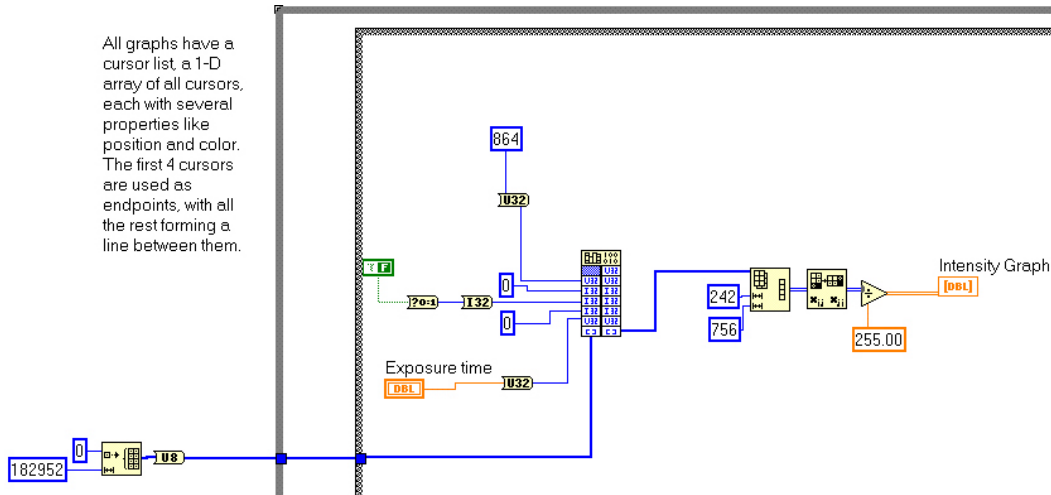
1. Number of Horizontal Pixels
2. Number of Vertical Pixels
3. Half Height (Half of the physical dimension of the ccd detector in mm)
4. Half Width (Half of the physical dimension of the ccd detector in mm)
5. A color table for the 2D intensity graph (for a black and white image)

Inside the main while loop is a case structure that allows the user to toggle between 'Preview Mode' and 'Analyze Mode'.



## Preview Mode

The code for preview mode is located on the “True” side of the main case structure. This is the code that calls the .DLL file, receives the images from the EDC-1000HR and updates the preview image on the front panel.



The input parameters for the HRCAM32.DLL are on the left side of the function. They are: hrcam32 (unsigned int,int,int,int,unsigned int,pixel\*)

1. The card's base address (in Hex is 360h)
2. Anti-Blooming Flag (not 0 for anti-blooming)
3. Interlace Mode (0 for no interlace)
4. Field Flag (0 for no interlace)
5. Exposure time (in milliseconds)
6. An empty array (buffer pointer must be 756\*242)

The output is a one-dimensional array of intensity values that must be reshaped into a 2D array that is 756 pixels wide and 242 pixels high. Intensity values initially range from 0 to 255 but they are converted to a scale from 0 to 1.

## Analyze Mode

The code for Analyze Mode is located on the “False” side of the case structure. The functions in this mode are described in this manual from the top down.

The first couple of code clusters located at the top of the Analyze Mode case structure are the axis controllers and the primary and secondary axis intensity graph generators. They run continually whenever the program is in Analyze Mode. Below them are a number of case structures which each run different analysis tools.

## Primary and Secondary Axis Cursors Concept

All graphs in LabVIEW have a cursor list, a 1-D array of clusters, each with several properties like position and color. In this specific program, a series of cursors are used to describe the primary and secondary axis on the intensity graph in the top left corner.

The cursors are stored in a 1-D array of clusters in which the first four clusters describe the 2 primary axis endpoint cursors and the 2 secondary axis endpoint cursors respectively. The rest of the array describes the cursors that make up the lines between the endpoints. The number of cursors between the two endpoints is a variable ('Sample Points in Crossection'), which is defaulted to be 200:

Intensity Graph



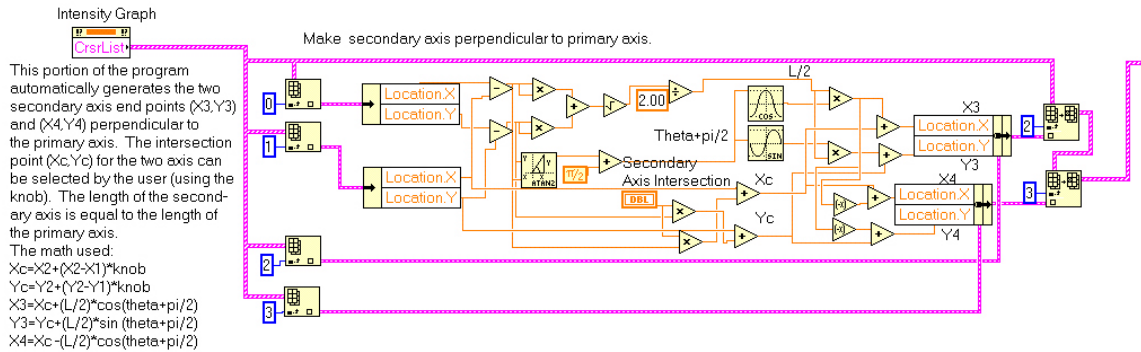
Index Number	Description
0	First Primary axis endpoint
1	Second Primary axis endpoint
2	First Secondary Axis endpoint
3	Second Secondary axis endpoint
4 thru 203	Points between Primary Axis endpoints
204 thru 403	Points between Secondary Axis endpoints

The first two cursor clusters (i.e. the endpoints for the primary axis) dictate all of the rest of the cursors in Cursor List (CrsrLst) because the secondary axis is automatically perpendicular to the primary axis and the rest of the cursors just fill in the lines between the endpoints.

**The first step this program takes once in Analyze Mode is to define all axis cursors using only the first 2 cursors.**

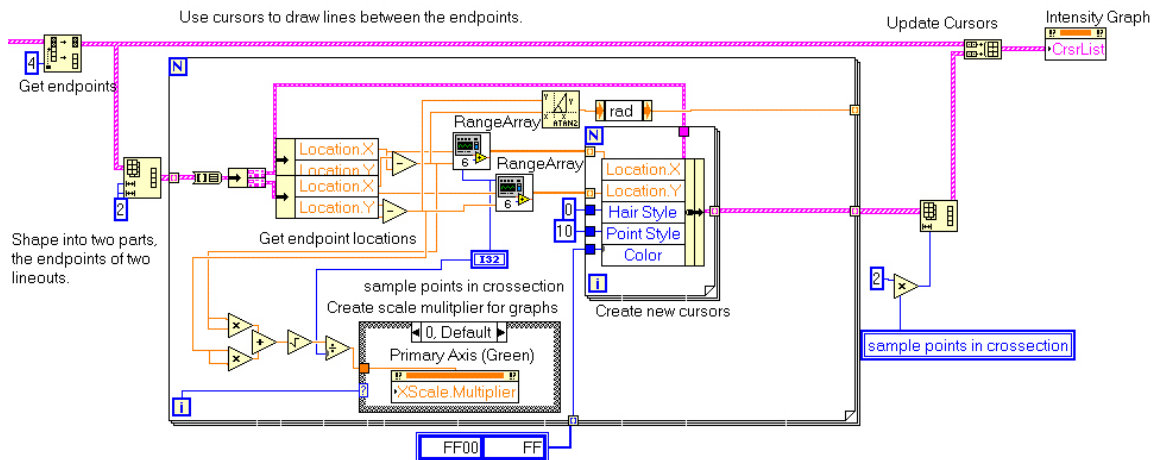
## Make Secondary Axis Perpendicular to Primary Axis

The following code generates the secondary axis endpoints perpendicular to the primary axis endpoints:



## Use the 4 endpoints to draw lines

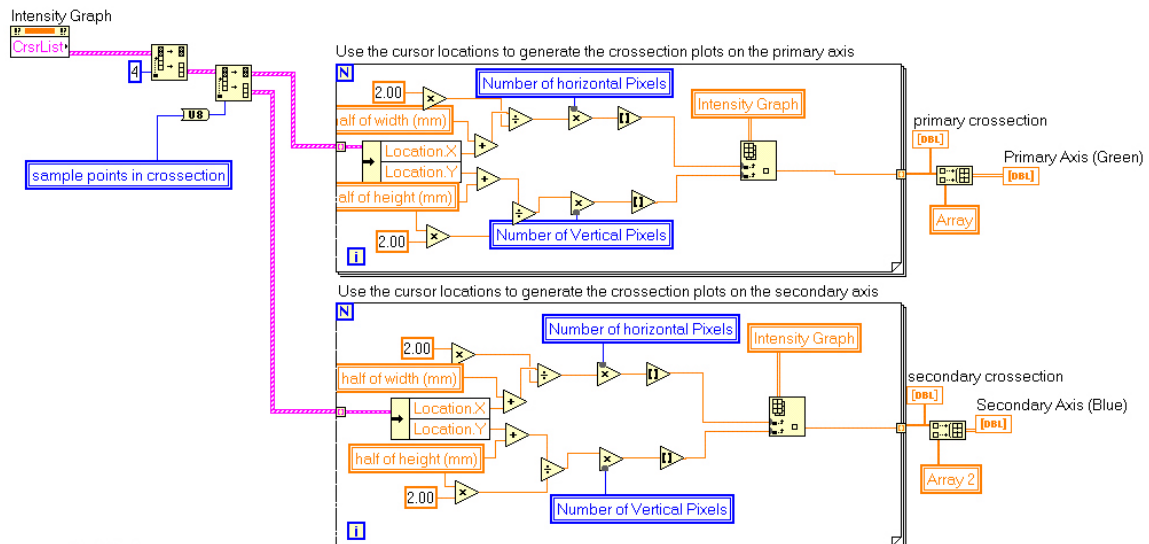
The following code generates the line (series of cursors) between the primary and secondary axis endpoints. The code also uses the endpoints to create a scale multiplier for the cross-section graphs:



This process uses a separate VI called RangeArray.vi. 'RangeArray' simply generates an array of numbers given a starting number, an ending number, and the number of points in between.

## Generate Primary and Secondary Axis Cross-Sections

The following code generates the primary and secondary axis intensity cross-sections:



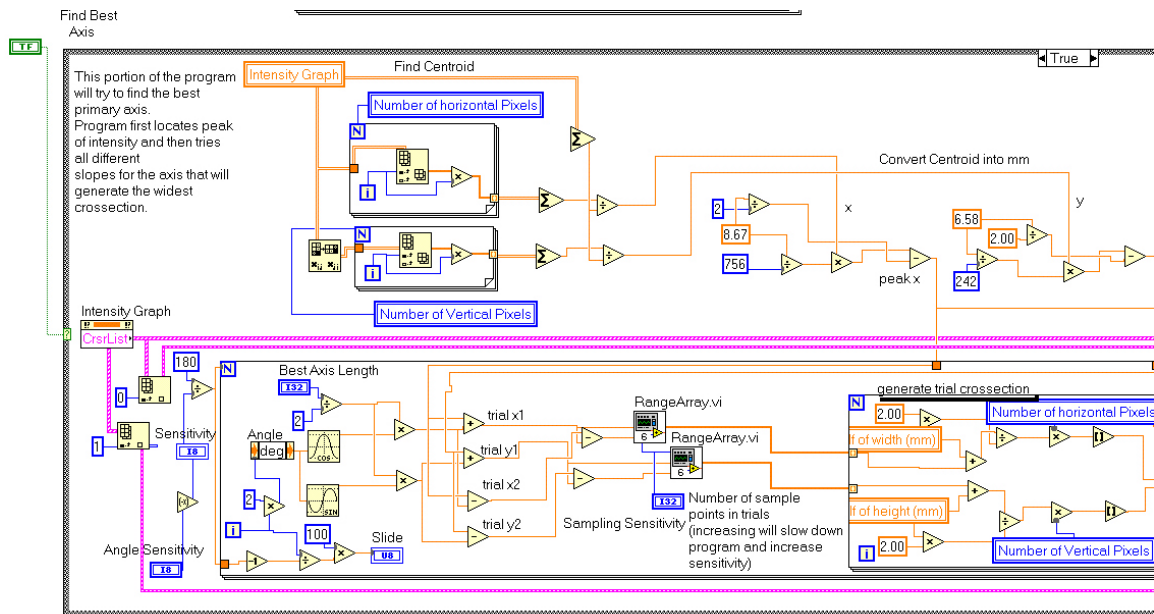
The code works by correlating each cursor location in the Cursor List (CrsrLst) with an intensity pixel from the 'Intensity Graph.' The positions of the cursors are stored in units of millimeters while the intensity array is stored in units of pixels. So this code simply converts each cursor location to a pixel value and then creates a 1-D intensity array to define the cross-section intensity.

## Find Best Axis Tool

The find best axis tool is a case structure with a true/false controller. When the user presses the 'Find Best Axis' button, the case structure switches to true for one iteration and then returns to false. Thus, the code will only run once when pressed.

The code first locates the center of the beam using a 'center of mass' method:

$$X_{center} = \frac{\sum A_{xy} * x}{\sum A_{xy}} \quad Y_{center} = \frac{\sum A_{xy} * y}{\sum A_{xy}}; \quad A_{xy} \text{ is the intensity at pixel (x,y)}$$

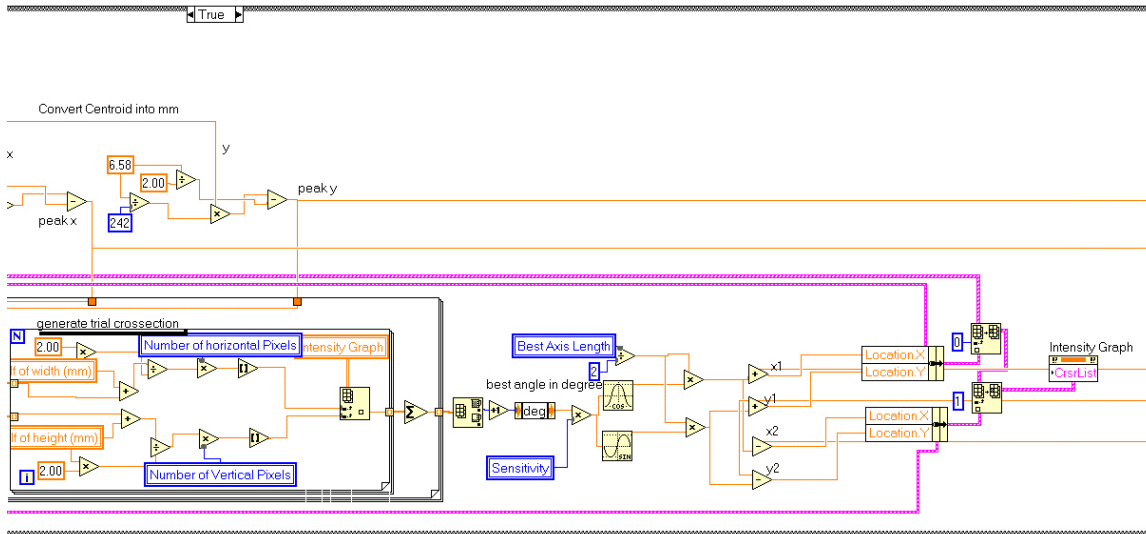


The 'Find Best Axis' code then generates a trial primary and secondary axis for a variable number of angles of rotation. The trial axis are always centered at Xcenter and Ycenter with their lengths controlled by the 'Best Axis Length' variable.

Note that this code is very similar to the 'Generate Primary and Secondary Axis Cross-Sections' code. The only difference is that the code is placed inside a 'For Loop' and each iteration uses a slightly different rotation angle. Then the intensities for each trial angle are summed up and placed in an array.



## Find Best Axis Case structure continued...

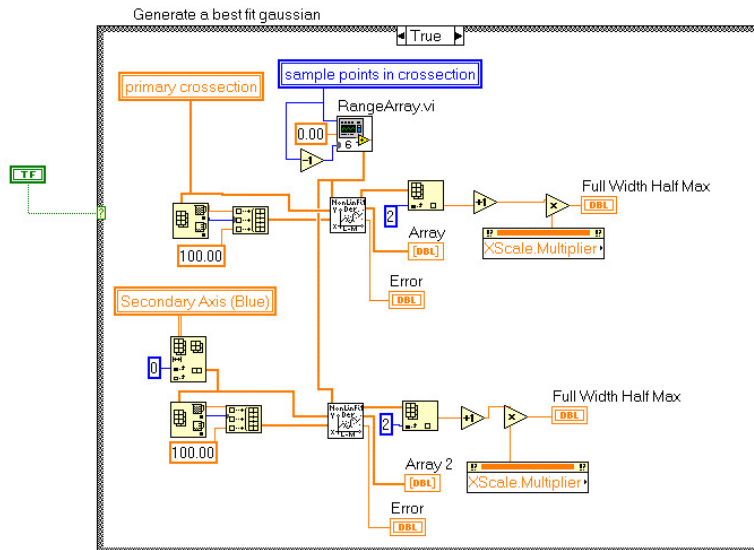


The loop's output array is then analyzed and the index of the maximum intensity summation is determined to be the ideal rotation angle.

Finally, the first two cursors' locations (which correspond to the primary axis endpoints) are replaced with the newly found ideal endpoint locations.

## Generate Best Fit Gaussian Tool

The Generate Best-Fit Gaussian tool is a case structure with a true/false controller. When the user presses the 'Perform Gaussian Fit' button, the case structure switches to true for one iteration. Thus, the code will only run once when pressed.



This portion of code uses a modified version of the Lev-Mar curve-fitting tool (which comes with LabVIEW). The original Lev-Mar VI attempts to fit an

exponential curve. However, the target equation for the Lev-Mar VI was modified for this program to specifically generate a Gaussian curve. The modified VI is called JBC\_LevMar.VI and is saved in 'BeamProfiler.llb.'

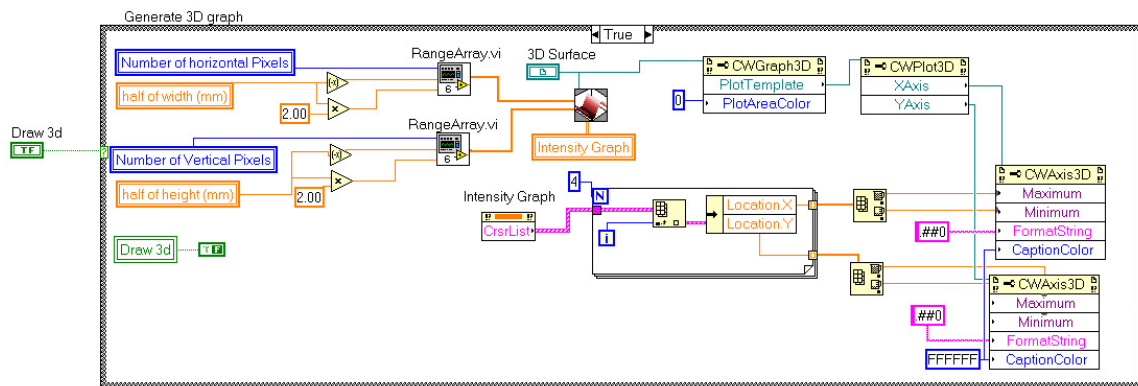
The inputs to the JBC\_Lev-Mar curve fit VI are:

1. X values for input graph (array)
2. Y values for input graph (array)
3. Initial guesses array for parameters (PeakValue, PeakLocation, FWHM)

For more information on the Lev-Mar VI, see the help index in LabVIEW.

### Generate 3-D Intensity Graph

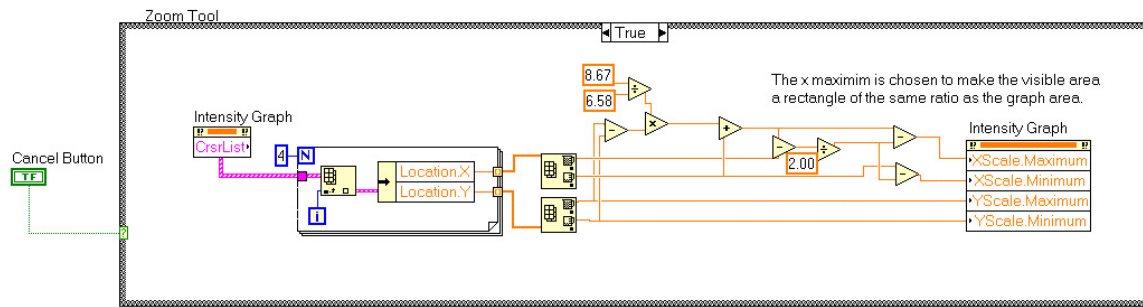
The Generate 3-D Intensity Graph tool is a case structure with a true/false controller. When the user presses the 'Generate 3D graph' button the case structure switches to true for one iteration. Thus, the code will only run once when pressed.



This code is a straightforward use of the 3D surface tool built into LabVIEW. The code at the bottom right of the case structure defines the axis maximum and minimum based on the endpoints of the primary and secondary axis.

## Zoom on Intensity Graph

Zooming in on a graph means simply changing the maximum and minimum values for the axis. The following code will set the maximum and minimum y-axis values to be the maximum and minimum y locations of the endpoints of the primary axis. The x-axis limits are then set to maintain the correct aspect ratio for the preview window.



The initial 'For loop' makes an array of all four x locations and all four y locations for the primary and secondary axis endpoints. Then, the largest and smallest x and y values are used to define the new limits on the intensity graph.

When the case structure is switched to false, the intensity graph limits are returned to the initial default settings.

## Chapter 5

# EDC-1000HR Camera Specifications

Image size (Sensing area)	8.67x6.58 mm
Pixel size (Non-interlaced) (Interlaced)	11.5x27 microns 11.5x13.5
Spatial Resolution (Non-interlaced) (Interlaced)	753 (H)x244 (V) 753 (H)x488 (V)
Non-uniformity	15% maximum over central 92% of image
Peak Quantum Efficiency	70% @ 0.68 microns
Saturation Signal (Single Scan)	200,000 electrons
R.m.s. noise	140 electrons/pixel
Exposure time	1 msec to 10secs