Document Title:	**Bringing the Dispatcher to the Scene With Panoramic Imaging and Remote Video Transmission, Final Report**
Author(s):	**Andrew Hohmann**
Document No.:	**190132**
Date Received:	**10/10/2001**
Award Number:	**1999-IJ-CX-K020**

ISI-TM01073001.150 190132 c.1

Bringing the Dispatcher to the Scene with Panoramic Imaging and Remote Video Transmission

Final Report
Contract #: 1999-IJ-CX-K020

Submitted By:

Andrew Hohmann

InterScience, Inc
105 Jordan Road
Troy, NY 12180

July 30, 2001

Prepared for:

U. S. Department of Justice
Office of Justice Programs
810 Seventh Street, N.W.
Washington, DC 20531

Table of Contents

Restatement of Phase I Specific Aims

Throughout the world, police agencies are equipping their vehicles with video recording systems to assist their agencies in the apprehension of criminals and in their prosecution. These video systems also play an important role in the protection of the officers by acting as a deterrent to violence directed towards the officer. Despite the contributions the video systems have made to law enforcement, a major drawback of the present-day system is that the majority of the systems only record the activity in front of the vehicle. To remove the forward only viewing capabilities, several systems are being developed utilizing multiple cameras to view all sides of the vehicle, or a pan and tilt mechanism either operated manually by an officer inside the vehicle or programmed to follow the officer outside the vehicle. The use of multiple cameras increases the overall cost of the system, and the pan and tilt systems also add cost to the overall system and only record in the direction the camera is pointed.

The development of a panoramic imager, consisting of a 360-degree mirror in a clear housing with a color camera aimed at the mirror, was proposed as a Phase I improvement to the current video camera systems used in law enforcement. Development of the proposed 360-degree panoramic imager would significantly enhance the viewing capabilities of today's law enforcement officers. The Phase I effort was aimed at demonstrating the feasibility of the 360-degree panoramic imager as applied to mounting on a police car. The overall goal of the development was to improve the viewing capabilities of officers in a police car, thereby reducing the risks that police officers face in the field.

The overall objective of the Phase I effort was to demonstrate the enhanced capabilities a 360-degree panoramic imager can provide and evaluate how this tool can be used in the field. The Phase I work concentrated on the development of a prototype 360-degree panoramic imager, the integration of the imager into a police car, developing the image processing software, and testing the resulting prototype. The specific objectives for the Phase I work were:

1. Determine the optimal viewing geometry for the panoramic imager.

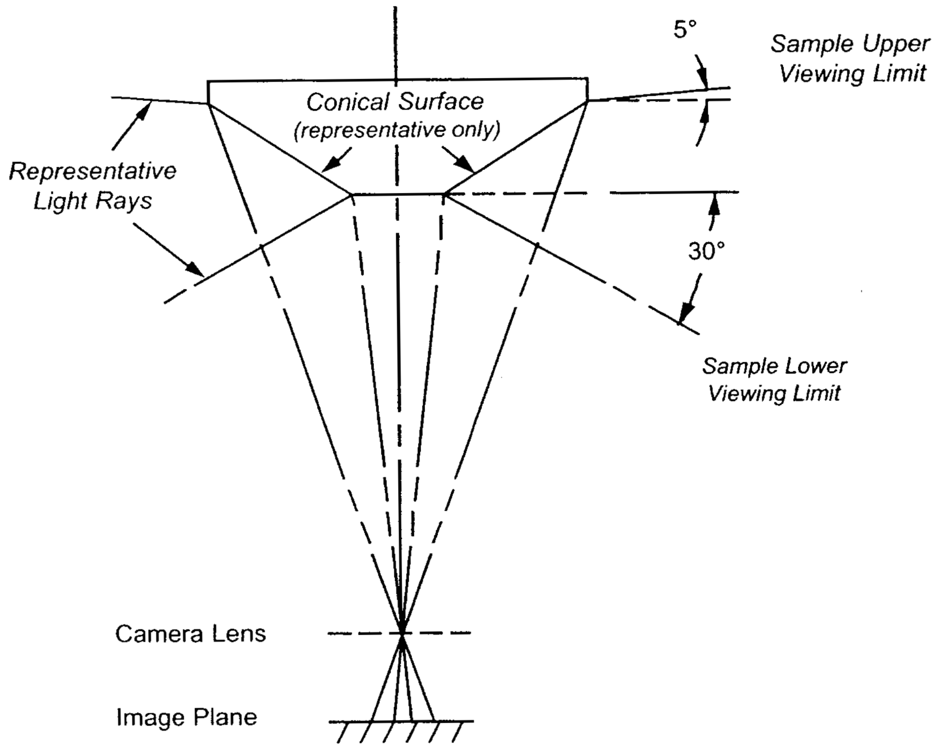
2. Design, develop and fabricate the prototype 360-degree panoramic imager and develop the associated image processing software.
3. Demonstrate the capabilities of the 360-degree panoramic imager through bench and road testing.

As presented below, these specific aims of the Phase I work were successfully accomplished. The final report details the work performed during the Phase I grant. In Phase I, we successfully designed and developed a prototype 360-degree panoramic imager and associated digital image remapping software. The capabilities of the prototype system were demonstrated in a series of optical tests in the laboratory and in a vehicle on the road.

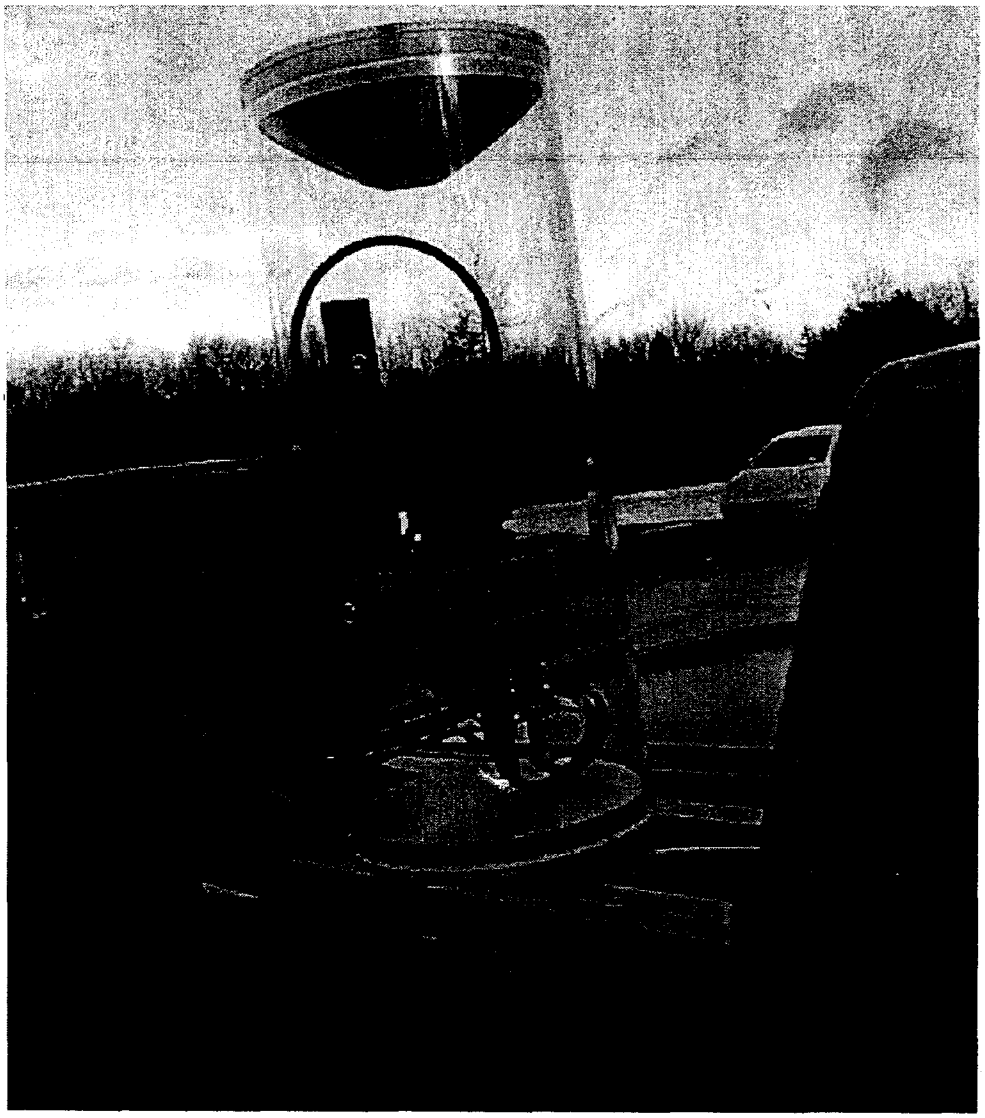
Phase I Work Results

Summary of Panoramic Imager Design

The panoramic imager consists of four main parts: the parabolic mirror, video camera, housing, and associated software. The mirror is a custom-designed, CNC machined block of aluminum.



The cone's surface is the mirror, and the image plane is where the camera's CCD rests. The mirror fits on top of the Lexan housing, and the camera is attached via an optical rail, which is bolted onto the housing. Below is a picture showing the completed panoramic imager.



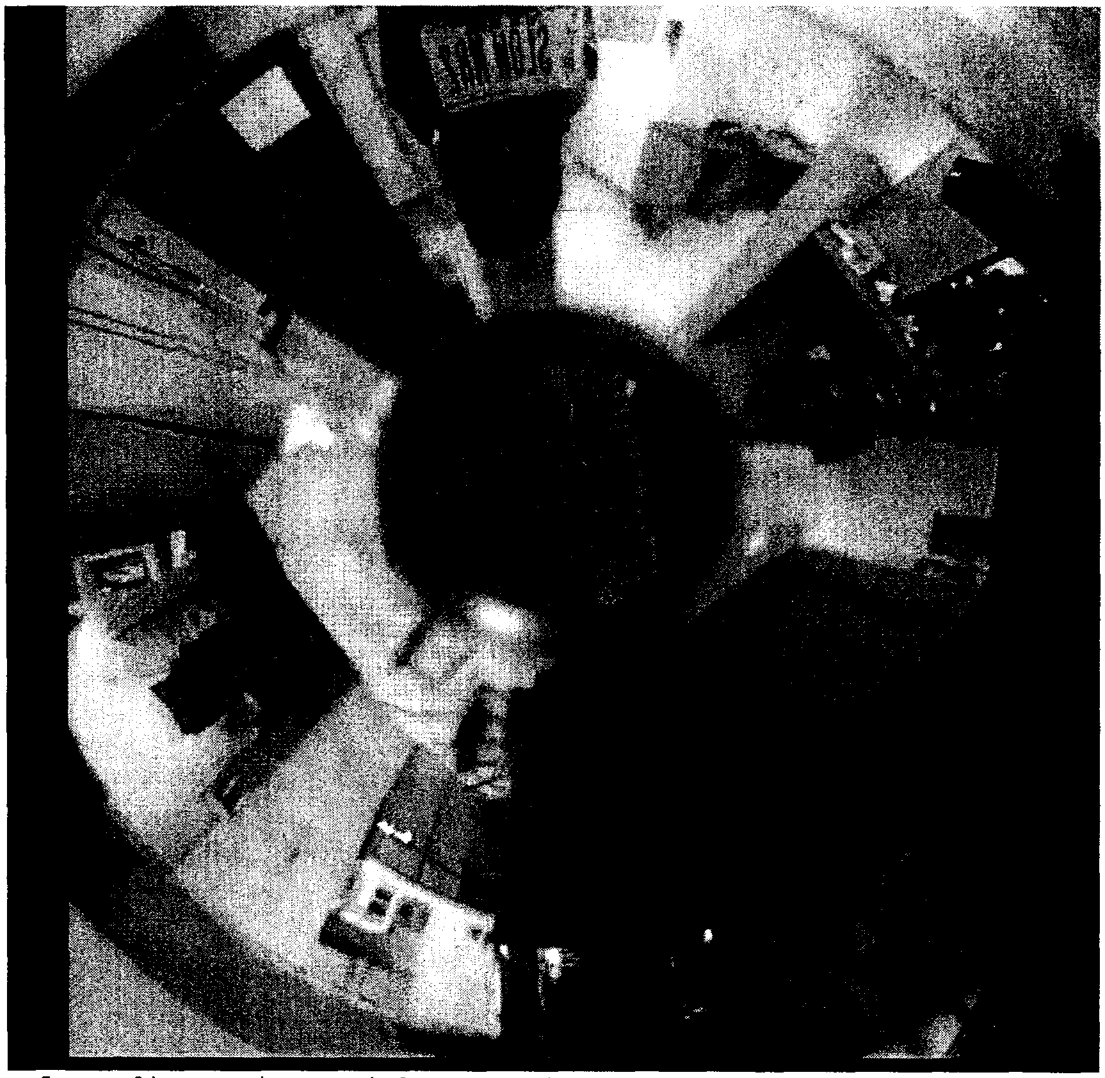
Housing

The mirror and camera are housed in a 1/8″ thick Lexan polycarbonate tube, 4¼″ in diameter and 11″ long. Lexan polycarbonate offers the highest impact strength of any transparent glazing product-250 times the impact strength of glass and 30 times that of acrylic. Lexan polycarbonate has abrasion resistance and clarity rivaling glass. This minimizes the risk of scratches and breakage. The base plate of the tube is a ½" thick machined aluminum piece, which snugly fits into the bottom of the Lexan tube. The mirror seals into the top of the tube, creating a weatherproof environment for the camera as shown in the above photo.

The camera is mounted on an optical rail, which is bolted to the side of the Lexan housing. Wires from the camera are run through a ½" hole in the side of the Lexan housing. This hole can be sealed with hot glue or a similar substance to make the housing weatherproof.

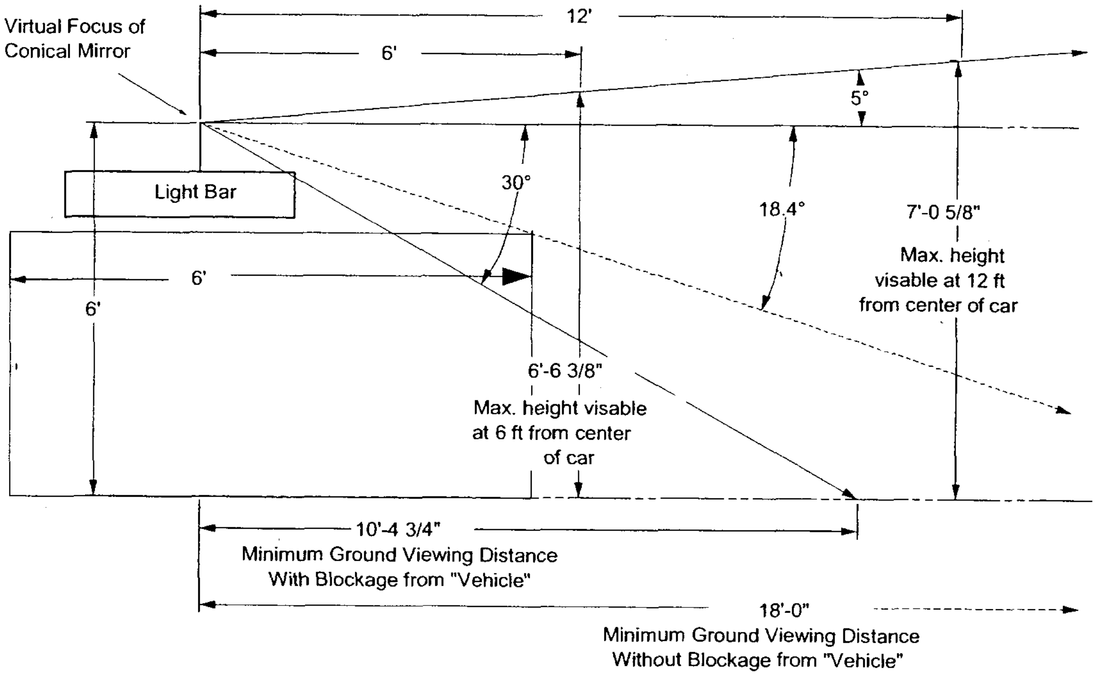
Mirror

The mirror was fabricated from a block of aluminum, on a CNC machine. The mirror was then silver plated for a better surface finish. Due to the plating process, the mirrors became pitted. We proceeded to polish the mirrors using various abrasives and polishing compounds. The polishing of the mirrors did slightly improve the image quality, but did give the mirrors a warping effect. This warping effect can be seen in the picture below.



After discussions with our subcontractor, we purchased another mirror, however this new mirror was not plated. By keeping the aluminum surface we could evenly smooth the surface to our desired specifications.

The viewing angle for the panoramic mirror is approximately -30 degrees, +5 degrees. The following diagram shows the viewing heights at different distances from the panoramic imager.

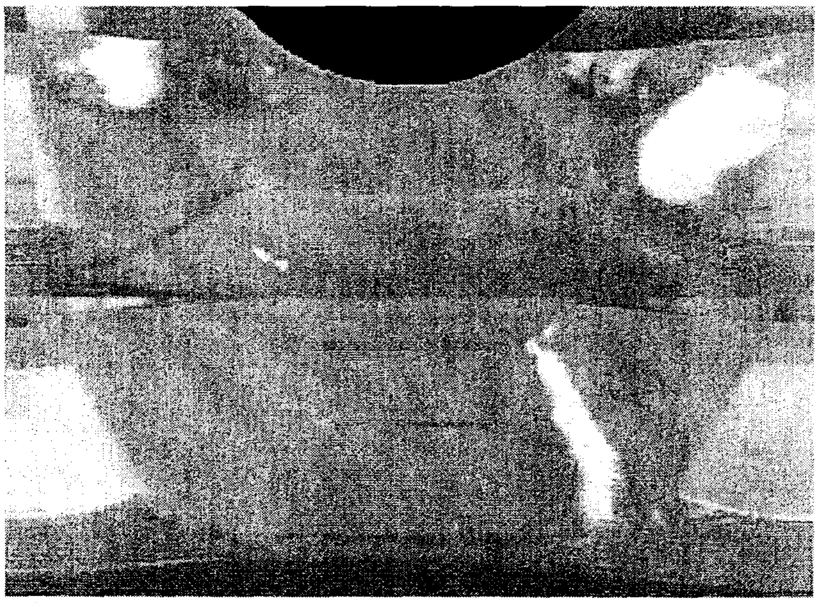


Camera

The camera we chose for this project was the Kowa PX-70KST camera. This camera outputs standard NTSC video through either an S-video output or a BNC output. The camera is extremely small: less than 2″ x 2″ x 2″. Additional information about this camera can be found in Appendix A. The lens used was the Kowa LM8PB.

Software

The software for this project performs three main functions: display the live video feed from the camera; save the video to a movie; and remap the live video feed. Displaying the video feed and saving the video to a movie is fairly standard, so the tricky part was getting the remapping to work correctly. In order to perform the remapping algorithm efficiently, we pre-computed the location of each pixel. This allowed us to significantly reduce the amount of processing needed. Following is a picture showing the processed image.



There are several ways to remap the image, each with their own tradeoffs. One way is eliminate the black center circle (some of which is shown at the top of the image), however this reduces your top view, especially in the top corners. After road and bench testing we concluded that the current remapping algorithm, although it leaves a small circle in the center of the image, works best. A detailed analysis of the software can be found in Appendix B.

Optical Tests

Several optical tests were performed, in both the lab and on the road. In the lab we took snapshots of each mirror at 3', 6', and 10' with both the Kowa NTSC camera and a high-resolution digital camera. Video was also taken in the lab, both remapped and non-remapped video from each mirror. On the road we took video from each mirror, in both remapped and non-remapped formats. Several videos were also taken with the camera-rotated 90 degrees. All of the pictures and videos can be found on the accompanying compact disks. As can be seen, both the warped and pitted mirrors introduced enough distortion to make the license plate significantly more difficult to read than the original mirror from the Air Force 360-degree project. However, our new mirror solved these problems and allowed us to achieve the clarity of the Air Force mirror, but with the new viewing angles.

In our project we found that the main factor limiting resolution was the camera. Due to our large viewing angle, each pixel in the camera corresponds to a large area, making small details difficult to see.

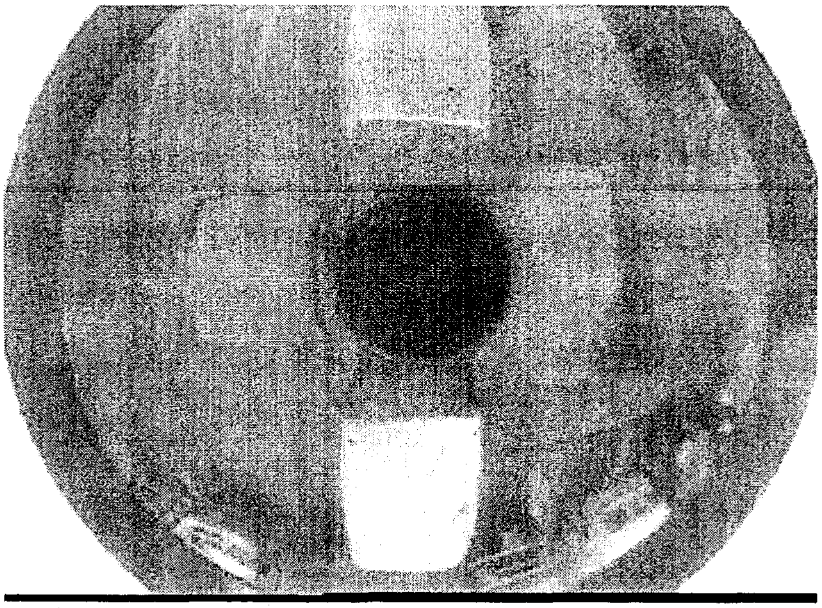
This complicates the issue, because law enforcement needs a relatively inexpensive panoramic imager. The least expensive solution is to use NTSC video, since law enforcement already has the equipment to record and broadcast NTSC video. At the moment, high-resolution digital cameras are expensive. Fortunately technology improves quite rapidly, and a new camera coming to the market should allow a law enforcement ready application. This camera is the Silicon Imaging MegaCamera, with 3.17 million pixels. The MegaCamera's resolution is over ten times NTSC resolution, and will enable viewing of license plates and other small objects from a distance of fifty feet away or more. Most importantly, the MegaCamera represents a breakthrough in CMOS technology, which means that the camera will cost approximately $500 for the entire system. By having a high-resolution camera and an image processing box, it will be possible to store 170 hours of high-resolution video. This will allow us to overcome the current problem, which is that license plates are unreadable from extended distances. Specifications for the Silicon Imaging MegaCamera can be found in Appendix C.

Transmitting video back to the dispatcher remains a problem, as wireless transmission technology still needs time to mature. Within five to ten years wireless technology should be standardized and mature enough to allow high resolution, interactive video from a remote scene to a dispatcher. A prototype demonstration of remote wireless interaction can be done right now, but the hardware necessary to accomplish the task is too expensive for law enforcement at the moment. Also, most of the long-range wireless systems employ proprietary standards, making them unsuitable for broad distribution in law enforcement.

Analysis of Results and Conclusion

Our first goal was to determine the optimal viewing geometry for the panoramic imager. Road tests showed that we could view surrounding cars and people within approximately twenty feet of the vehicle. However, in the sample picture below, notice that a large percentage of the image consists of the car and the lightbar. By removing most of the automobile and lightbar, we can significantly increase the resolution. Also, the view of the horizon should probably be expanded. This would require a non-conical mirror. By optimizing the viewing field of the panoramic imager the black center circle can be reduced, if not eliminated.

There are drawbacks to using a non-conical mirror, however. The software algorithm to remap the image would be quite complicated. Also, depending on the mirror design, some areas of the image would have greater



resolution than other areas. Thirdly, with a conical imager, the raw video is quite understandable, and easy to make sense of. With a non-conical imager, the raw video would be very difficult to view and understand. Our original design turned out to be a good compromise for the first prototype, and demonstrates some of the complications that 360-degree viewing may entail for law enforcement.

The second goal was to design, develop and manufacture the prototype 360-degree panoramic imager and develop the associated remapping software. This goal was successfully completed.

The third goal was to demonstrate the capabilities of the 360-degree panoramic imager through bench and road testing. This goal was successfully completed, with more than an hour of video recordings accumulated, and numerous still photos. Due to our subcontractor, the final mirror was delivered approximately one month after the project was officially completed. Therefore we were unable to road test the final mirror, but bench testing showed that it has the same quality as the original Air Force mirror, but with the field of view and viewing characteristics of the other mirrors.

The capabilities of the 360-degree panoramic imager are substantial, and this prototype proves the concept can work well in law enforcement applications. The panoramic imager successfully demonstrated the ability to view the entire surroundings simultaneously, and that mounting it in a vehicle is easily achievable. This project generated a good understanding of how a panoramic imager could be used in law enforcement, the modifications that need to be made in order to build a second, police car ready prototype, and the technologies that need to mature for large scale application of the 360-degree panoramic imager in law enforcement.

Appendix A

Kowa Camera Specifications

Kowa Optimed, Inc. *20001 S. Vermont Ave. Torrance, CA 90502 • 1-800-966-5692*
Levitch@kowa.com • Jonathan@kowa.com

SPECIFICATIONS

1/3" IR COLOR CAMERA

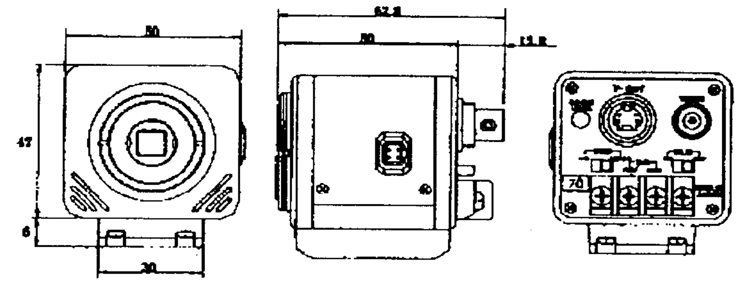
PX-70KST/KLT(NTSC)
PX-75KST/KLT(PAL)

MODEL	PX-70K(NTSC)	PX-75K(PAL)
PICK-UP DEVICE	1/3 INCH IR CCD IMAGE SENSOR	
[illegible]ERALL ELEMENTS	811(H)×508(V)	786(H)×586(V)
[illegible]FFECTIVE ELEMENTS	768(H)×494(V)	752(H)×582(V)
CELL SIZE	8.36 μm(H)×7.40 μm(V)	6.50 μm(H)×6.25 μm(V)
LENS MOUNT	CS MOUNT	
SYNC SYSTEM	INTERNAL CRYSTAL CONTROL	
SCANNING SYSTEM	2:1 INTERLACE	
VIDEO OUT	1Vp-p 75Ω	
H RESOLUTION	MORE THAN 480TV LINES	
MINIMUM SCENE ILLUMINATION	0.3LUX F1.2(AGC:HI-ON);1LUX F1.2(AGC:ON)	
S/N RATIO	50dB UP (AGC OFF)	
SHUTTER SPEED	SELECTABLE NTSC:1/100,1/250,1/500,1/1000,1/2000,1/4000,1/10000 PAL:1/120,1/250,1/500,1/1000,1/2000,1/4000,1/10000	
GAMMA	γ=0.45, γ=1	
AGC	ON/OFF/HI/LOW SELECTABLE	
CAMERA MODE CONTROL	RS-232C CONTROL	
POWER SOURCE	DC+12V±10% 180mA MAX	
STORAGE TEMPARATURE	-30℃~+70℃	
OPERATING TEMPARATURE	-10℃~+40℃	
DIMENSIONS	S 50mm×47mm×50mm/L 50mm×47mm×86mm	
WEIGHT	S 105g/L 130g	

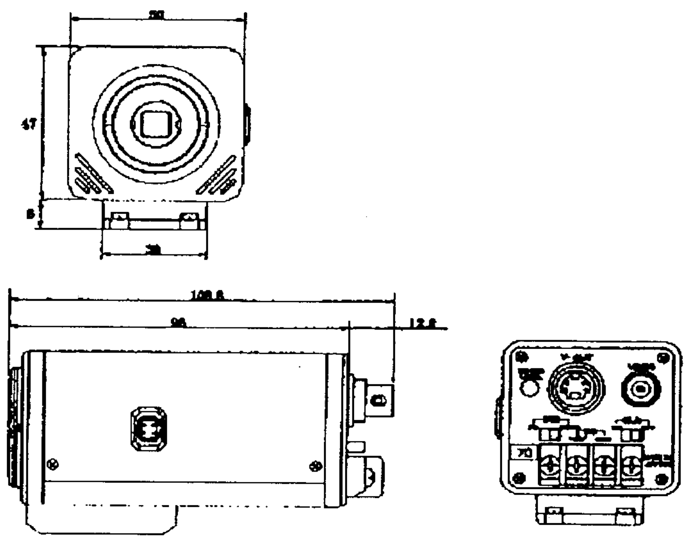
SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE
DC000208 VER1.0

■DIMENSION

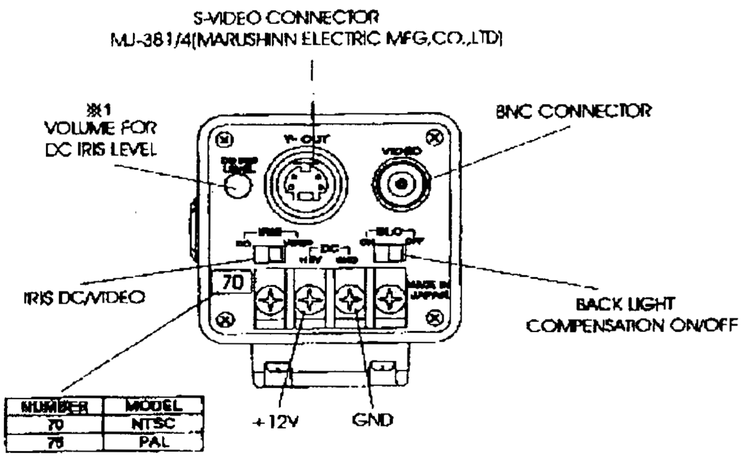
◆PX-70/75KST(SHORT CASE)



◆PX-70/75KLT(LONG CASE)



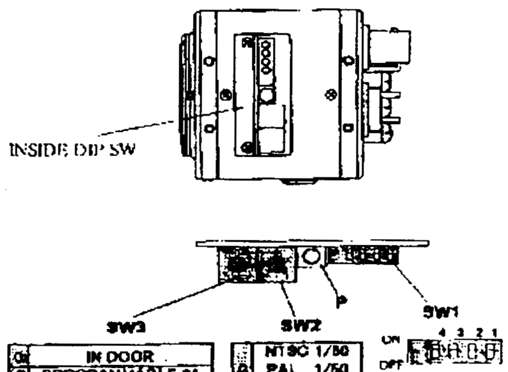
BACK PANEL



NUMBER	MODEL
70	NTSC
75	PAL

※1 THIS DC IRIS VOLUME ADJUSTS THE BRIGHTNESS WHILE SETTING THE IRIS DC/VIDEO SWITCH TURNS TO "DC"

■DIP SWITCH



SW3

0	IN DOOR
1	PROGRAMMABLE *1
2	FLURESCENT LIGHT
3	OUTDOOR
4	Don't Care
5	Don't Care
6	Don't Care
7	Don't Care
8	Don't Care
9	AUTO WHITE BALANCE

*P:RS-232C CONTROL

SW2

0	NTSC 1/60 PAL 1/50
1	1/100
2	1/250
3	1/500
4	1/1000
5	1/2000
6	1/4000
7	1/10000
8	Don't Care
9	Electronic Iris

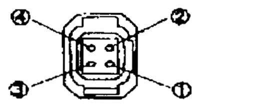
SW1

1	ON	AGC ON
	OFF	AGC OFF
2	ON	AGC LOW
	OFF	AGC HI
3	ON	GAMMA-0.45
	OFF	GAMMA-1
4	ON	BLC ON
	OFF	BLC OFF

■INITIAL SETTINGS

SWITCH	MODE
SW1-1	AGC ON
SW1-2	AGC HI
SW1-3	γ=0.45
SW1-4	BACKLIGHT COMPENSATION
SW2	0
SW3	9

■IRIS CONNECTOR



AUTO IRIS CONNECTOR
D4-15IN-100(CHUOMUSEN CO.,LTD)

PIN	DC	VIDEO
1	DAMP-	+12V OUT
2	DAMP+	NC
3	DRIVE+	CONTROL
4	DRIVE-	GND

Kowa

Kowa Optimed, Inc. 20001 S. Vermont Ave. Torrance, CA 90502 • 1-800-966-5692
Levitch@kowa.com • Jonathan@kowa.com

Appendix B

Software Analysis

Following is the remapping portion of the software, and it's analysis. Comments start with //, and refer to the code below the comment.

```
//Declare the constant PI.
const double PI = 3.141592;

//Declare the maximum X-Axis value.
const int Xmax = 640;
//Declare the maximum Y-Axis value.
const int Ymax = 480;

//Declare our lookup table.  This table consists of all possible X and
Y axis //points, and after it is built tells us where each pixel should
go.  For //example, say that pixel 0,0 needs to be moved to 36,123.  At
position 0,0 in //the lookup table, we will have 36,123.
int LookupTable[640*480];

//////////////////////////////////////////////////////////////////////
////////
//This function determines the angle, given X and Y
//////////////////////////////////////////////////////////////////////
////////
float getTheta(int y, int x) {

      //This gives us the angle for a given point.  For example, in a
      unit //circle, the point .866, .5 has an angle of 30 degrees.
      //theta is our angle
      float theta = (float) atan2(y, x);  //find the arctan of Y/X

      if (theta < 0)
            theta = theta + (float) PI * 2; //if it's -, add 2PI

      return theta;
}

//////////////////////////////////////////////////////////////////////
////////
//This function finds the nearest non-zero pixel to k
//////////////////////////////////////////////////////////////////////
////////
int FindNearest(int i, int k) {

      //This function horizontally stretches the image, in other words
      when we //find the nearest non-zero pixel, we are only looking
      horizontally, not //vertically or diagonally.  In the future,
      this algorithm can use a lot //of improvement.

      //i is our row in the image.  We multiply the row number by the
      length of //each row in order to find our offset into the image.
      In other words,  //the offset is equal to (row number * length of
      row + column number)
      int offset = i * Xmax;        //find the offset into the image

      int j;                        //a temp counter to loop through each
row
      int MinDist = Xmax;           //set the min distance to the max row
length
```

```
        for (j = 0; j < Xmax; j++) {            //loop through the row

              if (LookupTable[offset+j] != 0) {   //if it's not a non-
                                                  //zero pixel

                    if (abs(j - k) < abs(MinDist))      //find the
                                                        //distance to k

                          MinDist = j - k;              //if it's less
                                                        than the //min
                                                        dist, set it to
                                                        the
              }                                         //min dist.
        }
        return MinDist;               //return the closest pixel
}

/////////////////////////////////////////////////////////////////////////
////////
//This procedure builds a lookup table, in other words, this is a
template
//telling us where to remap each pixel too.
/////////////////////////////////////////////////////////////////////////
////////
void BuildLookupTable(void) {

        //Note that the standard coordinate system places 0,0 in the
        center of the //image, while the computer places 0,0 in the upper
        left corner of the //image, so we have to correct for that.

        //the size of the image
        const int Size = Xmax*Ymax;

        //the center X position (coordinate)
        const int Cx = Xmax / 2;

        //the center Y position (coordinate)
        const int Cy = Ymax / 2;

        //the outer radius.  This is the radius of the mirror.
        const int Rmax = 240;

        //the inner radius of the mirror.
        const int Rmin = 120;

        //the total radius.  This is useful to us for when we check to
        see where //the pixel is located in the image.
        int Rnewmax = Rmax + Rmin;

        //The radius of the current pixel.
        int R;

        //temporary variables
        //Xi and Yi are the converted X and Y values.
        int Xi;
        int Yi;
```

```
//current row
int i;
//current column
int j;

//a copy of Yi
int Yi2;

//the angle of the current pixel
double b;

//our offset into the image
int offset = 0;

//loop through our lookup table and reset it
for (i = 0; i < Size; i++)
      LookupTable[i] = 0;

//loop through each row
for (i = 0; i < Ymax; i++) {
      //loop through each column
      //note that we skip every other column.  This is because
      the video //capture card that we use uses a video format in
      which two //neighboring horizontal pixels can not be
      separated.  This reduces //our remapped resolution, and can
      be solved by buying a new video //capture card.
      for (j = 0; j < Xmax; j+=2) {

            Xi = j - Cx;      //calculate our actual x value
            Yi = Cy - i;      //calculate our actual y value

            //find the current radius
            R = (int) (sqrt((Xi*Xi) + (Yi*Yi)));

            Yi2 = Yi;   //make a copy of the current Y value

            Xi = abs(Xi);     //move our position into the 1st
                              //quadrant
            Yi = abs(Yi);

            b = atan2(Yi, Xi);      //find the current angle of
                                    //our pixel
            b = b * 639.0 / PI;     //convert this into the X
                                    //axis value

            if (j < Cx)             //set the new X axis value
                  Xi = (int) b;
            else
                  Xi = Xmax - (int) b;

            //if the pixel is between the inner and outer radius
            if (R > Rmin && R < Rnewmax) {
                  if (Yi2 > 0)      //if we are above the Y axis
                                    //remap the top half
                        LookupTable[(R-Rmin) * Xmax + Xi] =
                        offset;
```

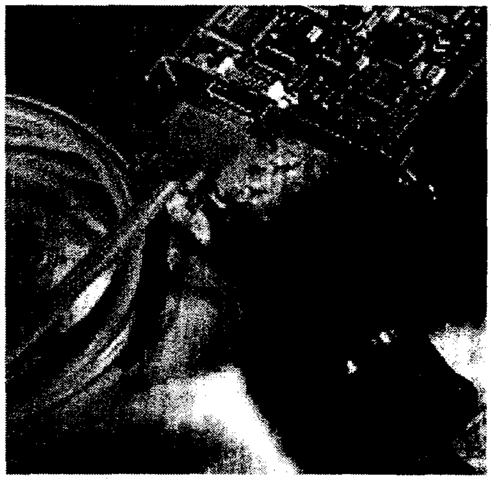
```
                        else                //else remap the bottom half
                              LookupTable[((R-Rmin) + Rmax)
                               * Xmax + Xi] = offset;
                  }
                  offset+=2;                //increase our offset by 2,
                                            //because we have to
            }                               //skip every other pixel.
      }

      //flips the lookup table upside down
      for (i = 0; i < Size/2; i++) {
            R = LookupTable[i];
            LookupTable[i] = LookupTable[Size - i];
            LookupTable[Size - i] = R;
      }

      //there's a lot of 'blank' pixels in our remapped image.
      //this algorithm fills in the blank pixels, with their
      //nearest horizontal neighbor.  This algorithm can be improved.
      offset = 0;
      for (i = 0; i < Ymax; i++) {          //loop through each row
            for (j = 0; j < Xmax; j++) {    //loop through each column
                  if (LookupTable[offset] == 0) {      //if the pixel is
                                                       //blank
                        R = FindNearest(i, j);         //find the
                                                       //nearest non-
                                                       //blank pixel
                        LookupTable[offset] = LookupTable[offset+R];
                  }
                  offset++;
            }
      }

      //Allocate space for the images.

      TempImage.lpData  = (LPSTR) malloc(Size*2);
      TempImage2.lpData = (LPSTR) malloc(Size*2);
      if ((TempImage.lpData == NULL) || (TempImage2.lpData == NULL))
            MessageBox(NULL, "Couldn't allocate the memory\n", "DOH",
MB_OK);

}

//This is where we actually remap the live video.
LRESULT CALLBACK capVideoStreamCallbackProc(HWND hWnd, LPVIDEOHDR
lpVHdr) {

      //creat a pointer
      int *ptr;

      //get the size of the image
      const int Size = lpVHdr->dwBytesUsed;

      //creat a temporary integer
      int i = 0;
```

```
        //loop through the lookup table, and copy each pixel to its
        correct //location in a temporary image
        for (ptr = &LookupTable; i < Size; ptr+=2) {
              memcpy((void *) &TempImage.lpData[i],
                          (const void *) &lpVHdr->lpData[*ptr * 2], 4);
              i += 4;
        }

        //copy the temporary image back to the original image
        memcpy(lpVHdr->lpData, TempImage.lpData, lpVHdr->dwBytesUsed);

        return 0;
  }
```

Appendix C
MegaCamera Specifications

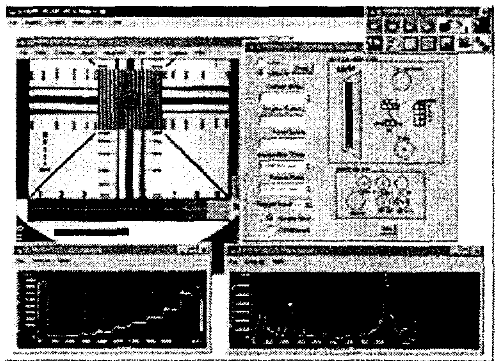
Silicon Imaging MegaCamera™
3.17 Megapixel Remote Head
30 FPS High Definition Digital Camera

Silicon Imaging has introduced the world first 3.17 Million pixel High Definition CMOS all-digital camera capable of running at video rates of 30 frames/second at its full 2056 x 1560 resolution. The entire micro-head package is 48 x 42 x 35mm and is small enough to be hand held for medical video instruments or placed on a robot for machine vision inspection.



High-Definition CMOS Technology breakthrough
CMOS imagers are breaking technical barriers in noise, sensitivity and dynamic range. Driven by the growing demand for consumer Digital Still Cameras, CMOS sensors have been developed which surpass the performance characteristics of CCD's in many photonic, imaging and consumer applications. By utilizing a single highly integrated CMOS device, which incorporates Megapixel sensing areas, timing generation, signal processing and high bandwidth outputs, Silicon Imaging has developed a compact high definition digital camera system.

12-Bit Pixel Clock Sampling – No Jitter
The SI-3170 Micro-HD camera uses 12-Bit digitizers to sample 2 output taps, each at 50MHz to achieve a 100Mpixel/sec or 30 Frames/sec data throughput and uses the 10 most significant bits for further processing. Converting the pixel data directly to digital at the CMOS sensor head eliminates pixel-sampling jitter and enables accurate sub-pixel metrology, image analysis and improved live video reconstruction.



16:9 HDTV Aspect Ratio & High Frame Rates
For HDTV applications, the aspect ratio can be switched from the traditional 4:3 to 16:9 by utilizing a region-of-interest (ROI) readout. A reduced size ROI enables readout rates in excess of 1000 frames per second, ideal for motion analysis or object tracking. A sub-sampled mode outputs a 640 x 480 pixel image representing the entire image on the sensor.

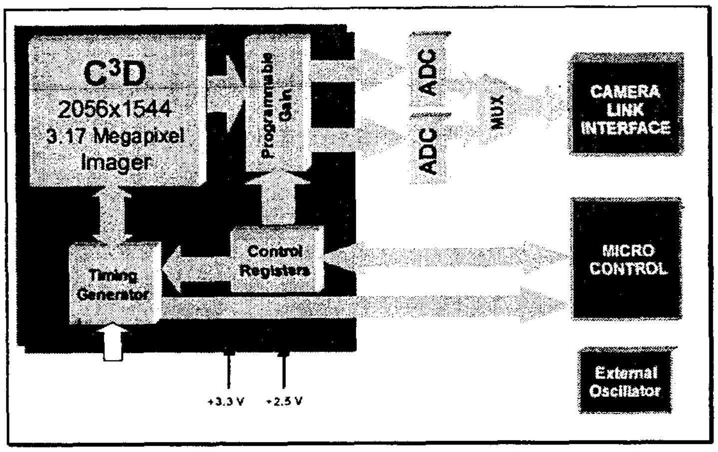
All-Digital Interface - CameraLink™
For 10 to 12 bit per pixel resolution and multi-tap systems the number of parallel digital signals becomes cumbersome to cable and physically large to connect. Therefore, high Speed digital multiplexers are used to serialize the image data and transmit the 1.2Gb/sec data, clock and triggers over just a few twisted pairs, thereby minimizing the cable size and increased flexibility. An industry standard forum has adopted this method, called CameraLink™, for camera and Frame Grabber connectivity and low cost cabling.

High Speed Transfer to PCI Bus
Silicon Imaging provide a high speed PCI Frame Grabber board to receive the 10 bit per pixel image data from the camera. As a PCI bus master, it can transfer the image data to the host computer's memory or a PCI bus target at rates up to 132 megabytes per second. Imaging applications which were once restricted to custom image processing hardware, or limited by expensive image memory, can now be performed by the SI-3170 Camera, frame grabber, imaging software, a Pentium processor, and a suitable PCI bus motherboard.

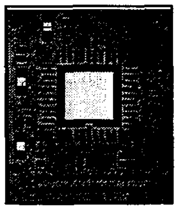
FEATURES

- 2056 x 1544 (3.17 Million Pixels)
- ½ Imaging Format, 3.3um Pixel
- 10 Bits per Pixel (12 Bit A/D Conversion)
- 100Mpixels/sec Throughput
- 30 FPS Sequence Capture at full Resolution
- CameraLink™ Digital Interface
- High-Speed Progressive Rolling Shutter
- 80$u$seconds to 8 minutes Integration
- Triggered Image Sequence Capture
- Monochrome & Color Models
- 32 Bit PCI Bus Master/Target
- 132 Megabyte per Second PCI Burst Transfers
- Software Supports up to 8 Boards/Computer
- Win98, WinNT, and DOS Software

CMOS Camera Block Diagram Acquisition



PCB OEM Version



44 x 33 x 14mm - 2PCB
Actual size

LIVE VIDEO DISPLAY

Image display is provided by the host computer's video graphics adapter (VGA) and monitor(s) or flat panel(s). Display resolution is a function of the capabilities of these devices. Some Super VGA (SVGA) monitors can display the full resolution of the camera image. A software look-up-table feature allows display of 8 bits from the 10 bit gray level image.

COLOR BAYER CONVERSION & CORRECTION

Bayer color Image data from the SI-3170C is converted to 24Bit RGB using the most advanced signal color conversion algorithm available. AWB Automatic White Balance functions are implemented to match the true world color to the calculated and displayed values. These functions are also available in the Imaging Library (S-LIB) for OEM applications.

SOFTWARE

A free ready-to-run interactive capture program (GRAB) that features camera control screens, line and column pixel plots, image sequence capture, and image sequence display. Images can be saved to disk for processing by other programs. ANALYZE is a ready-to-run interactive image processing and analysis program for qualitative and quantitative image operations. A few mouse clicks select FFTs, histograms, morphology, measurements, edge detection, correlations, 3D plots, arithmetic operations, and many other functions. The tools provide analysis of the SI-3170 images using 10 bits of grey level per pixel. GRAB can control up to 8 Frame Grabber boards in a single computer

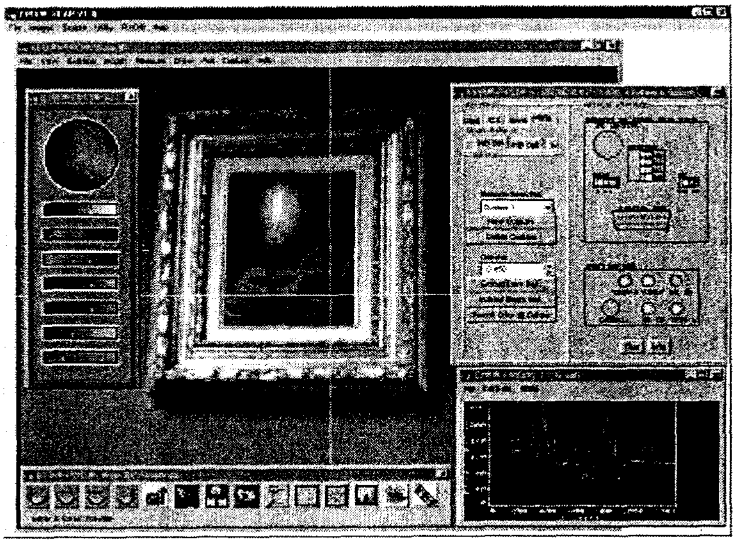


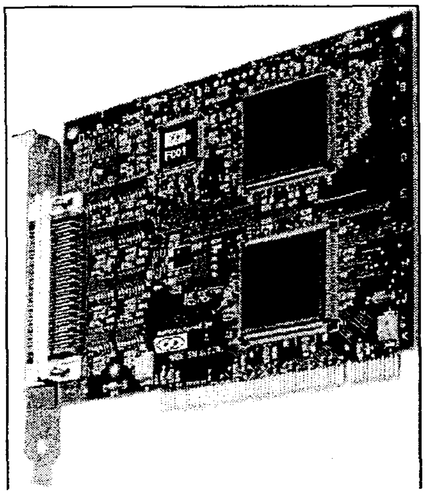
IMAGE PROCESSING LIBRARY

An extensive programming library is available which includes sample code for image processing and allows application-specific code to be added to the large selection of functions already developed. No royalties are required for software developed using our libraries.

SOLUTIONS and SUPPORT

The Silicon Imaging team has provided vision solutions and support for OEM machine vision manufacturers, camera manufacturers, radiologists, astronomers, biologists, and engineers for 15 years.

SPECIFICATIONS



FG-3170 PCI-Bus Frame Grabber

Features

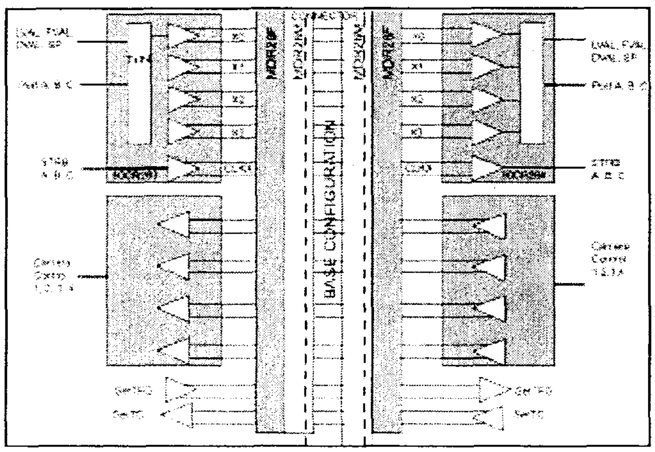
- 32 bits, 33 MHz PCI slot.
- 1.55 Amps @ +5 Volts.
- 4.913 inches long by 4.20 inches high (short slot)
- CameraLink™ MDR-26 pin Connection
- Camera Power at 12VDC, 1A included

Software Requirements

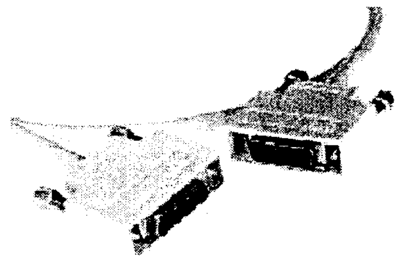
Requires Windows 16/24/32 bit RGB SVGA compatible display system. Display resolution as per installed VGA device driver. Direct Draw with hardware overlay recommended.

Motherboard

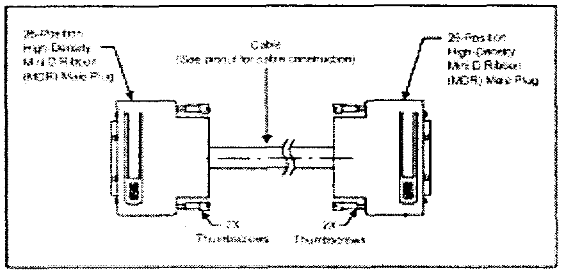
Requires a PCI motherboard capable of sustained transfer rates of at least 100 MB per second for full resolution image capture to motherboard DRAM. Advanced Graphics Port (AGP) PCI motherboard and AGP VGA is recommended.



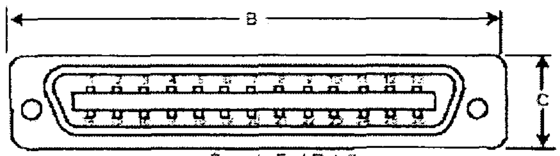
CameraLink™ Signals (Data, Clock, Serial & Control)



CameraLink™ Cable with 26-Pin MDR



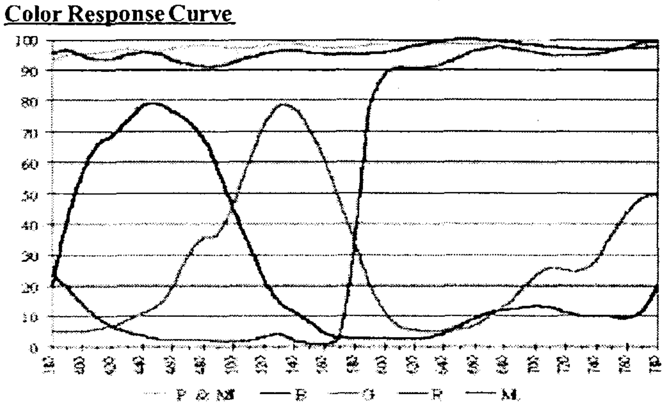
CameraLink™ Cable Diagram



Generic End Detail
MDR 26 Position Plug

Shell Option	B	C
Thumbscrew Overmold Shell	1.58 [40.2]	.55 [14.0]
Thumbscrew Shell Kit	1.55 [39.4]	.51 [12.8]

Color Response Curve



SI-3170 Camera Specifications (PRELIMINARY)

Sensor:

Active Pixels	2056 (H) x 1544 (V)
Optical Imaging Format	½
Pixel Size (pitch)	3.3um x 3.3um
Pixel Type	CMOS Active Pixel
Aspect Ratio	1 : 1
Spectral Response	350 ~ 1100 nm (see curve)
Dynamic Range	66dB (Vsat/Read Noise)
Fill Factor	38%
Fill Factor with MicroLens	80%
Sensitivity	0.5 Lux @ F1.0
Linearity (5-70%)	+/- 2.5% SAT
QE @ 540nm	0.58 e-/photon
Read Noise	20 e-
Dark Current Noise at 295K	< 3e-
Saturation Capacity	35,000 e-
Conversion Gain	36.0 uV/e-
Vsat	2.7 V
Shutter	Rolling Shutter
Shutter Speed / Integration	Variable, 4 to 4091 Line times
Readout	Progressive Scan, windowed, Sub-sampled

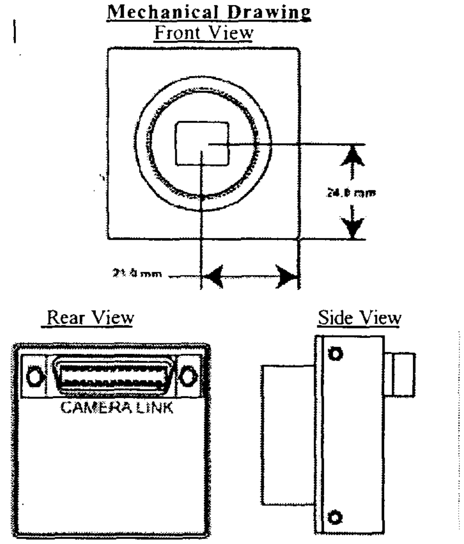
A/D Conversion & Sampling Clock Synthesizer

A/D Conversion	2ch @ 50Mhz (Nominal)
Vertical Resolution	12 Bit (10 MSB are for processing)
Pixel Clock Frequency	20 ~ 100Mhz
Adjustment Method	Serial command Protocol
A/D SNR	67.5dB
Output Noise	0.2 LSB rms

Digital Video Output: 12 Bit Multiplexed LVDS (CameraLink)

Readout Rate	100 MHz @ 12 Bit (8 or10 Bit optional)
Readout Format	10/12 Bit Dual Channel, (Ports A, B, C)
Frame Rate	2056 x 1544 @ 30fps 1600 x 1200 @ 38fps UXGA 1920 x 1080 @ 48fps (16:9) 1280 x 1024 @ 48fps SXGA 1280 x 720 @ 60fps (16:9) 1024 x 768 @ 60fps XGA 640 x 480 @ 100fps VGA 640 x 240 @ 200fps 256 x 128 @ 360fps 24 x 6 @ 2500fps
Sub-sampled mode	686 x 512 @ 30fps
Signal-to-Noise	> 60dB (fc=20MHz, Gains=1.0)
Connector	MDR 26-pin connector (3M 10226-6212VC)

Mechanical Drawing



Frame Grabber Control & Communication:

Serial Communication	RS-232 Protocol 112Kbps
Signaling	TX & RX (LVDS)
GP Asynchronous Triggers	3 (LVDS)
General Purpose I/O	4 TTL (on Camera PCB header)
High Speed Shutter	40nsec ~ 2msec
Long Integration	1 sec ~ 5 min
Region-of -Interest	24 x 6 to 2056 x 1544 in 24x6 steps
R,G,B Independent Gains	4 Settings each (1x, 1.2x, 1.5x, 2x)
Overall Gain	3 Settings (1x, 2x, 4x)
Setting Timing	Next top of Frame
Data/Power/Trigger/RS-232	MDR-26

Power

Input Voltage	6 ~ 12 VDC
Power	6 Watts
Power Connection	CameraLink Connector Using CTRL +4 (+V) & CTRL –4 (GND)

Mechanical

Lens Mount	C-Mount
Enclosure Size	1.8" W x 2.0" H x 1.5" L
Weight	12 oz.
Camera Mount	¼" x 20standard tripod mount

ORDERING INFORMATION

SI-3170RGB-S	3.17MP Digital Camera, 2M Cable, PCI Frame Grabber & Win 98/NT Imaging Software System
SI-3170RGB or 3170M	3.17MP Digital Camera (RGB for Color, M for Monochrome)
FG-3170	PCI bus Frame Grabber for the SI-3170
CL-2M	2 meter Digital Camera Cable
CL-5M	5 meter Digital Camera Cable
CL-10M	10 meter Digital Camera Cable

***** Pricing & Specifications subject to change*