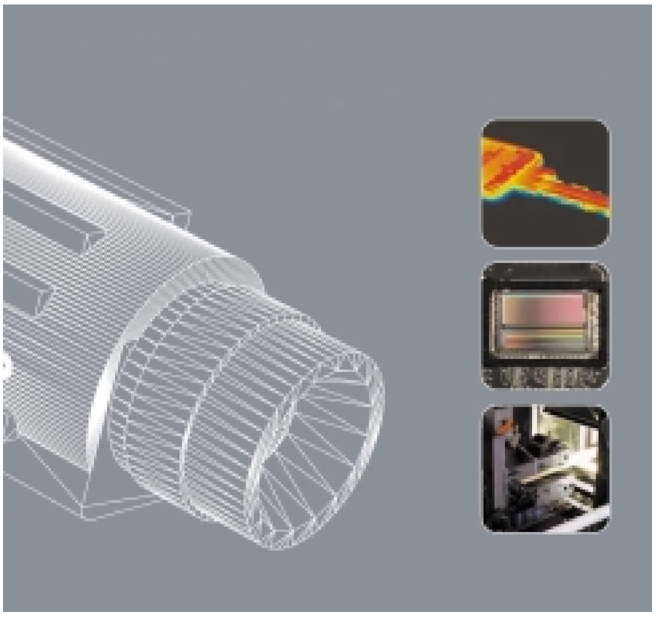
IVP Ranger M50

Product Information



CONTENTS

1 Introduction

IVP Ranger M50 is a 3D-vision sensor system, based on the IVP Smart Vision Sensor, M12. The system is designed for industrial measuring applications requiring high speed and accuracy. IVP Ranger M50 can be used in advanced inspection systems, in areas like: electronic, wood and automation. Following items are included in the IVP Ranger M50 system:

- The IVP Ranger M50 Camera hardware including Ranger M50 3D profiling software
- The HSSI Adapter Board, a PCI-bus adapter, for high speed transfer of the 3D data from the IVP Ranger M50 Camera to a PC
- The IVP Ranger M50 Development Software, including an application programmer's interface and an advanced graphical user interface for Windows NT, Windows 2000 Professional or Windows XP

The profiling software in the Camera consists of a number of algorithms for 3D profiling, optimized for different applications. The IVP Ranger M50 also provides various data buffering options as well as various possibilities to synchronize the 3D measurement with motion or external signals.

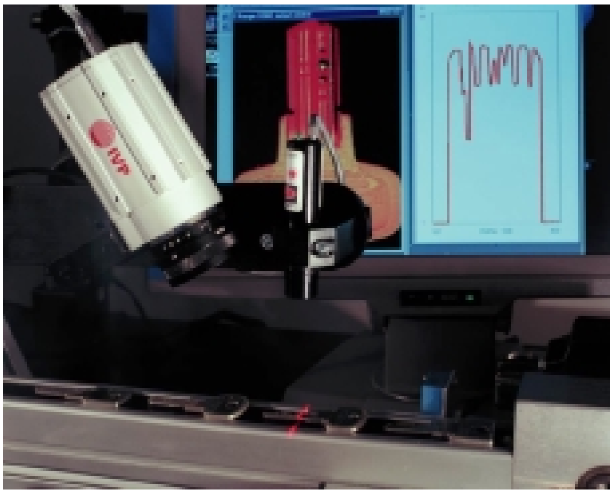


Figure 1-1 IVP Ranger M50 provides high accuracy at extreme speed.

1.1 Measuring Principle

The measuring principle of IVP Ranger M50 to acquire 3D shapes is based on the method called laser triangulation, or sheet-of-light range imaging.

The object is illuminated from one direction with a laser line projector and viewed with the camera from another. The illumination angle, the viewing angle, and the baseline between the illuminator and the camera define the triangulation geometry, from which the camera calculates the 3D shape.

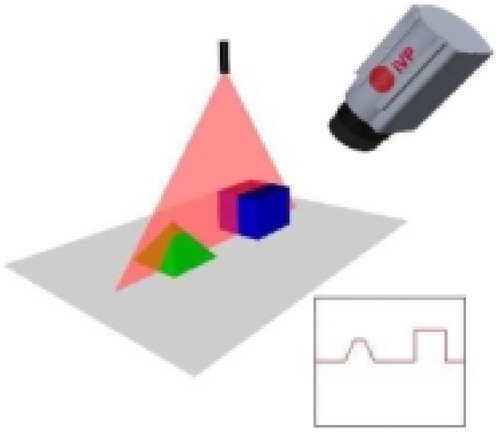


Figure 1-2 Laser triangulation principle, laser line projected over an object gives the profile of the object.

1.2 High Speed Data Acquisition

The processing to convert the image of the laser line into a vector with 3D shape data is very time consuming in traditional machine vision systems. Thanks to the unique IVP Smart Vision Sensor with built-in RISC processors, the conversion from 2D images to 3D shape data can be done already on the sensor chip. The image of the laser line never leaves the sensor chip, only the calculated 3D data itself, this is the key to the extremely high speed of IVP Ranger M50.

1.3 High Accuracy Measuring

The IVP Smart Vision Sensor M12 has a width and height resolution of 1536 x 512 pixels. With the sophisticated Ranger M50 3D profiling software it is possible to obtain 3d data with a resolution of up to 1/2048 of the height measurement range. Since the camera can be fitted with any 1" C-mount lens, the user can define his own measurement range, best suited for a specific application. Metric resolution of the measurements is thereby allowed to vary from micrometer range and up, just by choice of optics.

2 IVP Ranger M50 Camera

2.1 Camera Hardware

The core of the IVP Ranger M50 Camera is the Smart Vision Sensor M12 and the Intel StrongArm processor. The architecture is built of three main components: the M12 sensor, the processor and the high-speed serial interface (HSSI) module. The sensor has two parallel 32 bit wide buses, one is used for feeding instructions from the Strong Arm processor, and one is used for outputting data over the HSSI-link.

The HSSI link can send and receive data at high speed (Max. 330 Mbit/s) to the PC. In addition to the HSSI interface there is a RS232 serial port and 5 digital in and 1 digital output on the camera. The digital I/O can be used to synchronize the measurements with the movement of the object or the sensor system.

The IVP Ranger M50 Camera is built to withstand industrial environments with compact dust-resistant aluminum housing, with shielded cabling and connectors. The Camera dimensions are 97 x 50 x 50 (l x h x w) mm and the unit is threaded for standard 1" C lenses. The Camera unit supports power supply between 12 and 24VDC.

There are three cables used in an IVP Ranger M50 system: the SAH HSSI Cable, SAH Power and I/O Cable and the SAH RS-232 Cable. The distance between the PC and the Camera is maximum 50 meters with a twisted pair SAH HSSI cable, for longer distance optical fiber communication cables can be offered by IVP.

2.2 Camera Software

The profiling software in the Camera consists of algorithms for 3D profiling, data buffer options and possibilities to synchronize the 3D measurement. IVP can also develop customized algorithms to support unique requirements of an inspection application, please contact IVP for further discussions.

2.2.1 Algorithms

The standard algorithms are: *Horizontal thresholding*, *Horizontal maximum and thresholding* and *Image*. The two first are 3D-range algorithms and the last one is a grey scale algorithm.

Measurement Techniques

All standard algorithms works with the laser line aligned horizontally with the rows on the sensor. That means, when viewing an image of a flat object illuminated with the laser a flat horizontal laser line appears in the image.

The algorithms support region of interest windowing, and thereby height measurement range can be traded for speed. Using a smaller region on the sensor enables measurements at a higher profiling rate. The *ROI*-height (Region Of Interest) is defined by selecting the number of sensor rows to be use, See *Figure 2-1*.

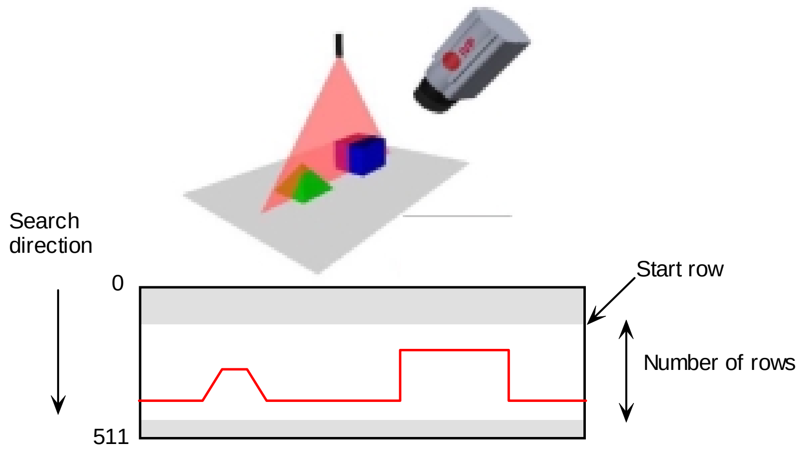


Figure 2-1 The ROI (Region Of Interest) is defined by the selecting the sensor rows used by the measurement algorithms.

The basic function of all the measurement algorithms is to compute the impact position of the laser line for all columns on the sensor. The light intensity distribution from the laser line along a sensor column can be described as in *Figure 2-2* below.

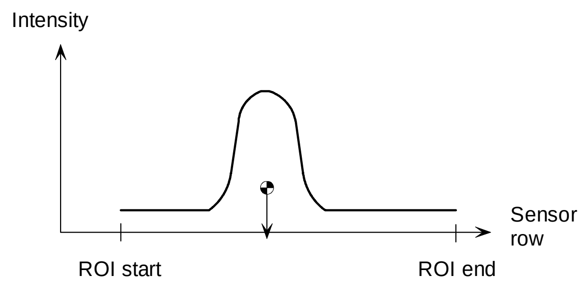


Figure 2-2 Light distribution of a laser line within one sensor column.

The laser line will form a distinct light peak distributed over a number of pixels within the sensor column. The center of this peak will be defined as the *impact position* of the laser line on that sensor column.

The measurement with the 3D algorithms is made by using *thresholds*. The usage of thresholds increases the height resolution of the measurement, with one threshold a ½ - pixel resolution can be obtained and with two thresholds a ¼ - pixel resolution can be obtained.

The *integration time* is defined as the time used to collect light on the sensor for one profile sample. With a fixed light intensity from the laser, two different integration times will result in different intensity values on the sensor.

The result from one measurement (scan) is a *range profile* of 3D-values (*Rangels*) and optionally also an *intensity profile* of intensity-values.

The IVP Ranger M50 algorithms are using parameters to set all the measurement techniques described above. The behavior of the selected algorithm is defined by the current parameter setting for that particular algorithm at the time when the algorithm is started.

HorThr Algorithm

The *Horizontal thresholding* algorithm is the fastest of the standard algorithms. The impact position of the laser in each column is computed as described in section above. Multiple thresholds may be used to achieve higher range resolution. A (3x1) median filter can be applied to the range data to smoothen the data, and to reduce laser speckle influences.

The acquisition speed of the HorThr algorithm is dependent on the current setting of the algorithm and the synchronization parameters. In the figures below some performance examples are given.

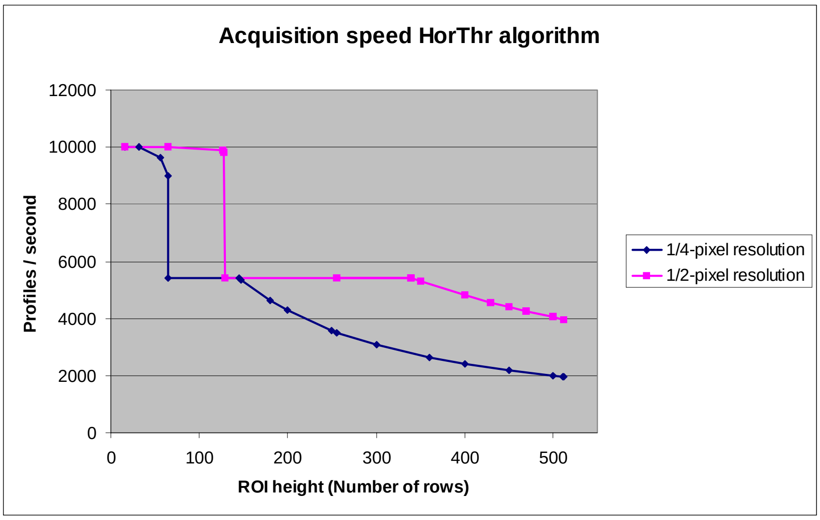


Figure 2-3 Acquisition speed diagram for the HorThr algorithm (trig mode: free-running).

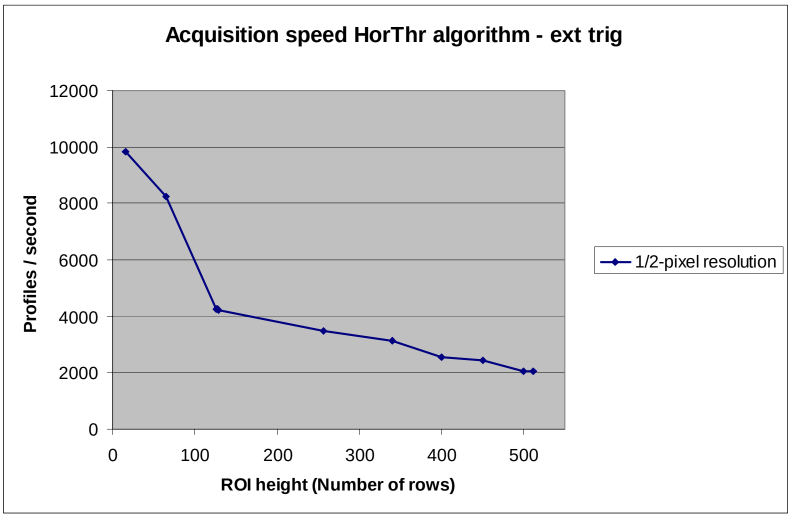


Figure 2-4 Acquisition speed diagram for the HorThr algorithm (trig mode: ext trig).

HorMaxThr Algorithm

The *Horizontal maximum (and) thresholding* algorithm computes maximum intensity values related to each Rangel. These values represent the intensity of the laser line at the impact position in each sensor column. The result from this algorithm is one range profile and one intensity profile. 3x1 Median filtering can be applied on this algorithm too.

Also for this algorithm, the acquisition speed of the algorithm is dependent on the current setting of the algorithm and the synchronization parameters. In the figure below some performance examples are given.

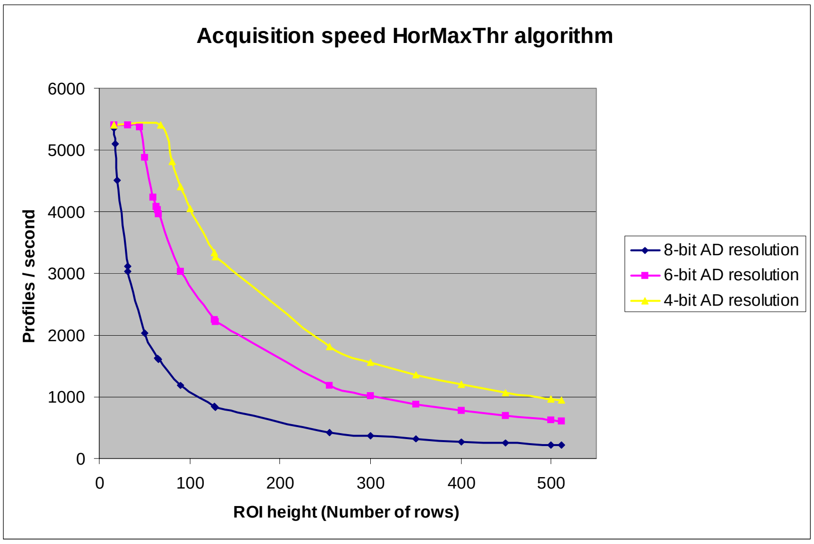


Figure 2-5 Acquisition speed diagram for the HorMaxThr algorithm (trig mode: free-running).

Image Algorithm

The *Image* algorithm is a traditional 2D grey-scale imaging algorithm. ROI windowing can be used in the same fashion as in the Range algorithms, i.e. the user can define the size of the window to read out and the position of it. This algorithm is useful when setting the system up, , aligning the laser, focusing lenses etc.

User Algorithm

The *User* algorithm is where a application specific algorithm fits into the IVP Ranger M50 system. This algorithm can be customized by IVP to meet the specific requirements of a particular customer application.

A default User algorithm is delivered among the standard algorithms. This User algorithm includes more advanced settings compared to the standard algorithms above (HorThr, HorMaxThr). Two special features are implemented in the default user algorithm, morphological filtering and range direction. The morphological filters are noise reduction operations that pre-process the binary image data used in the profile detection

The Range direction feature is used to change the notation of up and down in the Ranger M50 profile data. Thus, if the profile appears the wrong way in the Ranger GUI this feature will put it right. Apart from visualization, this might also be useful if using two Ranger M50 cameras that measure the thickness of an object.

2.2.2 Data Buffering

The IVP Ranger M50 camera software supports three different buffer modes: Live, Fifo and Record. The buffer area is placed in the memory on the PC side.

2.2.3 Profile Synchronization

IVP Ranger M50 supports different methods to synchronize the measurements with the movement of the object or the sensor system. Example of this could be to synchronize the measurements with the movement of the object on a conveyer belt. Another situation could be to synchronize the measurements obtained from several views of the object.

Trig mode: Free-running

In the *Free-running* mode the camera runs without synchronization. Profiles are acquired and sent to the PC at a frequency derived from the integration time. Timing wise everything runs by the cameras internal clock.

Trig mode: Data enabled

In *Data enable* mode the camera runs unsynchronized in the same fashion as in the free running case, but data is not sent to the PC until there is an object with a user definable (set by parameter) height and width in camera's filed if view.

Trig mode: Ext clock

In *Ext clock* mode an external clock is used define the integration time and data is sent to the PC at the external clock frequency. The external clock can be scaled down to a lower frequency.

Trig mode: Ext trig

In *Ext trig* mode an external signal is used to start the data acquisition but the integration time is still controlled by the internal clock. The sensor is always reset before the acquisition in this mode and the results is that data is sent to the PC after each external signaling. The external trig can also in this mode be scaled down to a lower frequency.

Other Synchronization Possibilities

A complement to the trig-methods above is to mark the acquired measurements with information related to when the measurements were made. There are two position-counters holding a 32-bit signed value, one position-counter for the X-axis and one position-counter for the Y-axis (scan direction).

A second dimension of synchronization is to enable/disable the trig modes with an external signal. This option is designed for synchronizing several cameras.

3 HSSI Adapter Board

The HSSI Adapter Board is a half size PCI board that performs the high speed communication between the camera and the PC. Each HSSI Adapter Board is equipped with two full duplex Hot Link ports for simultaneous access to two cameras. Up to four boards can be used in one PC. Each board is also configured with a general I/O connector with 5 in and 1 out pins (TTL).

The software drivers for the HSSI Adapter Board are delivered in the "Ranger M50 Development SW" and compatible with the Microsoft Windows NT/2000/XP environments.

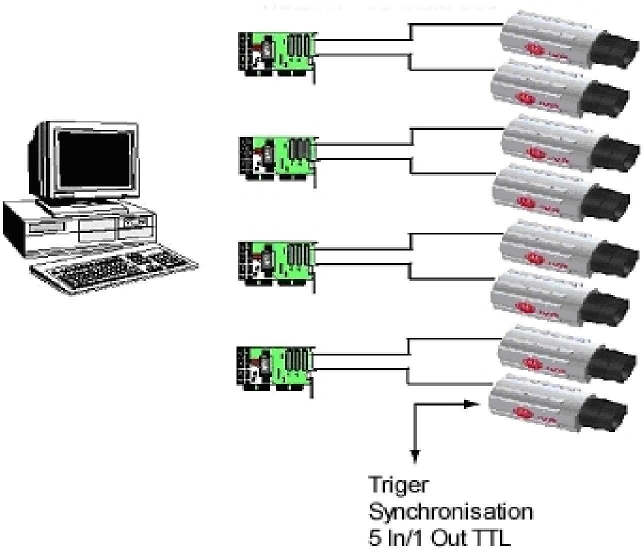


Figure 3-1 Possible HSSI Adapter Board set-up for one PC.

4 IVP Ranger M50 Development Software

The Ranger M50 Development Software is a package for application development in a Windows NT, Windows 2000 Professional or Windows XP environment. It includes an evaluation environment with a graphical user interface "Ranger GUI", example data, example program as well as libraries and header files for application development.

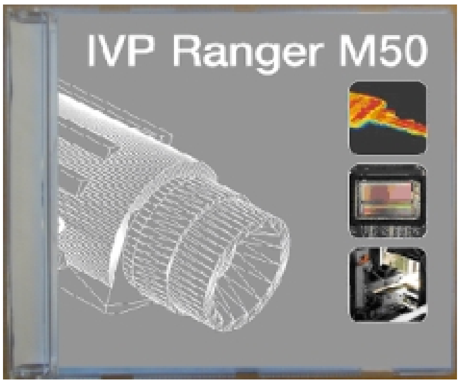


Figure 4-1 Ranger M50 Development SW.

4.1 Ranger GUI

The Ranger GUI, Graphical User Interface, is an evaluation platform that is built using the Ranger API, Application Program Interface. The purpose of the Ranger GUI is to get the user of a Ranger system familiar with the possibilities of the system. The user can test the different algorithms and evaluate which one is most suitable for his industrial application. It is then straight forward to use the Ranger API to build the application.

4.1.1 Ranger GUI Layout

The Ranger GUI is built as a standard Windows based application which makes it very easy to understand and use. The Ranger GUI consists of four main areas: the System Interaction Area, the Display Area, the Parameter Control Area, and the Status Output Area see *Figure 4-2*.

The System Interaction Area is used to interact with the camera/cameras. The Display Area is used to display plots, gray-scale pictures or 3D pictures of the measured object. The Parameter Control Area is used to set the parameter set-up of the running algorithm. The Status Output Area displays massages from the system when the system is running.

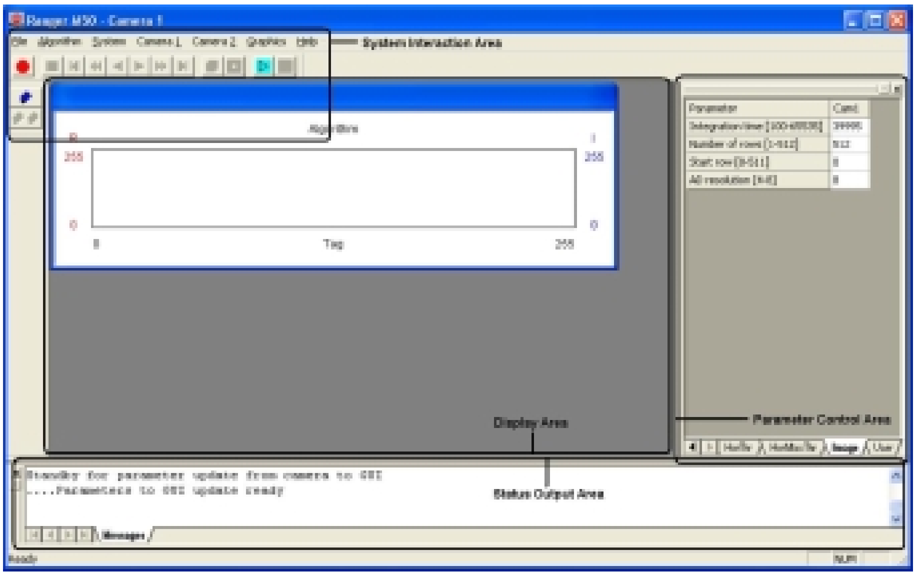


Figure 4-2 Ranger GUI layout.

4.1.2 Ranger GUI Functions

The Main functions in the Ranger GUI are to run the algorithms and show the result in a plot or a picture. It is very easy to try out different parameter set-ups to find the right set-up for the specific measured object. The parameter set-up can be stored to be used at a later occasion.

The measurement can either be recorded or viewed live. When recorded, it can also be stored for later usage in the GUI or for another viewer. The Ranger GUI can handle up to two cameras simultaneous.

4.1.3 PC Requirements

These are the recommended minimum requirements on a PC to be operated with the IVP Range M50 system:

- Windows NT 4 service pack 6, Windows 2000 Professional or Windows XP
- 800 MHz processor
- 256 MB RAM
- 1024*768 @ 24 bits of colors

4.2 Application Development

The application development is done on a well defined Ranger API using a C/C++ programming environment. The API contains calls for the Ranger algorithms as well as functions for camera communication and system setup. It also contains functions for buffering and access of the data received from the camera via the HSSI Adapter Board.

5 Technical Specification

Table 5-1 Technical Specification

System Part	Specification
Maximum profile sampling rate	≤ 10 000 profiles per second
Algorithms	3D range with horizontal, threshold and maximum finding and 2D image
Trig modes	Free running, Data enabled, Ext clock and Ext trig
Sensor size	512 x 1536 pixels
Optics	1 inch, C-mount
Camera I/O	5 in, 1 out, TTL level
Communication link	Cypress HOTLink™
Cable length between PC and Camera	Max 50 meter with shielded twisted pair
Max number of Cameras to one interface board	Max 2 Cameras to one interface board
Max number of boards in one PC	Max 4 interface boards in one PC
Host platform	Windows NT/2000/XP
PC requirements	Min Pentium III, 800 MHz, half-length PCI slot, 256 Mbyte RAM
Load on the PC @ 5000 profiles/s (PIII , 2,4 GHz)	< 10%
Power	12-24 VDC
Camera house temperature	5-50 °C

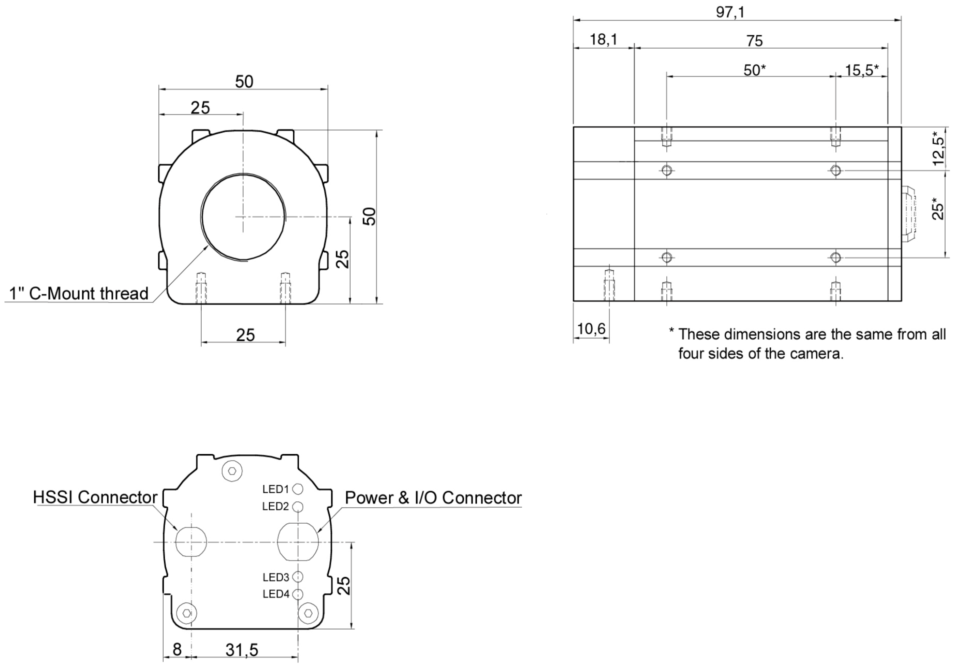


Figure 5-1 Dimensions of the Ranger M50 camera house.

6 Appendix 1: Program Example

There are typically three stages in an application: Camera communication set-up, Algorithm and parameter set-up and Measurement control. Following sections examine how to realize this with some simple pseudo code examples.

```
 1: RangerInitCamera(id, board, chnl, NULL); // Starts the API and
                                             // contacts the camera.
 2: RangerLoadCameraParameters(id, fname);   // Loads algorithm
                                             // parameters from file
                                             // and sends them
                                             // to the camera.
 3: RangerStartLive(id);                   // Starts a receiver buffer
                                           // on the PC side.
 4: RangerRunAlgorithm(id, RANGERHORTHR);  // Sends a message to the
                                           // camera, telling it to
                                           // run the requested algorithm
 5: while(cont && !error) {
      // Poll the Live buffer for measurement data
 6:   error = RangerGetLive(id, timeOut, &pHeader, &pData, &nrOfMissed);
 7:   cont = MyEvaluationFunction(pHeader, pData); // Process data
    }
 8: RangerSendStop(id); // Send a stop message to the camera
 9: RangerStop(id);     // Stop the receiving buffer
10: RangerReleaseAPI(); // Release resources & exit API
```

The first statement sets up the communication with the camera.

The second statement points the Ranger API to a parameter file to be sent to the camera.

The third statement starts preparing the measurement loop by setting up a buffer that will receive the camera data when the camera algorithm is started.

The fourth statement sends a message to the camera telling it to start an algorithm. The Live buffer is now ready to be polled and the resulting data can be evaluated.

In this example it is assumed that the data evaluation function returns false when having all needed samples are acquired, whereby the measurement loop, consisting of row 5-7, is exited. The first thing to do after this is to send a stop message to the camera. This is done by statement 8 and the camera enters idle mode waiting for new commands.

When the camera has stopped sending data to the application the receiving buffer can safely be taken down, as done by statement 9.

Before finished it is needed to tell the Ranger API to release the allocated resources. This is done in the last statement with a call to the RangerReleaseAPI function. If the application uses multiple cameras, then all cameras must be in idle mode before calling this function. A single camera can in the meantime be released with the RangerReleaseCamera function.