Smart Camera Surveillance System

PROJECT REPORT 2002-2003

Submitted By

PREETHI VENKATESAN

RAVI SHANKAR . S

SHARAD MURALI

SIVA SELVI ABIRAMI . S

Under the Guidance of

Asst. Prof. Mr. K. RAMPRAKASH, B.E., M.E.,

IN PARTIAL FULFILMENT OF THE REQUIRMENTS

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN ELECTRONICS AND COMMUNICATION ENGINEERING

OF THE BHARATHIYAR UNIVERSITY, COIMBATORE

Department Of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641006

Dedicated to
Our
Beloved Parents,
Teachers & Friends

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006

Department Of Electronics and Communication Engineering

Certificate

This is to certify that the Report titled

Smart Camera Surveillance System

has been submitted by

Mr/Ms ..PREETHI...VENKATESAN...RAVI SHANKAR S.... ..SHARAD MURALI, SIVA SELVI ABIRAMI......*in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Electronics and Communication Engineering Branch of the Bharathiar University, Coimbatore – 641006 during the academic year 2002- 2003.*

Guide 14/3/03

Head of Department

Dept. of Electronics & Communication Engineering
Kumaraguru College of Technology
COIMBATORE - 641 006

Certified that the candidate was examined by us in the project work Viva-Voce Examination held on ..18.3.2003....... *and whose University Register Number is* ..9927D0148, 155, 165, 166

18/3

(Internal Examiner)

(External Examiner)

Soliton
True engineering

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is a bonafide certificate for the successful completion of the project work titled "**Smart Camera Surveillance System**" done by **Miss. Preethi Venkatesan, Mr. Ravi Shankar. S, Mr. Sharad Murali, and Miss. Siva Selvi Abirami. S**, students of Electronics and Communications Dept., Kumaraguru College of Technology, from May 2002 to Feb 2003 as a part of their curriculum for the degree of Bachelor of Engineering in our organization.

The students have displayed tremendous energy and initiative and have performed outstanding work on a challenging project. We take this opportunity to congratulate them and wish these promising engineers all the very best for a bright future.

T. Sasikala
Senior Project Engineer,
External Project Guide.

Ganesh Devaraj, Ph.D.
Managing Director.

Coimbatore
March 10, 2003

Soliton Automation India Pvt. Ltd., Classic Towers, 1552/B7, Trichy Road, Coimbatore 641 018, INDIA
Tel : 91 422 302374, 302371 • Fax : 91 422 302375 • E-mail : mail@solitonautomation.com
Web : http://solitonautomation.com

ACKNOWLEDGEMENT

"A man may not stand alone but for the team that guided him to victory." Our deepest gratitude to all those who have helped us in this project and made it a memorable and successful one. First and foremost we wish to express our profound thanks to our respected Principal Dr.K.K.Padmanabhan, B.Sc., (Engg), M.Tech., Ph.D., who has given consent to this project.

We are forever indebted to our beloved Head of Department, Prof. M. Ramasamy, M.E., M.IEEE (USA), MIE, MISTE, and C.Eng. (I), MIBMESI. for his valuable suggestions and support. It has been a privilege to work under the guidance of our discerning guide Prof. K. Ramprakash M.E., (PhD) who is a constant source of inspiration. We sincerely thank him for his encouragement, undying patience and constant evaluation. Our particular thanks to our Class Advisor Mrs. Amrita Gowri for her good support and good will at various instances. We are very grateful to all the staff members who have extended their support during all times. Thanks to all the Lab Technicians for their full cooperation.

We shall be failing in our duties if we do not place on record the efforts of Soliton Automation Pvt. Limited. We express our immense gratitude to Dr. Ganesh Devaraj Ph.D., Managing Director of Soliton Automation Pvt. Limited for his faith in our abilities and for his constant encouragement. We are forever grateful to our project guide Ms. T. Sasikala, Project Engineer, Soliton Automation Pvt. Limited, for her ever supportive nature, constant encouragement, valuable suggestions and for all the help provided to us. We shall never be able to thank her enough. Very special thanks to Mr. Senthil Kumar, Soliton Automation Pvt. Limited, for all the help he has provided us. Finally thanks to all the people in Soliton for their encouragement and support.

Last but not the least we wish to thank all our friends who have helped us make this project a successful one.

ABSTRACT

ABSTRACT

Security systems around the world are facing increasing challenges and the concept of security as compared to a few years back has increased in complexity and demand. There is not one security system that has yet offered what we term as 'Ultimate Security', though a few of them have come very close to attaining that level. This is where the concept of image processing comes. Perhaps the answer to all security systems is image processing. But as of now not one perfect algorithm that detects human presence has been established.

In this project our aim is to provide an algorithm for providing security using image processing techniques. Our objective is to design and implement an algorithm that detects the presence of a human being. The system provides facilities to differentiate between human beings and other disturbances such as pet dogs or cats.

A monochrome 12V CCIR camera is used to capture images from the environment under consideration. Initially a reference image is obtained and stored in the system. The images acquired further are compared with the reference image and further processing is carried out to obtain the intelligence information to determine whether there is any intrusion by a human. If there is any intrusion, an alarm is sounded. If the intrusion is due to any other pet, no alarm is sounded and the system carries out process in normal fashion.

The advantage of this technique is that even if the human being bends or even crawls in order to fool the system, he does not succeed in his attempts. In other words the system is fool proof. Another advantage of this system is that there is no need for a human operator to continuously monitor the cameras since it is fully automated.

Contents

1 Soliton Automation India Pvt. Ltd. – A Profile

Introduction

Soliton Automation India Pvt. Ltd., a high technology company based at Coimbatore, India, was started with the aim of becoming a leader in the development of test and measurement and industrial automation software and systems. Soliton Automation specializes in providing comprehensive test & measurement and industrial automation solutions that result in enhanced quality and productivity for our customers around the world. Soliton is one of the few companies in India to offer test & measurement solutions based on the latest Virtual Instrumentation technologies. Today Soliton Automation is India's No.1 provider of Virtual Instrumentation solutions.

Milestones & Achievements

Soliton Automation is committed to developing original and innovative solutions for its customers. This approach has resulted in high levels of customer satisfaction and led to international recognition. It has established a solid presence in the domestic and international Virtual Instrumentation market with its award-winning LabVIEW expertise. The wide acceptance of Soliton's capabilities is reflected in the impressive list of customers and the string of international awards:

- Customers and end-users include industry leaders like DaimlerChrysler, IBM and Nortel Networks in the U.S., and Ford, GE Medical Systems, Exide, Pricol, and others in India.
- In 1999 Soliton became the first Indian company to exhibit in NIWeek, the worldwide conference on Measurement and Automation.
- In 2000, Soliton's entry in the NIWeek 2000 Best Application Contest - 'Large Scale Battery Test System' was selected the 'Most Outstanding Application of Measurement & Automation' becoming the first Indian company to win the most prestigious international Virtual Instrumentation award.
- In 2001, our entry in the NIWeek 2001 Best Application Contest - 'Networked Furnace Monitoring system, won for Soliton the overall best

application award for the second year in a row. Soliton became the first company in the world to win the awards twice and in successive years.

- In 2001, Soliton signed an agreement with Cal-Bay Systems, Inc., a Silicon Valley company specializing in Virtual Instrumentation Systems Integration Services to establish the 'Cal-Bay Soliton Development Center', a dedicated Virtual Instrumentation Software Development Center for Cal-Bay at Coimbatore.

Inherent Strengths

National Instruments Alliance Advantage

Soliton Automation is an Alliance Member of National Instruments, Inc., headquartered in Austin, Texas, USA. National Instruments is one of the most innovative automation companies in the world and are the pioneers of the concept of Virtual Instrumentation. They are the leading global suppliers of computer based data acquisition, control and instrumentation products. Soliton has invested heavily in building up expertise in National Instruments products like LabVIEW™. This helps us design and build advanced automation systems quickly to customer specifications.

Excellent Human Resources

Soliton's core strength is its expertise in the design & development of Virtual Instrumentation systems. The basis for this is a team of highly skilled and motivated engineers with vast experience in building virtual instrumentation and automation systems. They bring together a wide range of skills and expertise that complement each other. The team is led ably by Dr. Ganesh Devaraj, who has extensive experience in the automation field, having executed many projects, while working in the US. He holds a Ph.D. in Theoretical Physics and an MS in Electrical Engineering both from the University of Michigan, Ann Arbor. Strong leadership in the technology area is also given by Mr. Anand Prasad Chinnaswamy the Director & Chief Technology Officer of the company. He has an MS in Electrical Engineering from Clemson University, SC, U.S.A & an MS in Plastics Engineering from the University of Massachusetts, Lowell, U.S.A.

Sound Project Management

At Soliton, we have implemented strong project management and effective quality control measures, to consistently meet our delivery commitments to our customers while maintaining our high quality standards.

Technology

Virtual Instrumentation

Virtual Instrumentation, a concept pioneered by National Instruments, is today the preferred platform for building measurement and automation systems. Virtual Instrumentation leverages the high performance and low cost of today's computers for building powerful automation systems very cost-effectively. Our expertise with the software and hardware products from National Instruments allows us to build customized automation systems for our customers extremely quickly. The major benefits of Virtual Instrumentation systems over traditional instrumentation are

- Higher performance and lower cost by using high volume PC components
- Highly integrated and customized to the requirement
- Modular, reusable, and easy to reconfigure as requirements change
- Built in networking and communications capabilities for enterprise connectivity
- Continuously upgradeable with the latest technologies
- Highly scaleable for future expandability

LabVIEW

LabVIEW, the flagship product from National Instruments, is the leading high-productivity software development platform for Virtual Instrumentation. LabVIEW is the primary development platform at Soliton to build feature-rich, customized automation systems to meet the short development cycles required by our customers.

Awards

The Virtual Instrumentation solutions that we have developed for our customers have been judged to be among the best in the world in National Instruments' Annual Best Applications in Measurement & Automation Contest.

- PXI-Based Oscilloscope Breaks New Ground in Real Time Monitoring of X-Ray Tube Spits – *Winner* – Biomedical Category – 2002
- High-Voltage Multi-Output Power Supply Test System – *Winner* – Manufacturing Functional Test Category – 2002
- Integrated PXI-System for Multi-head X-Ray Tube Stress Testing – *Finalist* – R&D Lab Automation Category – 2002
- Networked Furnace Monitoring System – *Winner* – Overall Best Application and Manufacturing Category – 2001
- Functional Testing of High Voltage X-Ray Tanks – *Winner* – Biomedical Category – 2001
- Motorbike On-Road Brake Test – *Semifinalist* – Automotive Category – 2001
- Large Scale Battery Test System – *Winner* – Overall Best Application and Test & Quality Category – 2000
- PC-based System Results in Faster Testing of Electronic Components – *Winner* – Semiconductor Category – 2000
- Automated Speedometer Calibration – *Semifinalist* – Automotive Category – 1999

The application 'Automated Speedometer Calibration' based on our innovative GaugeVIEW software was featured as a full-length article in the 15th February 2000 issue of the Test & Measurement World magazine, the leading publication of the Test & Measurement Industry. The same application was also featured in the November 2000 issue of Evaluation Engineering – The Magazine of Electronics Evaluation & Test. An article on the 'Large Scale Battery Test System' has been published in the June 2001 issue of Control Solutions – the control systems & instrumentation magazine published from the US. In addition, Dr. Ganesh Devaraj, the Managing Director of Soliton

Automation, was profiled in February 19th issue of India Today magazine, in the section called 'Adventure Capitalists'. Soliton was selected as one of the nine 'New Economy Players to Watch'.

These prestigious international awards and recognitions have motivated us to think and work harder to build innovative, cost-effective and premier quality solutions for our customers.

Customized Solutions

Test & Measurement Automation

- Production Testing System
- Prototype Testing System
- Automated Testing Equipment

Industrial Automation

- HMI/SCADA
- Distributed Data Acquisition & Control
- Process Monitoring Systems

Machine Vision & Image Processing

- Vision based Automation – Calibration, Control, Assembly, Positioning etc.
- Vision based Sorting, Fault Inspection
- Vision based Dimensioning

Virtual Instrumentation Software

Software development using –

- LabVIEW, LabWindows™/CVI, Measurement Studio, Lookout™, and HiQ™ from National Instruments
- C/C++, Visual Basic, and Visual C++

Device / Instrument Drivers

Driver development using –

- LabVIEW, LabWindows/CVI, C/C++

e-Factory

- Instrumenting Production Lines to Monitor Production and Quality
- Networking Production and Quality Monitoring Systems
- Online Production and SPC reports
- Interfacing Factory Floors with ERP Systems

Energy Saving System

- Micro-controller or computer controlled variable speed motor drives for energy saving.

Soliton's product development efforts have achieved significant breakthroughs in developing state-of-the-art products in energy saving, machine vision and data acquisition. The energy saving device for textile autoconers saves up to 60% energy. New products are being developed, of which the Energy Saving System for Humidification Plants in Textile Mills will lead to considerable savings in energy consumption.

Industries Served

- Automotive
- Medical Equipment
- Electronics & Semiconductor
- Battery Manufacturing
- Textile

2 INTRODUCTION

2.1 What is Digital Image Processing?

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer.

The digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels. Pixel is a term most widely used to denote the elements of the digital image.

Digital image processing encompasses processes whose inputs and outputs are images and in addition, encompasses processes that extract attributes from images, up to and including the recognition of individual objects. Digital image processing may be considered to have three types of computerized processes: low level processes, midlevel processes, and high level processes. Low level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement and image sharpening. A low level process is characterized by the fact that both its inputs and outputs are images. Mid-level processing on images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally, high-level processing involves "making sense of an ensemble of recognized objects as in image analysis, performing the cognitive functions normally associated with vision.

2.2 Components of an Image Processing System

The basic components of an image processing system are

Image Sensors:

Two elements are required to acquire digital images. The first is the physical device that is sensitive to the energy radiated by the object. The second is a digitizer, a device for converting the output of the physical sensing device into digital form. In a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

Specialized Image Processing Hardware: It usually consist of a digitizer and hardware to perform other primitive operations such as arithmetic logic unit which performs arithmetic and logical operations in parallel on entire Images.

Computer:

It is an image processing system which can range from a PC to a super computer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance.

Software:

It consists of specialized modules that perform specific tasks. A well designed package also includes the capability for the user to write codes. More sophisticated software packages allow the integration of those modules and general purpose software commands from at least one computer language.

Mass storage capability:

It is mandatory in image processing applications. When dealing with thousands or millions of images providing adequate storage in an image processing system can be challenging. Digital storage for image processing applications falls into three principal categories.

1. Short term storage for use during processing
2. Online storage for relatively fast recall
3. Archival storage characterized by infrequent access.

Image Displays:

Monitors are driven by the outputs of the image and graphics display cards that are an integral part of the computer system.

Hardcopy:

It is used to make a permanent copy of the images and data relating to the processed images. Devices used for recording images are laser printers, film cameras, heat sensitive devices, inkjet units and digital units such as optical CD ROM disks.

2.3 About LabVIEW

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution.

In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

All the toolsets integrate seamlessly in LabVIEW. LabVIEW is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and data acquisition control, vision, and motion control devices. LabVIEW also has built-in features for connecting ones application to the Internet using the LabVIEW web server and software standards such as TCP/IP networking and ActiveX. Using LabVIEW, one can create 32-bit compiled applications that give you the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. One also can create stand-alone executables and shared libraries, like DLLs, because LabVIEW is a true 32-bit compiler.

LabVIEW contains comprehensive libraries for data collection, analysis, presentation, and storage. LabVIEW also includes traditional program development tools. One can set breakpoints, animate program execution, and single-step through the program to make debugging and development easier. LabVIEW also provides numerous mechanisms for connecting to external code or software through DLLs, shared libraries, ActiveX, and more. In addition, numerous add-on tools are available for a variety of application needs.

Functions of LabVIEW

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.

A VI contains the following three components:

Front panel: The front panel is the user interface of the VI. One builds the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, pushbuttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

Block diagram: After one builds the front panel, he can add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures.

Icon and connector pane: Identifies the VI so that one can use the VI in another VI. A VI within another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages.

Tools Palettes: The Tools palette is available on the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. When one selects a tool, the cursor icon changes to the tool icon. Use the tools to operate and modify front panel and block diagram objects. Select Window»Show Tools Palette to display the Tools palette. One can place the Tools palette anywhere on the screen. If automatic tool selection is enabled and one moves the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the Tools palette.

Controls Palette: The Controls palette is available only on the front panel. The Controls palette contains the controls and indicators one use to create the front panel. Select Window»Show Controls Palette or right-click the front panel workspace to display the Controls palette. One can place the Controls palette anywhere on the screen.

Functions Palette: The Functions palette is available only on the block diagram. The Functions palette contains the VIs and functions one use to build the block diagram. Select Window»Show Functions Palette or right-click the block diagram workspace to display the Functions palette. One can place the Functions palette anywhere on the screen.

3 FUNDAMENTAL STEPS IN IMAGE PROCESSING

There are various different methodologies that can be applied to an image for different purposes and for different objectives. The fundamental steps that may be applied to a digital image may be classified as follows.

Image Acquisition

This is the first process of any system. It may be a simple process like obtaining an image that is already in digital form or a slightly complex process of converting an analog image to a digital one. Generally image acquisition involves preprocessing such as scaling. Scaling is used to create a series of approximations of an image.

Image Enhancement

The basic idea of enhancement is to bring out detail that is obscured or simply to highlight certain features of interest in an image. For example one may want to increase the contrast of an image which forms a part of enhancement process. Enhancement involves a lot of mathematical parameters and may be termed as one of the most complex processes of image processing.

Image Restoration

This process also involves improving the appearance of an image. However, unlike image enhancement which is subjective, image restoration is objective, in the sense that image restoration tends to be based on mathematical or probabilistic models of image degradation.

Compression

It deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Storage technology has improved significantly over the past decade with the introduction of file storage forms such as jpeg (Joint Photographic Experts Group).

Morphological Processing

This deals with tools for extracting image components that are useful in the representation and description of shape. It involves such processes as filtering, gradient, region filling, smoothing etc.

Segmentation

This involves partitioning of an image into constituent parts or objects. A rugged segmentation procedure rings the process a long way toward successful solution of imaging problems that require objects to be identified individually. On the other hand an erratic segmentation procedure almost always results in failure. Therefore the more accurate the segmentation procedure, the greater the likelihood of success.

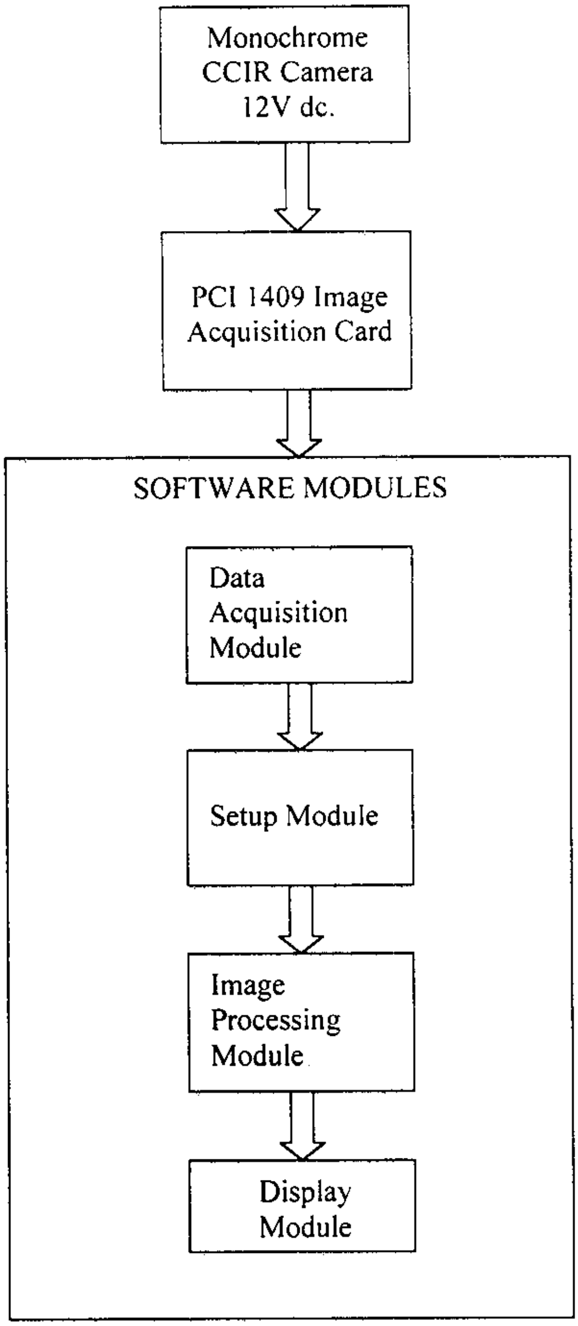
Representation and Description

This process involves the conversion of data to a form suitable for the computer to process. The first decision to be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties such as texture or skeletal shape.

Recognition

It is a process that assigns a label to an object based on descriptors. It is the final step and the output depends on the accuracy of the previous steps and processes.

4 SYSTEM MODEL:



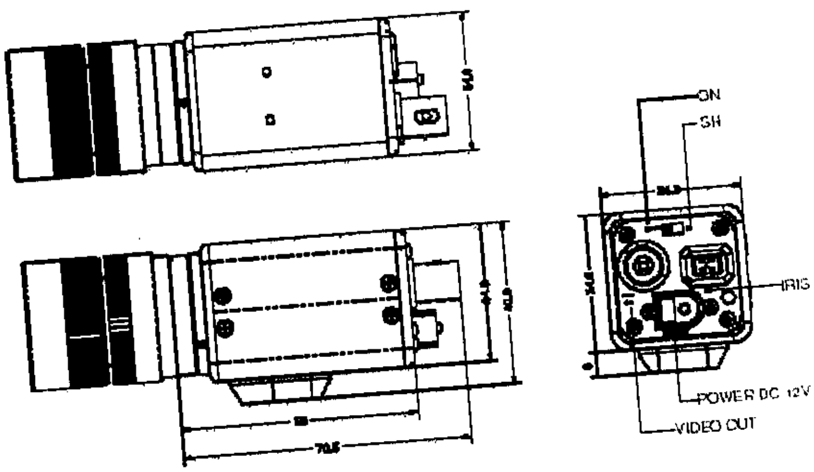
Description:

The system consists of a camera through which images are acquired. The IMAQ card is used to interface the camera with the computer. All the processing takes place in the computer. The computer analysis the images obtained from the camera for any intrusion and displays the result on the screen provided.

5 Hardware Specifications:

5.1 CAMERA FEATURES:

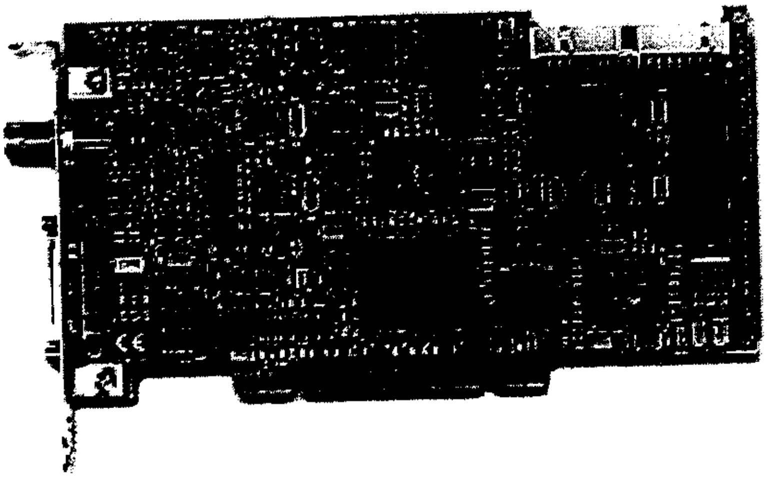
CAMERA INTERNAL DIAGRAM



5.1.1 Specifications:

Model	TCD 02C2 (Mid Resolution, SONY CCD camera)
Image Device	1/3" Inter line B/W CCD Sensor
Picture Elements	Total - CCIR 537(H) X 597(V) Effective - CCIR 500(H) X 582(V)
Scanning system	CCIR standard 625 lines, 25 frames/sec
Synchronizing system	Internal
Horizontal Resolution	More than 420 TV lines
Electronic shutter control	Switch ON/OFF, Auto-Electronic shutter (1/60 ~ 1/100000) EIA (1/50 ~ 1/100000) CCIR
Mechanical iris	Video Drive Auto Iris Lens
Gamma	0.45
Minimum illumination	0.15 Lux./F1.2-TCD-02C2
Video output level	1.0Vp-p/75 ohms
Video S/N ratio	46dB (AGC OFF)
Backlight compensation	Switch ON/OFF
Operating temperature	-10°C ~50°C
Storage temperature	-30°C ~80°C
Power requirement	12VDC ± 1.2VDC
Power consumption	DC12V Typical 100mA-TCD-02C2
Dimensions	75.5(L)X 34.6 (W)X 40.6(H)mm
Weight	142g

5.2 PCI 1409 Image Acquisition Card



The card used is a NI1409 card. It has high resolution, 10-bit digitized for cameras and sensors that offer 60 dB dynamic range. In addition, one can easily configure the NI1409 to work with standard monochrome cameras. It can also acquire colour images of stationary objects from NTSC, PAL and RGB cameras in still colour mode. The NI1409 is a calibrated device resulting in highly repeatable measurements.

5.2.1 Overview and Applications

For machine vision and scientific imaging developers who need to acquire high-resolution, measurement-quality images from standard and nonstandard cameras and sensors, the NI 1409 image acquisition hardware offers quick and easy configuration. NI 1409 devices have high-resolution, 10-bit digitization for

cameras and sensors that offer 60 dB dynamic range. In addition, you can easily configure the NI 1409 Series to work with standard monochrome cameras (RS-170 or CCIR), slow or variable rate pixel clock cameras, double-speed progressive scan cameras, and analog line-scan cameras.

5.2.2 Measurement-Quality

Image Acquisition with Calibration

NI 1409 devices are gain calibrated and delivered with a calibration certificate. Calibration ensures repeatable, consistent image acquisition for your machine vision and scientific imaging applications.

Variable Scan and Nonstandard Video

With NI 1409 devices, you can acquire images from nonstandard video, such as analog line-scan and variable-scan devices with pixel clocks that range from 2 to 40 MHz. The NI 1409 devices work with double-speed (60 frames/s) progressive scan or noninterlaced monochrome cameras from vendors such as Sony and JAI.

10-Bit Image Acquisition

Many high-resolution analog monochrome cameras have dynamic range greater than 48 dB, which means that the camera has resolution capabilities greater than 8-bit or 256 gray scales. For these high-resolution cameras, you should use an image acquisition device that offers comparable or greater resolution. The NI 1409 Series works with the highest quality monochrome cameras and delivers up to 60 dB of dynamic range, which corresponds to 10 bits or 1,024 gray scales. With NI 1409 devices, you have the option to acquire in 8-bit or 10-bit mode at very high rates, such as 60 frames/s with double-speed progressive scan cameras.

Interchange Cameras with NI-IMAQ

NI-IMAQ driver software for image acquisition can scale between many types of cameras and acquisition methods. With NI-IMAQ, you can begin by using a low-cost RS-170 camera and image acquisition device and then upgrade to a faster, higher-resolution camera and acquisition hardware with minimal software changes. Because NI-IMAQ driver software uses one set of function calls for a wide variety of cameras, there is no need to rewrite your software.

5.2.3 4-Channel Monochrome Image Acquisition

Quick and Easy Camera Configuration

You can easily configure standard and nonstandard video capture with NI Measurement & Automation Explorer, delivered with NI-IMAQ. This utility is an interactive tool for setting the camera type (RS-170, CCIR, NTSC, and PAL), programmable ROI, aspect ratio, and antichrominance filter. Use this interactive utility to set up acquisition from noninterlaced progressive-scan cameras and to create your own camera configurations for nonstandard video. Use the external lock feature to set variable-scan acquisition for microscopes and other sources that generate their own pixel clock and horizontal and vertical synchronization signals.

Image Acquisition Performance Gains with Partial Image Scanning

For improved throughput and processing bandwidth, the acquisition and ROI control circuitry monitors the incoming video signal and routes the active pixels to the FIFO. In addition to being able to digitize an entire frame, an NI 1409 can perform pixel and line scaling (decimation) and ROI acquisition. With pixel and line scaling, multiples (2, 4, and 8) of pixels and/or lines can be transferred to the PCI bus. Using ROI acquisition, you select an area in the acquisition window that will be transferred to the PCI bus.

5.2.4 Hardware PCI Interface

The PCI bus is the electrical interface for both NI PCI-1409 and NI PXI-1409. The PCI interface, implemented with the National Instruments MITE ASIC, can transfer data at a maximum sustained rate of 100 Mbytes/s in master mode to maximize the use of the available bandwidth. The interface logic ensures that the NI 1409 meets the loading, driving, and timing requirements of the PCI specification.

Scatter-Gather DMA Controllers

An NI 1409 has three independent onboard DMA controllers. The DMA controllers can transfer data between the Host memory and the onboard FIFO via the PCI bus. Each of these controllers performs scatter-gather DMA, which means the DMA controller can reconfigure on the fly, and thus perform continuous image transfers to either contiguous or fragmented buffers.

Onboard Memory

An NI 1409 has 16 MB of onboard memory used to temporarily store the image being transferred to the PCI bus. With 16 MB, you can buffer large images during image acquisition.

DAQ Synchronization

The RTSI bus or PXI trigger bus provide a flexible synchronization scheme between any National Instruments IMAQ, DAQ, or motion device and your PCI-1409 or PXI-1409, respectively.

Trigger Control and Mapping Circuitry

The trigger control and mapping circuitry routes, monitors, and drives the external and RTSI or PXI trigger bus lines. You can configure each of these lines to start or stop acquisition on a rising or falling edge, or use each line as digital I/O. You can also map the lines to onboard status values (CSYNC, Acquisition in Progress, and Acquisition Complete).

Antichrominance and Lowpass Filter

NI 1409 devices have an antichrominance filter, which can remove chrominance information from a color video signal. There are two software selectable antichrominance filters – one for NTSC color -coded signals (a 3.58 MHz notch filter), and another for PAL color -coded signals (a 4.43 MHz notch filter). Use the lowpass filter at 9 MHz to remove unwanted noise.

10-Bit A/D and Look-Up Table (LUT)

An A/D converter performs the image digitization, the result of which passes to a 1,024 by 10 RAM LUT. You can configure the LUT to implement simple imaging operations, such as gamma manipulation, contrast enhancement, data inversion, or any nonlinear transfer function.

Programmable Gain and Offset

The NI 1409 has programmable gain and offset circuitry for optimizing the input signal range.

VCR Circuitry

Because many commercial and home VCRs have noisy synchronization signals or no synchronization at all, VCRs are difficult to use with most image acquisition hardware. However, with its built-in circuitry and logic for handling video signals from VCRs, the NI 1409 can acquire monochrome images from VCRs.

Digital I/O

The NI 1409 Series includes four general-purpose digital input/output lines for general-purpose triggering and pulse generation capabilities. In addition, each NI 1409 includes four digital output lines for lighting, shutter, camera control, and pulse generation.

BNC Connector

The BNC connector supplies an immediate connection to the VIDEO0 input of The NI 1409. Use the 2 m BNC cable shipped with the NI 1409 to connect a camera to VIDEO0. You can configure the BNC connector only for RSE mode.

NI 1409 Cable Accessories

The IMAQ-A6822 cable provides 22 BNC connectors to all video signals (VIDEO0, VIDEO1, VIDEO2, and VIDEO3), the external digital I/O lines and triggers, and external signals. The IMAQ A6804 provides four bus connectors to three video signals and one trigger line. The IMAQ SCB-68 provides access to all of the signals on the NI 1409 68-pin connector.

5.2.5 4-Channel Monochrome Image Acquisition Vision

5.2.6 Specifications

Typical for 25 °C unless otherwise stated

Available Formats

RS-170/NTSC .... 30 frames/s interlaced

CCIR-601/PAL... 25 frames/s interlaced

Progressive scan....................................... 60 frames/s noninterlaced maximum

Variable scan .... 2 to 40 MHz pixel clock

Quantity.. 4 monochrome

Video 0.. Single-ended (BNC)

Video (0:3).. Single-ended or differential (D-Sub)

Input impedance .... 75. ± 1%

Bandwidth... Typical 30 MHz (-3 dB)

Input range (black and white).................... 700 MV (calibrated) or

50mV to 1.4V (variable gain)

Antichrominance filter............................... Programmable

3.58 MHz notch filter (<-25 dB)

4.43 MHz notch filter (<-25 dB)

Gray levels .... 256 (8 bits) or 1,024 (10 bits)

DNL.. ±1 LSB maximum

RMS noise .... < 0.5 LSB rms

SNR.. Typical 56 dB

Sampling rate... 2 to 40 MHz

External Synchronization and Trigger Signals

Trigger lines.. 4

Trigger sense.. TTL

Trigger polarity.. Programmable (rising or falling)

PCLKIN sense... Selectable (TTL or RS-422)

PCLKIN polarity... Programmable (direct or invert)

HSYNCIN sense.. Selectable (TTL or RS-422)

HSYNCIN polarity...................................... Programmable (rising or falling)

VSYNCIN sense.. Selectable (TTL or RS-422)

VSYNCIN polarity...................................... Programmable (rising or falling)

CSYNCIN sense.. Selectable (TTL or RS-422)

CSYNCIN level programmable (rising or falling)

Pulse width .... 20 ns minimum detectable

VIH (TTL) .... 2 V

VIL (TTL)... 0.8 V

Pixel Clock

Internally generated

Frequency range................................... 11.0 to 25.8 MHz

Adjustable pixel aspect ratio

RS-170/NTSC.. ±5%

CCIR/PAL... ±5%

Pixel jitter .... <2 ns

Lock time .... <1 frame

External PCLK frequency range................. 2 to 40 MHz

Power Requirements

+5 VDC (±5%) .... 1.25 A

+12 VDC (±5%).. <100 mA

–12 VDC (±5%).. <100 mA

Physical Dimensions

PCI .... 10.7 by 17.5 cm (4.2 by 6.9 in.)

PXI.. 10 by 16 cm (3.9 by 6.3 in.)

Environment

Operating temperature .... 0 to 55 °C

Storage temperature.................................. -20 to 70 °C

Relative humidity .... 5 to 90%, noncondensing

6 Software Modules

The software modules used may be classified as

- Data Acquisition Module
- Setup Module
- Image Processing Module
- Result Display Module

6.1 Data Acquisition Module

The images are acquired continually from a camera and are stored in a directory. The camera is a CCIR, black and white camera. The camera is focused on an area of interest. Images are continually snapped using the IMAQ card fitted into the system.

6.1.1 Image Acquisition driver software

Whether one is using LabVIEW or Measurement Studio, NI-IMAQ gives high level control of National Instruments IMAQ acquisition devices. NI IMAQ is a robust application programming interface for image acquisition. NI IMAQ performs all the computer and board specific tasks for straight forward image acquisition without register kevel programming.

NI IMAQ is compatible with NI DAQ and all other National Instruments driver software for integrating imaging into any National Instruments solution. It is an extensive library of functions that one can call from the application programming environment. These functions include routines for video configuration, image acquisition, buffer allocations, trigger control and board configuration. It provides an interface path between LabVIEW.

6.1.2 Specifications

- Comprehensive high-level driver software included with all IMAQ hardware
- Interchange cameras with minimum or no code rewrite
- Measurements and Automation Explorer for easy camera configuration
- Native operating system drivers for Windows 2000/NT/Me/9x
- Monochrome, colour, RGB and digital acquisition
- High level routines for
  Snap- single image acquisition
  Grab- Single buffer continuous acquisition
  Ring- multiple buffer continuous acquisition
- Session control
- Low level routines for
  Interface control
  Triggers
  Memory locking
  Buffer copy
- Controls up to 8 IMAQ boards
- Synchronous data acquisition and motion with RTSI

The data acquisition module is responsible for obtaining data from the camera and feeding it into the processing unit. This is done with the help of the snap function.

IMAQ Snap

This function acquires a single frame image. Used in a loop for repeated image acquisition. On initialization, it triggers the IMAQ card to obtain the latest frame that is in view from the camera. The speed of snapping may be adjusted depending on the speed of the camera.

Figure 1: Reference image

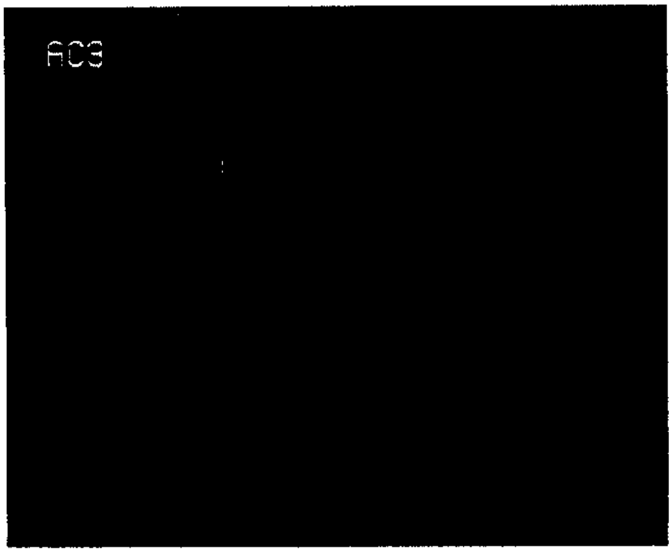


Figure 2 : Intrusion by a dog

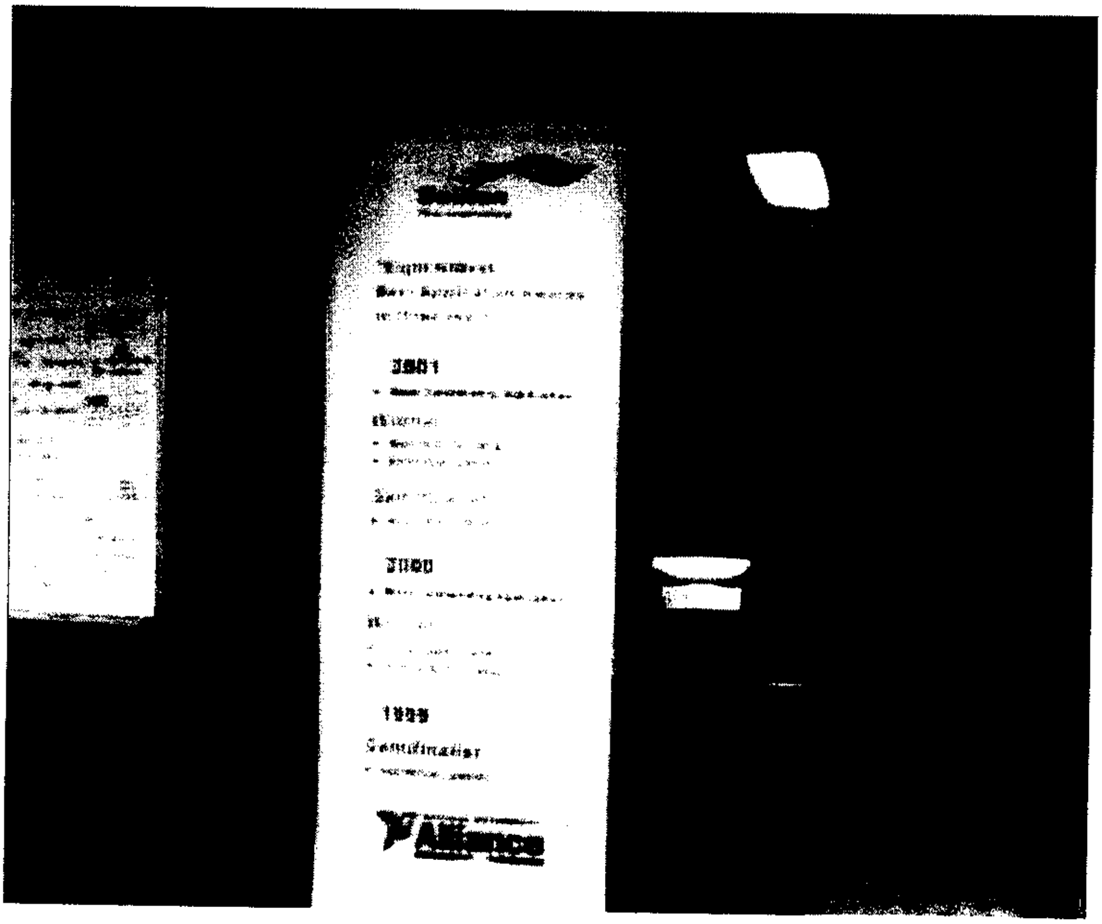


Figure 3 : Reference image

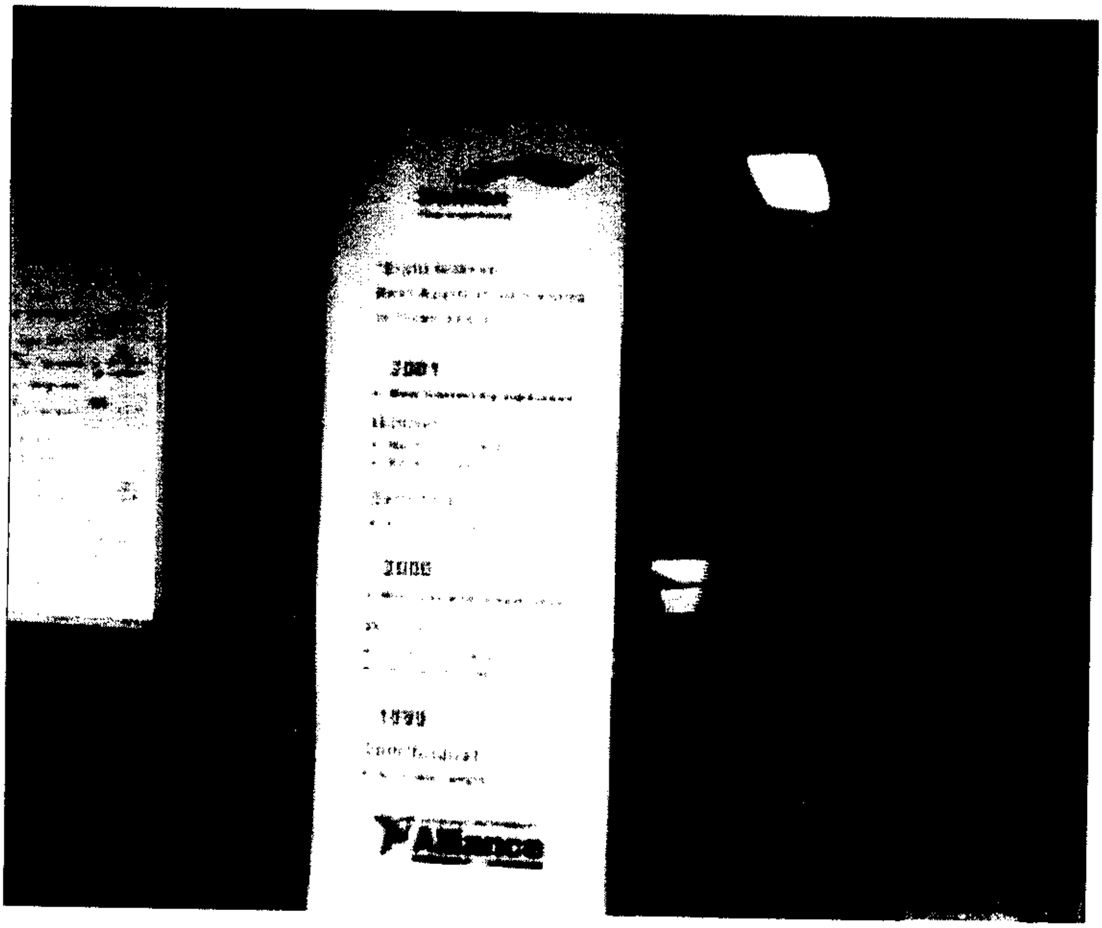


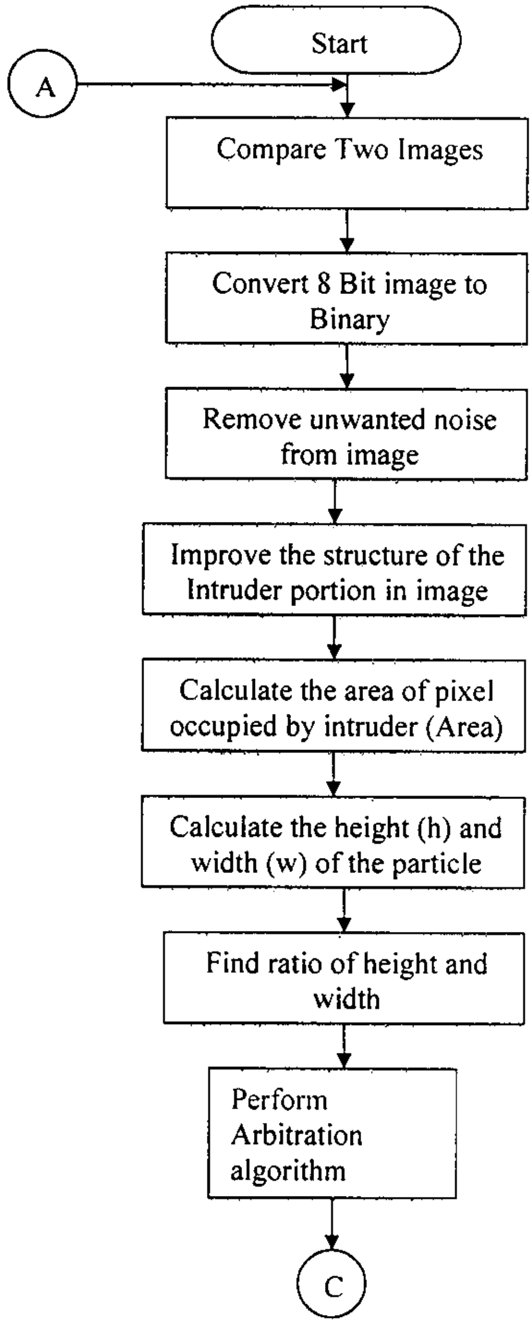
Figure 4: Intrusion by human

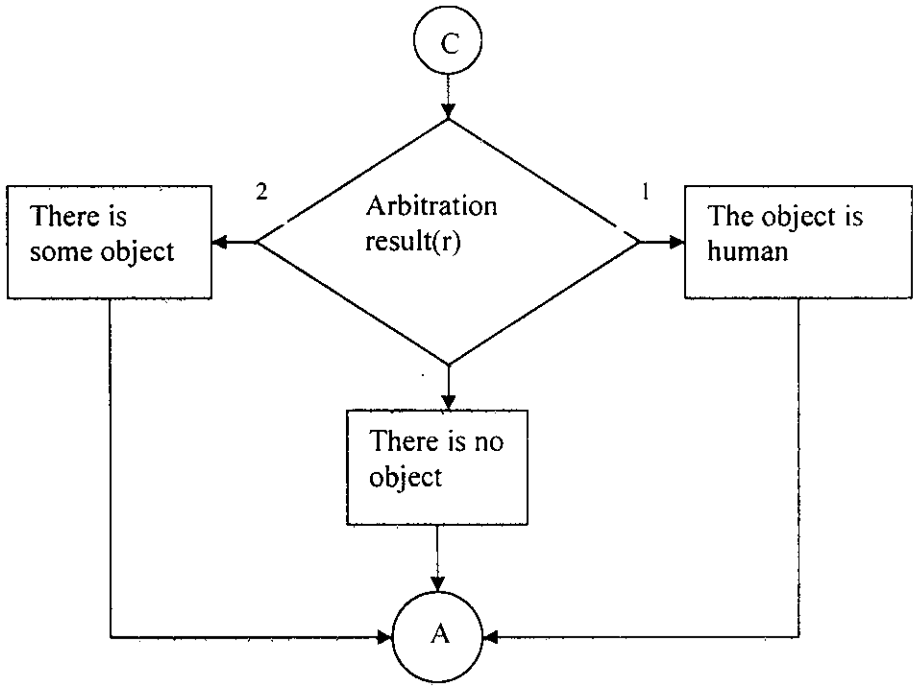
6.2 Setup Module

The setup module allows the user to obtain a reference image through the camera to be compared with the forthcoming images. While obtaining a reference, one must ensure that there is no foreign presence in the background. One must also ensure that once the reference image is obtained, the position of the camera must not be changed. This is to minimize any kind of disturbance that may enter the system otherwise.

6.3 Image Processing Module

6.3.1 Flowchart:





The objective of this project is to acquire images from a camera focused on a particular predetermined area and to check for intrusion of any kind. It also involves differentiating between a human intrusion and intrusion by animals. The code consists essentially of four basic units namely

- Comparison Unit
- Image Processing Unit
- Arbitrary Unit
- Display Unit

The images acquired from the camera are compared with a reference image that is stored previously or set using the Set Reference button and to extract any difference in the images. This difference is then processed to determine whether the intrusion is of a human being or any pet.

6.4 Comparison Unit

The fundamental purpose of this sub VI is to compare the images obtained from the camera with that of the reference image. The value of each pixel is in the form of 8-bit value (i.e., the value of each pixel ranges from 0-255 decimal value). This unit gives out the absolute difference of the images.

Compared image = |V (image reference) – V (acquired image)|

It consists essentially of three VIs.

IMAQ Create- Used to create an image file.

IMAQ Read file- Used to read an image file which may be of various formats.

IMAQ Absolute Difference- Compared the two images and returns the absolute value

It is however not always very accurate. For a moving background, there may be a lot of noise that may occur in the background. These noise particles must be filtered before any processing takes place

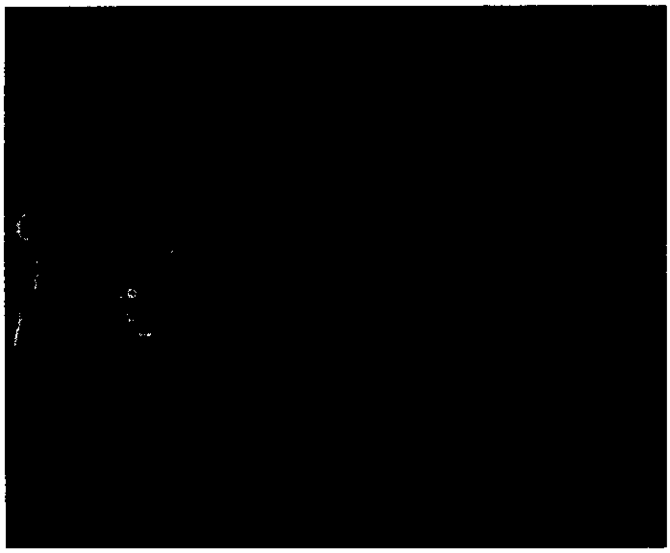


Figure 5: Difference image (for dog)

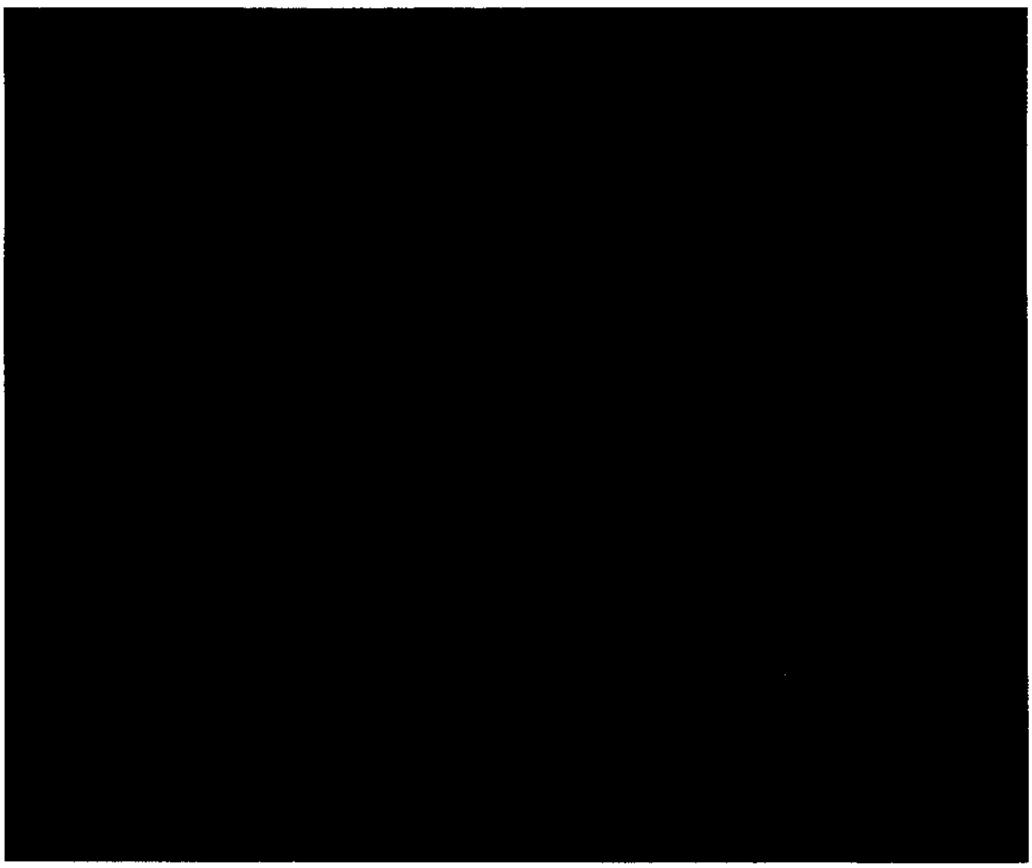


Figure 6: Difference image (for man)

6.5 Image Processing Unit

This is the unit wherein most of the processing takes place and where the extraction process of any intrusion is separated from the rest of the image. Further image enhancement is also carried out here.

6.5.1 Threshold:

The difference image that is obtained is an 8-bit image. The value of each pixel varies from 0-255. Consider that a foreign object is added to the background. The difference value of the object with the background is greater than with the rest of the image where the image is unchanged. Hence pixels corresponding with the object in the image occupy a value towards the 255 range whereas the pixels in which there is no change in the background occupy a value closer to the 0 range.

Therefore in order to extract the foreign object, a threshold T is applied. For all points having value greater than T, the pixel values are converted to binary 1. For all points having a value less than T, the values are converted to binary 0. In other words the object portion is converted to an array of 1s and the undisturbed portion is converted to an array of 0s.

Mathematically it may be calculated as

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T \\ 0, & \text{if } f(x, y) < T \end{cases}$$

Where,

f (x, y): 8 bit pixel value of difference image

g (x, y): threshold value of the difference image.

The type of thresholding carried out here is called as global thresholding. The threshold value is obtained by trial and error method. There is however another method to obtain the threshold value by heuristic approach. The method involves computing the average pixel values for the different groups of previously computed image and then obtaining the average of the two values to obtain the new threshold.

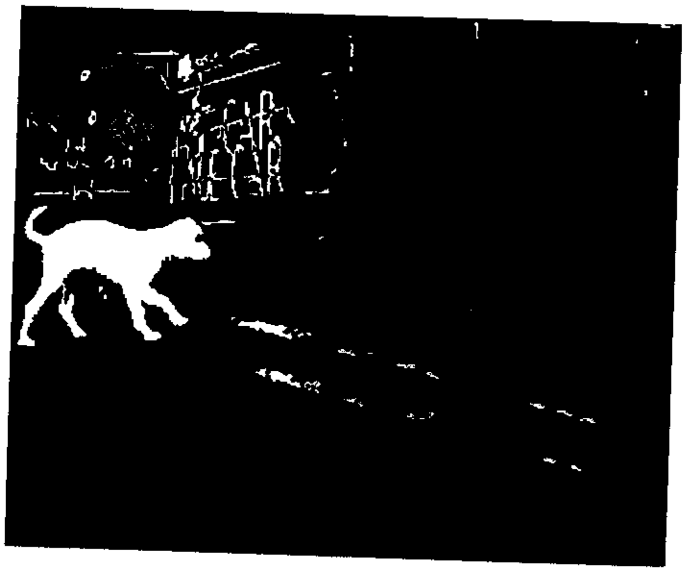


Figure 7: Threshold image (for dog)

Figure 8: Threshold image (for man)

IMAQ Cast type

This VI converts the current image type of an image to the image type specified by Image Type. If a lookup table is specified, IMAQ Cast Image converts the image, using a lookup table.

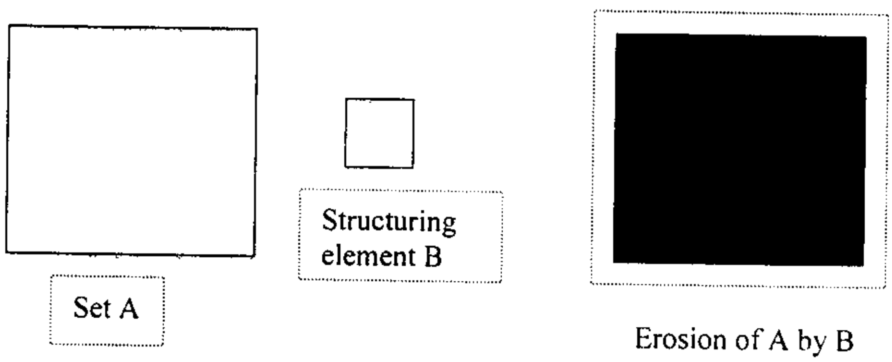
6.5.2 Remove Particles:

This VI removes particles subjected to 3X3 erosions. An erosion of set A by B may be denoted as

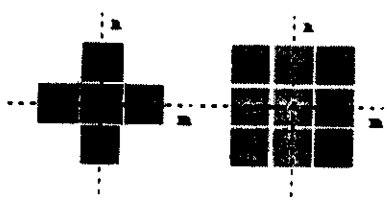
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

Erosion is also called Minkowski subtraction. The erosion of A by B is the set of all points z such that B is contained in A. Consider the set A and structuring element B as shown in the figure. The locus of the points within a boundary constitutes erosion of A by B. The most common use of this is removing irrelevant details.

This can be done by eroding the image with the structuring element of a size somewhat smaller than the objects one wishes to keep. A structuring element of 3X3 is used in this module which signifies that all objects within this boundary are removed.



The structuring elements commonly used are



(a) N_4 (b) N_8

The standard structuring elements N_4 and N_8.

In simple words erosion may be explained as each binary object pixel (with value "1") that is *C*-connected to a background pixel is chosen and the object pixel value is set to "0".

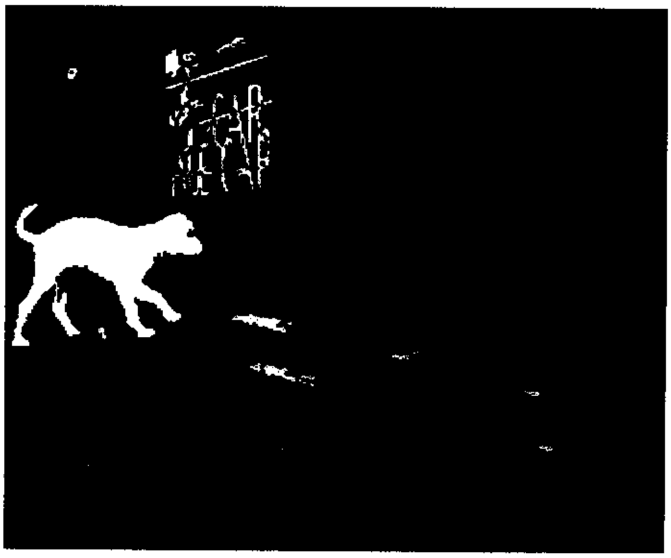


Figure 9: Image after removing particles (for dog)

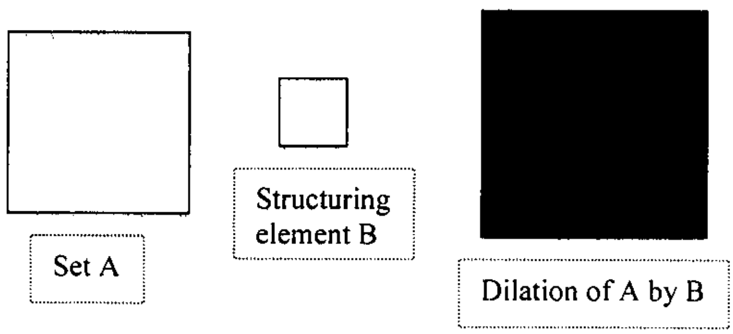
Figure 10: Image after removing particles (for man)

6.5.3 Morphology (Dilation):

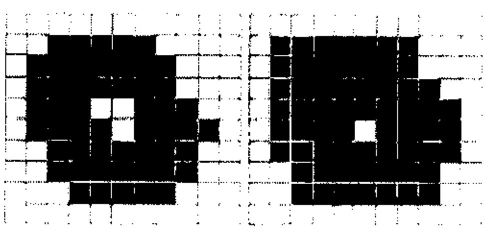
The word morphology denotes that the processing involved is on the structure and the shape of the object. Dilation may be explained as follows. Consider two sets A and B. The dilation of A by B is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq 0\}$$

Dilation is also called Minkowski addition. The dilation of A by B is the set of all displacements z such that B and A overlap by at least one element. The set B is called structuring element. In simple terms dilation is the process of 'flipping' B about it origin and then successively displacing it so that it slides over image A.



The need for dilation is to bridge broken gaps between images. Sometimes the image that has been captured may not be very distinct. Or due to a small amount of disturbance, the image may get bifurcated into two objects. In order to correct such mistakes, dilation is undertaken.



(a) $B = N_4$ **(b)** $B = N_8$

Illustration of dilation: Original object pixels are in gray; pixels added through dilation are in black.

In simple words dilation may be explained as each binary object pixel (with value "1") is chosen and all background pixels (with value "0") that are *C*-connected to that object pixel are set to the value "1".

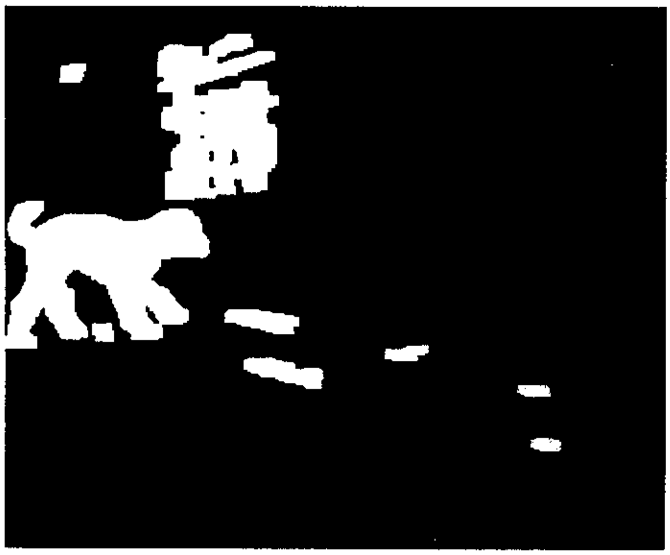


Figure 11: Dilated image (for dog)

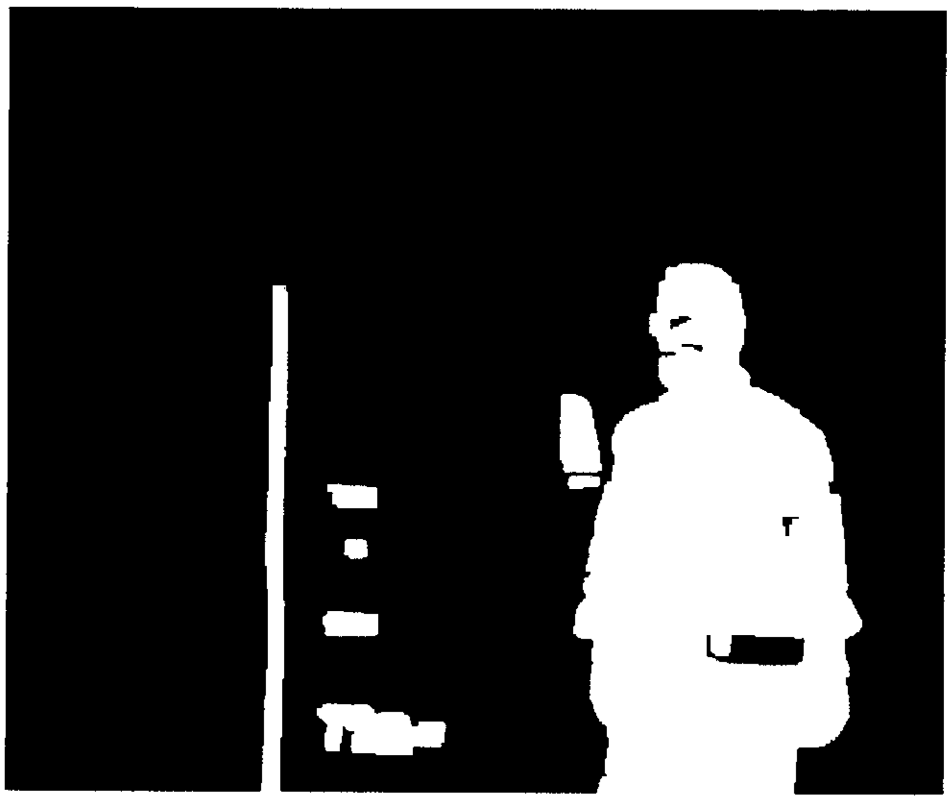


Figure 12: Dilated image (for man)

6.5.4 Fill holes

This VI's function is to fill areas surrounded by a particular value of pixels with the same. For example if there is an area of zeroes surrounded by area of 1, then the pixels with value zero are converted to one. The process is similar to dilation except that the filling is limited to within the border.

Beginning at a point p inside the boundary the objective is to fill the entire region with 1s.

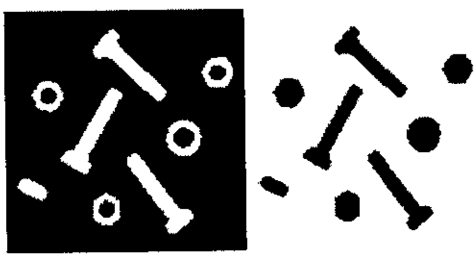
$X_k = (X_{k-1} \oplus b) \cap A^c$, k=1, 2, 3....

Where

X_0 = p, and B is the symmetric structuring element. The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The algorithm is similar to the dilation process except that here there is a limit A^c set so that the filling is limited within the boundary. This condition is called as conditional dilation.

To fill holes in objects we use the following procedure

i) Segment image to produce binary representation of objects
ii) Compute complement of binary image as a mask image
iii) Generate a seed image as the border of the image
iv) Propagate the seed into the mask.
v) Complement result of propagation to produce final result



a) Mask and Seed images b) Objects with holes filled

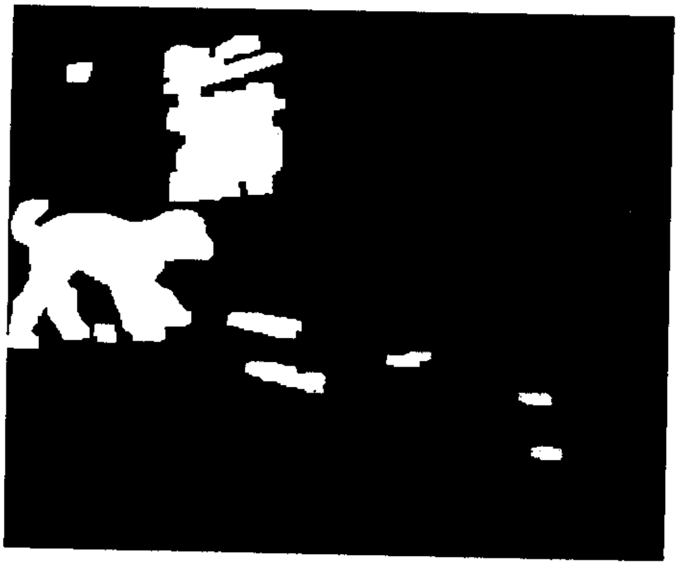


Figure 13: After filling holes (for dog)

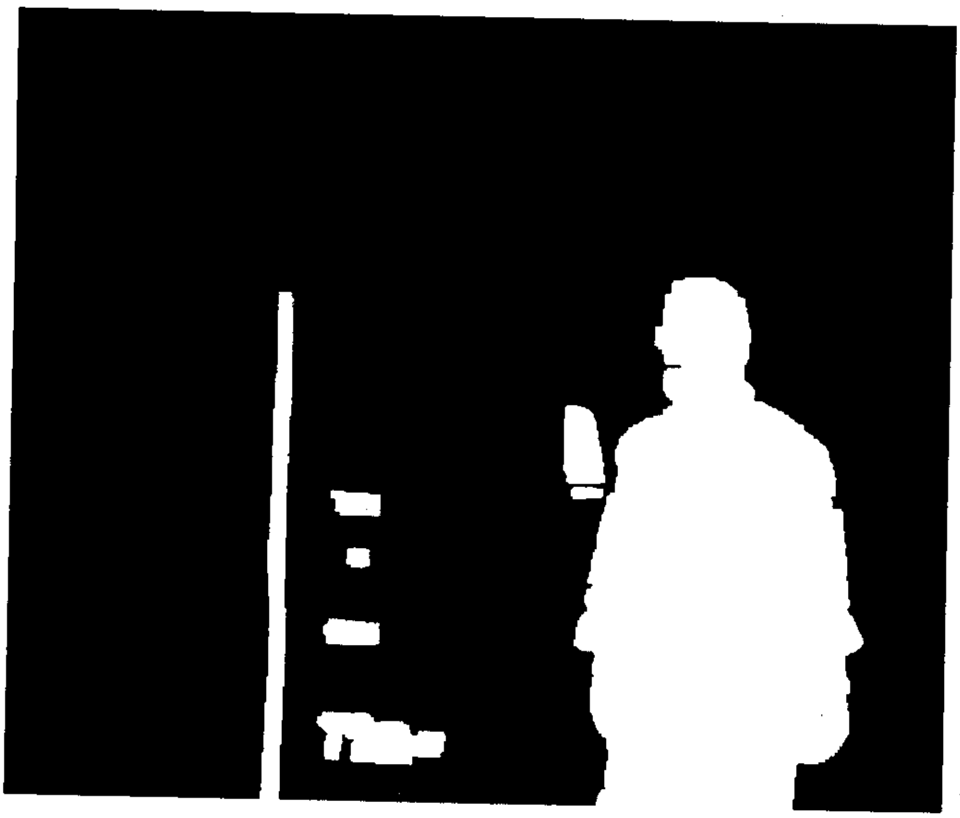


Figure 14: After filling holes (for man)

6.5.5 Particle filter

This VI performs a particle filter by removing objects of minute areas defined in the selection values. This is performed by comparing the area of each object with that of the selection values. If the selection values are greater than that of the object, the object is removed or in other words replaced with 0s. This helps to filter out any unnecessary noise that may be present. In other words only large objects are allowed to be kept and smaller objects are deleted.

Mathematically it may expressed as

If $A < T_0$, $A = 0$

Where,

A – The area of the selected (number of pixels)

T_0 – The selection value

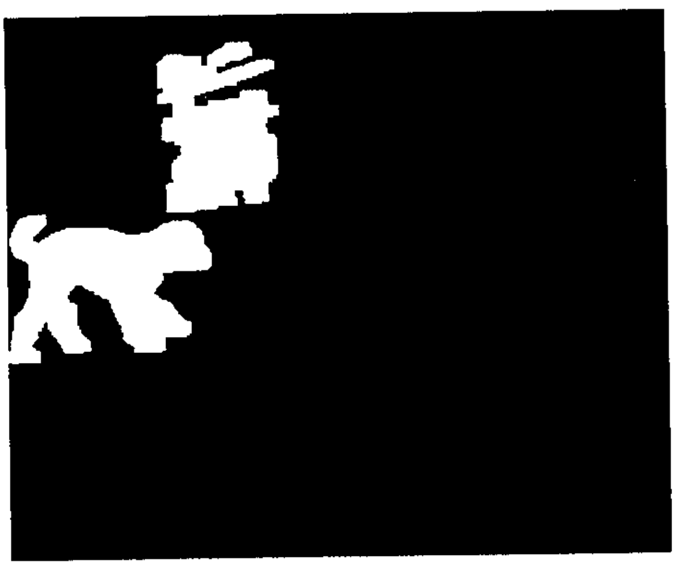


Figure 15: After particle filter (for dog)

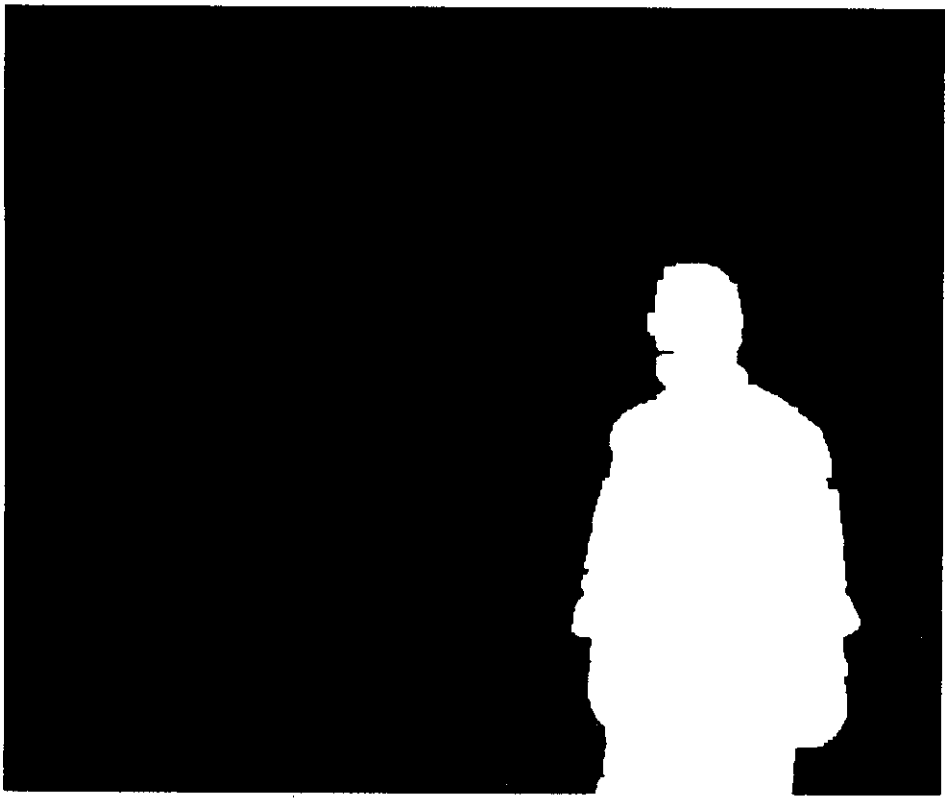


Figure 16: After particle filter (for man)

IMAQ Complex Particle

This function detects objects of various sizes and returns the number of objects present along with their pixel count.

IMAQ Complex Measure

This VI returns the measure of the objects based on the type of measure selected. The measure may be number of pixels, number of holes, width of the particle, height of the particle etc.

IMAQ Get Palette

This VI selects the palette required to display the image. The palettes available are gray palette, binary palette, rainbow palette, gradient palette, and temperature palette.

IMAQ Windraw

The function of this VI is to display the image that is input to it.

6.6 Arbitrary Unit

After the intrusion is detected, and image enhancement is undertaken, the next step is to determine whether the intrusion is that of a human being or not. This is where the arbitrary unit comes into the picture.

The area (A) of the object (in terms of number of pixels) is known from the pixel count obtained previously. Also the height and the width (h/w) of the object are known. If an area of Th1 pixels or greater is encountered, then it is straightaway evident that the intruder is a human being. This value is reached by trial and error method that has been performed on various people. A pet cannot have such a large area.

$A \geqslant Th1$, the object is human

Where

A – Area in terms of number of pixels

Th1 – threshold pixels

The next check is for pixels of size less than Th1. Here two possibilities arise. The human being is small or the intrusion is from a pet. Therefore we check for the height and width ratio. If the height and width ratio is considerably greater and the area is greater than Th2 pixels then the intruder is a human. This measure is also ascertained by trial and error method. If these conditions are not satisfied then the intrusion is that of another object.

$A \geqslant Th2 \cap h/w > Th3$, the object is human

Where

A – area of the object

Th2 – threshold pixels

h/w – height to width ratio of object

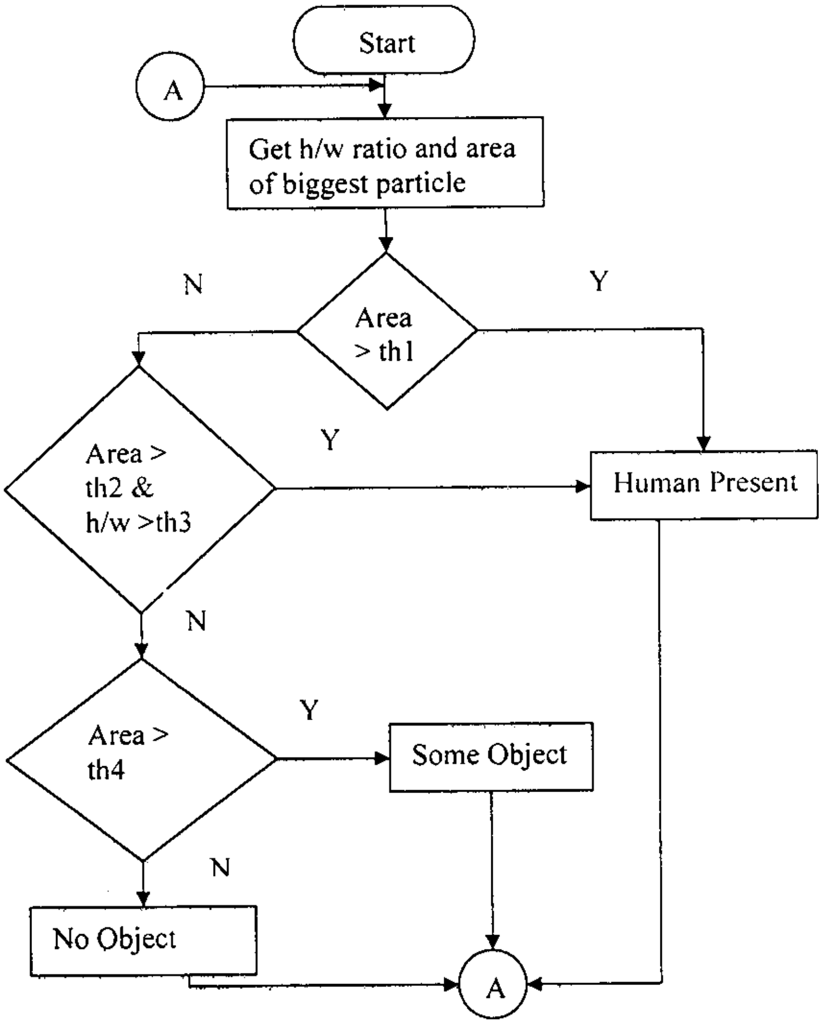
Th3 – threshold height to width ratio.

To ensure whether the object is of any significance, another threshold th4 is used to ascertain this.

$A > th4$, Object is present

Else there is no Object.

6.6.1 Flowchart for Arbitrary Unit



6.7 Display Unit

The purpose of this unit is to display the image that is input into it. In the demo mode, the images are already stored in a predefined directory. The system displays the images one by one by reading from the directory. In case of the normal mode, the system displays the images acquired from the camera through the IMAQ card.

IMAQ overlay rectangle:

The display unit also differentiates any intrusion from the rest of the image by enclosing the intruder in a red rectangle called as overlay. This is done by considering the end points of the image – top edge, bottom edge, left edge, and right edge of the image. Then the system draws a rectangle over this portion thereby enclosing the object within the rectangle.

6.8 End User Interface:

The system is designed to be very user friendly. Even a layman can operate the system without any difficulty. The end user interface presents a screen to the user. Initially when the program is run, the system offers two options.

- Demo Mode
- Normal Mode.

Demo Mode:

In this mode the system shows the user how it actually works by running a predefined demo. An example program is run so that one may get an idea as to what the system does when it encounters a human being or a pet. A pull down menu enables the user to select the type of demo to be run either human or pet.

Normal Mode:

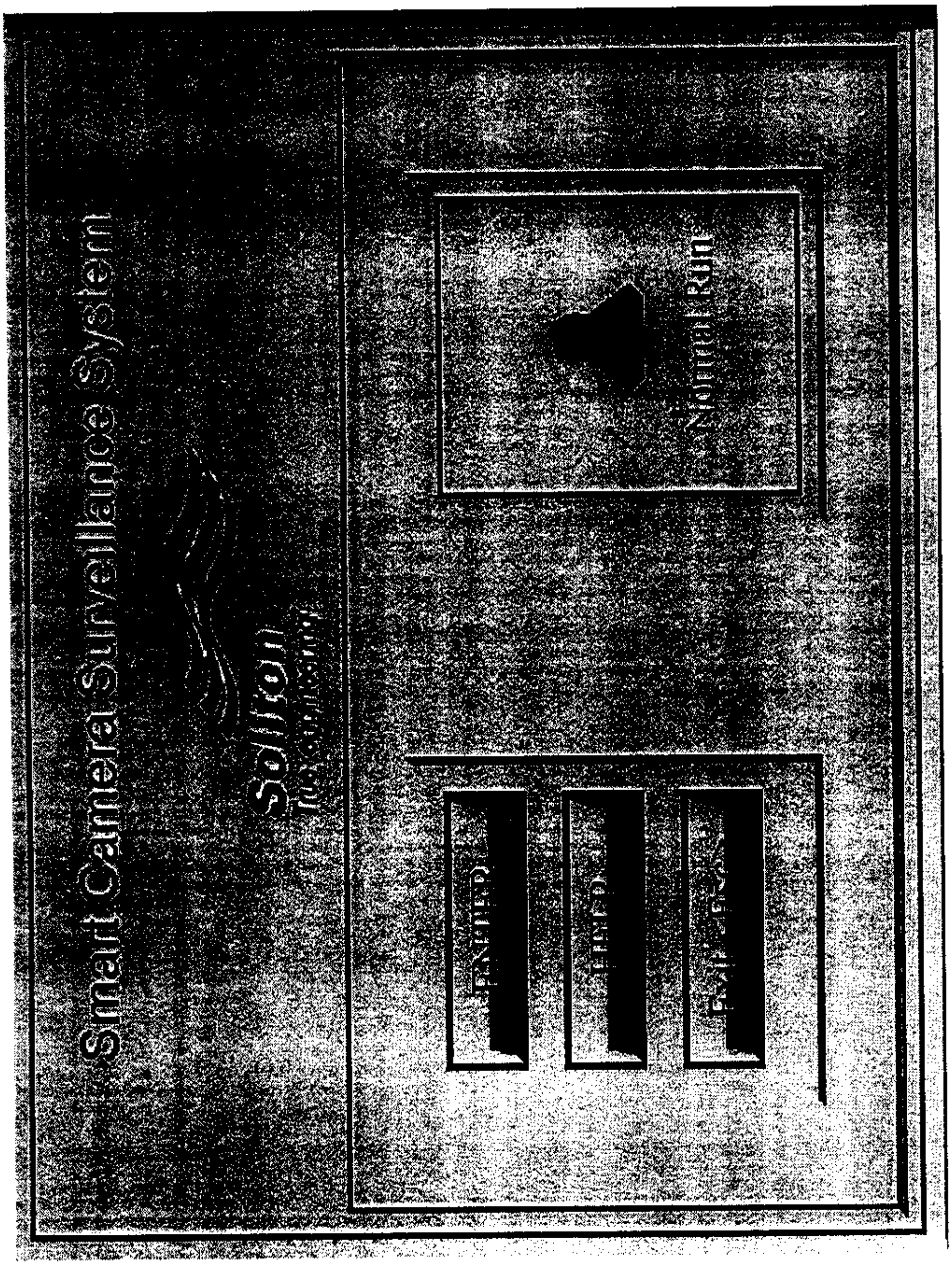
In this mode, one may actually interface a camera and operate the system. Before entering into this mode, one must ensure that the camera is connected to the system. The reference image may be selected by the user by selecting the Set Reference option. The camera is focused on the area of interest and on depressing the Set Reference button, the image acquired at that instant is used as the reference image. Now if any person walks around the area where the camera is focused, then the system immediately identifies him as an intruder.

Help:

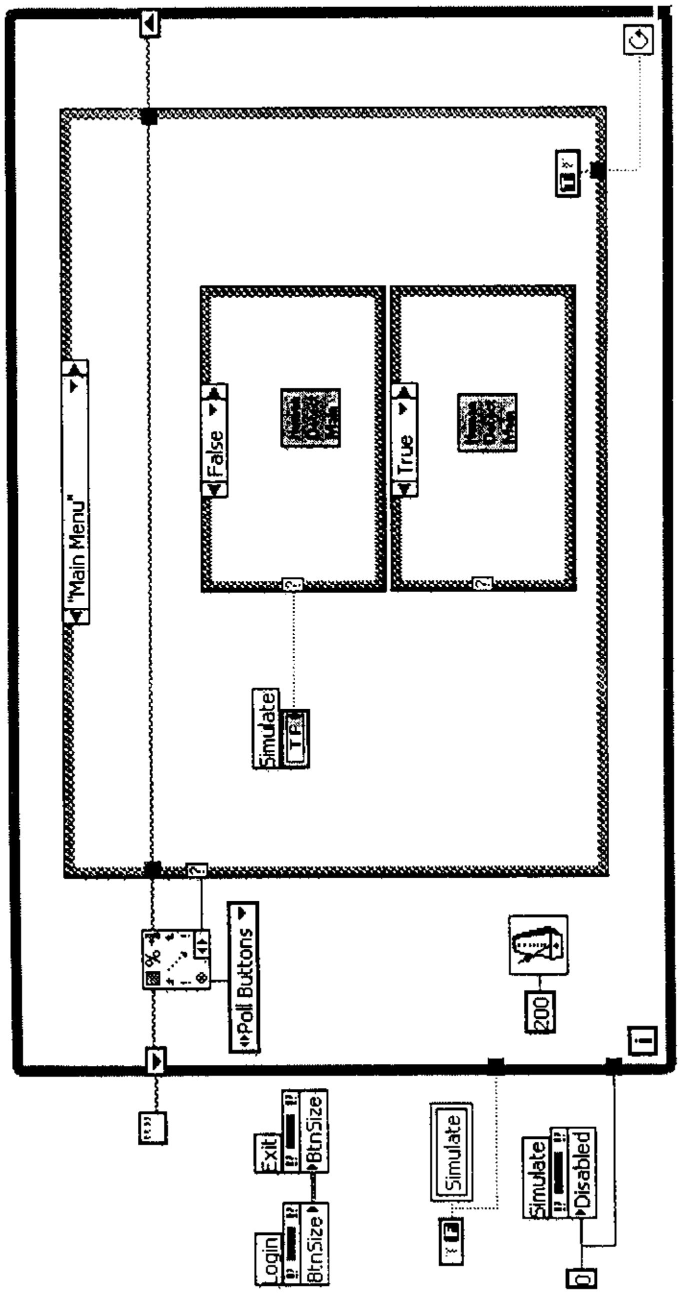
Even if one is unable to understand how to operate the system, he/she may use the help option provided in each screen.

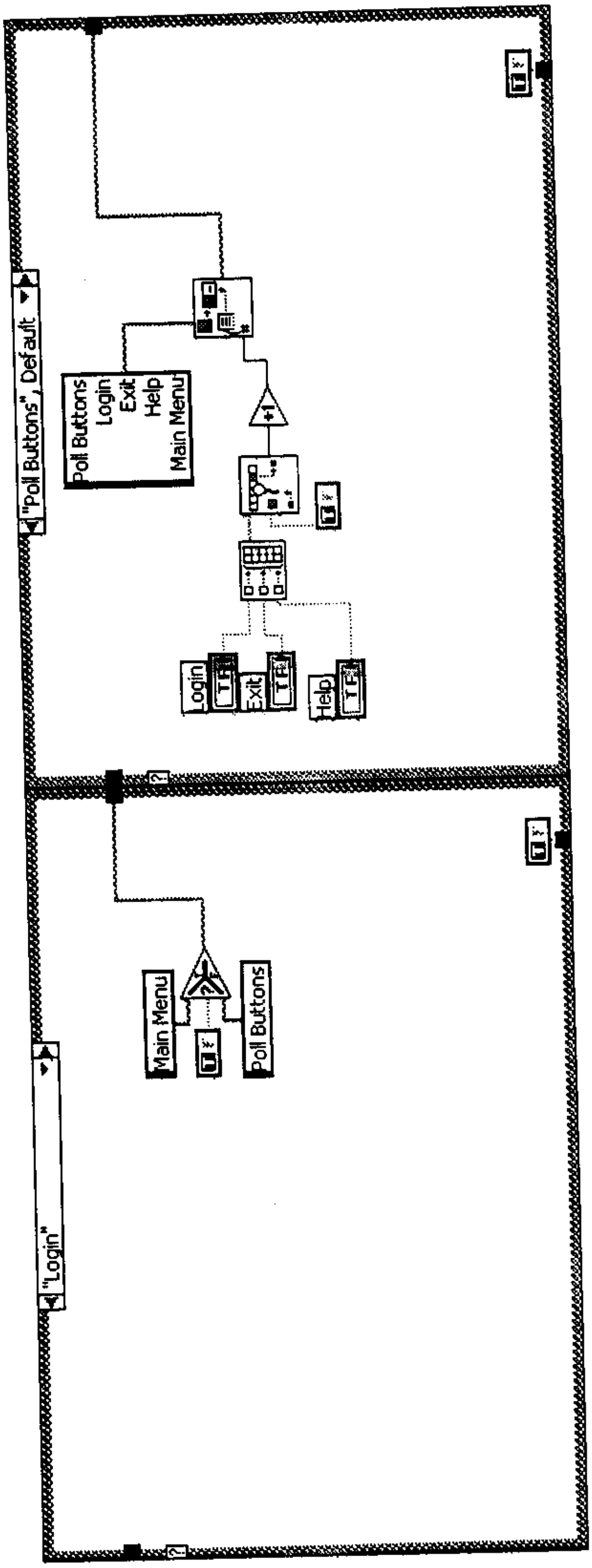
7 Software Codes:

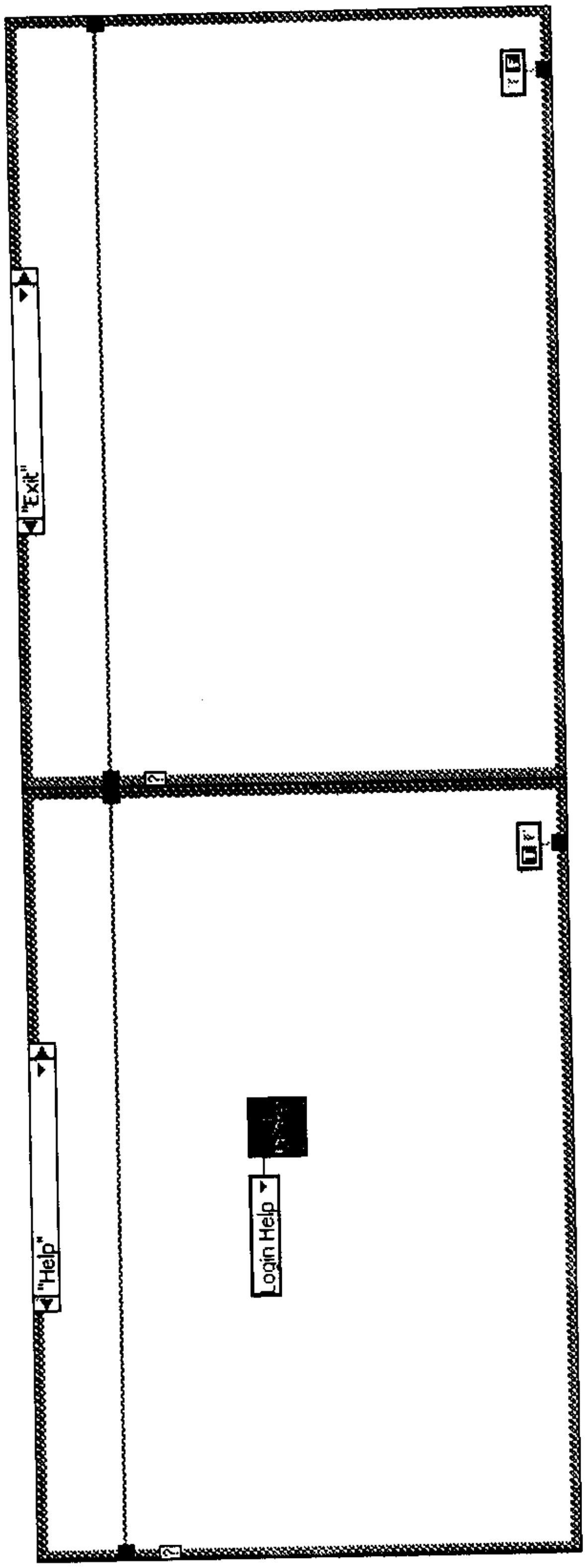
Front Panel



Block Diagram

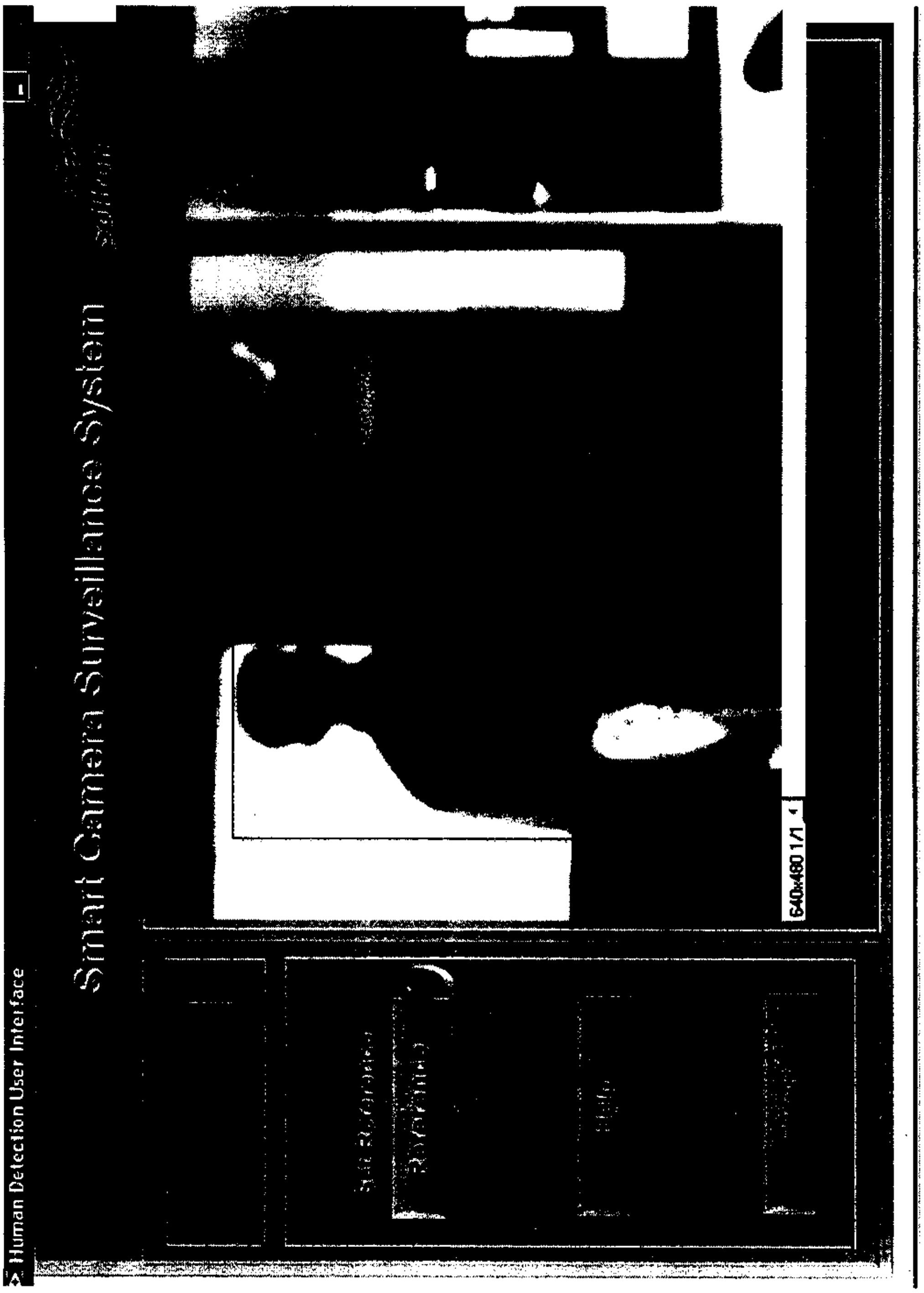



"Poll Buttons", Default
Poll Buttons
Login
Exit
Help
Main Menu
Login
Exit
Help
"Login"
Main Menu
Poll Buttons

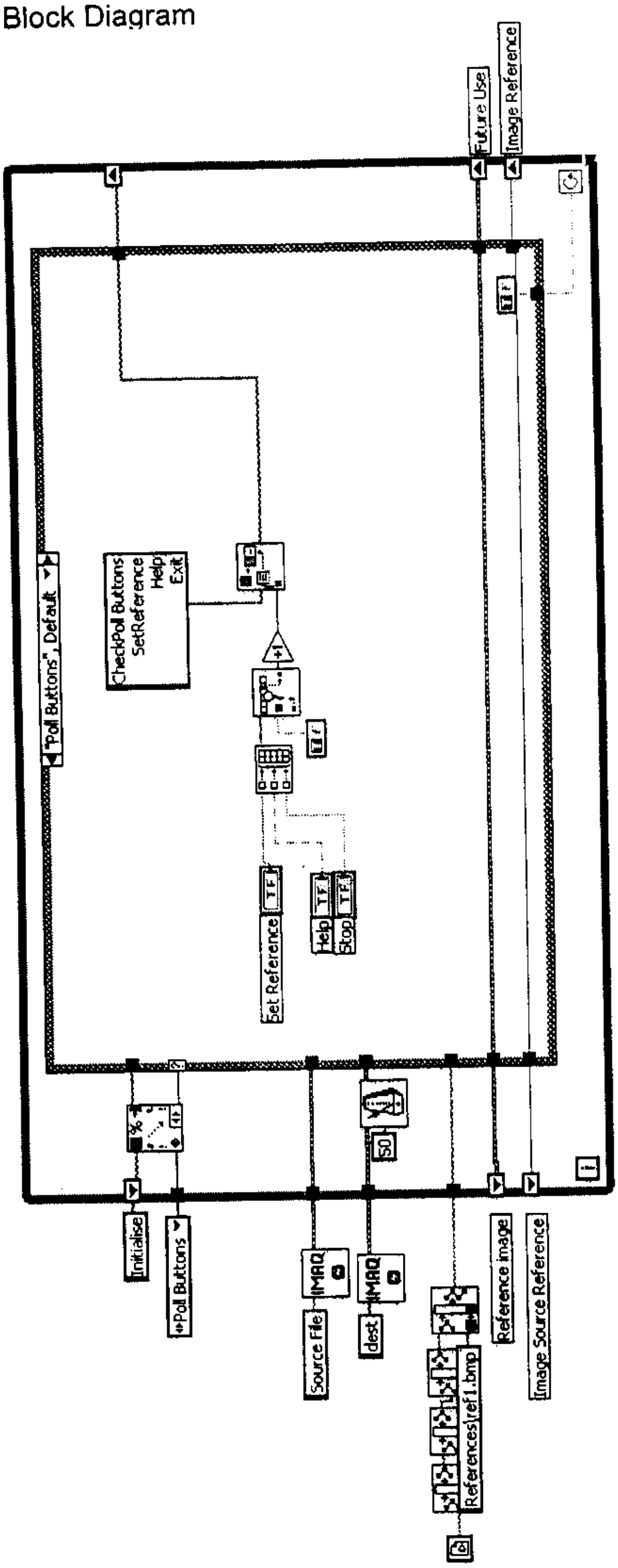

"Exit"
"Help"
Login Help

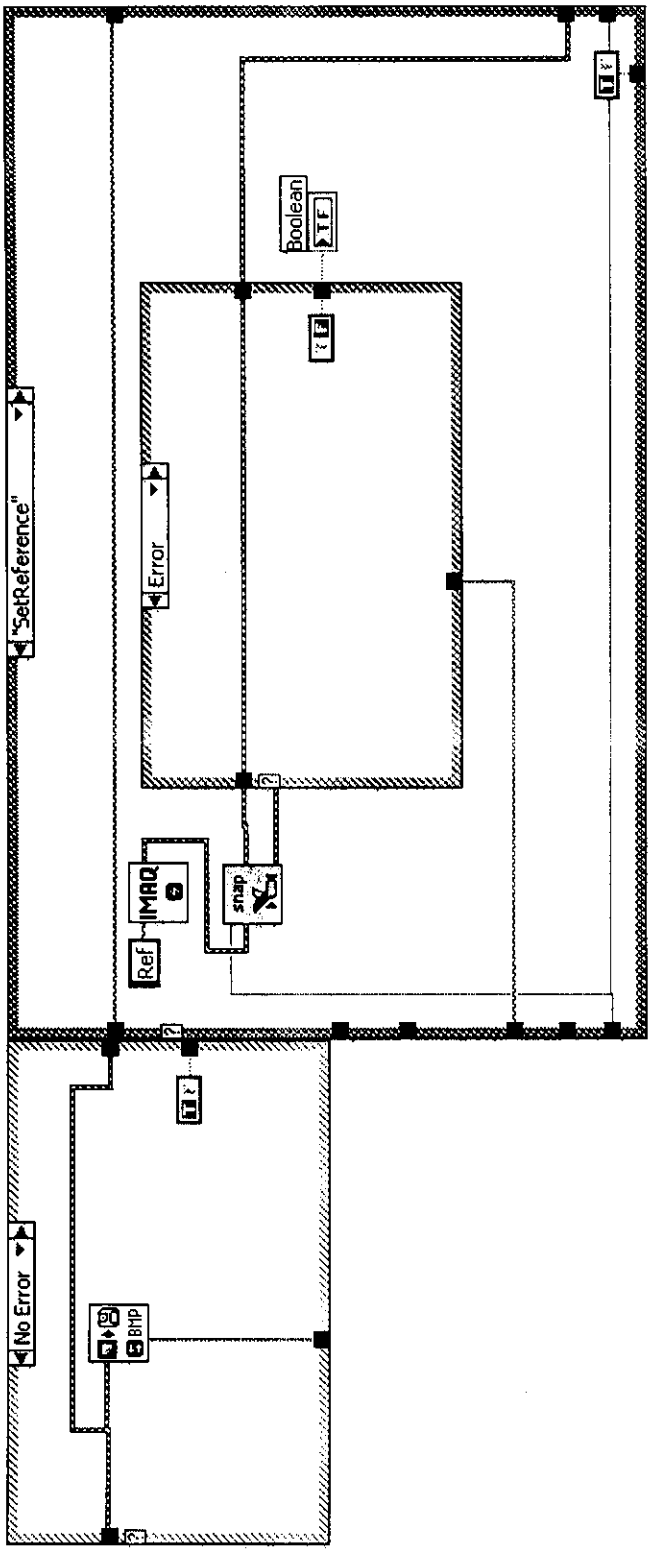
Human Detection Main User Interface. vi

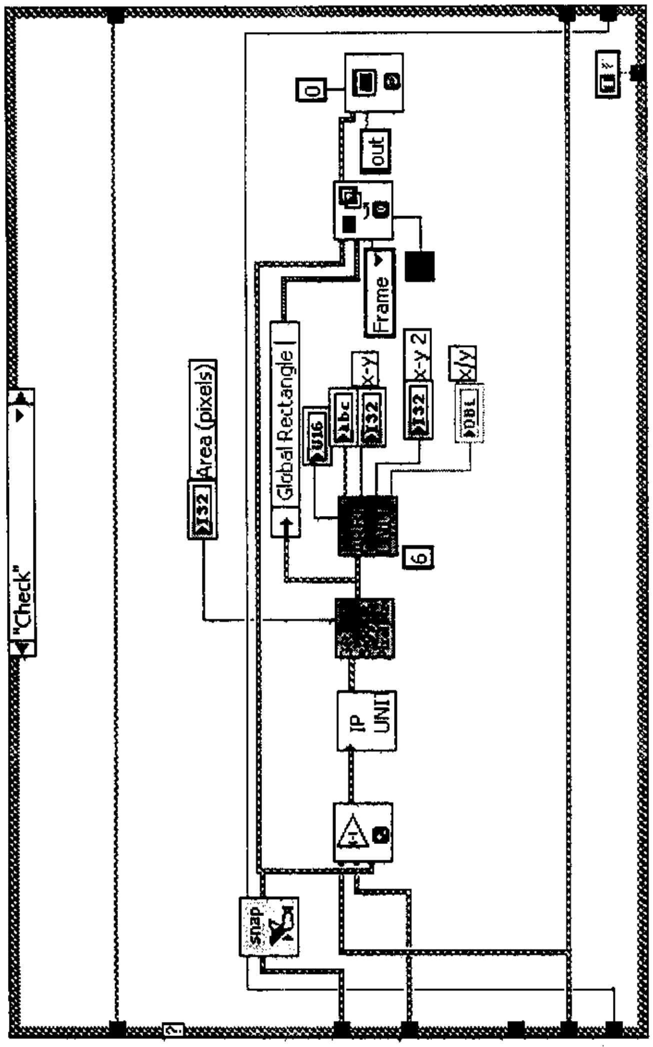
Front Panel

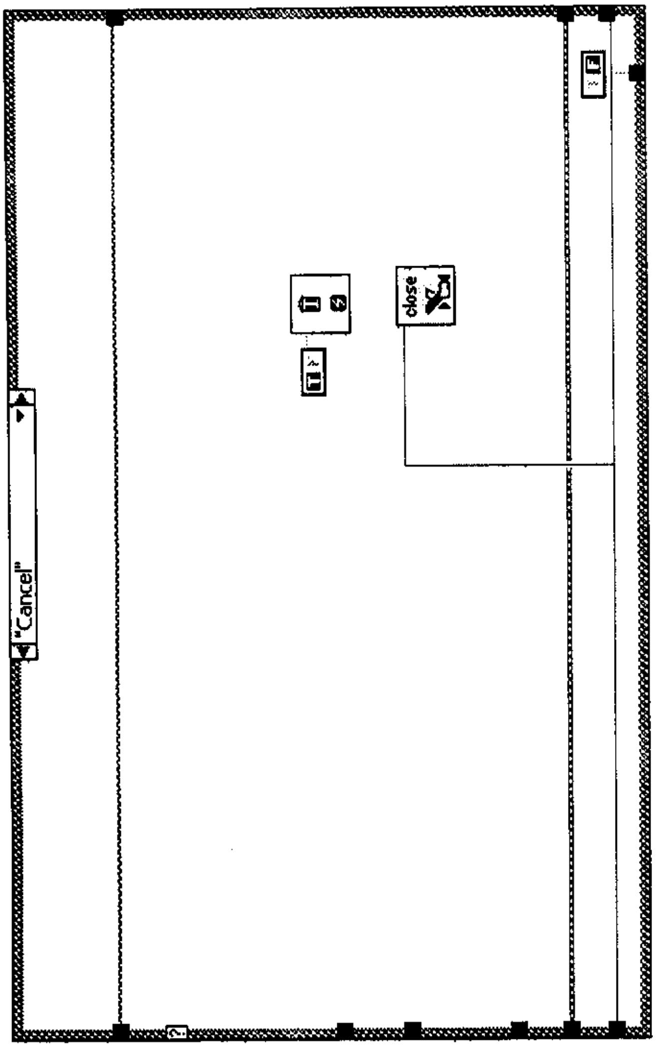


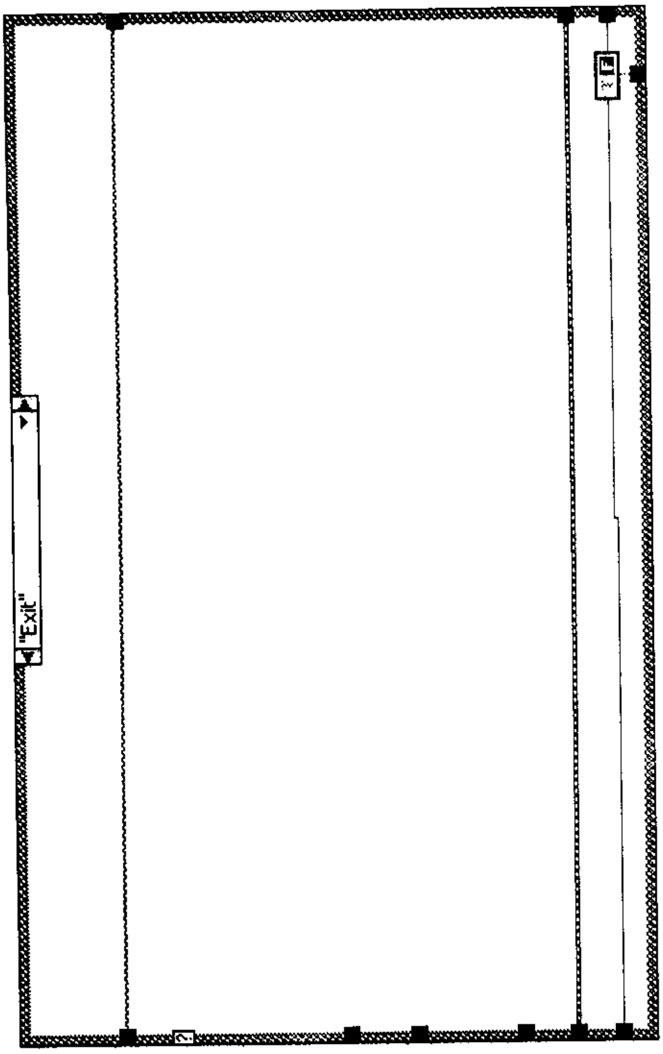
Block Diagram

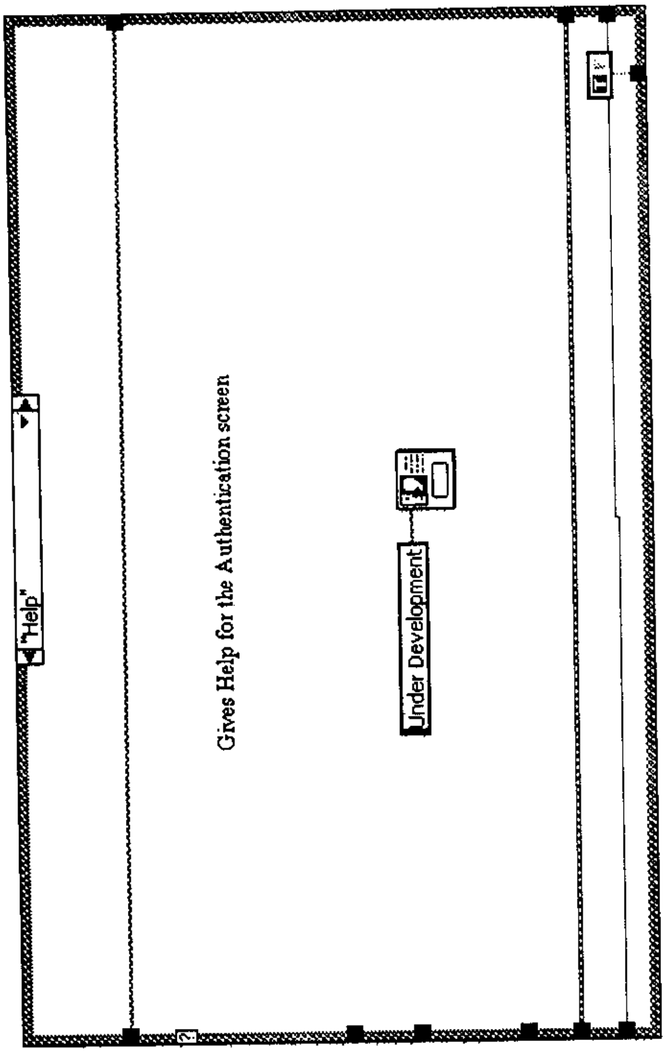


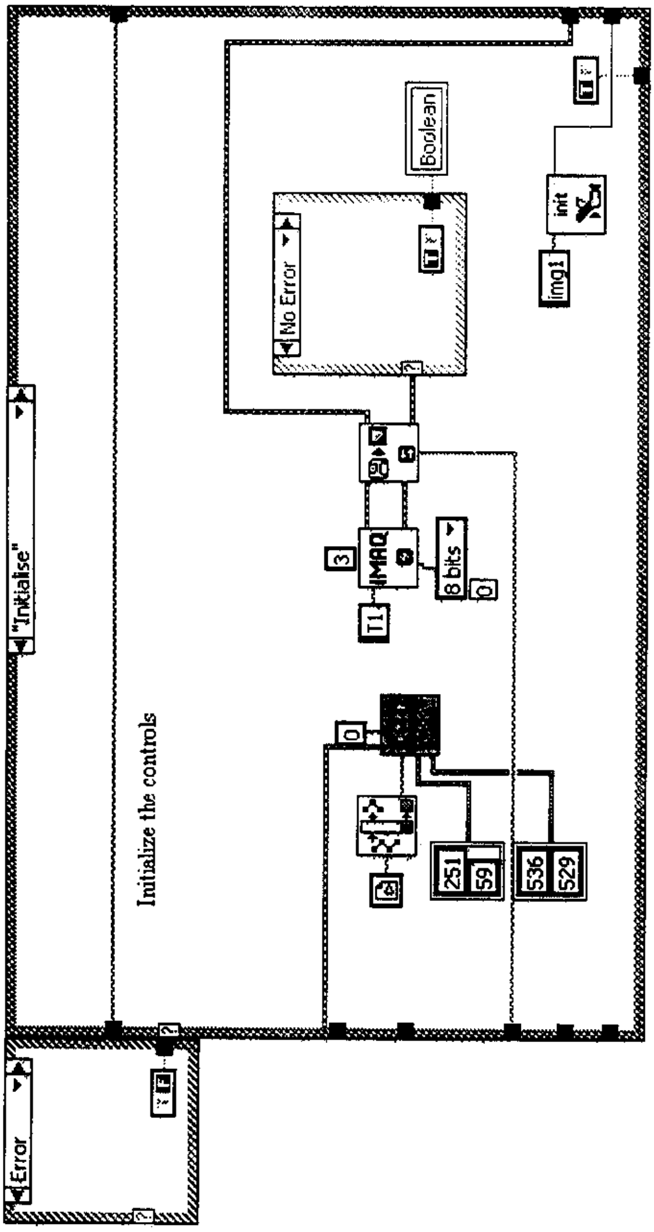

"SetReference"
Boolean
Error
IMAQ
Ref
snap
No Error
BMP


"Check"
Area (pixels)
I32
Global Rectangle 1
U16
abc
I32
x-y
I32
x-y 2
DBL
x/y
6
Frame
out
0
snap
IP
UNI


"Cancel"
close

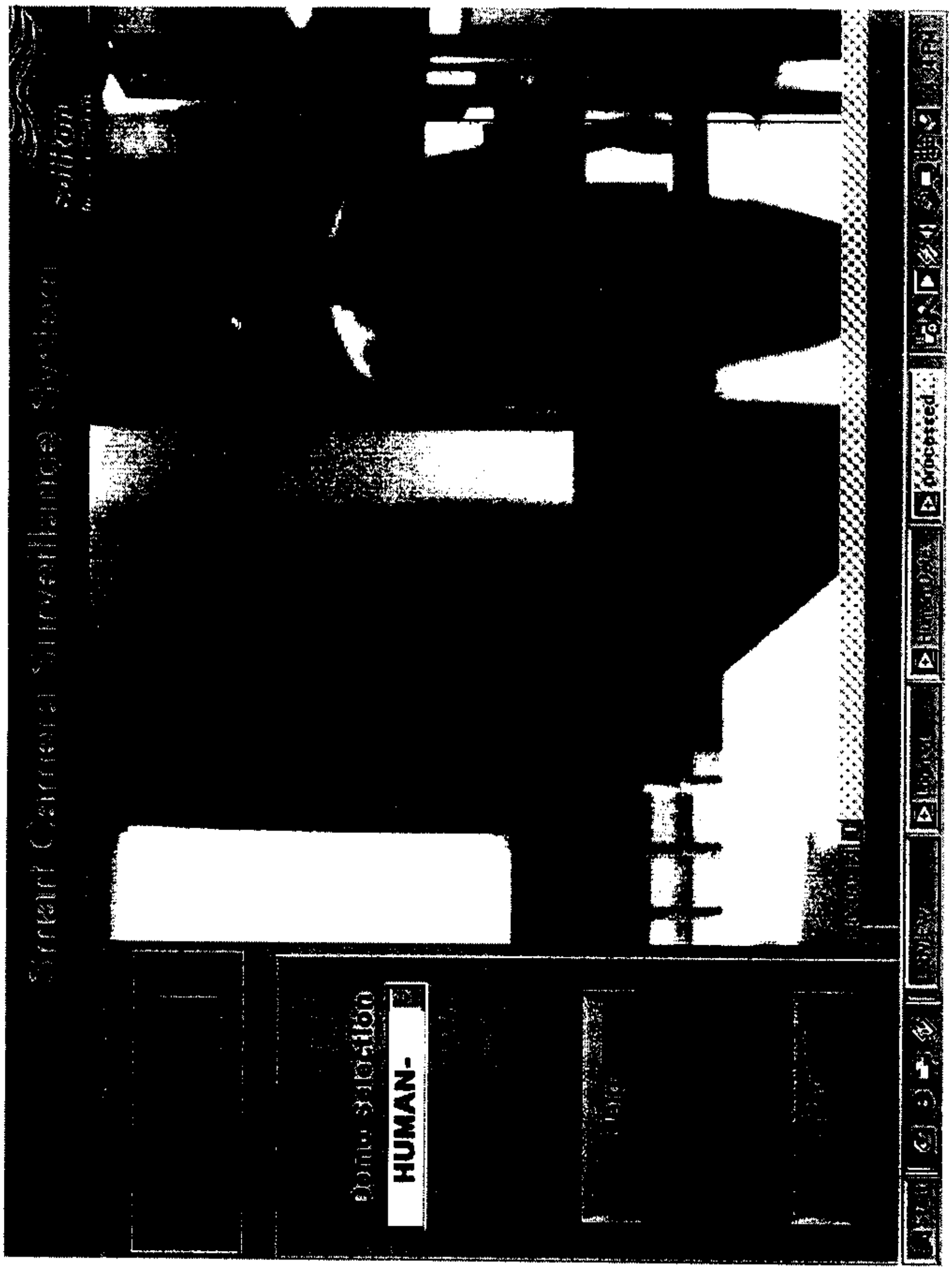

"Exit"


"Help"
Gives Help for the Authentication screen
Under Development

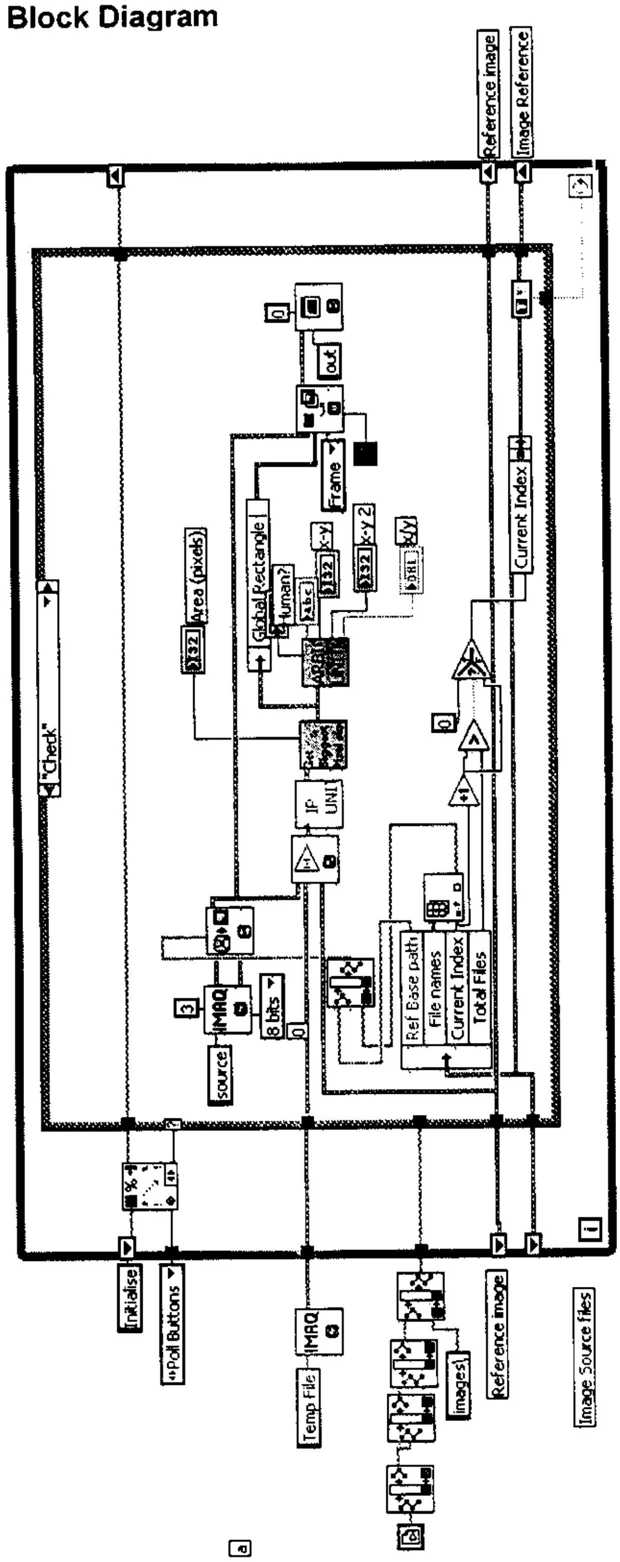

"Initialise"
Initialize the controls
No Error
Boolean
Error
IMAQ
8 bits
T1
3
0
251
59
536
529
img1
init

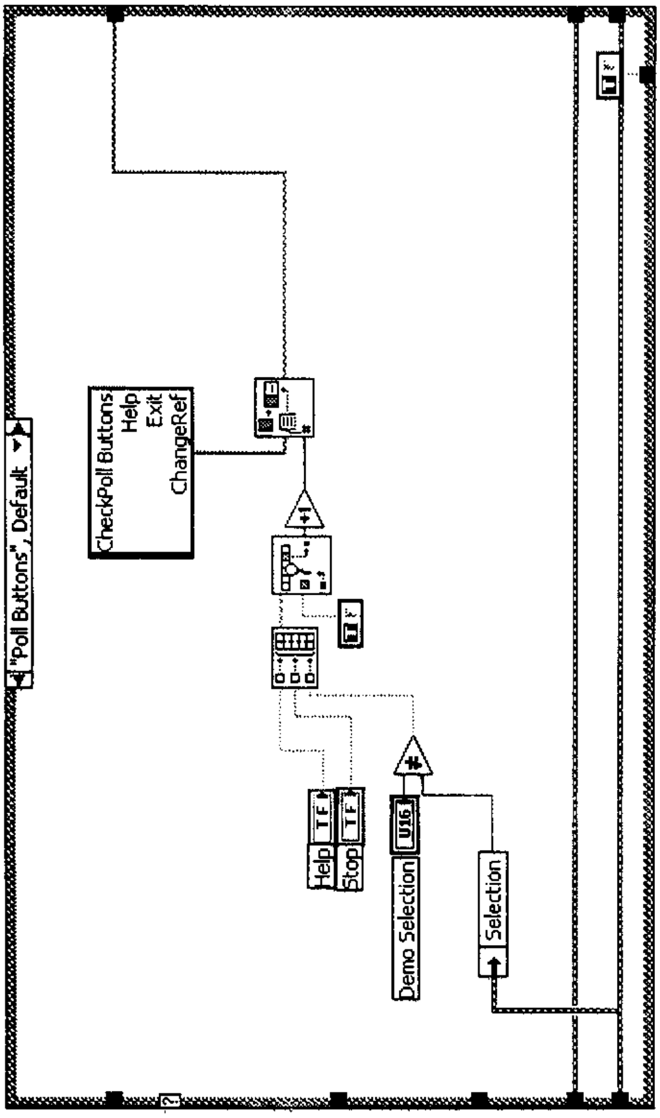
Human Detection Main User Interface Demo.vi

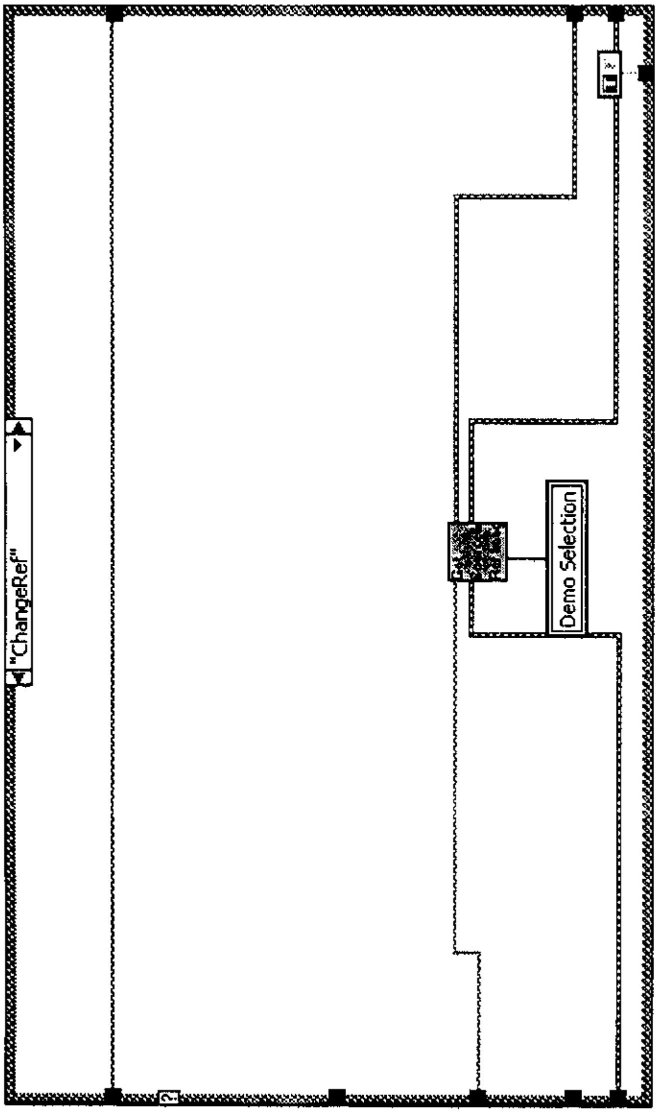
Front Panel

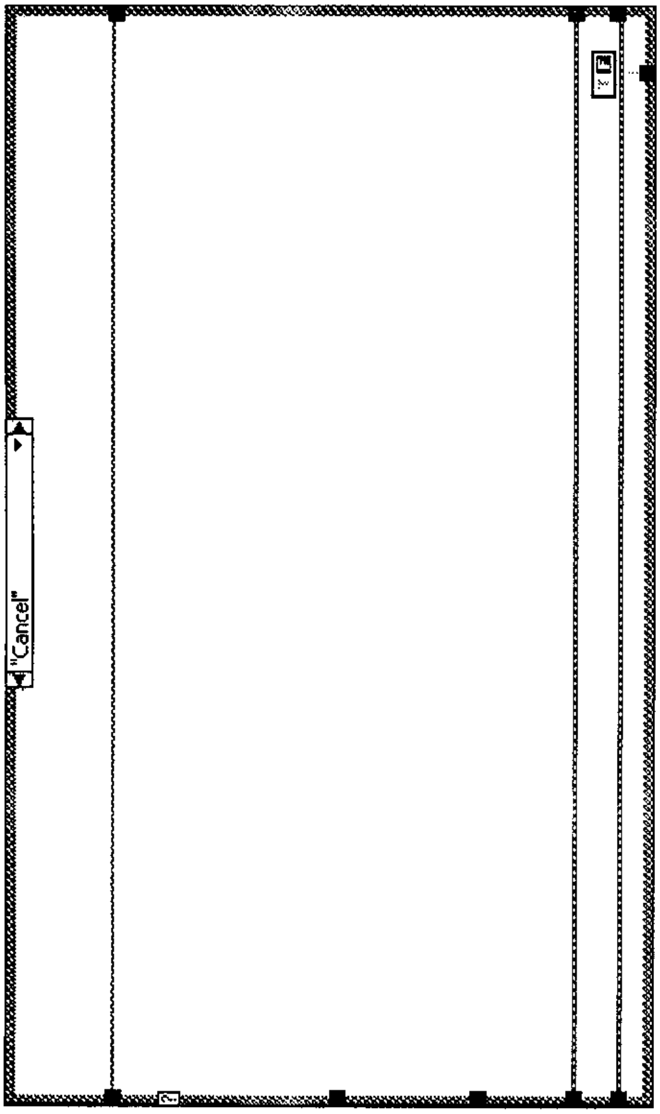


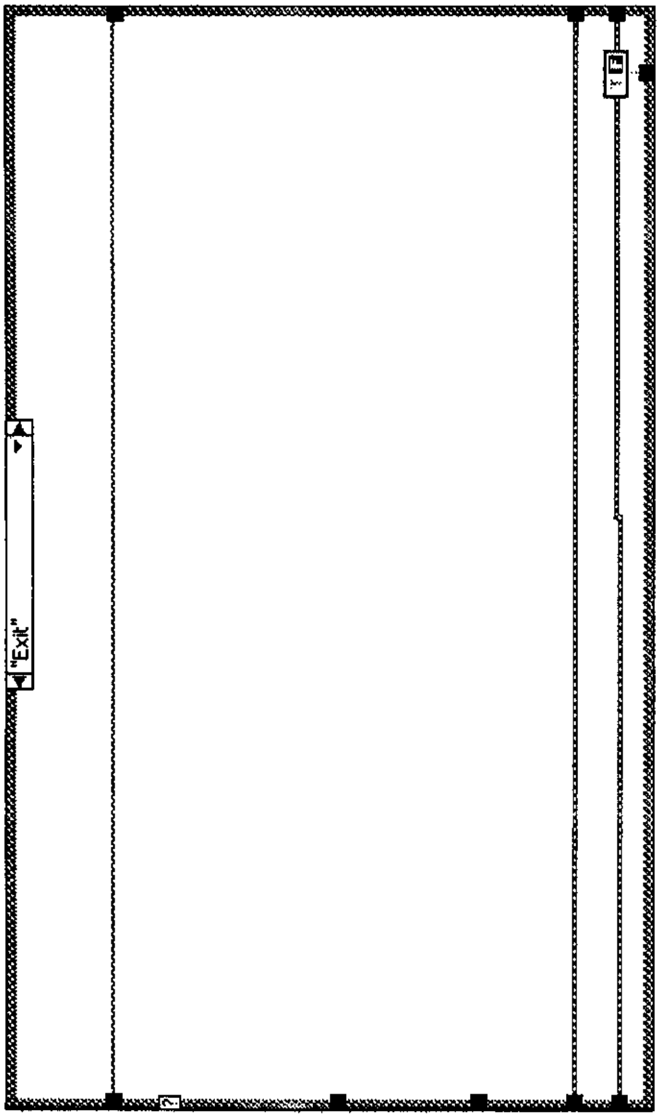
Block Diagram

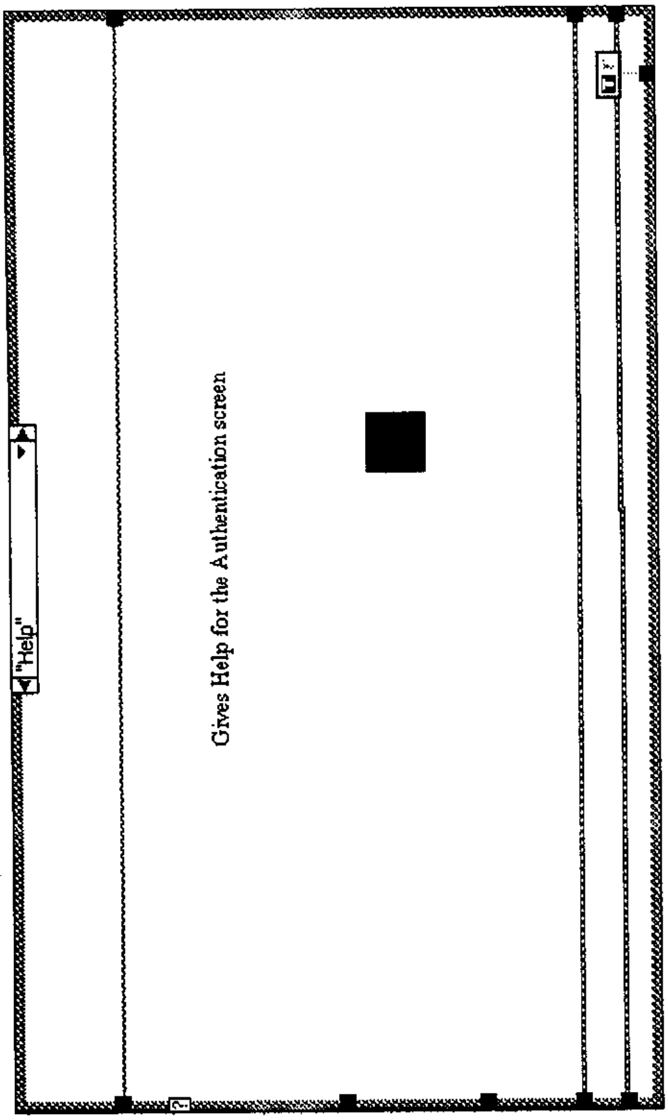


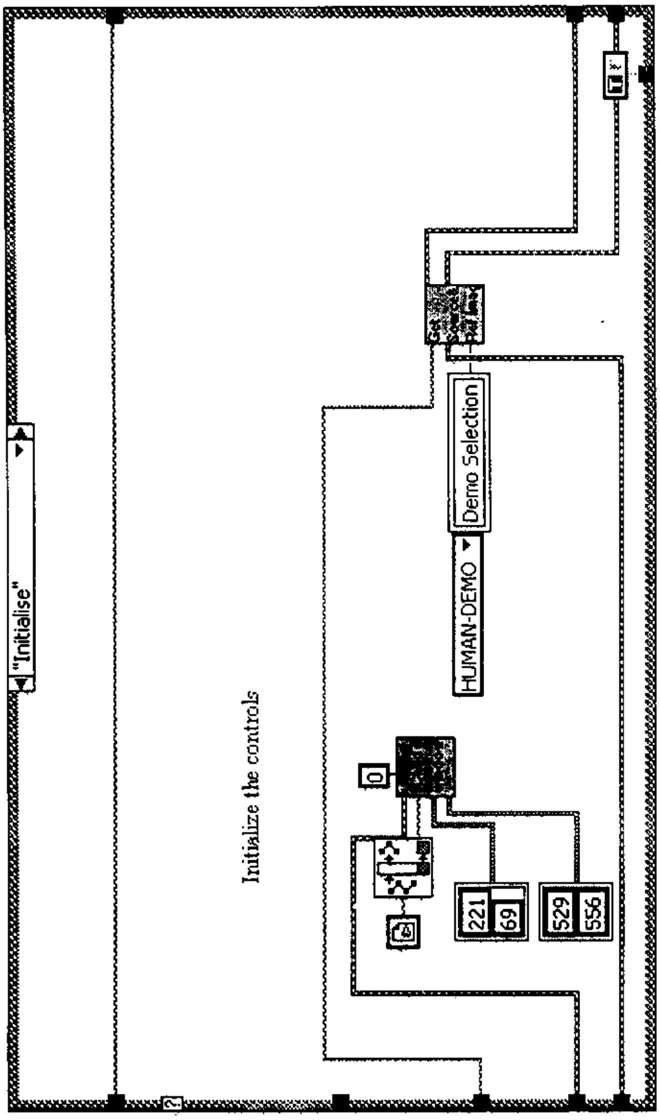

"Poll Buttons", Default
CheckPoll Buttons
Help
Exit
ChangeRef
Help
Stop
U16
Demo Selection
Selection


"ChangeRef"
Demo Selection


"Cancel"

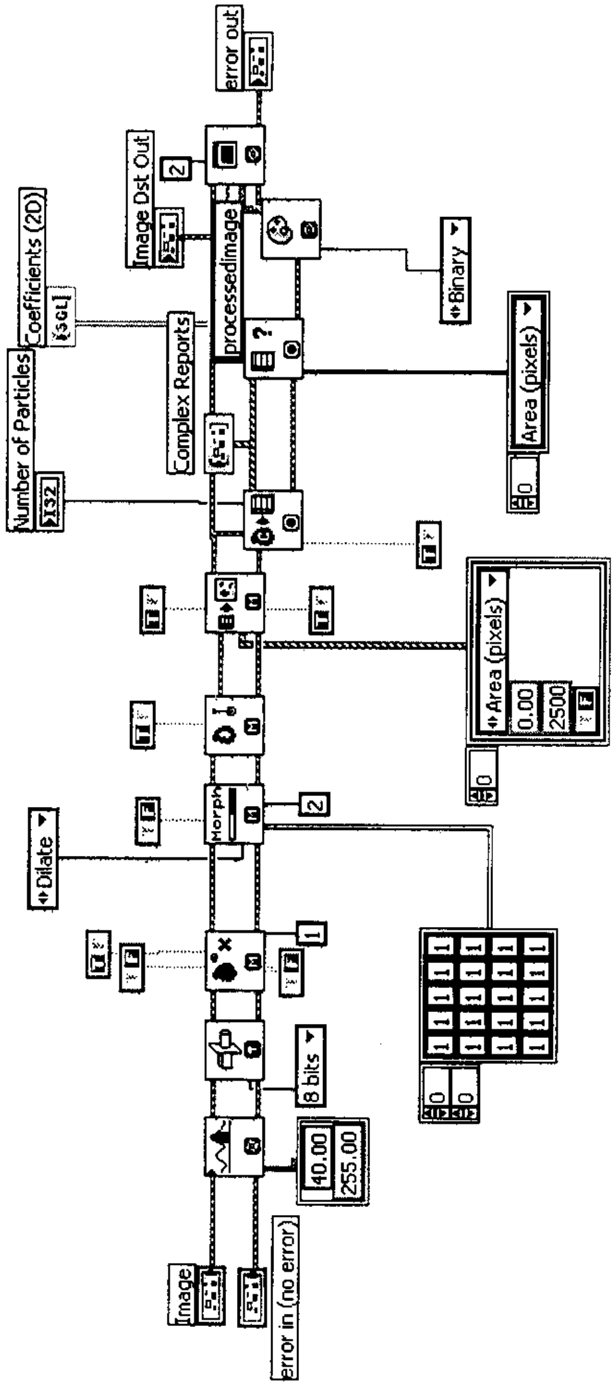

"Exit"


"Help"
Gives Help for the Authentication screen


"Initialise"
Initialize the controls
HUMAN-DEMO
Demo Selection
0
221
69
529
556

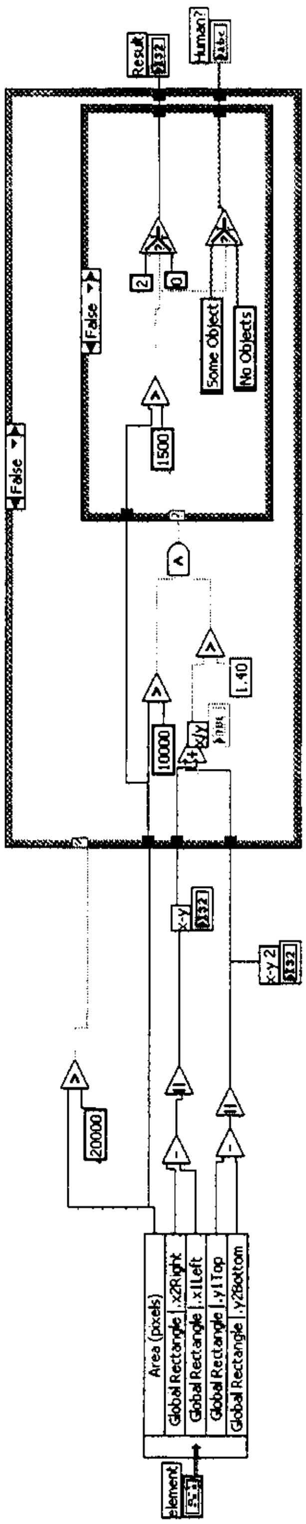
ip.vi

Block Diagram



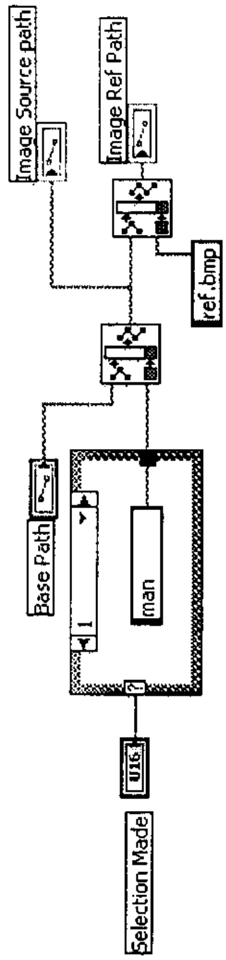
ARBITUNIT.vi

Block Diagram



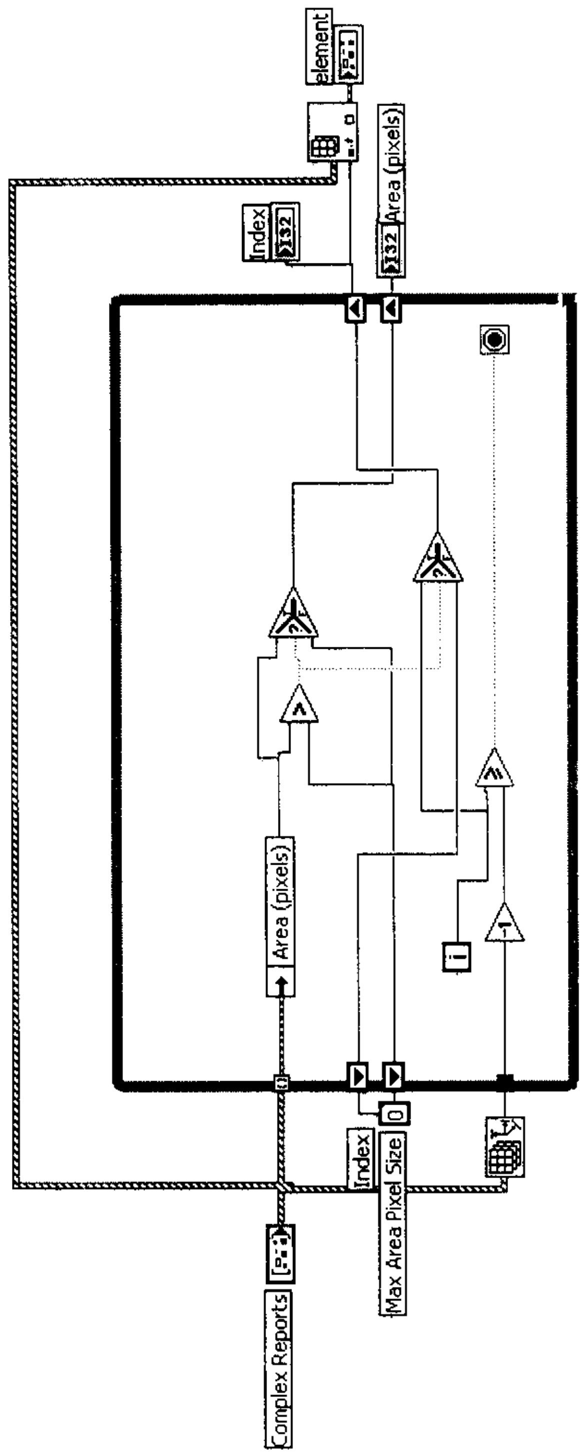
Get Appropriate Image Ref Path.vi

Block Diagram



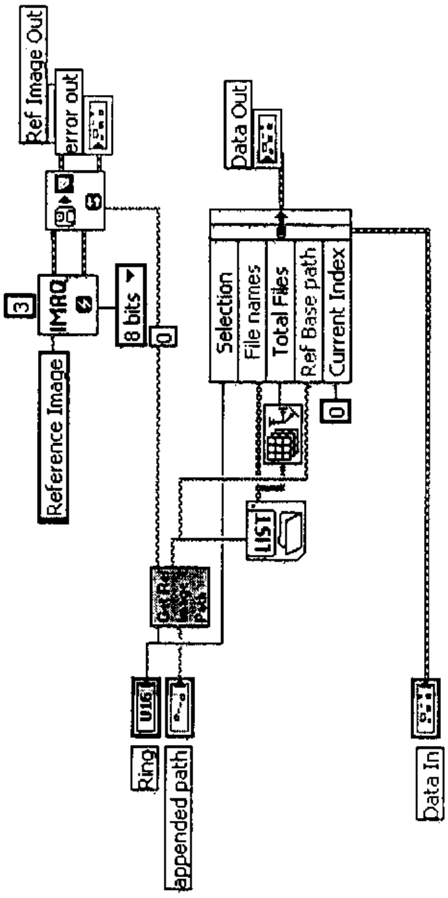
Get_the_biggest_pixel_area.vi

Block Diagram



Get Source Images and Ref Image.vi

Block Diagram

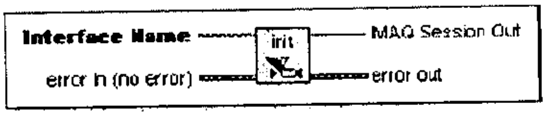


8 CONCLUSION

The prototype algorithm for the security system has been developed. This system has proved to be effective and user friendly, provided all the protocols for proper operation are followed strictly.

There is a lot of scope for improvement which can enhance the performance of the system. Currently our system proves to be an efficient one. Adding more features like creating a more user friendly options and setup features which can make the system adapt to any environment at any time will make the system an choice for surveillance.

IMAQ Init



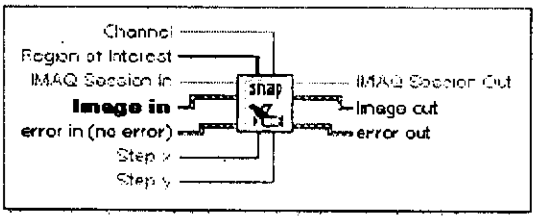
Loads an NI-IMAQ configuration file and configures the IMAQ device.

Interface Name is the name of the interface to be loaded. The name must match the configuration file name used in Measurement & Automation Explorer (MAX) for IMAQ. The default value is [illegible].

IMAQ Session Out is a unique identifier to an IMAQ session that identifies the Interface file.

Note See Error Handling for error input and output information and error codes.

IMAQ Snap



Acquires a single image into **Image out**. If necessary, this VI performs a system initialization using IMAQ Init before the acquisition. When you invoke a snap, the VI initializes the board and acquires the next incoming video frame (or field) to a buffer. A snap is appropriate for low-speed or single-capture applications where ease of programming is essential.

Channel specifies from which channel on the IMAQ device to acquire.

Region of Interest specifies a rectangular portion of the image to be captured. **Region of Interest** is defined by an array of four elements [Left, Top, Right, Bottom]. You must set the width [Right-Left] to a multiple of eight. If **Region of Interest** is not connected or empty, the entire acquisition window is captured.

IMAQ Session In is a unique identifier that identifies the interface file. If you are using the default interface, `img0`, no connection is needed.

Image In is the reference to the image that receives the captured pixel data.

Step x is a horizontal sampling step or horizontal reduction factor. If it is set to its default value of 1, each column of the image is transferred. If **Step x** is set to another value *n*, only one column every *n* columns is transferred. **Step x** only accepts values of 1, 2, 4, or 8.

Step y is a vertical sampling step or vertical reduction factor. If it is set to its default value of 1, each line of the image is transferred. If **Step y** is set to another value *n*, only one line every *n* lines is transferred. **Step y** only accepts values of 1, 2, 4, or 8.

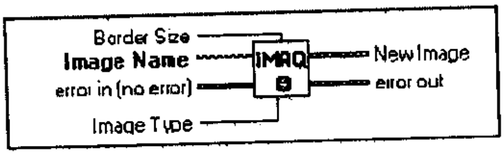
IMAQ Session Out has the same value as **IMAQ Session In**.

Image Out is the reference to the captured image.

Note See Error Handling for error input and output information and error codes.

IMAQ Create

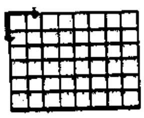
Creates an image.



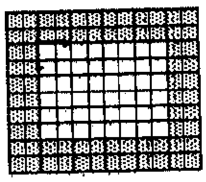
Note Use IMAQ Create in conjunction with the IMAQ Dispose VI to create or dispose of IMAQ Vision images in LabVIEW.

Border Size determines the width, in pixels, of the border to create around an image. These pixels are used only for specific VIs. Create a border at the beginning of your application if an image is to be processed later using functions that require a border (for example, labeling and morphology). The default border value is 3. With a border of three pixels, you can use kernels up to 7 × 7 with no change. If you plan to use kernels larger than 7 × 7 in your process, specify a larger border when creating your image.

The following graphic illustrates an 8 × 6 image with a border equal to 0.



In the following 8 × 6 image, the border equals 2, allowing the use of kernels up to 5 × 5.



Note The border of an image is taken into account only when the image is processed. It is never displayed or stored to a file.

Image Name is the name associated with the created image. Each image created must have a unique name.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

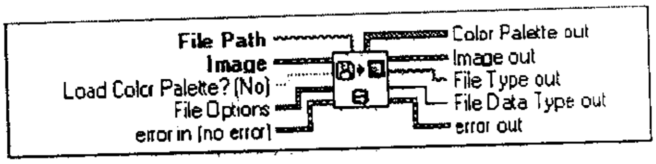
Image Type specifies the image type. This input can accept the following values:

8	**8 bits**	8 bits per pixel (unsigned, standard monochrome)
16	**16 bits**	16 bits per pixel (signed)
F	**float**	32 bits (floating point) per pixel
C	**complex**	2 × 32 bits (floating point) per pixel (native format after an FFT)
	RGB	32 bits per pixel (RGB, color)
	HSL	32 bits per pixel (HSL, color)

New Image is the **Image** structure that is supplied as input to all subsequent (downstream) functions used by IMAQ Vision. Multiple images can be created in a LabVIEW application.

IMAQ ReadFile

Reads an image file. The file format can be a standard format (BMP, TIFF, JPEG, PNG, and AIPD) or a non-standard format known to the user. In all cases, the read pixels are converted automatically into the image type passed by **Image**.



File Path is the complete pathname, including drive, directory, and filename, for the file to be loaded.

Image is the reference to the image structure to which the data from the image file is applied.

Load Color Palette? (No) determines whether to load the color table present in the file (if it exists). If loaded, this table is read and made available to the output **Color Palette**. The default is FALSE.

File Options is a cluster of user-optional values that you can use to read non-standard file formats. The file structure must be known to the user. This cluster consists of the following elements:

Read Raw File indicates whether the file to be read has a non-standard file format. If so, the remaining options in this cluster describe how to read the data.

File Data Type indicates how the image file is encoded.

Offset to Data specifies the size, in bytes, of the file header. This part of the file is not taken into account when read. The pixel values are read from the byte immediately after the offset size. The default is 0.

Use Min Max determines if the user is using a predetermined minimum and maximum. The technique to determine this minimum and maximum depends on the following input values:

0	**Don't use min max**	Minimum and maximum are dependent on the type of image. For an 8-bit image, min = 0 and max = 255.
1	**Use file values**	Pixel values from the file are scanned one time to determine the minimum and maximum, and a linear interpolation is performed before loading the image.
2	**Use optional values**	Uses the two optional values described below.

Optional Min Value is the minimum value of the pixels if **Use Min Max** is selected in mode 2 (**Use optional values**). In this case, pixels with a smaller value are altered to match the chosen minimum. The default is 0.

Optional Max Value is the maximum value of the pixels if **Use Min Max** was selected in mode 2 (**Use optional values**). In this case, pixels with a greater value are truncated to match the chosen maximum. The default is 255.

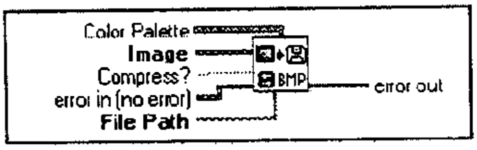
Byte Order determines if the byte weight is to be swapped (Intel or Motorola). The default is FALSE, which specifies big endian (Motorola). TRUE specifies little endian (Intel). This input is useful only if the pixels are encoded with more than 8 bits.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Color Palette out contains the RGB color table (if the file has one) read from the file when the user passes the value TRUE for the input **Load Color Palette? (No)**.

IMAQ Write BMP File

Writes an image to a file in BMP format.



Color Palette is used to apply a color palette to an image. **Color Palette** is an array of clusters constructed by the user or supplied by IMAQ GetPalette. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have the identical value, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

Image is the reference to the image structure to be written as an image file.

Compress? designates whether to compress the BMP file. By default, the image file is not compressed. If it is compressed, it uses the **Run Length Encoded** compression type.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

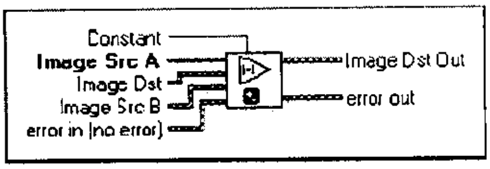
File Path is the complete pathname, including drive, directory, and filename, of the image file to be written. This path can be supplied by either the user or the **File Dialog** VI from LabVIEW.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

IMAQ Absolute Difference

Subtracts one image from another or a constant from an image and returns the absolute value of the difference.

8 16 F



Constant is the value subtracted from the input **Image Src A** for image-constant operations. The constant is rounded down in the cases in which the image is encoded as an integer. The default is 0.

Image Src A is the reference to the source (input) image A.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src A**.

Image Src B is the reference to the source (input) image B.

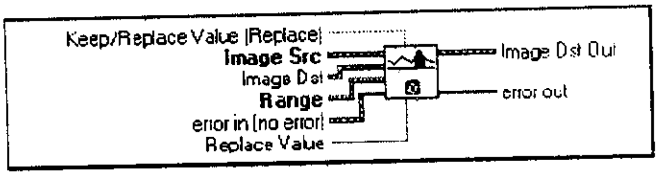
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

IMAQ Threshold

Applies a threshold to an image.



Keep/Replace Value (Replace) determines whether to replace the value of the pixels existing in the range between **Lower value** and **Upper value**. The default TRUE replaces these pixel values, and the status FALSE keeps the original values.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Range is a cluster specifying the threshold range. It is composed of the following elements:

Lower value is the lowest pixel value used during a threshold. The default is 128.

Upper value is the highest pixel value used during a threshold. The default is 255.

All pixels outside the range between **Lower value** and **Upper value** are set to 0. All values found between this range are replaced by the value entered in **Replace Value** if **Keep/Replace Value (Replace)** is TRUE.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Replace Value is the value used to replace pixels between the **Lower value** and **Upper value**. This operation requires that **Keep/Replace Value (Replace)** is TRUE.

Note Use a binary palette when you plan to visualize an image to which a threshold has been applied in Replace mode. However, which palette to use for visualization depends on the value of Replace Value. For example, the visualization of a threshold image could be performed with a gray palette. However, in this case it is advised that you use a replacement value of 255 (white) to see the threshold image better.

Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**

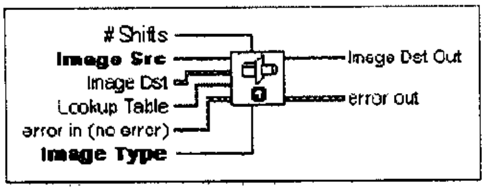
.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

For examples using this VI, see the BasicParticle Example and the Particle Orientation Example.

IMAQ Cast Image

Converts the current image type of an image to the image type specified by **Image Type**. If you specify a lookup table, IMAQ Cast Image converts the image, using a lookup table. If converting from a 16-bit image to an 8-bit image, the VI executes this conversion by shifting the 16-bit pixel values to the right by the specified number of shift operations and then truncating to get an 8-bit value.



 # Shifts specifies the number of right shifts by which each pixel value in the input image is shifted. Valid only when converting from a 16-bit image to an 8-bit image.

Image Src is the reference to the source (input) image to be converted.

Image Dst is the reference to the destination image.

Lookup Table is an array consisting of a maximum of 256 elements if **Image Src** has an 8-bit image or a maximum of 65,536 elements if the **Image Src** has a 16-bit image. The array contains values equal to the index if there are fewer than the amount determined by the image type in **Image Src**. **Lookup Table** can be used to calculate a function that gives a relation between a gray-level value and a user value. This input is valid only when converting from an 8-bit image to a 16-bit image, or when converting from an 8-bit or 16-bit image to a 32-bit floating point image.

VIs capable of analyzing floating-point type images can be used to directly quantify an image, or regions from an image, in user values after converting the image into a floating-point type image.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Image Type specifies the image type into which the input **Image** is converted. The following values are valid:

8 bits	8 bits per pixel (unsigned, standard monochrome)
16 bits	16 bits per pixel (signed)
float	32 bits (floating point) per pixel
complex	2 × 32 bits (floating point) per pixel
RGB	32 bits per pixel (RGB, standard color)
HSL	32 bits per pixel (color)

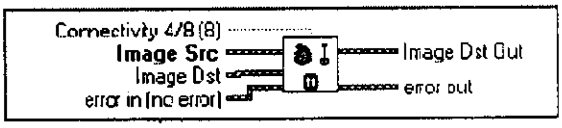
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

For en example using this VI, see the Image Averaging Example.

IMAQ FillHole

Fills the holes found in a particle. The holes are filled with a pixel value of 1. The source image must be an 8-bit binary image. The holes found in contact with the image border are never filled because it is impossible to determine whether these holes are part of a particle.



Connectivity 4/8 (8) specifies how the algorithm determines whether an adjacent pixel belongs to the same or to a different particle. The default is **Connectivity 8**.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

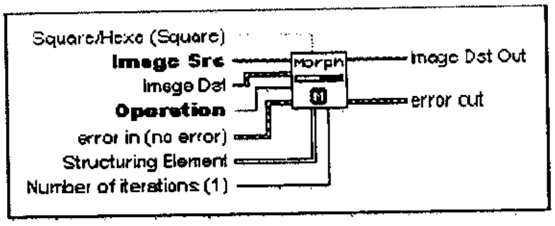
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, then **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

For an example using this VI, see the Blob Analysis Example.

IMAQ Morphology

Performs primary morphological transformations. All source and destination images must be 8-bit binary images. The connected source image for a morphological transformation must have been created with a border capable of supporting the size of the structuring element. A 3 × 3 structuring element requires a minimal border of 1, a 5 × 5 structuring element requires a minimal border of 2, and so on. The border size of the destination image is not important.



A structuring element must have odd-sized dimensions so that it contains a central pixel. If one of the dimensions for the structuring element is even, the function does not take into account the odd boundary, farthest out on the matrix. For example, if the input structuring element is 6 × 4 ($X = 6$ and $Y = 4$), the actual processing is performed at 5 × 3. Both the sixth line and the fourth row are ignored. The processing speed is correlated with the size of the structuring element. For example, a 3 × 3 structuring element processes nine pixels, and a 5 × 5 structuring element processes 25 pixels.

Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as the **Image Src**.

Operation specifies the type of morphological transformation procedure to use. The default is **AutoM**. You can choose from the following values:

AutoM	(Default) Auto median
Close	Dilation followed by an erosion
Dilate	Dilation (the opposite of an erosion)
Erode	Erosion that eliminates isolated background pixels
Gradient	Extraction of internal and external contours of a particle
Gradient out	Extraction of exterior contours of a particle
Gradient in	Extraction of interior contours of a particle
Hit miss	Elimination of all pixels that do not have the same pattern as found in the structuring element
Open	Erosion followed by a dilation
PClose	A succession of seven closings and openings
POpen	A succession of seven openings and closings
Thick	Activation of all pixels matching the pattern in the structuring element
Thin	Activation of all pixels matching the pattern in the structuring element

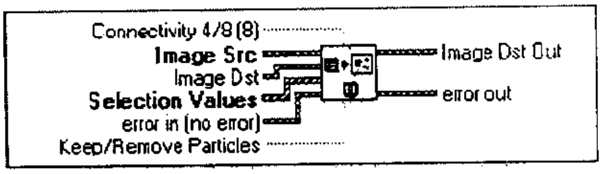
error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Structuring Element is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of 3 × 3 is used if this input is not connected.

Number of iterations (1) is the number of times the VI performs a dilate or erode

IMAQ Particle Filter

Filters (keeps or removes) particles in an image according to their morphological measurements.



Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default value is **Connectivity 8**. The following values are possible:

TRUE	**Connectivity 8**	(Default) Particle detection is performed in connectivity mode 8.
FALSE	**Connectivity 4**	Particle detection is performed in connectivity mode 4.

Image Src is the reference to the source (input) image.

Image Dst is the reference to the destination image. If it is connected, it must be the same type as **Image Src**.

Selection Values controls the criteria that will be used to filter the particle in the image. This control is made of an array of clusters composed of the following values:

- **Parameter** selects the measurement.
- **Lower Value** specifies the lower value of the range for the chosen parameter.
- **Upper Value** specifies the upper value of the range for the chosen parameter.
- **Interval** specifies if the process selects particles inside or outside the range.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Keep/Remove Particles sets the action that is performed on the objects filling the criteria specified by the control **Selection Values**. TRUE removes the particles filling the criteria. FALSE keeps the objects filling the criteria.

Image Dst Out is the reference to the destination (output) image that receives the processing result of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise **Image Dst Out** refers to the image referenced by **Image Src**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

For an example using this VI, see the Blob Analysis Example.

IMAQ ComplexParticle

Detects and measures particles. This VI returns a set of measurements made from particles in a binary image.

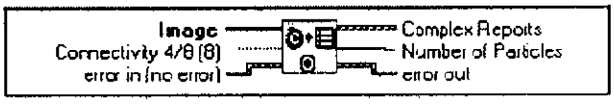


Image is the input source image used for detecting and measuring the particles. The image must be binary. A particle consists of pixels that do not contain a 0 value. The source image must have been created with a border size of at least 2.

Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is Connectivity 8. The following values are possible:

8(T) (Default) Particle detection is performed in connectivity mode 8.

4(F) Particle detection is performed in connectivity mode 4.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Complex Reports is an array that returns a set of measurements from the detected particles. This cluster contains the following elements:

- **Area (pixels)** indicates the surface area of a particle in number of pixels.
- **Area (calibrated)** indicates the surface area of a particle in user-defined units.
- **Perimeter** is the perimeter size in user units.
- **Number of Holes** is the number of holes in the particle.
- **Hole's Area (pixels)** is the total surface area of all the holes in a particle, in pixels.
- **Hole's Perimeter** is the total perimeter size calculated from all the holes in a particle, in user-defined units.
- **Global Rectangle** is a cluster that contains the coordinates of a bounding rectangle for the object detected in the image. This cluster includes the following parameters:
  - **x1Left** indicates the x coordinate of the top-left corner of the rectangle.
  - **y1Top** indicates the y coordinate of the top-left corner of the rectangle.
  - **x2Right** indicates the x coordinate of the bottom-right corner of the rectangle.
  - **y2Bottom** indicates the y coordinate of the bottom-right corner of the rectangle.
- **Σ x** is the sum of the X-axis for each pixel of the particle.
- **Σ y** is the sum of the Y-axis for each pixel of the particle.
- **Σ xx** is the sum of the X-axis squared for each pixel of the particle.
- **Σ xy** is the sum of the X-axis and Y-axis for each pixel of the particle.
- **Σ yy** is the sum of the Y-axis squared for each pixel of the particle.
- **Longest Segment Length** is the longest segment length of the particle.
- **Longest Segment Coordinates** are the coordinates of the left-most pixel in the **Longest Segment Length** of the particle. The top-most segment coordinates are used in a case in which more than one **Longest Segment Length** exist. This cluster contains the following parameters:

IMAQ Overlay Rectangle

Overlays a rectangle on an image.

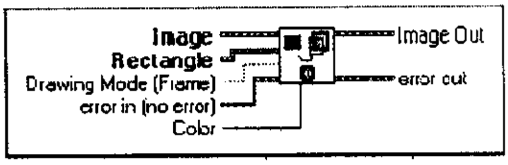


Image is a reference to the source image.

Rectangle is the coordinates that define the rectangle.

- **Left** is the value for the left side of the rectangle.
- **Top** is the value for the top of the rectangle.
- **Right** is the value for the right side of the rectangle.
- **Bottom** is the value for the bottom of the rectangle.

Drawing Mode (Frame) specifies whether to draw the frame of the rectangle only, to fill in the rectangle, or to highlight the rectangle.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Color is the color of the overlay.

Image Out is a the output image.

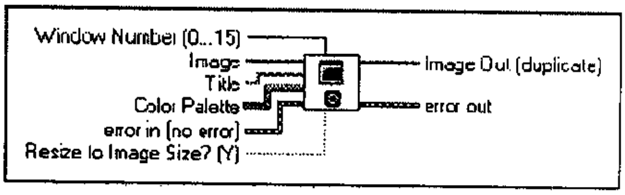
error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

For examples using this VI, see the Circle Distance Example and the Gauging Example.

IMAQ WindDraw

Displays an image in an image window. The image window appears automatically when the VI is executed. The default image window does not have scrollbars. You can add scrollbars by using the IMAQ WindSetup VI.

Use the IMAQ WindDisplayMapping VI to set the pixel mapping policy for displaying 16-bit images.



Window Number (0...15) specifies the image window in which the image is displayed. As many as 16 windows can be displayed simultaneously. Each window is specified with an indicator ranging from 0 to 15. Only the specified image window is affected, and all other image windows remain the same. The default value is 0.

Image specifies the image reference for the displayed image.

Note By default, floating-point and 16-bit images are displayed by scaling the data to 8 bits, as specified by the IMAQ WindDisplayMapping VI.

Title is an image window name. When a string is connected to this input, the image window automatically takes that name.

Color Palette is used to apply a color palette to the image window. **Color Palette** is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). The color palette can be used only for 8-bit images.

Tip For best results, set your video adapter to high color or true color.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

Resize to Image Size? (Y) specifies whether the user wants to resize the image window automatically to fit the image size. The default is TRUE (yes), in which case the user does not have to know the size of a source image before displaying it.

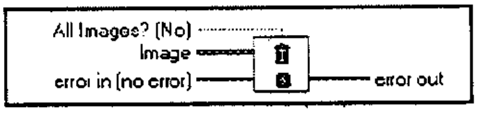
Image Out (duplicate) is a reference to the connected input **Image**.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

For an example of how to use this VI, see 1st Example and or the Histogram Example.

IMAQ Dispose

Destroys an image and frees the space it occupied in memory. This VI is required for each image created in an application to free the memory allocated to IMAQ Create. Execute IMAQ Dispose only when the image is no longer needed in your application. You can use IMAQ Dispose for each call to IMAQ Create or just once for all images created with IMAQ Create.



All Images? (No) specifies whether to destroy a single image or all previously created images. Giving a TRUE value on input destroys all images previously created. The default is FALSE. Be sure to use this function at the end of an application to free the memory occupied by the images.

Image specifies the reference to the image to destroy.

error in (no error) is a cluster that describes the error status before this VI executes. For more information about this control, see IMAQ VI Error Clusters.

error out is a cluster that describes the error status after this VI executes. For more information about this indicator, see IMAQ VI Error Clusters.

Note When a LabVIEW application is aborted, the image space remains occupied.

For an example of using this VI, see the ReadFile Example and Histogram Example..

BIBLIOGRAPHY

- "Digital Image Processing" by Rafael C. Gonzalez, Richard E. Woods. Pearson Education Asia.
- "The Measurement and Automation catalog 2001", National Instruments.
- "LabView6.1 Manual", National Instruments
- "IMAQ VISION 6.1 Manual", National Instruments
- www.Ni.com
- www.google.com