



P-3516



**REAL TIME IMAGE PROCESSING  
USING DSP PROCESSOR**

By

**A.P.PRIYANKA**

**Reg. No. 0920107015**

of

**KUMARAGURU COLLEGE OF TECHNOLOGY**

( An Autonomous Institution affiliated to Anna University of Technology, Coimbatore)

**COIMBATORE - 641049**

**A PROJECT REPORT**

*Submitted to the*

**FACULTY OF ELECTRONICS AND COMMUNICATION  
ENGINEERING**

*In partial fulfillment of the requirements*

*for the award of the degree*

of

**MASTER OF ENGINEERING**

**IN**

**COMMUNICATION SYSTEMS**

**APRIL 2011**



Stay Ahead

# BANNARI AMMAN INSTITUTE OF TECHNOLOGY

SATHYAMANGALAM - 638401 ERODE DISTRICT TAMILNADU

(Autonomous Institution Affiliated to Anna University of Technology, Coimbatore  
Approved by AICTE - Accredited by NBA and NAAC with A Grade)



## DRDO SPONSORED NATIONAL CONFERENCE ON INNOVATIONS IN INFORMATION TECHNOLOGY (NCIIT 2011)

### CERTIFICATE

This is to certify that Mr/Ms. \_\_\_\_\_

of \_\_\_\_\_ has presented a paper

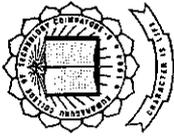
on \_\_\_\_\_ in the National Conference on

Innovations in Information Technology (NCIIT 2011) held on 24<sup>th</sup> and 25<sup>th</sup> February 2011.

ORGANIZING SECRETARY

CONVENOR

PRINCIPAL



# KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)



COIMBATORE - 641049, TAMIL NADU, INDIA.

## CERTIFICATE CITEL 2011

This is to certify that Mr./Ms. A.P. PRIYANKA, ME [COMMUNICATION SYSTEMS]

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE has attended / presented a paper  
titled REAL TIME IMAGE PROCESSING USING DSP PROCESSOR in

the 3<sup>rd</sup> National Conference on **COMMUNICATION, INFORMATION AND TELEMATICS  
(CITEL 2011)** on 3<sup>rd</sup> & 4<sup>th</sup> March 2011, organized by the Department of Electronics and  
Communication Engineering, Kumaraguru College of Technology, Coimbatore.

  
Dr. Rajeswari Mariappan  
HOD - ECE

  
Dr. S. Ramachandran  
Principal

  
Dr. J. Shanmugam  
Director



# K.S.R. RANGASAMY COLLEGE OF TECHNOLOGY

NBA Accredited, ISO Certified Autonomous Institution  
Affiliated to Anna University of Technology, Coimbatore  
KSR Kalvi Nagar, Tiruchengode-637 215

## DEPARTMENT OF INFORMATION TECHNOLOGY

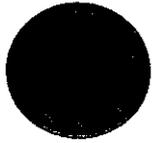
NATIONAL CONFERENCE

ON

RECENT TRENDS AND ADVANCES IN INFORMATION TECHNOLOGY - NCRAIT '11

This is to certify that **Mr./Ms. A.P. PRIYANKA**.....  
of **.....Kumaraguru College...of...Technology**..... has participated  
and presented a paper on **Real-time Image Processing...Using**.....  
**.DSP..Processor**.....in the National Conference on Recent Trends  
and Advances in Information Technology (NCRAIT '11) held on  
05<sup>th</sup> March 2011. He / She has won the best paper award.

Sponsored by



**NCRAIT '11**

*A. S. G. M.*

*[Signature]*

**CONVENOR**

**H.O.D**

**PRINCIPAL**

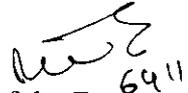
## BONAFIDE CERTIFICATE

Certified that this project report entitled “**REAL TIME IMAGE PROCESSING USING DSP PROCESSOR**” is the bonafide work of A.P.PRIYANKA [Reg. no. 0920107015] who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.



Project Guide  
6411

Dr. Rajeswari Mariappan



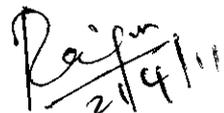
Head of the Department  
6411

Dr. Rajeswari Mariappan

The candidate with university Register no. 0920107015 is examined by us in the project viva-voce examination held on 21.04.2011...



Internal Examiner



External Examiner

Grams : CONSEARCH, CHENNAI  
Fax : 22541889 / 22541973  
e-mail : sic@ceerichennai.org  
Web : ceerichennai.org



☎ 044 - 22541923  
044 - 22542430  
044 - 22541233  
044 - 22542281

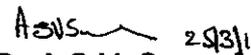
**केन्द्रीय इलेक्ट्रॉनिकी अभियांत्रिकी अनुसंधान संस्थान- चेन्नै केन्द्र**  
सी. एस. आई. आर. मद्रास कॉम्प्लेक्स, टी. टी. टी. आई. पी. ओ. तारमणी चेन्नै -600 113.

**CENTRAL ELECTRONICS ENGINEERING RESEARCH INSTITUTE CHENNAI CENTRE**  
**CSIR MADRAS COMPLEX, T.T.T.I. P.O., Taramani, Chennai - 600 113, India.**

No.EE/14(2)-2011  
Date:25.03.2011

**TO WHOMSOEVER IT MAY CONCERN**

This is to certify that Ms. A.P. Priyanka, M.E. (Communication Systems) final year student of Kumaraguru College of Technology, Coimbatore, worked on the project "REAL TIME IMAGE PROCESSING USING DSP PROCESSOR" at this Centre from 16<sup>th</sup> December, 2010 to 25<sup>th</sup> March, 2011 and successfully completed her project work. During this period, her conduct and character was good.

  
[Dr.A.S.V. Sarma]  
Scientist-in-charge

**Scientist - in - charge**  
**CEERI Centre**  
**CSIR Complex**  
**Chennai - 600 113**

**Head Office : Central Electronics Engineering Research Institute, Pilani - 333 031 (Rajasthan), Tel No. 01596-242119**  
मुख्यालय : केन्द्रीय इलेक्ट्रॉनिकी अभियांत्रिकी अनुसंधान संस्थान, पिलानी 333031 (राजस्थान), टेलीफोन : 01596 242119

## ACKNOWLEDGEMENT

First of all I thank the Lord Almighty for bestowing strength, patience, confidence and perseverance in me to complete this project.

I express my profound gratitude to our Director **Dr.J.Shanmugam, M.E, Ph.D.**, for giving this opportunity to pursue this course.

I wish to acknowledge my sincere gratitude and thanks to our beloved Principal **Dr.S.Ramachandran Ph.D.**, for having given me the adequate support and opportunity for completing this project work successfully.

I would like to thank **Dr.Rajeswari Mariappan Ph.D.**, Professor and Head of the Department of Electronics and Communication Engineering (ECE), Kumaraguru College of Technology, who rendering us all the time by helps throughout this project.

I express my deepest gratitude to my guide **Dr. Rajeswari Mariappan Ph.D.**, Professor and Head of the Department of Electronics and Communication Engineering (ECE), for her expert guidance, technical assistance and patience for the project. I would also like to express my sincere thanks to **Mr.Ramprakash M.E., (Ph.D)** Professor, Department of ECE for providing me all the necessary information and support throughout the project.

My sincere thanks to our Project Coordinator **D.Mohana Geetha M.E., (Ph.D)** Associate Professor, Department of ECE, for her encouragement and support throughout the project.

I express my utmost gratitude to **Dr.A.S.V.Sarma, Scientist-In-Charge, CEERI, Chennai** for providing the project, sparing the resources of the organization and for his support throughout the duration of the project.

I sincerely thank my external guide **Mr .R.Sridhar, Scientist, CEERI, Chennai** for his valuable guidance and persistent encouragement throughout the project work. I would

also like to express my sincere thanks to **Mr.A.Sada Siva Sarma, Scientist,CEERI, Chennai**, for his encouragement and support throughout the project.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering department for their encouragement and support throughout the course of this project.

## ABSTRACT

The objective of this project work is image capture, defect detection and analysis of fruit surfaces using TMS320C6711. The real time fruit quality detection with machine vision based is an attractive and prospective subject for improving marketing competition and post harvesting value of fruit products. In view of existing situations of fruit quality detection and broad application aspect of machine vision in quality evaluation of agricultural products, the methods to detect the external quality of apple by machine vision is being done in this project work. Machine vision based technology is an alternative to unreliable manual sorting method. It aims at solving the problems such as fast processing the large amount of image information, processing capability and increasing precision of detection, etc. This can be used for fruit grading by external qualities like size, shape, colour and surface defects.

The project work uses Texas Instrument's Code Composer Studio (CCS) which is a user-friendly, Windows-based debugger for developing and debugging software for the TMS320C6711. It allows users to write and debug code in C or in TI assembly language. The C6711 has a complete set of development tools which includes: a new C compiler, an assembly optimizer to simplify programming and scheduling and a windows debugger interface for visibility into source code execution

## TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	<b>ABSTRACT</b>	v
	<b>LIST OF FIGURES</b>	ix
	<b>LIST OF ABBREVIATIONS</b>	x
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Overview of the project	1
	1.2 Tools Used	2
<b>2</b>	<b>MACHINE VISION AND DIGITAL IMAGE PROCESSING</b>	<b>3</b>
	2.1 Manual Sorting Of Fruits	3
	2.1.1 Drawbacks	3
	2.2 Analysis Of Fruits Using DSP Processor	4
	2.3 Machine Vision	5
	2.3.1 Components Of A Machine Vision System	5
	2.3.2 Processing Methods	7
	2.4 Digital Image Processing	9
	2.4.1 Digital Images	9
	2.4.2 Image Representation	10
	2.4.2.1 Monochrome Image Representation	10
	2.4.2.2 Colour Image Representation	10
	2.5 Fundamental Steps In Digital Image Processing	11
	2.5.1 Image Acquisition	11
	2.5.2 CCD Camera	12
	2.5.3 Image Enhancement	12
	2.5.4 Histogram Processing	14

2.5.5	Image Restoration	15
2.5.6	Colour Image Processing	16
2.5.7	Colour Models	16
2.5.7.1	RGB Model	17
2.5.7.2	CMY Model	18
2.5.7.3	HSI Model	18
2.5.8	Image Compression	18
2.5.9	Morphological Processing	19
2.5.9.1	Dilation And Erosion	19
2.5.9.2	Opening And Closing	20
2.5.10	Segmentation	20
2.5.11	Representation And Description	20
2.5.12	Object Reorganization	21
2.6	Digital Signal Processing	21
2.6.1	Features Of Digital Signal Processors	22
2.6.2	Applications	23
<b>3</b>	<b>PROJECT DESCRIPTION</b>	<b>24</b>
3.1	Determination Of Colour And Size	24
3.1.1	Background Removal	25
3.1.2	Conversion Of RGB Model To One Dimensional HUE Space	26
3.1.3	Finding The Size Of The Apple	27
3.1.4	Finding The Centroid And AOI	27
3.1.5	Finding The Hue Histogram	28
3.1.6	Finding The Median Hue From The Hue Histogram	28
3.1.7	Finding The Colour Of The Apple	28
3.2	Determination Of Shape	29
3.2.1	Getting The Binary Image	30
3.2.2	Edge Detection	31
3.2.3	Finding The Radius And Theta Values	31
3.2.4	DFT And Shape Heuristic Separator	31

<b>4</b>	<b>SYSTEM SPECIFICATION</b>	<b>34</b>
	4.1 Hardware Requirements	34
	4.2 Software Requirements	34
<b>5</b>	<b>PROCESSOR, CAMERA AND EMULATOR DESCRIPTION</b>	<b>35</b>
	5.1 Architecture	36
	5.2 The Camera	38
	5.2.1 Principle Of Line Scan Camera	38
	5.2.2 Features Of Line Scan Camera	39
	5.3 Overview Of XDS560R JTAG Emulator	40
	5.3.1 Key Features Of Emulator	40
	5.3.2 Key Items Of Emulator	41
	5.4 Stabilite Features	41
<b>6</b>	<b>SOFTWARE TOOLS – DSP C COMPILER</b>	<b>42</b>
	6.1 Code Composer Studio	42
	6.2 Code Composer Studio Development	42
	6.3 Integrated Development Environment	43
	6.4 Programming Code Editing Features	43
	6.5 Hardware Emulation And Real Time Data Exchange	44
	6.6 A Note On User-Defined Files	45
	6.7 Key Features	45
<b>7</b>	<b>RESULTS AND DISCUSSION</b>	<b>46</b>
<b>8</b>	<b>CONCLUSION &amp; FUTURE ENHANCEMENTS</b>	<b>50</b>
	<b>BIBLIOGRAPHY</b>	<b>51</b>

## LIST OF FIGURES

<b>FIGURE NO</b>	<b>NAME OF THE FIGURE</b>	<b>PAGE NO</b>
1.1	Overall Block Diagram Of The Project	1
2.1	Steps Involved In Implementing The Algorithm	8
2.2	Fundamentals in digital image processing	12
2.3	Histogram Of An Image	15
2.4	Visible Spectrum	17
2.5	RGB Colour Cube	17
2.6	HIS Model	18
3.1	Flowchart to determine the size and colour of the apple	25
3.2	Flowchart to determine the shape of the apple	30
5.1	Board diagram of TMS320C6711	36
5.2	Memory map of TMS320C6711	37
5.3	Block diagram of line scan camera	38
6.1	Block Diagram Of CCS	42

## LIST OF ABBREVIATIONS

MV	-----	Machine Vision
DSP	-----	Digital Signal Processor
LED	-----	Light Emitting Diode
CCD	-----	Charged Couple Device
CCS	-----	Code Composer Studio
IDE	-----	Integrated Development Environment
JTAG	-----	Joint Technology Action Group
FPGA	-----	Field Programmable Gate Array
RTDX	-----	Real Time Data Exchange
RTA	-----	Real Time Analysis
MIPS	-----	Million Instruction Per Second
ISR	-----	Interrupt Service Routine
API	-----	Application Program Interface
DMA	-----	Direct Memory Access
EDMA	-----	Enhanced Direct Memory Access
McBSP	-----	Multi Channel Buffered Serial Port
HPI	-----	Host Post Interface
EMIF	-----	External Memory Interface
GPIO	-----	General Purpose Input/output
A/D	-----	Analog To Digital Converter
COFF	-----	Common Object Format File
TI	-----	Texas Instruments
EVM	-----	Evaluation Module
VLIW	-----	Very Long Instruction Word

## CHAPTER 1

### INTRODUCTION

The early detection of damages in fruits is especially important in agriculture products processing because a very small number of injured fruits can cause rottenness infected by microbes and spread the infection to the whole batch and thus causing great economic loss and it also affects further storage and sale. The most important post harvest damage in fruit picking, transport and storage is mechanical bruise caused by external forces which causes physical changes in texture. The detection accuracies are also greatly affected by many factors such as time, bruise type, bruise severity, fruit variety, and fruit pre- and postharvest conditions.

### OVERVIEW OF THE PROJECT

One of the components of the machine vision system for the apple sorting and grading system is the PC platform which acts as the host and a software which is used for inspection. The overall working of this application which would interact with the user and which has been simulated in this project is given below

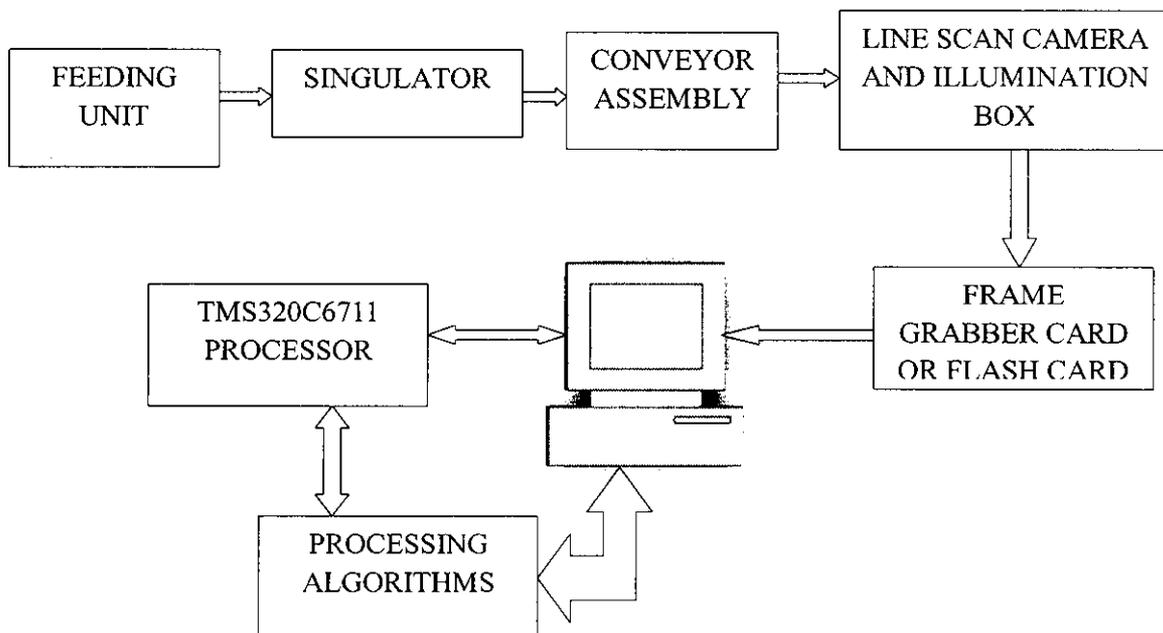


Figure 1.1 Overall Block Diagram of the Project

- Initially the image of the apple is captured using line scan camera
- A frame grabber (National Instruments IMAQ 1428 hardware) card or the flash card receives the digital data from the line scan camera and stores it in a buffer in the PC
- Digital data is transferred from the host computer to the DSP'S memory for processing through the Host Port Interface (HPI) with the help of PCI
- In the front end VC++ application interacts with the user and it does the function of grabbing the image and transferring the data to the the DSP through HPI
- For accomplishing the transfer of data between the host computer and the EVM through HPI
- Certain host software components are provided along the EVM board. These host support software components consist of an operating system specific low level driver and a user mode Win32 DLL. These components are used to create and execute user mode applications for the EVM board.
- C code built with the help of CCS runs at the backend on the Digital Signal Processor to process the digital image and the result is stored in a memory location
- The result is accessed at the front end and displayed to the user

The line scan cameras field of view covers three apples. In a real time system a total of 4 views would be captured for an apple (for each 90 degree rotation). So all 4 views are processed for a single and the final result is displayed. Given below is the sequence of steps followed for the determination of colour, size and shape for each view of the apple by using digital image processing algorithms.

## **TOOLS USED**

The project work uses Texas Instrument's Code Composer Studio (CCS) which is a user-friendly, Windows-based debugger for developing and debugging software for the TMS320C6711. It allows users to write and debug code in C or in TI assembly language.

## CHAPTER 2

### MACHINE VISION AND DIGITAL IMAGE PROCESSING

Inspection of fruits and vegetables is an important procedure for marketing, storing and processing as their appearance affects the consumer acceptance. Colour provides valuable information in estimating the maturity and examining the freshness of fruits and vegetables. Uniformity in size and shape of fruits and vegetables are some of the other important factors in deciding overall quality for buyer's acceptance and value addition.

#### 2.1 MANUAL SORTING OF FRUITS

The early detection of damages in fruits is especially important in agriculture products processing because a very small number of injured fruits can cause rottenness infected by microbes and spread the infection to the whole batch and thus causing great economic loss and it also affects further storage and sale. At present manual sorting of fruits and vegetables is carried out at many places. The most important post harvest damage in fruit picking, transport and storage is mechanical bruise caused by external forces which causes physical changes in texture. The detection accuracies are also greatly affected by many factors such as time, bruise type, bruise severity, fruit variety, and fruit pre- and postharvest conditions. Nowadays, the detection of fruit early bruise has been predominantly performed manually in the past and in some current sorting applications by people trained in the standards of the quality characteristics of the fruit.

##### 2.1.1 DRAWBACKS

The manual sorting of fruits has the following disadvantages

- Plenty of labours needed
- Great labour intensity
- Low productivity
- Grading standard difficult to carry out
- Grading precision instable

## 2.2 ANALYSIS OF FRUITS USING DSP PROCESSOR

The fruit quality can be improved through lots of methods, among which quality detection and sorting operations are the most important ones to increase fruit quality and the profits. Many high quality fruits intermixed with low quality ones are exported or on sale at low price due to the laggard means of quality detection and sorting operations. With the development of computer technology, machine vision grows rapidly. The manual sorting replaced by machine vision with the advantages of high precision, high automatization and belonging to non-contact detection is an inevitable trend of the development of automatic sorting.

In the proposed method using machine vision technology the apples are passed in a linear fashion on a conveyer assembly and when the system is triggered the image of an apple is acquired with the aid of a line scan camera. A flash card receives the image data and then with the help of File Transfer Protocol the digital image is stored in the host computer's buffer. The image captured in the computer's buffer is fed to the TMS320C6711 DSP processor, which is a 32 bit floating point VLIW processor and algorithms for determination of size, shape and colour are applied by using appropriate machine vision techniques. These feature extraction methods are implemented using the Code Composer Studio (CCS) software that serves as an Integrated Development Environment (IDE), comprising tools for highly optimized code generation, a C compiler, an assembler and a linker.

Digital Image Processing forms the fundamental methodology adopted in this project. Digital Image Processing is defined as subjecting the numerical representation of an object to a series of image processing routines to obtain the desired results. This field combines the techniques of image processing with the power of DSP to enable the achievement of Real Time Processing, which requires the computation to be kept in pace with the reception of input signals. It is achieved with the help of highly efficient Digital Signal Processors designed specifically to cater the computationally intensive problems.

## **2.3 MACHINE VISION**

Machine vision (MV) is the application of computer vision to industry and manufacturing. Machine vision, being an engineering discipline, is interested in digital input/output devices and computer networks to control other manufacturing equipment such as robotic arms and equipment to eject defective products. Machine Vision is a subfield of engineering that is related to computer science, optics and industrial automation. One of the most common applications of Machine Vision is the inspection of manufactured goods such as semiconductor chips, automobiles, food and pharmaceuticals. Machine vision systems use digital cameras, smart cameras and image processing software to perform similar inspections.

Machine vision systems are programmed to perform narrowly defined tasks such as counting objects on a conveyor, reading serial numbers, and searching for surface defects. Manufacturers favour machine vision systems for visual inspections that require high-speed, high-magnification, 24-hour operation, and/or repeatability of measurements. Cameras are not equivalent to human optics and while people can rely on inference systems and assumptions, computing devices must 'see' by examining individual pixels of images, processing them and attempting to develop conclusions with the assistance of knowledge bases and features such as pattern recognition engines. Although some machine vision algorithms have been developed to mimic human visual perception, a number of unique processing methods have been developed to process images and identify relevant image features in an effective and consistent manner.

Machine vision and computer vision systems are capable of processing images consistently, but computer-based image processing systems are typically designed to perform single, repetitive tasks, and despite significant improvements in the field, no machine vision or computer vision system can yet match some capabilities of human vision in terms of image comprehension, tolerance to lighting variations and image degradation, parts' variability etc.

### **2.3.1 COMPONENTS OF A MACHINE VISION SYSTEM**

While machine vision is best defined as a process of applying computer vision to industrial application, it is useful to list commonly utilized hardware and software components.

A typical machine vision solution will include several of the following components

1. One or more digital or analog cameras (black-and-white or colour) with suitable optics for acquiring images
2. Camera interface for making the images available for processing. For analog cameras, this includes digitization of the images. When this interface is a separate hardware device it is called a "frame grabber"
3. A processor (often a PC or embedded processor, such as a DSP)
4. Machine Vision Software which provides the tools to develop the application-specific software program
5. Input/output hardware (e.g. digital I/O) or communication links to report results.
6. A Smart Camera, a single device which includes all of the above items.
7. Lenses to focus the desired field of view onto the image sensor.
8. Suitable, often very specialized, light sources (LED illuminators, fluorescent or halogen lamps etc.)
9. An application-specific software program to process images and detects relevant features.
10. A synchronizing sensor for part detection (often an optical or magnetic sensor) to trigger image acquisition and processing.
11. Some form of actuators used to sort or reject defective parts.

The sync sensor determines when a part (often moving on a conveyor) is in position to be inspected. The sensor triggers the camera to take a picture of the part as it passes beneath the camera and often synchronizes a lighting pulse to freeze a sharp image. The lighting used to illuminate the part is designed to highlight features of interest and obscure or minimize the appearance of features that are not of interest such as shadows or reflections. LED panels of suitable sizes and arrangement are often used to this purpose. The camera's image is captured by the frame grabber or by computer memory in PC based systems where no frame grabber is utilized.

A frame grabber is a digitizing device (within a smart camera or as a separate computer card) that converts the output of the camera to digital format (typically a two dimensional array of numbers, corresponding to the luminous intensity level of the

corresponding point in the field of view, called pixel) and places the image in computer memory so that it may be processed by the machine vision software.

The software will typically take several steps to process an image. Often the image is first manipulated to reduce noise or to convert many shades of gray to a simple combination of black and white (binarization).

As a final step, the software passes or fails the part according to programmed criteria. If a part fails, the software may signal a mechanical device to reject the part; alternately, the system may stop the production line and warn a human worker to fix the problem that caused the failure. Though most machine vision systems rely on "black-and-white" (gray scale) cameras, the use of colour cameras is becoming more common. It is also increasingly common for Machine Vision systems to include digital camera equipment for direct connection rather than a camera and separate frame grabber, which reduces cost and simplifies the system.

"Smart" cameras with built-in embedded processors are capturing an increasing share of the machine vision market. The use of an embedded (and often very optimized) processor eliminates the need for a frame grabber card and external computer, thus reducing cost and complexity of the system while providing dedicated processing power to each camera. Smart cameras are typically less expensive than systems comprising a camera and a board and/or external computer, while the increasing power of embedded processors and DSPs is often providing comparable or higher performance and capabilities than conventional PC-based systems.

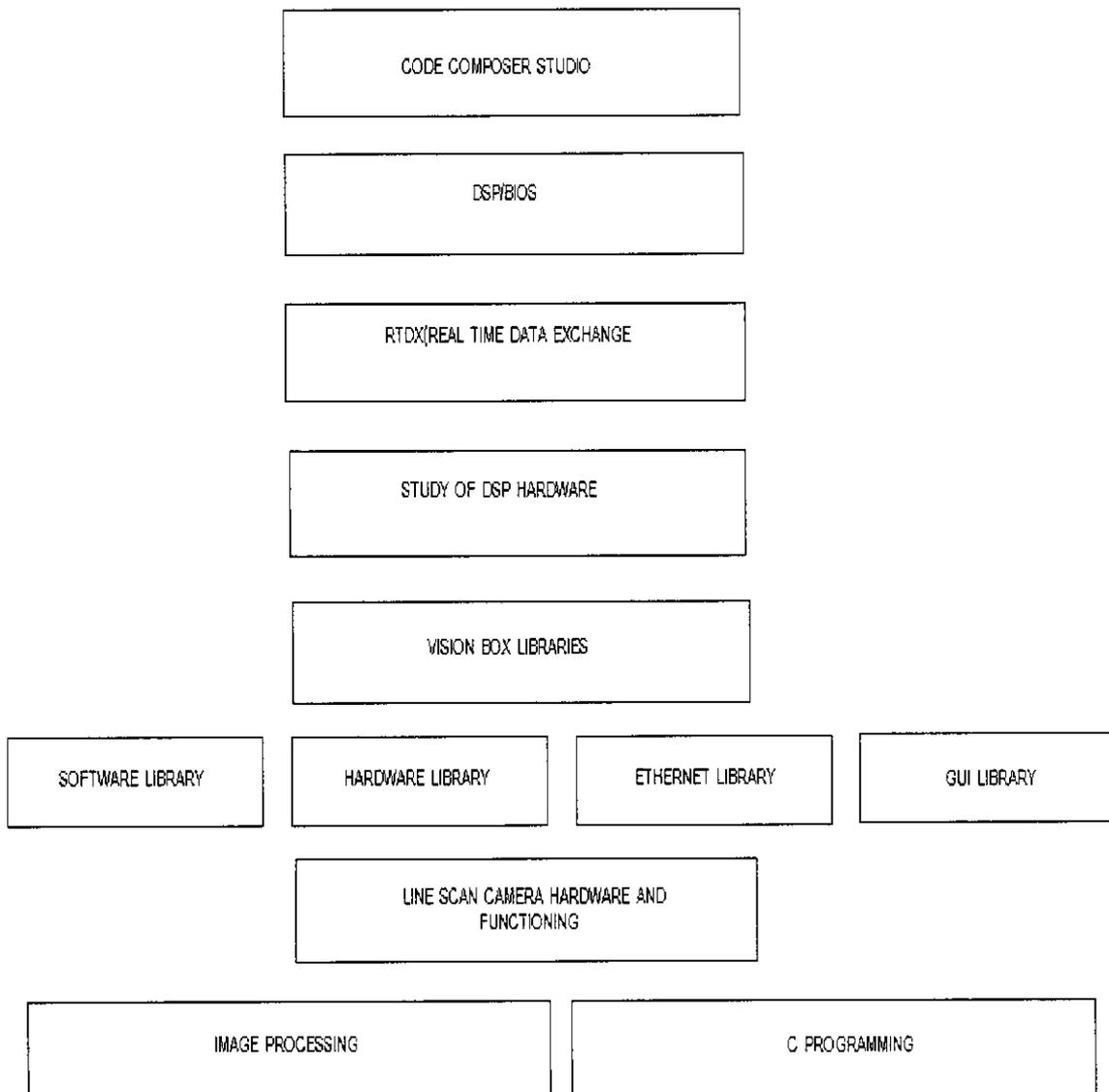
### **2.3.2 PROCESSING METHODS**

Commercial and open source machine vision software packages typically include a number of different image processing techniques such as the following:

- Pixel counting: counts the number of light or dark pixels
- Thresholding: converts an image with gray tones to simply black and white
- Segmentation: used to locate and/or count parts
- Barcode reading: decoding of 1D and 2D codes designed to be read or scanned by machines
- Optical character recognition: automated reading of text such as serial numbers

- Gauging: measurement of object dimensions in inches or millimetres
- Edge detection: finding object edges
- Template matching: finding, matching, and/or counting specific patterns
- Neural Net Processing: weighted and self-training multi-variable decision making

The problem statement is developing typical vision system applications. The hardware chosen for development is TMS320C6416 (Vision box processor). So the implementation and development involved steps demonstrated in the figure below



**Figure 2.1 Steps involved in implementing the algorithm**

## 2.4 DIGITAL IMAGE PROCESSING

The field of digital image processing refers to processing digital images by means of a digital computer. An image defined in the "real world" is considered to be a function of two real variables, for example, a  $(x,y)$  with "a" as the amplitude (e.g. brightness) of the image at the real coordinate position  $(x,y)$ . There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in the continuum - Low, mid and high-level processes. Low-level process involves primitive operations such as image pre-processing 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 and objects), description of those objects to reduce to a form suitable for computer processing and classification 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 (e.g. edges, contours and identity of the objects)

### 2.4.1 DIGITAL IMAGE

A digital image is a representation of a two-dimensional image as a finite set of digital values called picture elements or pixels. Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. These values are often transmitted or stored in a compressed form. Digital images can be created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines, seismographic profiling, airborne radar, and more. They can also be synthesized from arbitrary non-image data such as mathematical functions or three-dimensional geometric models; the latter being a major sub-area of computer graphics.

A digital image  $a[m,n]$  described in a 2D discrete space is derived from an analog image  $a(x,y)$  in a 2D continuous space through a sampling process that is frequently referred to as digitization. The 2D continuous image  $a(x,y)$  is divided into  $N$  rows and  $M$  columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates  $[m,n]$  with  $\{m=0,1,2,\dots,M-1\}$  and  $\{n=0,1,2,\dots,N-1\}$  is a  $[m,n]$

## 2.4.2 IMAGE REPRESENTATION

An image is a two-dimensional (2D) signal, typically a brightness function of two spatial variables. Common examples are photographs, still frames of video, radar and sonar images and x-rays. A digital image results from a sampling of the spatial domain and a quantization of the brightness values. The sampling produces a finite 2D array of values uniformly distributed over the field of view, while the quantization restricts the sample values to a finite integer range. These necessary operations convert real-world analog sensory data to a form suitable for computer processing and storage.

### 2.4.2.1 MONOCHROME IMAGE REPRESENTATION

A monochrome digital image  $a[m,n]$  is a 2D array of luminance values with  $a[m,n] \in Z$ , where  $Z$  is the domain of the integers and  $0 < a[m,n] < L-1$ , where typically  $L = 256$ .

$$a[m,n] = \begin{pmatrix} a[0,0] & a[0,1] & \dots & a[0,N-1] \\ a[1,0] & a[1,1] & \dots & a[1,N-1] \\ \dots & \dots & \dots & \dots \\ a[M-1,0] & a[M-1,1] & \dots & a[M-1,N-1] \end{pmatrix} \quad (2.1)$$

Each element of the array is called a pel (picture element) or more commonly pixel. Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. These values are often transmitted or stored in a compressed form. Values in this range can be efficiently represented by 8 binary digits (note that  $2^8 = 256$ ) and therefore each pixel occupies one byte in memory. Total storage requirements for an image are therefore of the order of  $M \times N$  bytes, where  $M$  and  $N$  are the number of rows and columns in the image array. A typical value for  $M$  and  $N$  is 512, resulting in 262144 bytes of memory for a single monochrome digital image.

There are however image acquisition devices on the market today that have far higher spatial resolutions. It is clear that the raw image data of a monochrome image may be conveniently represented by a matrix. As such, it may be easily manipulated and visualized.

### 2.4.2.2 COLOR IMAGE REPRESENTATION

A color digital image is typically represented by a triplet of values, one for each of the color channels, as in the frequently used RGB color scheme. The letters R, G and B stand for Red, Green and Blue. The individual color values are almost universally 8-bit values,

resulting in a total of 3 bytes (or 24-bits) per pixel. This yields a three-fold increase in the storage requirements for color versus monochrome images.

Naturally, there are a number of alternative data structures for multi-color image data. The so-called pixel-interleaved (or meshed) and color-interleaved (or planar) formats are the most widely used. Less frequently used are row-wise or column-wise interleaving methods. In a pixel-interleaved format the raw image data is contained in a 2D array, with every element being a list of three values:

$$a[m,n]= \left[ \begin{array}{l} r[0,0],g[0,0],b[0,0] \dots\dots\dots r[0,N-1],g[0,N-1],b[0,N-1] \\ r[1,0],g[1,0],b[1,0] \dots\dots\dots r[1,N-1],g[1,N-1],b[1, N-1] \\ r[M-1,0],g[M-1,0],b[M-1,0] \dots\dots\dots r[M-1,N-1],g[M-1, N-1],b[M-1,N-1] \end{array} \right] \quad (2.2)$$

In the color-interleaved format, the raw data is separated into three 2D matrices, one for each of the color channels:

$$a [m, n] =[r [m, n], g [m, n], b [m, n]] \quad (2.3)$$

The three primary colors may be used to synthesize any one of  $2^{24}$  or approximately 16 million colors. Equal amounts of each of the three colors result in a shade of gray.

## 2.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING

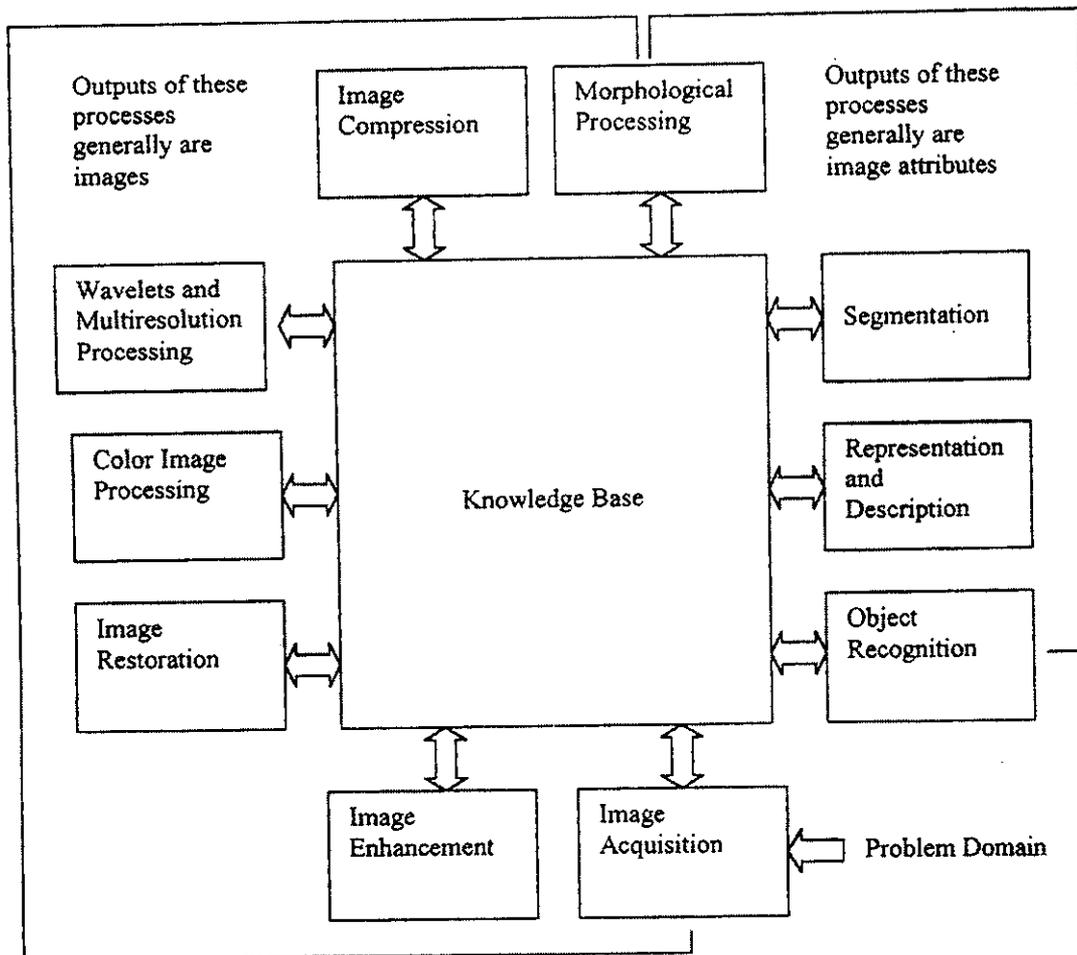
The fundamental operations in digital image processing are the following.

1. Image Acquisition
2. Image enhancement
3. Image Restoration
4. Color Image Processing
5. Wavelets and Multiresolution Processing
6. Image Compression
7. Morphological Processing
8. Segmentation



## 9. Representation and Description

## 10. Object Recognition



**Figure 2.2 Fundamental Steps in Digital Image Processing**

### 2.5.1 IMAGE ACQUISITION

Image acquisition is the first step in digital image processing. The data acquisition methods used in imaging play an important role in image formation. Optimized with the imaging instrumentation, the data collection methods become a decisive factor in determining the best temporal and spatial resolution. It is also crucial in developing strategies to reduce image artifacts through active filtering or post-processing methods. Most digital image capture devices will be based upon a Charge Coupled Device (or CCD), arranged in a matrix (area arrays) or as a line (linear arrays).

## 2.5.2 CCD CAMERA

CCD stands for charge coupled device. The image sensor in a CCD camera is the CCD chip. These range in size from 2 to 3 millimeters on a side for the less expensive cameras to 2 inches or more on a side for the latest generation of professional chips. The chips contain arrays of pixels (light sensitive spots) from appx. 128 x 128 pixels upto appx. 2048 x 2048 pixels. The pixels themselves range in size from about 6 microns to about 30 microns.

## 2.5.3 IMAGE ENHANCEMENT

The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers or to provide better input for other automated image processing techniques. The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. The word specific is important, because it establishes at the outset that the techniques discussed in this chapter are very much problem oriented. Thus, for example, a method that is quite useful for enhancing X-ray images may not necessarily be the best approach for enhancing pictures of Mars transmitted by a space probe. Regardless of the methods used, image enhancement is one of the most interesting and visually appealing areas of image processing.

Image enhancement approaches fall into two broad categories:

1. **Spatial domain methods**
2. **Frequency domain methods**

The term spatial domain refers to the image plane itself (the aggregate of pixels composing an image), and approaches in this category are based on direct manipulation of pixels in an image. Frequency domain processing techniques are based on modifying the Fourier transform of an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

$$g(x, y) = T[f(x, y)] \quad (2.4)$$

Where  $f(x,y)$  is the input image,  $g(x, y)$  is the processed image, and  $T$  is an operator on  $f$  defined over some neighborhood of  $(x, y)$ .

## 2.5.4 HISTOGRAM PROCESSING

A majority of the digital images captured in an optical device, such as a camera or microscope, require adjustments to either the look-up table or the image histogram to optimize brightness, contrast and general image visibility. Histograms of digital images provide a graphical representation of image contrast and brightness characteristics and are useful in evaluating contrast deficiencies such as low or high contrast and inadequate dynamic range. An image histogram is a graphical plot displaying input pixel values on the x-axis (referred to as a bin) versus the number (or relative number) of pixels for any given bin value on the y axis. Each bin in a grayscale histogram depicts a subgroup of pixels in the image sorted by gray level. The numeric range of input values or bins, on the x-axis usually corresponds to the bit depth of the captured image (0-255 for 8-bit images, 0-1023 for 10-bit images, and 0-4095 for 12-bit images). Mathematical operations may be performed on the histogram itself to alter the relative distribution of bins at any gray level. Manipulation of the histogram can correct poor contrast and brightness to dramatically improve the quality of digital images.

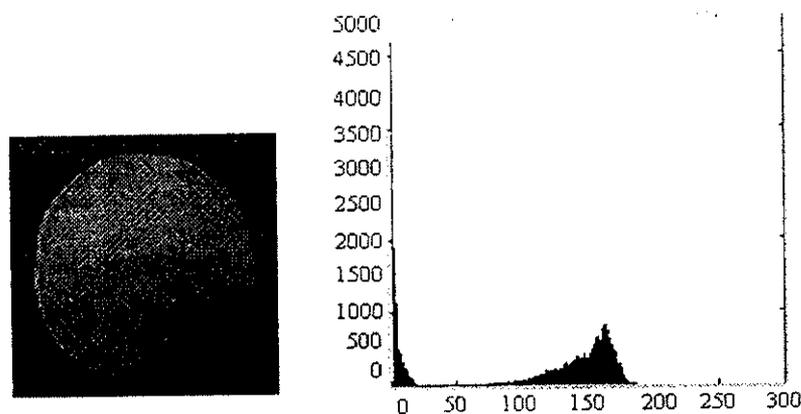
Histograms are the basis for numerous spatial domain processing techniques. Histogram manipulation can be used effectively for image enhancement, image compression and segmentation. Histograms are simple to calculate in software and also lend themselves to economic hardware implementations, thus making them a popular tool for real-time image processing.

Histogram stretching involves modifying the brightness (intensity) values of pixels in the image according to a mapping function that specifies an output pixel brightness value for each input pixel brightness value. For a grayscale digital image, this process is straightforward. The following mapping function is often utilized to compute pixel brightness values:

$$\text{Output } (x, y) = (\text{Input } (x, y) - B) / (W - B) \quad (2.5)$$

In the above equation, the intensity range is assumed to lie between 0.0 and 1.0, with 0.0 representing black and 1.0 representing white. The variable B represents the intensity value corresponding to the black level, while the intensity value corresponding to the white

level is represented by the variable  $W$ . In some instances, it is desirable to apply a nonlinear mapping function to a digital image in order to selectively modify portions of the image.



**Fig 2.3 Histogram of an image**

Histogram equalization (also referred to as histogram leveling) is a related technique, which results in the reassignment of pixel gray-level values so that the entire range of gray levels is utilized and the number of counts per bin remains constant. The process yields a flat image histogram with a horizontal profile that is devoid of peaks. Pixel values are reassigned to ensure that each gray level contains an equal number of pixels while retaining the rank order of pixel values in the original image. Equalization is often utilized to enhance contrast in images with extremely low contrast where a majority of the pixels have nearly the same value and which do not respond well to conventional histogram stretching algorithms. The technique is effective in treating featureless dark, flat-field frames and to rescue images with low-amplitude gradients. In contrast, histogram stretching spaces gray-level values to cover the entire range evenly. The auto-enhance or automatic levels (contrast) features of many image processing software packages utilize one of these histogram-based transformations of the image.

### **2.5.5 IMAGE RESTORATION**

As in image enhancement, the ultimate goal of restoration technique is to improve an image in some predefined sense. In many applications (e.g., agricultural imaging, satellite imaging, medical imaging, astronomical imaging, poor-quality family portraits) the imaging

system introduces a slight distortion. Often images are slightly blurred and image restoration aims at deblurring the image. Image restoration removes or minimizes some known degradations in an image. It can be seen as a special kind of image enhancement. The most common degradations have their origin in imperfections of the sensors, or in transmission. It is assumed that a mathematical model of the degradation process is known, or that it can be derived by an analysis of other input images.

### **2.5.6 COLOR IMAGE PROCESSING**

The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of gray. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color.

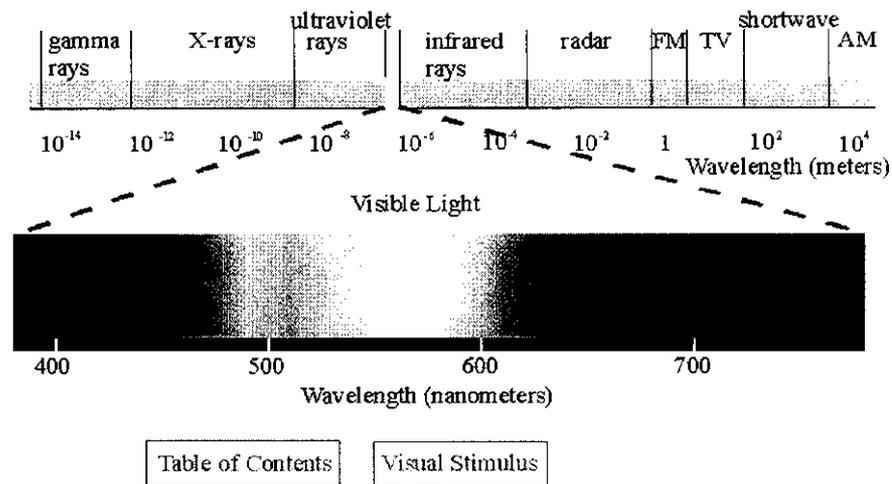
The RGB (red-green-blue) coordinate system is commonly used for representing digital color images. However, coordinate systems related to the human visual system's perceptual attributes (luminance, hue, and saturation) are often useful for processing color images. Much research has been done toward the development of color measurement techniques, but there is not any one-color coordinate system that is universally accepted as corresponding to human perception. There are several color coordinate systems, each having its particular merits, which are being used in color image processing. For examples, LHS, YIQ, CIELAB, and CIELUV coordinate systems are the most popular. Three independent quantities are used to describe any particular color. The hue is determined by the dominant wavelength.

The saturation is determined by the excitation purity, and depends on the amount of white light mixed with the hue. A pure hue is fully saturated, i.e. no white light mixed in. Hue and saturation together determine the chromaticity for a given color. Finally, the intensity is determined by the actual amount of light, with more light corresponding to more intense colors.

### **2.5.7 COLOR MODELS**

Color models provide a standard way to specify a particular color, by defining a 3D coordinate system, and a subspace that contains all constructible colors within a particular model. Any color that can be specified using a model will correspond to a single point within

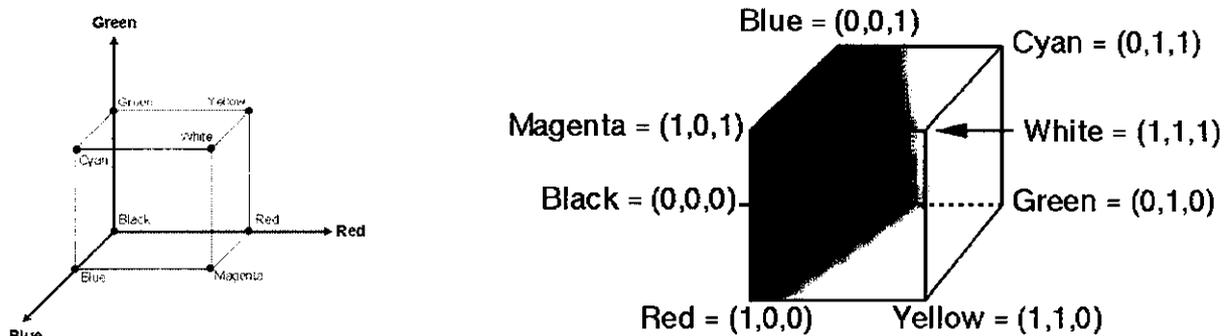
the subspace it defines. Each color model is oriented towards either specific hardware (RGB, CMY, YIQ), or image processing applications (HSI).



**Figure 2.4 Visible Spectrum**

### 2.5.7.1 THE RGB MODEL

In the RGB model, an image consists of three independent image planes, one in each of the primary colors: red, green and blue. Specifying a particular color is by specifying the amount of each of the primary components present. Figure 4.5 shows the geometry of the 3B color model for specifying colors using a Cartesian coordinate system. The grayscale spectrum, i.e. those colors made from equal amounts of each primary, lies on a line joining the black and white vertices.



**Schematic of the RGB Color Cube**

**RGB 24 bit Color Cube**

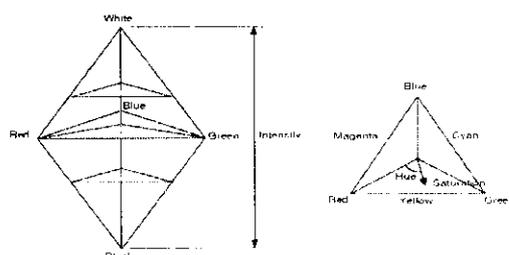
**Figure 2.5 RGB Color Cube**

### 2.5.7.2 THE CMY MODEL

The CMY (cyan-magenta-yellow) model is a subtractive model appropriate to absorption of colors, for example due to pigments in paints. In RGB model, primary color pigments are added to black to get a particular color, whereas in the CMY model, primary color pigments are subtracted from white. In this case, the primaries are cyan, magenta and yellow, with red, green and blue as secondary colors.

### 2.5.7.3 THE HSI MODEL

Color may be specified by the three quantities hue, saturation and intensity. This is the HSI model, and the entire space of colors is shown below



**Figure 2.6 HSI Model**

JHSI model, showing the HSI solid on the left, and the HSI triangle on the right, I by taking a horizontal slice through the HSI solid at a particular intensity. Hue is ed from red, and saturation is given by distance from the axis. Colors on the 5 of the solid are fully saturated, i.e. pure colors, and the grayscale spectrum is on s of the solid. For these colors, hue is undefined.

### 2.5.8 IMAGE COMPRESSION

The purpose of compression is to code the image data into a compact form, minimizing both the number of bits in the representation, and the distortion caused by the compression. So compression is the process of reducing or eliminating redundant and/or irrelevant data. The importance of image compression is emphasized by the huge amount of data in raster images: a typical gray-scale image of 512x512 pixels, each represented by 8 bits, contains 256 kilobytes of data. With the color information, the number of bytes is tripled. If we talk about video images of 25 frames per second, even a one second of color film requires approximately 19 megabytes of memory, therefore the capacity of atypical hard disk

of a PC-machine (540MB) can store only about 30 seconds of film. Thus, the necessity for compression is obvious.

## 2.5.9 MORPHOLOGICAL PROCESSING

Mathematical morphology is a set-theoretical approach to multi-dimensional digital signal or image analysis, based on shape. The signals are locally compared with the structuring elements  $S$  of arbitrary shape with a reference point. The aim is to transform the signals into simpler ones by removing irrelevant information. Morphological operations can be applied to binary and grey level signals.

### 2.5.9.1 DILATION AND EROSION

The main morphological operations are dilation and erosion. Dilation and erosion are related operations, although they produce very different results. Dilation adds pixels to the boundaries of objects (i.e., changes them from off to on), while erosion removes pixels on object boundaries (changes them from on to off).

Each dilation or erosion operation uses a specified neighborhood. The state of any given pixel in the output image is determined by applying a rule to the neighborhood of the corresponding pixel in the input image. The rule used defines the operation as dilation or erosion. With  $A$  and  $B$  as sets in  $Z^2$ , the Dilation of  $A$  by  $B$  is defined as (2.6)

$$A \oplus B = \{Z | ((B \wedge)_Z \cap A) \neq \emptyset\} \quad (2.6)$$

For sets  $A$  and  $B$  in  $Z^2$ , the Erosion of  $A$  by  $B$  is defined as (2.7)

$$A \ominus B = \{Z | ((B)_Z \subseteq A)\} \quad (2.7)$$

For dilation, if any pixel in the neighborhood of the input pixel is on, the output pixel is on. Otherwise, the output pixel is off. For erosion, if every pixel in the neighborhood of the input pixel is on, the output pixel is on. Otherwise, the output pixel is off.

The neighborhood for a dilation or erosion operation can be of arbitrary shape and size. The neighborhood is represented by a structuring element, which is a matrix consisting of only 0's and 1's. The center pixel in the structuring element represents the pixel of interest, while the elements in the matrix that are on (i.e., 1) define the neighborhood.

### 2.5.9.2 OPENING AND CLOSING

The two basic operations dilation and erosion can be combined into more complex Sequences. The most useful of these for morphological filtering are called opening and Closing. Opening consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are too small to contain the structuring element. In this case the structuring element is called a probe. The opening of set A by structuring element B denoted  $f \circ b$ , is defined as

$$f \circ b = (f \ominus b) \oplus b \quad (2.8)$$

Closing consists of a dilation followed by erosion and can be used to fill in holes and small gaps and it generally fuses narrow breaks and long thin gulfs. The closing of set A by structuring element B, denoted  $f \bullet b$ , is defined as

$$f \bullet b = (f \oplus b) \ominus b \quad (2.9)$$

Various other morphological operations include Boundary extraction, Region filling, Connected component extraction, Thinning, Thickening, Skeletons, and Pruning.

### 2.5.10 SEGMENTATION

Image segmentation is important in many computer vision and image processing applications. The goal of image segmentation is to subdivide an image into its constituent regions or objects. Division of the image into regions corresponding to objects of interest is necessary before any processing can be done at a level higher than that of the pixel. Thus, the level to which the subdivision is carried depends on the problem being solved. That is segmentation should stop when the objects of interest in an application have been isolated.

### 2.5.11 REPRESENTATION AND DESCRIPTION

After the images segmented into regions the resulting aggregate of the segmented pixels usually is represented and described in a form suitable for computer processing. Representing a region can be based on its external characteristics (boundary based) or internal characteristics (region based). The next step is to describe the region based on the chosen representation. An external representation is chosen when the primary focus is on

shape characteristics. An internal representation is selected when the primary focus is on regional properties such as color and texture. Sometimes it may be necessary to use both type of representation.

### **2.5.12 OBJECT RECOGNITION**

Object recognition otherwise called pattern recognition can be divided into decision theoretic object recognition and structural object recognition. The decision theoretic method deals with patterns described using quantitative descriptors such as length, area and texture. The structural method deals with patterns described by qualitative descriptors. The recognition methods make use of the concept of learning from sample patterns. Decision theoretic approaches to recognition are based on the use of decision functions. Recognition techniques based on matching represent each class by a prototype pattern vector. Using the Euclidean distance to determine the closeness to each pattern class is used in a minimum distance classifier. The concept of image correlation may also be used. Optimum statistical classifiers make use of a probabilistic approach to recognition. The minimum distance classifier establishes a hyper plane between every pair of classes. The Bayes classifier for Gaussian populations is specified by the mean vector and the covariance matrix of each class.

## **2.6 DIGITAL SIGNAL PROCESSING**

Digital signal processing (DSP) is concerned with the representation of signals by a sequence of numbers or symbols and the processing of these signals. Digital signal processing and analog signal processing are subfields of signal processing. DSP includes subfields like: audio and speech signal processing, sonar and radar signal processing, sensor array processing, spectral estimation, statistical signal processing, digital image processing, signal processing for communications, control of systems, biomedical signal processing, seismic data processing, etc.

The goal of DSP is usually to measure, filter and/or compress continuous real-world analog signals. The first step is usually to convert the signal from an analog to a digital form, by sampling it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of numbers. However, often, the required output signal is another analog output signal, which requires a digital-to-analog converter (DAC).

DSP algorithms have long been run on standard computers, on specialized processors called digital signal processors (DSPs), or on purpose-built hardware such as application-specific integrated circuit (ASICs). Today there are additional technologies used for digital signal processing including more powerful general purpose microprocessors, field-programmable gate arrays (FPGAs), digital signal controllers (mostly for industrial apps such as motor control), and stream processors, among others.

### 2.6.1 FEATURES OF DIGITAL SIGNAL PROCESSORS

By the standards of general purpose processors, DSP instruction sets are often highly irregular. One implication for software architecture is that hand-optimized assembly is commonly packaged into libraries for re-use, instead of relying on unusually advanced compiler technologies to handle essential algorithms.

Hardware features visible through DSP instruction sets commonly include:

- Hardware modulo addressing, allowing circular buffers to be implemented without having to constantly test for wrapping.
- Memory architecture designed for streaming data, using DMA extensively and expecting code to be written to know about cache hierarchies and the associated delays.
- Driving multiple arithmetic units may require memory architectures to support several accesses per instruction cycle
- Separate program and data memories (Harvard architecture), and sometimes concurrent access on multiple data busses Special SIMD (single instruction, multiple data) operations
- Some processors use VLIW techniques so each instruction drives multiple arithmetic units in parallel
- Special arithmetic operations, such as fast multiply-accumulates (MACs). Many fundamental DSP algorithms, such as FIR filters or the Fast Fourier transform (FFT) depend heavily on multiply-accumulate performance.
- Bit-reversed addressing, a special addressing mode useful for calculating FFTs
- Special loop controls, such as architectural support for executing a few instruction words in a very tight loop without overhead for instruction fetches or exit testing

- Deliberate exclusion of a memory management unit. DSPs frequently use multi-tasking operating systems, but have no support for virtual memory or memory protection. Operating systems that use virtual memory require more time for context switching among processes, which increases latency.

## 2.6.2 APPLICATIONS

The main applications of DSP are audio signal processing, audio compression, digital image processing, video compression, speech processing, speech recognition, digital communications, RADAR, SONAR, seismology and biomedicine. Specific examples are speech compression and transmission in digital mobile phones, room correction of sound in hi-fi and sound reinforcement applications, weather forecasting, economic forecasting, seismic data processing, analysis and control of industrial processes, medical imaging such as CAT scans and MRI, MP3 compression, computer graphics, image manipulation, hi-fi loudspeaker crossovers and equalization, and audio effects for use with electric guitar amplifiers.

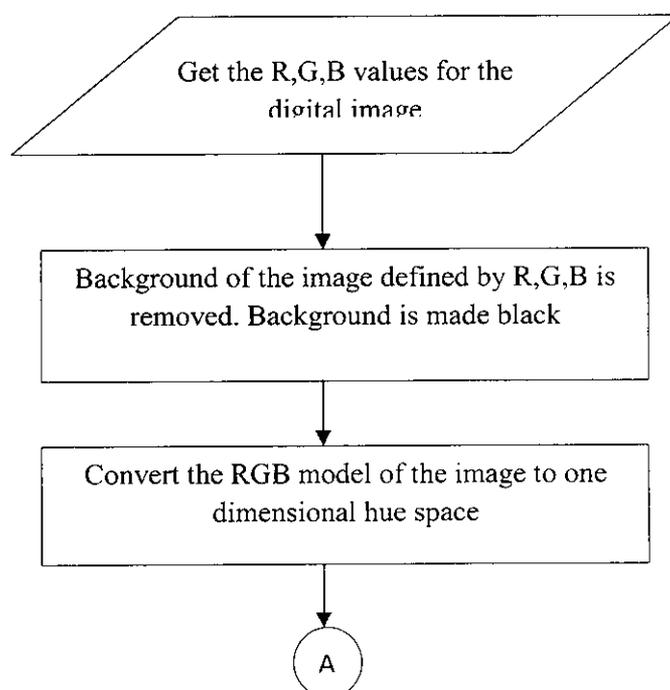
## CHAPTER 3

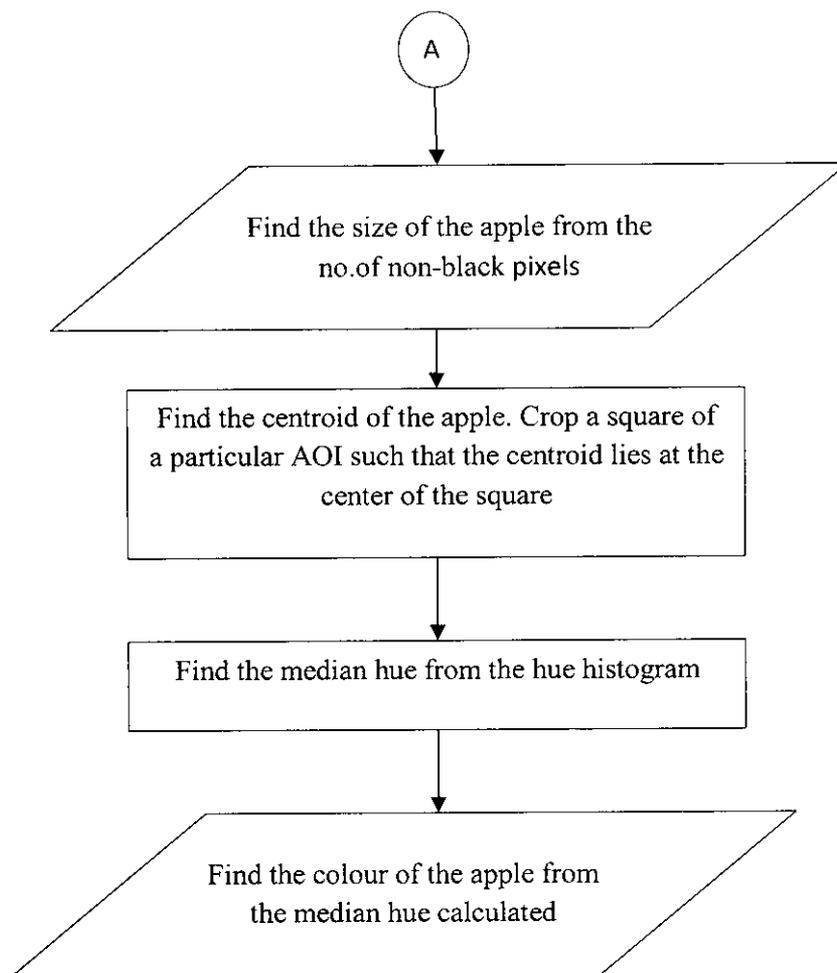
### PROJECT DESCRIPTION

The objective of this project work is capture image capture, defect detection and analysis of fruit surfaces. This is a machine vision based technology in image processing. Machine vision based technology is an alternative to unreliable manual sorting method. It aims at solving the problems such as fast processing the large amount of image information, processing capability and increasing precision of detection, etc. This can be used for fruit grading by external qualities like size, shape, colour and surface defects.

Grading and sorting apples ensures that the products meet defined grade and quality requirements for sellers and provided expected quality for buyers. For this purpose software has been developed in this project based on Machine Vision Technology to extract the features of apples in real time. So once the features like colour, size and shape of apples are quantized the apples could be graded according to their quality.

#### 3.1 DETERMINATION OF COLOUR AND SIZE





**Figure 3.1 Flowchart To Determine Size And Colour Of Apple**

A brief description of the above flowchart is given below

The initial step is to get the details of the image for processing which has been stored in RGB format in the DSP'S memory. The R, G, B components of all the pixels are got separately to image processing algorithm in which the background removal which is segregating the object of interest(apple) from the background forms the first step.

### **3.1.1 BACKGROUND REMOVAL**

During the image acquisition of apples, the CCD camera captures the entire scene within its field of view, which contains the image of the apple and other parts of conveyor system such as friction belt, rollers, etc. A blue or black background is chosen for ease of segmentation. In situations such as sorting of fruits and vegetables, at least some measure of

control over the environment is possible. Either Simple Global Thresholding can be used to accomplish thresholding. In this project Simple global Thresholding or binary contrast enhancement is used to remove the background, as the background is uniformly blue.

There is no universally applicable segmentation technique that will work for all images and so on a trial and error basis, the following threshold was developed for background removal with sufficient accuracy.

$$\text{IF } (R-B) < 20 \quad (6.1)$$

Then  $R=0, G=0, B=0$  (make the background pixel black )

Each image pixel is determined to be above or below a predetermined threshold value. If the pixel brightness is lesser than the threshold, the resulting pixel brightness is set to 0 i.e. made darker. This is used to create a very high- contrast image by highlighting an object of interest, thus separating the object from its background.

### 3.1.2 CONVERSION OF RGB MODEL TO ONE DIMENSIONAL HUE SPACE

The R,G,B components of the object can be converted to H,S,I colour space and then the hue histogram will instantly show the predominant hue. The human perception of vision is important in colour analysis and needs to be pertained in the system devised. So a standard scale or model that would satisfy the human perception of vision has to be selected for colour analysis. Hence we chose HIS space which is more intuitive than the RGB model.

Given a colour defined by (R,G,B) an equivalent HIS colour can be determined by a series of formulas. Let MAX equal the maximum of the (R,G,B) values and MIN equal the minimum of those values. Let DELTA be the difference of the two. The formula can then be written as

```

If (R==MAX)
{
If (G<B)
G+=360
HUE=0+ ((G-B)*60)/DELTA
}
Else

```

```

{
If (G==MAX)
{
If (B<R)
B+=360
HUE=120+ ((B-R)*60)/DELTA
}
Else
{
If (B==MAX)
{
If (R<G)
R+=360
HUE=240+ ((R-G)*60)/DELTA
}
}
}

```

The resulting values are in HIS form, where HUE varies from 0.0 to 360 indicating the angle in degrees around the colour where the hue is located. Thus the hue value of all the pixels of the image are got in a separate array and then further processing is carried on.

### **3.1.3 FINDING THE SIZE OF THE APPLE**

In the next step the size of the apple is determined from the number of non black pixels i.e. the area of the apple in terms of the no of pixels gives direct estimate of the size of the apple. This method gives accurate results for arbitrary shapes as it is unaffected by the concerns of the symmetry.

### **3.1.4 FINDING THE CENTROID AND AOI**

The centroid of the apple is calculated by summing up the row and the column positions of all the apple pixels separately and then dividing it by the area of the apple. The

entire image is scanned pixel by pixel, and if a non-black pixel is found, the above computation is done.

Once the centroid of the apple is calculated an Area Of Interest (AOI) in the form of a square is cropped from the apple such that the centroid is in the center of the square. The AOI used here is a 40 \* 40 square. This is done mainly to minimize the number of computations required for determining the colour and also at the same time get accurate results. After performing this step, the left, right, top and bottom positions of the square is obtained for further processing.

### 3.1.5 FINDING THE HUE HISTOGRAM

Once the AOI is obtained a hue histogram is found for the apple pixels in the AOI. The hue histogram is found as the frequency of the hue value vs the hue values. So at the end of this step a histogram array of 360 values is obtained corresponding to the no. of pixels for each hue value ranging between 0 and 360. The hue histogram is plotted to find out the hue concentration which is present in the apple.

### 3.1.6 FINDING THE MEDIAN HUE FROM THE HUE HISTOGRAM

The next step is to calculate the median hue value of the apple from the hue histogram. The median value is found as the weighted arithmetic mean of the hue histogram.

The median is found by applying the following formula

$$\text{Median As Weighted Arithmetic Mean} = \left( \sum_{i=1}^{i=359} h_i n_i \right) / \left( \sum_{i=1}^{i=359} n_i \right) \quad (6.2)$$

Where  $h_i$  indicates the hue value and  $n_i$  indicates the number of pixels with the corresponding hue value.

### 3.1.7 FINDING THE COLOUR OF THE APPLE

The final estimation of the colour of the apple is done from the median hue value which is calculated. The following condition was applied for segregating the colour of the apple in accordance with the median value as either red, yellowish red or green.

If (median >60)

The colour of the apple is GREEN

If (median >22)

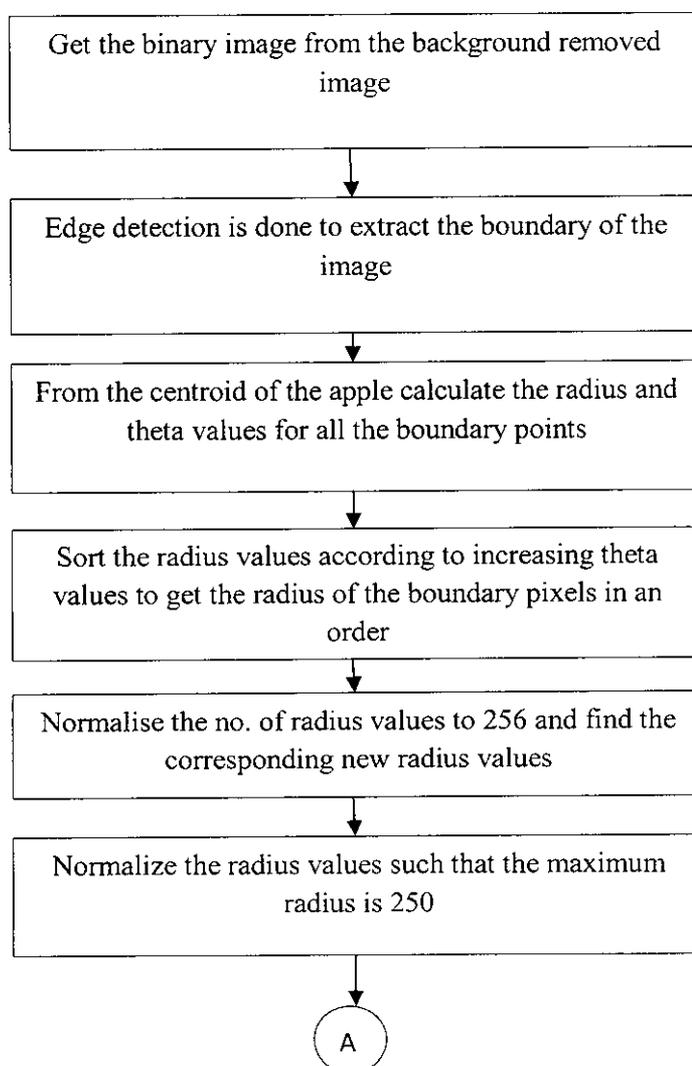
The colour of the apple is YELLOWISH RED

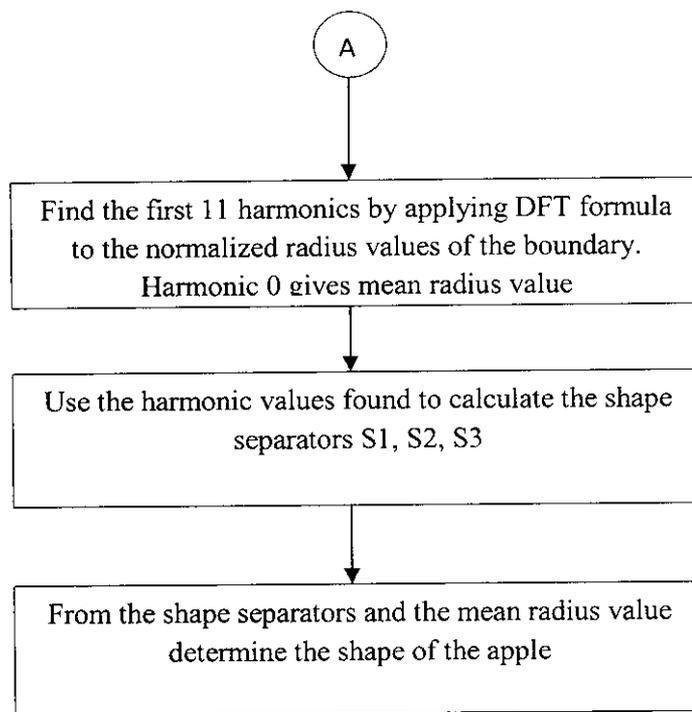
Else

The colour of the apple is RED

After the determination of the colour and the size of the apple, the processing for the shape of the apple is done. The sequence of steps followed for determining the shape of the apple is given below

### 3.2 DETERMINATION OF SHAPE





**Figure 3.2 Flowchart to Determine Shape The Apple**

### 3.2.1 GETTING THE BINARY IMAGE

A binary image is essentially a black and white image that has only 2 components pure black (grayscale intensity=0) and pure white (grayscale intensity=255). The background removed colour image, obtained at the end of the colour analysis routine forms the input for the creation of the binary image. If a particular pixel's hue value is non-zero, it implies that it will belong predominantly to the apple, though the noise from the boundary may not have been completely removed. The intensity value for such a pixel is assigned as 255, corresponding to that of white in the grayscale. As a result, the entire image is converted to a 2-intensity image with each pixel having intensity value of either 255 (logic value 1), white colour, or 0 (logic value 0), black colour. This logic is the reason why the image obtained is termed as binary. This procedure is similar to thresholding and the relationship between the input and the output pixels is represented below

$$g(x,y) = \begin{cases} 0, & f(x,y) = 0 \\ 255, & otherwise \end{cases} \quad (6.3)$$

Where  $g(x,y)$  is a pixel in the output image and  $f(x,y)$  is a pixel in the input image.

### 3.2.2 EDGE DETECTION

Fundamentally an outline is composed of a single-pixel wide sequence of pixels that follow the perimeter of an object. Each pixel adjoins a neighbouring pixel either to its left, right, top or bottom, or on one of its four diagonals. In this way the outline is fully connected into an unbroken progression of boundary pixels.

In the edge detection process, the image is scanned pixel by pixel and if an object pixel is found, its eight neighboring pixels are checked for the number of white pixels surrounding it. A pixel is classified as an interior pixel and is made black if it is surrounded by at least 7 white pixels. In this way at the end of this step the boundary of the image is obtained.

### 3.2.3 GETTING THE RADIUS AND THETA VALUES

The next step is to find the radius and theta values of all points on the boundary from the centroid of the apple using the following formula

$$Radius = \sqrt{((j - x_c)^2 + (i - y_c)^2)} \quad (6.4)$$

$$\theta = \arctan\left(\frac{i - y_c}{j - x_c}\right) \quad (6.5)$$

Where  $i, j$  are the row and column position of each boundary point.

These radius values are not in the same order as the boundary. So in order to get the radius values in the same sequences as the boundary points, the theta values which is unique for each point is arranged in the ascending order and the corresponding change is made in the radius values also.

To reduce the number of computation and at the same time maintain the accuracy the number of boundary points is normalized to 256 and find the corresponding new radius values is found out. In the next step the radius values are normalized such that the maximum radius value is 250.

### 3.2.4 DFT AND SHAPE HEURISTIC SEPARATOR

The one-dimensional radial boundary needs to be further processed to compress the information so that the shape information can be extracted toward shape grading or separation. The transformation of the one-dimensional boundary to Fourier frequency domain

is performed to achieve the compression. The shape information is the obtained by the analysis of the components in the Fourier frequency domain. The radial boundary signature  $r(k)$  was treated as a 1D digital signal and translated to the Fourier domain using the discrete Fourier transform (DFT)

$$F(h) = \frac{1}{n} \sum_{k=0}^{k=n-1} r(k) * \exp\left(-j\left(\frac{k2\pi h}{n}\right)\right) \quad (6.6)$$

Where  $h=0,1,2,\dots,(n-1)$  and  $n$  is the normalized number of boundary points

Where  $f(h)$  is the magnitude at the harmonic  $h$  in Fourier domain

The first 11 harmonics (0-10) are taken into consideration for effectively representing the shape of an apple. Harmonic 0 represents the mean radius of the apple. Harmonic components in the Fourier domain represent the magnitude of boundary frequency variation in the radial boundary sequence. The interpretation of the first few Fourier coefficients is given below

Fourier coefficients	Implied shape information
F(0)	Average radius
F(1)	Bending
F(2)	Elongation
F(3)	Triangle
F(4)	Square

Generally low harmonic components represent the slow changes in the boundary; radius and high harmonic components represent high frequency change of the boundary radius. Although  $f(0)$  to  $f(10)$  contains most of the information about basic shape, further data compression was used for shape grading. A separator  $S$  was defined for the degree of the shape irregularity

$$S(i) = \sum_{h=0}^{h=10} \frac{F(h)}{F(0)} * 100 * i^h \quad (6.7)$$

Where  $i=0,1,2$

The values of  $s$  for  $i=3$  or greater have significant enhancement to the higher frequency components, i.e., the noise components.

So once three shape separator components are obtained, the shape of the apple can be classified into different categories like circular, elliptical, conical and asymmetric by considering the mean radius and the separator values.

After finding the size, shape and colour of the apple they are graded accordingly as defected or good apple.

## **CHAPTER 4**

### **SYSTEM SPECIFICATION**

#### **4.1 HARDWARE REQUIREMENTS**

- 1) Processor :TMS320C6711
- 2) Camera : SM2 CL Camera Link Line Scan Camera
- 3) XDS560R JTAG Emulator
- 4) Vision Box
- 5) Stabilite (Model 40 DCD Illuminator)
- 6) Flash Card ( Sandisk Extreme III 4Gb)
- 7) PC

#### **4.2 SOFTWARE REQUIREMENTS**

Software: Code Composer Studio (CCS)

Version: V3.1

Platform : Windows XP

## CHAPTER 5

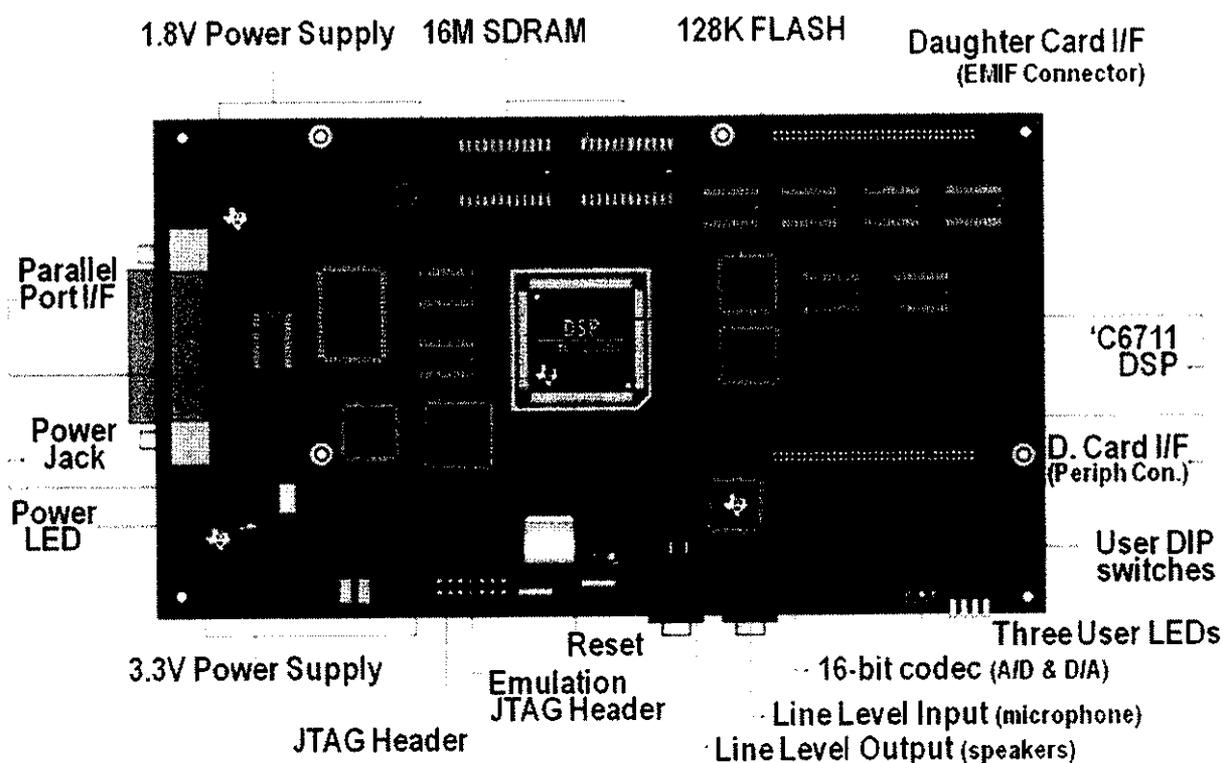
### PROCESSOR, CAMERA AND EMULATOR DESCRIPTION

The TMS320C67xx DSPs (including the TMS320C6711, TMS320C6711B, TMS320C6711C, TMS320C6711D devices†) compose the floating-point DSP family in the TMS320C6000 DSP platform. The C6711, C6711B, C6711C, and C6711D devices are based on the high-performance, advanced very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making these DSPs an excellent choice for multichannel and multifunction applications.

With performance of up to 900 million floating-point operations per second (MFLOPS) at a clock rate of 150MHz, the C6711 device offers cost-effective solutions to high-performance DSP programming challenges. The C6711 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. This processor has 32 general-purpose registers of 32-bit word length and eight highly independent functional units. The eight functional units provide four floating-/fixed-point ALUs, two fixed-point ALUs, and two floating-/fixed-point multipliers. The C6711 can produce two MACs per cycle for a total of 300 MMACS. The C6711 DSPs also have application-specific hardware logic, on-chip memory, and additional on-chip peripherals. The C6711 uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 32-Kbit direct mapped cache and the Level 1 data cache (L1D) is a 32-Kbit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 512-Kbit memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or combinations of the two. The peripheral set includes two multichannel buffered serial ports (McBSPs), two general-purpose timers, a host-port interface (HPI), and a glue less external memory interface (EMIF) capable of interfacing to SDRAM, SBSRAM and asynchronous peripherals.

The C6711 has a complete set of development tools which includes: a new C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows debugger interface for visibility into source code execution.

## 5.1 ARCHITECTURE



**Figure 5.1: Board Diagram of TMS320C6711**

The CPU fetches advanced very-long instruction words (VLIW) (256 bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. The VLIW architecture features controls by which all eight units do not have to be supplied with instructions if they are not ready to execute. The first bit of every 32-bit instruction determines if the next instruction belongs to the same execute packet as the previous instruction, or whether it should be executed in the following clock as a part of the next execute packet. Fetch packets are always 256 bits wide; however, the execute packets can vary in size. The variable-length execute packets are a key memory-saving feature, distinguishing the C67x CPU from other VLIW architectures.

The highest-priority interrupt is INT\_00 (dedicated to RESET) while the lowest-priority interrupt is INT\_15. The first four interrupts (INT\_00–INT\_03) are non-maskable and fixed. The remaining interrupts (INT\_04–INT\_15) are maskable. The interrupt source for

interrupts 4–15 can be programmed by modifying the selector value (binary value) in the corresponding fields of the Interrupt

Selector Control registers: MUXH (address 0x019C0000) and MUXL (address 0x019C0004).

Internal memory is always located at address 0 and can be used as both program and data memory. The C6711 configuration registers for the common peripherals are located at the same hex address ranges. The external memory address ranges in the C6711 devices begin at the address location 0x8000 0000.

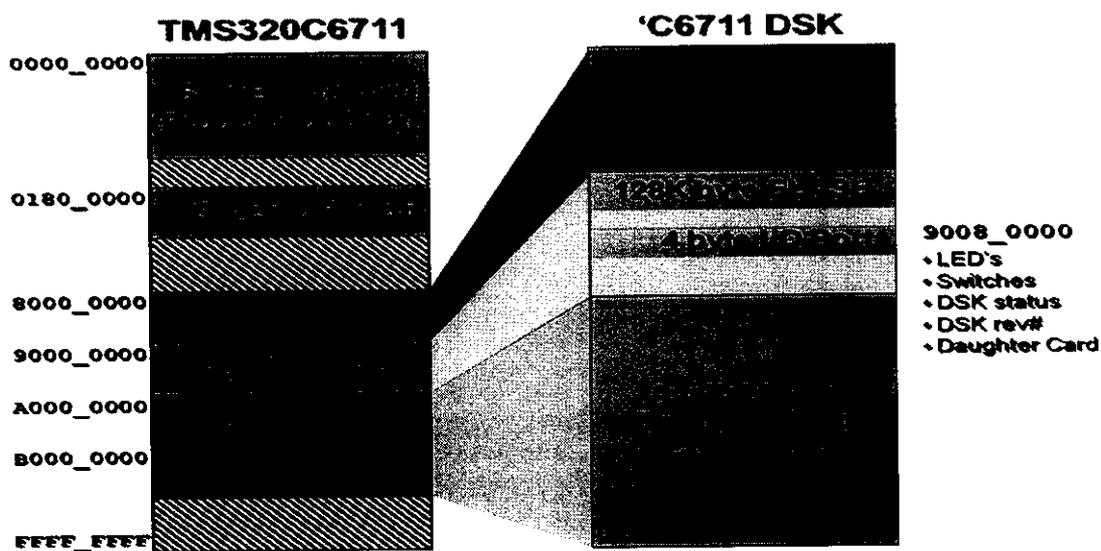


Figure 5.2: Memory Map of TMS320C6711

The C67x EDMA on the C6711 device supports up to 16 EDMA channels. Four of the sixteen channels (channels 8–11) are reserved for EDMA chaining, leaving 12 EDMA channels available to service peripheral devices. For the C6711/11B, the association of an event to a channel is fixed; each of the EDMA channels has one specific event associated with it.

All of the internal C6711 clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which multiplies the source clock in frequency to generate the internal CPU clock, or bypasses the PLL to become the internal CPU clock. To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. To minimize the clock jitter, a single clean power supply should power both the C6711 device and the external clock oscillator circuit. Noise coupling into PLLF will directly impact PLL clock jitter. The minimum CLKIN rise and fall times should also be

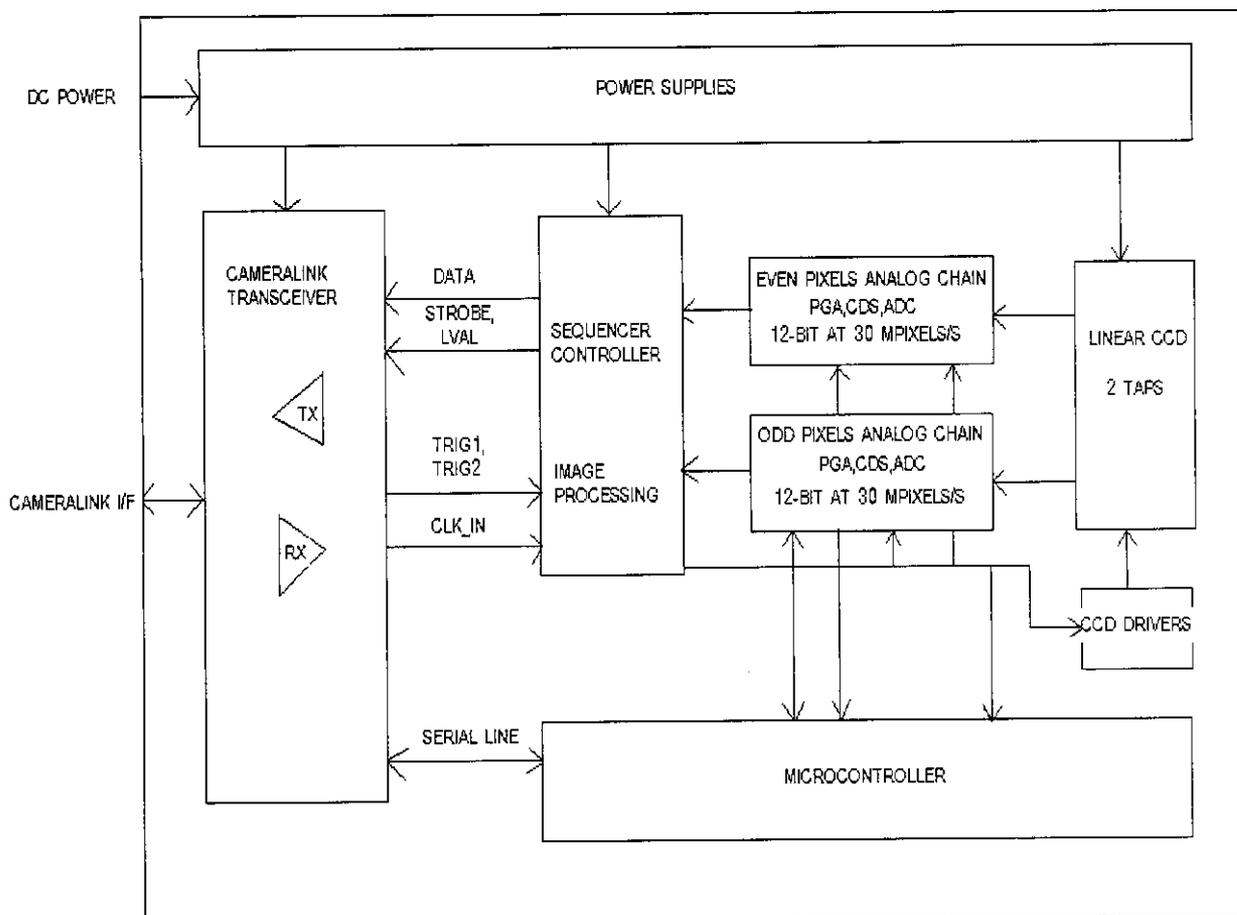
observed. For the input clock timing requirements, see the input and output clocks electrical section. Rise/fall times, duty cycles (high/low pulse durations), and the load capacitance of the external clock source must meet the DSP requirements

## 5.2 THE CAMERA

The camera used is AViiVA SM2 CL Camera Link Line Scan Camera. The camera is based on a tow –tap linear CCD.

### 5.2.1 PRINCIPLE OF LINE SCAN CAMERA

CCD sensor converts the light on a line of photo diode array to electrical signal and reads out its storage amount. When light is on the photo diode, +/- electric charge is stored, parallel transfer signal is imposed and transferred to CCD analog shift register at one time and the image is output . Video signal is picked up as time series pulses, however, it perform average scanning at constant pitch.



**Figure 5.3 Block Diagram of the Line Scan Camera**

Therefore, two analog chains process the odd and even pixel outputs of the linear sensor. The CCD signal processing encompasses the correlated double sampling (CDS), the dark level correction (dark pixel clamping), the gain (PGA) (Programmable Gain Array) and offset correction, and finally the analog-to-digital conversion in 12 bits. A FPGA has been implemented for image processing (flat-field correction, dynamic selection and test pattern generation).

The camera is powered by a single Dc power supply from 12 to 24v. The functional interface (data and control) is provided with the Camera Link interface. The camera uses the base configuration of the Camera Link standard with DVAL=1 and FVAL=0;

Data can be delivered on two channels or on a single multiplexed channel. The data format can be configured in 8, 10 or 12 bits. The camera can be used with external triggers (TRIG1 and TRIG2 signals) in different trigger modes. The camera can also be clocked externally allowing system synchronization or multi-camera synchronization.

The camera configuration and settings are done via a serial line, used for

- Gain and offset settings.
- Dynamic range data rate setting.
- Trigger mode setting: free-run or external trigger modes.
- Integration time setting: in free-run and external trigger modes.
- Flat field correction, contrast expansion.

### 5.2.2 FEATURES OF LINE SCAN CAMERA

Following features are provided in the comparison between line scan camera and area camera.

#### (A) High Resolution

Simple comparison of resolution between area camera with 512 pixel and line scan camera with 5000 pixel in 100 mm field of view.

Area camera  $100/512 = 195 \text{ um/pixel}$

Line scan camera  $100/5000 = 20 \text{ um/pixel}$

Resolution of line scan camera is approx. 10 times ( $10^2$  times in two dimensional) higher than area camera.

**(B) High Speed**

Line scan camera can scan at 20 MHz per 1 scan (50 nsec/pixel). Image capture speed by line scan camera is considerably fast compared with 10 MHz (33.3 msec/ frame) of area camera.

**(C) Continuous processing**

In the inspection of the object which moves continuously like sheet, it's difficult to get synchronization by area camera, however, continuous processing is easily done by line scan camera because of its video output.

**5.3 OVERVIEW OF THE XDS560R JTAG EMULATOR**

The XDS560R JTAG Emulator is designed to be used with digital signal processors (DSPs) and microprocessors which operate from +1.0 to +5 volt levels on the JTAG interface. The power for the emulator comes from the provided supply. This means no power is drawn from the target system or host PC.

**5.3.1 KEY FEATURES OF THE XDS560R JTAG EMULATOR**

The XDS560R JTAG Emulator has the following features:

- Supports Texas Instrument's Digital Signal Processors with JTAG interface (IEEE 1149.1)
- Compatible with Texas Instrument's XDS560 emulator.
- Modular tail connector for alternate JTAG headers.
- Advanced emulation controller provides high performance.
- Compatible with USB 2.0 interface on host PC, no adapter card required.
- Supports +1.0 volt to +5 volt JTAG interfaces.
- 5 LEDs for operational status.
- User accessible RESET switch
- Power provided by supplied power supply
- Compatible with Texas Instruments Code Composer Studio, DSP BIOS,

### 5.3.2 KEY ITEMS ON THE XDS560R JTAG EMULATOR

The key items identified are:

- Status LEDs
- JTAG connector
- Tail
- USB connector to the host PC or hub
- Power connector to power supply
- Cooling vents

### 5.4 STABILITE FEATURES

Stabilite Model 40 DCD Illuminator is from StockerYale,

#### SPECIFICATIONS

Illumination Type      Line

Light Source            Lamp

#### ILLUMINATOR PERFORMANCE

Active Area Length      1.00 inches

Lifetime                    10000 hours

Color Temperature        3250 K

Voltage                    240 volts

Illuminator Output        150 watts

Illuminator Features      Fiber Optic

Intensity Control          0 to 100%

## CHAPTER 6

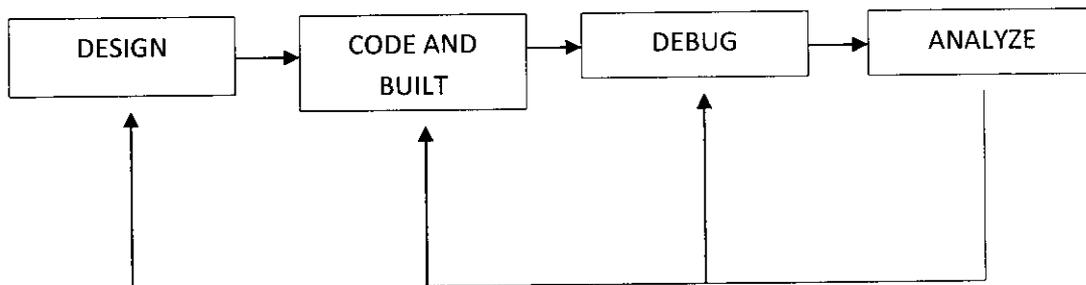
### SOFTWARE TOOL – DSP C COMPILER

#### 6.1 CODE COMPOSER STUDIO

Code Composer Studio speeds and enhances the development process for programmers who create and test real-time, embedded signal processing applications. Code Composer Studio extends the capabilities of the Code Composer Integrated Development Environment (IDE) to include full awareness of the DSP target by the host and real-time analysis tools.

#### 6.2 CODE COMPOSER STUDIO DEVELOPMENT

Code Composer Studio extends the basic code generation tools with a set of debugging and real-time analysis capabilities. Code Composer Studio supports all phases of the development cycle shown here:



**Figure 6.1: Block diagram**

Code Composer Studio includes the following components:

- TMS320C6000 code generation tools
- Code Composer Studio Integrated Development Environment (IDE)
- DSP/BIOS plug-ins and API
- RTDX plug-in, host interface, and API

The following list describes the tools

- The C compiler accepts C source code and produces assembly language

source code. The assembler translates assembly language source files into machine language object files. The machine language is based on common object file format

- The assembly optimizer allows you to write linear assembly code without being concerned with the pipeline structure or with assigning registers. It assigns registers and uses loop optimization to turn linear assembly into highly parallel assembly that takes advantage of software pipelining.
- The linker combines object files into a single executable object module. As it creates the executable module, it performs relocation and resolves external references. The linker accepts relocatable COFF object files and object libraries as input.
- The archiver allows you to collect a group of files into a single archive file, called a library. The archiver also allows you to modify a library by deleting, replacing, extracting, or adding.
- You can use the library-build utility to build your own customized run-time-support library.
- The run-time-support libraries contain ANSI standard run-time- support functions, compiler-utility functions, floating-point arithmetic functions, and I/O functions that are supported by the C compiler.
- The hex conversion utility converts a COFF object file into TI-Tagged, ASCII-hex, Intel, Motorola-S, or Tektronix object format.
- The cross-reference lister uses object files to cross-reference symbols, their definitions, and their references in the linked source files.

### **6.3 CODE COMPOSER STUDIO INTEGRATED DEVELOPMENT ENVIRONMENT**

The Code Composer Studio Integrated Development Environment (IDE) is designed to allow you to edit, build, and debug DSP target programs.

### **6.4 PROGRAM CODE EDITING FEATURES**

The integrated editor provides support for the following activities:

1. Highlighting of keywords, comments, and string in colour
2. Marking C blocks in parentheses and braces, finding matching or next parenthesis

3. Increasing and decreasing indentation level, customizable tab stops
4. Finding and replacing in one or more files, find next and previous, quick search
5. Undoing and redoing multiple actions
6. Getting context-sensitive help
7. Customizing keyboard command assignments

## **6.5 HARDWARE EMULATION AND REAL-TIME DATA EXCHANGE**

TI DSPs provide on-chip emulation support that enables Code Composer Studio to control program execution and monitor real-time program activity. Communication with this on-chip emulation support occurs via an enhanced JTAG link. This link is a low-intrusion way of connecting into any DSP system. An emulator interface, like the TI XDS560, provides the host side of the JTAG connection.

Evaluation boards like the C6x EVM provide an on-board JTAG emulator interface for convenience.

The on-chip emulation hardware provides a variety of capabilities:

- Starting, stopping, or resetting the DSP
- Loading code or data into the DSP
- Examining the registers or memory of the DSP
- Hardware instruction or data-dependent breakpoints
- A variety of counting capabilities including cycle-accurate profiling
- Real-time data exchange (RTDX) between the host and the DSP

Code Composer Studio provides built-in support for these on-chip capabilities. In addition, RTDX capability is exposed through host and DSP APIs, allowing for bi-directional real-time communications between the host and DSP. RTDX provides real-time, continuous visibility into the way DSP applications operate in the real world. RTDX allows system developers to transfer data between a host computer and DSP devices without stopping their target application. The data can be analyzed and visualized on the host using

any OLE automation client. This shortens development time by giving designers a realistic representation of the way their systems actually operate.

RTDX consists of both target and host components. A small RTDX software library runs on the target DSP. The designer's DSP application makes function calls to this library's API in order to pass data to or from it. This library uses the on-chip emulation hardware to move data to or from the host platform via an enhanced JTAG interface. Data transfer to the host occurs in real time while the DSP application is running.

## **6.6 A NOTE ON USER-DEFINED FILES, LINKER COMMAND FILES AND SOURCE FILES**

User defined files - "include" (\*.i) and "header" (\*.h) are the files that are common to most programs. Functionally, both the files are same. Both types of files can define constants, macros (user defined callable functions), or variables. The linker command file (\*.cmd) is vital to the proper building of your code. It specifies where in the program memory to place sections of the program code, defines memory blocks, contains linker options, and names input files for the linker, names the (.out) etc. The linker command file also specifies memory allocations. Without a proper linker command file, CCS will not build the program properly. In our case, the linker command file is named 'realtime.cmd'.

## **6.7 KEY FEATURES**

- User-friendly Windows environment
- Ability to use code written in C and assembly
- Memory displays and on-the-fly editing capability
- Disassembly window for debugging
- Source level debugging that allows stepping through and setting breakpoints in original source code
- CPU register visibility and modification
- Real-time debugging with watch windows and continuous refresh
- Various single step/ step over/ step-into command icons
- Ability to display data in graph for

## CHAPTER 7

## RESULTS AND DISCUSSION

This Image processing Algorithm was implemented on several apple Images and the results obtained were satisfactory. The following screen shots gives the results obtained after implementing the algorithm on the various apple images captured.

## Snapshot 1

The screenshot displays the Code Composer Studio interface. The top window shows assembly code for the file 'easwar123.pjt' in the 'Debug' mode. The code includes instructions like LDW.D2T1, MV.D1, NOP, CSSEXIT, \_fflush, and STW.D2T1. The bottom window shows the execution results, including the sum of hue values, number of pixels in the apple, average hue value, color of the apple (Red), size of the apple (4380), and various geometric parameters like axis lengths, centroid, and circularity. The status bar at the bottom indicates 'CPU HALTED' and 'Ln 60, Cld 23'.

Address	Hex	Dec	Label	Instruction	Comment
8000C428	053C92E5			LDW.D2T1	+++SP[0x4],A10
8000C42C	02280940			MV.D1	A10,A4
8000C430	00006000			NOP	4
8000C434			CSSEXIT:		
8000C434	00000000			NOP	
8000C438	00000310			B.S1	0x8000C438
8000C43C	00008000			NOP	5
8000C440			_fflush:		
8000C440	00900940			MV.D1	A4,A1
8000C444	80001A10	[ A1]		B.S1	0x8000C510
8000C448	063C94F4			STW.D2T1	A12,*SP--[0x4]

```

Sum of Hue values = 134719.781250
No.of pixels in Apple = 9995
Average Hue Value(color) = 13.478718

Colour of the apple : Red

Size Of Apple : 4380

Vert Axis Length : 66
Hori Axis Length : 76
Centroid of the image : 50 , 42
Major Axis : 76 , Minor Axis : 66
Circularity = 0.868421 Eccentricity = 0.495827
Average of C and E : 0.682124

Shape of Apple : Near Ellipse

Grade of Apple : Grade B apple

```

## Snapshot 2

• /ACE/0 C6x1x/CPU\_1 - C6711 - Code Composer Studio - [Disassembly]

File Edit View Project Debug Profiler GEL Option Tools Window Help

easwar123.pjt | Debug

8000C428	059C92E5		LDW.D2T1	***SP[0x4].A10
8000C42C	02280940		MV.D1	A10,A4
8000C430	00006000		NOP	4
8000C434		CSSEXIT:		
8000C434	00000000		NOP	
8000C438	00000310		B.S1	0x8000C438
8000C43C	00008000		NOP	5
8000C440		_fflush:		

Sum of Hue values = 144529.968750  
 No. of pixels in Apple = 5179  
 Average Hue Value(color) = 27.906927

Colour of the apple : Yellowish-Red

Defect area Size = 21

Size Of Apple : 4400

Vert Axis Length : 66  
 Hori Axis Length : 76  
 Centroid of the image : 50 , 42  
 Major Axis : 76 , Minor Axis : 66  
 Circularity = 0.868421 Eccentricity = 0.495827  
 Average of C and E : 0.682124

Shape of Apple : Near Ellipse

Grade of Apple : Grade H(defected) apple

Build Stdout

CPU HALTED For Help, press F1 Ln 60, Col 23 NUM

start

## Snapshot 3

/ IACE#0 C6x1x/CPU\_1 - C6711 - Code Composer Studio - [Disassembly]

File Edit View Project Debug Profiler GEL Option Tools Window Help

easwar123.pjt Copy Debug

8000C428	053C92E5		LDW.D2T1	***SP[0x4],A10
8000C42C	02280940		MV.D1	A10,A4
8000C430	00006000		NOP	4
8000C434		CSSEXIT:		
8000C438	00000000		NOP	
8000C43B	00000310		B.S1	0x8000C438
8000C43C	00008000		NOP	5
8000C440		_fflush:		
8000C440	00900940		MV.D1	A4,A1
8000C444	80001A10	[ A1]	B.S1	0x8000C510

Sum of Hue values = 251025.406250  
 No. of pixels in Apple = 10000  
 Average Hue Value(color) = 25.102541  
 Colour of the apple : Yellowish-Red  
 Defect area Size = 139  
 Size Of Apple : 3637  
 Vert Axis Length : 63  
 Hori Axis Length : 67  
 Centroid of the image : 47 , 49  
 Major Axis : 67 , Minor Axis : 63  
 Circularity = 0.940298 Eccentricity = 0.340351  
 Average of C and E : 0.640325  
 Shape of Apple : Near Circle  
 Grade of Apple : Grade H(defected) apple

CPU HALTED Copy the selection and put it on the Clipboard Ln 60, Col 23 NUM

start

Snapshot 4

Code Composer Studio - [Disassembly]

File Edit View Project Debug Profiler GEL Option Tools Window Help

owncode.pjt | Debug

Files	8000F870	0210E297		LEBU.D2I2	++B4[0x7],B4
GEL files	8000F874	90000790	[IA1]	B.S1	0x8000F89C
Projects	8000F878	00006001		NOP	4
owncode.pjt (D)	8000F87C	00000000		NOP	
Dependent Pr...	8000F880	0001A1A1		ADD.S1	13,A0,A0
DSP/BIOS Co...	8000F884	02018236		STB.D1I2	B4,++A0[0xC]
Generated Files	8000F888	020C3615		LEBU.D1I1	*A3++[0x1],A4
Include	8000F88C	20000112	[B0]	B.S2	0x8000F888
Libraries	8000F890	00006000		NOP	4
Source	8000F894	02003635		STB.D1I1	A4,*A0++[0x1]
c6xdsknit.c	8000F898	200029C2	[B0]	STB.D2	B0,0x1,B0
newdawn.c	8000F89C	00000000		NOP	
vectors.asm	8000F8A0			CSSIOSS:	
C6xdsk.cmd	8000F8A0	00000000		NOP	
	8000F8A4	000C0362		B.S2	B3
	8000F8A8	00008000		NOP	5
	8000F8AC			_readmsg:	
	8000F8AC	000EBE28		MOV.S1	0x1d7c,A0
	8000F8B0	00000068		MVRR.S1	0x0000,A0
	8000F8B4	02808214		LEBU.D1I1	++A0[0x4],A5
	8000F8B8	01800264		LDR.D1I1	++A0[0x0],A3
	8000F8BC	00901058		MOV.L1X	B4,A1
	8000F8C0	00002000		NOP	2
	8000F8C4	02900234		STB.D1I1	A5,++A4[0x0]

Colour of the apple : Green  
 Green Thresholding  
 Average hue value (mean estimate) is 61.48  
 The size of the image is 3474 pixels out of 10000  
 0.942857  
 264 pixels are present along the boundary  
 Extent of match with horizontal circle - 18 percent  
 Extent of match with vertical circle - 19 percent  
 Extent of match with ellipse - 11 percent

Build | Stdout

CPU HALTED | For Help, press F1 | Ln 16, Col 21 | NUM

start

## CHAPTER 8

### CONCLUSION AND FUTURE ENHANCEMENTS

It is testified that machine vision is an alternative to unreliable manual sorting of fruits. The system can be used for fruit grading by the external qualities of size, shape, colour and surface defects. The machine vision system can be developed to quantify quality attributes of various fruits and vegetables such as mangoes, cucumbers, tomatoes, potatoes, peaches and mushrooms. The exploration and development of some fundamental theories and methods of machine vision for apple quality detection and sorting operations would accelerate the application of new techniques to the estimation of agricultural products' quality.

The work in this project has resulted in a clear-cut and systematic sequence of operations to be performed in order to obtain the end result of an apple by categorizing them as good or defected fruit. The proposed steps are based on the assumption that the images were taken under proper illumination, due to which some regions with improper illumination are considered defects. Future work might include a small modification in the presented algorithm in order to adapt to this irregularity and the work can be extended to obtain the internal defects of the fruits so as to improve the quality of agricultural products. This algorithm was tested with several images and the results were encouraging.

**BIBLIOGRAPHY**

- [1] Bin Fang, Wynne Hsu, and Mong Li Lee “On the Accurate Counting of Tumor Cells”, IEEE transactions on nano-bioscience, vol. 2, no. 2, June 2003.
- [2] Blasco J, Aleixos N, Molto E, “Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm”, Journal of food engineering ISSN 0260-8774, 2007, vol. 81, n<sup>o</sup>3, pp. 535-543 [9 page(s) (article)] (3/4 p.)
- [3] Code Composer Studio V3.1 IDE Getting Started Guide, Users Guide, TI Literature Number SPRU509, Texas Instruments, May 2005.
- [4] Deabes W.A, Abdelrahman M.A, “An Image processing approach for surface characterization of the Foam Patterns”, 39th Southeastern Symposium on System Theory MA3.4, Mercer University, Macon, GA, 31207, March 4-6, 2007
- [5] DSP-Based Electromechanical Motion Control by Toliyat and Campbell, 2008
- [6] DSP/BIOS User's Guide, TI Literature Number SPRU423, Texas Instruments, April 2004.
- [7] Hocenski Z, Novak I, “Texture feature Extraction for visual Inspection of Ceramic Tiles” IEEE ISIE 2005, June 20-23, 2005, Dubrovnik, Croatia.
- [8] Jarimopas B, Jaisin N. An Experimental Machine Vision System for Sorting Sweet Tamarind. Journal of Food Engineering, 2008, 89 (3): 291-297.
- [9] Li Q, Wang M, “A fast identification method for fruit surface defect based on fractal characters”, Chinese Journal of graphic Images, vol.5, pp 144-148, 2000.
- [10] Lu.R Detection of bruises on apples using near infrared hyper spectral imaging Transactions of the ASAE, vol.46, pp.523–530, 2003.

- [11] Nasser Kehtarnavaz, Real Time Digital Signal Processing Based on the TMS320C6000, Elseiver, 2005
- [12] Rafael. C. Gonzalez and Richard E. Woods, "Digital Image Processing" second edition, ISBN 0-201-18075-8,pp. 443-447,738-761
- [13] Texas Instruments, TMS320C6000 Code Composer Studio Tutorial, Literature ID# SPRU 301C, 2000.
- [14] TMS320C6000 DSP/BIOS Users Guide, TI Literature Number SPRU303, Texas Instruments, May 2005.
- [15] TMS320C6000 Programmer's Guide, TI Literature Number SPRU198, Texas Instruments, May 2005
- [16] TMS320C6000 DSP/BIOS Application Programming Interface (API) Reference Guide, TI Literature Number SPRU403, Texas Instruments, May 2005.
- [17] TMS320C64x Image/Video Processing Library Programmer's Reference, TI Literature Number SPRU023, Texas Instruments, April 2002.
- [18] TMS320C6000 DSP Peripherals Overview Reference Guide, TI Literature Number SPRU190, Texas Instruments, February 2001.
- [19] TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide, TI Literature Number SPRU234, Texas Instruments, July 2003
- [20] Wen.Z, Tao.Y, Dual-camera NIR/MIR imaging for stem-end/calyx identification in apple defect sorting.Transaction of the ASAE, vol.43, pp.449-452, 2000.
- [21] Xu J H. Image Processing and Analysis. Science Press, Beijing, 1992.
- [22] Zhou Jianmin, Zhou Qixian, Liu Juanjuan, Xu Dongdong,Design of on-line detection system for apple early bruise based on thermal properties analysis International Conference on Intelligent Computation Technology and Automation, 2010