ABSRACT

In the present study, a machine vision based, sorting system was developed, the aim is to sort vegetables especially tomatoes based on the quality as fresh and rotten, to meet consumers' demands. The system comprises a conveying unit, illumination and capturing unit, and sorting unit. Physical and mechanical features were extracted from the samples provided, and the detection algorithm via Matlab by image processing was designed accordingly. An index based on color features was defined to detect tomatoes. Then they were fed on a conveyor belt in a row. When they were at the center of the camera's field of view, a snapshot was taken, the image was processed immediately through Matlab via Raspberrypi and the quality of the tomato was determined. When the tomato passes, a signal was sent to the interface circuit and an appropriate actuator positioned at the end of the conveyor belt, driven by a step motor, was actuated, leading the tomato towards an appropriate port. Thus, the rotten tomatoes are separated from the fresh ones. The algorithms used are ROI and SIFT to detect the cracks and shrinkages in it. The proposed method provides an effective and robust separation means for sorting and in a single-channel space, and it can be easily adapted for other imaging-based agricultural applications.

CHAPTER 1

INTRODUCTION

This topic was chosen from **'The Smart India Hackathon 2019'**.

The image processing and computer vision systems have been widely used for identification, classification, grading and quality evaluation in the agriculture area challenging task for the computer vision to achievenear human levels of recognition. The computer vision for the automatic sorting of vegetables and vegetables from a set, consisting of different kind of goods. In recent years, many types of research have been done on vegetables quality detection by using image processing technique, and many significant results have been obtained. There are many research reports, but so far they are in the experimental stage, and the analysis method is far from practical application. Particularly in the defect identification and maturity detection, the current approach used to deal with very slow. Many companies are moving to automated classification in many crops such as grading on potatoes and mushrooms. The purpose of the study was to implement sift algorithms that can help in automating the process of defected vegetables. Demand from the consumer for quality produces, the consistent behavior of machines in compare with humans, the insufficiency of labor and attempt to reduce labor costs are the primary motivations of proposed system. The primary objective of this work is to design an algorithm that can identify defect and maturity detection of vegetables based on shape and size features by digital image analysis.Matching features across different images in a common problem in computer vision. When all images are similar in nature (same scale, orientation, etc) simple corner detectors can work.

But when you have images of different scales and rotations, you need to use the Scale Invariant Feature Transform. Yield mapping for plant crops by mechanical harvesting requires automatic detection and counting of vegetables in plant canopy. However, partial occlusion, shape irregularity, varying illumination, multiple sizes and similarity with the background make vegetable identification a very difficult task to achieve. Therefore, rottenvegetable detection within a canopy is a challenging task due to all the above mentioned problems. A novel algorithmic technique was used to detect rotten vegetable in plant canopy under natural outdoor conditions. Shape analysis and texture classification were two integral parts of the algorithm. Shape analysis was conducted to detect as many vegetables as possible. Texture classification by a support vector machine (SVM), canny edge detection combined with a graph-based connected component algorithm and Hough line detection, were used to remove false positives. Next, Key points were detected using a scale invariant feature transform (SIFT) algorithm and to further remove false positives.

A majority voting scheme was implemented to make the algorithm more robust. The algorithm was able to accurately detect and count 80.4% of vegetable in a validation set of images acquired from a grove under natural outdoor conditions. The algorithm could be further improved to provide growers early yield estimation so that growers can manage grove more efficiently on a site-specific basis to increase yield and profit.

In this paper, we present a novel image sorting algorithm based on SIFT (Scale-Invariant Features Transform). In the proposed method, matching feature is firstly characterized by calculating the number of matching points between images to find the nearest image of a certain one. After calculating the matching points one by one, an accurate picture order could be finally obtained. The results show the effectiveness of the proposed algorithm.

CHAPTER 2

LITERATURE REVIEW

Fengyun Wang et al, 2018, [1] provided an automatic sorting hardware system is designed that consists of a conveying mechanism, an image acquisition system, a control module and an actuator. Second, we present an image algorithm based on the watershed method, Canny operator, OR operation, and closed operation to determine the pileus diameter of fresh white button mushrooms.

C. S. Nandi et al, 2014, [2] explained recursive feature elimination technique in combination with support vector machine (SVM)-based classifier has been employed to identify the most relevant features among the initially chosen 27 features. Finally, the optimum set of reduced number of features have been obtained and used for classification of the mangoes into four different classes according to the maturity level.

Chong-Yaw Wee et al, 2009, [3] showed the separation distance of one grain from another by the amount of time taken to process the image of each grain. The closer the separation distance of one grain from another will lead to higher output of grains imaged and processed. Hence, the fast computation of a reliable set of pattern features is required in a rice sorting system. In this study we propose a new application of the symmetrical property of Zernike polynomials to compute a set of Zernike moments in rice sorting system.

Ghobad Moradi et al, 2011, [4] studied the vegetables skin color defects. Removing of image background and extraction of vegetable shape, exactly, at presence of shadow and complex background is considered as an important preprocessing stage.

In proposed algorithm at first, background in image is omitted by using active counter model (ACM) algorithm. Finally, the image is segmented using modified FCM (MFCM) algorithm.

Qingsong Meng et al, 2015, [5] eliminated the mismatch points of image registration. At first initial matching with SIFT algorithm is done, and then use the feature of key points in spatial domain to eliminate the error matching, Finally, using RANSAC algorithm to do the final debugging. Experimental results show that the improved SIFT algorithm can better eliminate mismatch points.

Yogesh et al, 2016, [6] proposed the method is based on the use of speeded up robust feature (SURF) and extracts the local feature of the segmented image and describes the object recognition. The objective is to design the defect detection algorithm which will be used for feature extraction and descriptor having less processing time.

Hongshe Dang et al, 2010, [7] realized grading. Experiments show that this embedded grading system has the advantage of high accuracy of grading, high speed and low cost. Vegetable non-destructive detection is the process of detecting vegetables' inside and outside quality without any damage, using some detecting technology to make evaluation according some standard rules. The system takes ARM9 as main processor and develops the vegetables size detecting program using image processing algorithms on the QT/Embedded platform.

D. G. Lowe et al, 2004, [8] recognized matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters.

R. S. Lakshmi et al, 2014, [9] illustrated fruits kept in piles and stock houses need more sophisticated robotic manipulators for in-house inspection. The readings obtained from the sensors or the inline cameras are feed for image processing methods and algorithms for grading. A few to name them are classifiers like neural network and fuzzy based classifiers.

Y.-C. Chen et al, 2012, [10] describes SIFT accelerator which consists of two interactive hardware components, one for key point identification, and the other for feature descriptor generation. Development of a segment buffer scheme that could not only feed data to the computing modules in a data-streaming manner, but also reduce about 50% memory requirement than a previous work. With a parallel architecture incorporating a three-stage pipeline, the processing time of the key point identification is only 3.4ms for one video graphics array (VGA) image.

R. Alsharif et al, 2011, [11] defined the spatial FCM algorithm in pomegranate MR images' segmentation. The algorithm is performed with setting the spatial neighborhood information in FCM and modification of fuzzy membership function for each class. The segmentation algorithm results on the original and the corrupted Pomegranate MR images by Gaussian, Salt Pepper and Speckle noises show that the SFCM algorithm operates much more significantly than FCM algorithm.

H. Sardar et al, 2014, [12] estimated an logical updated algorithm for quality and its standard level of fruits/fruit by its visual external surface color using the non-destructive technique to automated quality verification systems further which can be implemented in image processing to determine quality (i.e. un ripe, partial ripe, ripe or over ripe (bad fruit),in this work color play as key role, other parameters like size, shape ,hardness, softness, day light, day temperature, colorization

J. Wang et al, 2013, [13] presented a low-cost embedded system based on a new architecture that successfully integrates FPGA and DSP. It optimizes the FPGA architecture for the feature detection step of SIFT to reduce the resource utilization, and optimizes the implementation of the feature description step using a high-performance DSP.

Jarimopas et al, 2008, [14] has done an experiment that involves the use of pods from two sweet tamarind cultivars: “Sitong” and “Srichompoo”. The sorting system involved the use of a CCD camera which was adapted to work with a TV card, microcontrollers, sensors, and a microcomputer. Analysis was performed with image processing software.

Pourdarbani R et al 2015, [15] experimented date fruits on a conveyor belt in a row. When they were at the center of the camera’s field of view, a snapshot was taken, the image was processed immediately and the maturity stage of the Date was determined. When the Date passed the sensor, positioned at the end of the conveyor belt, a signal was sent to the interface circuit and an appropriate actuator, driven by a step motor, was actuated, leading the Date toward an appropriate port.

CHAPTER 3

PROBLEM DESCRIPTION

3.1 PROBLEM IDENTIFICATION

Based on the literature survey, the following problems were identified in our project:

- Top and bottom side images of a vegetable
- Individual linear movement of vegetables
- Complexity of better-quality images
- Shadows of the vegetables
- Separator system after image processing
- Installing OpenCV to Raspberrypi

Top and bottom side images

When the conveyor belt is made up of a normal sheet containing the vegetables, the images are taken so that only the top side is focused, leaving the bottom side. Thus, the vegetable rotten rate could not be fully discovered leading to an improper sorting. Hence the bottom side of the vegetable should also be considered. For this we have planned to use a transparent sheet as the conveyor belt making the bottom side to be visible when looking from below. By this method of a transparent belt, the image can be captured either by two ways to fulfil the needs:

- Rotation of camera by a specific angle
- Using a reflection mirror below the belt

Using a servo motor the camera is made to rotate around the conveyor belt with a specific angle.

By doing so first the top side image of the vegetable is taken followed by the rotation of the camera and finally the bottom side image is also captured. Now both these images are processed by Matlab and as a result image processing is done for detecting the rotten goods

The other method is to use a mirror placed in a particular angle to obtain the bottom side image. In this case the camera doesn't need to move an inch, rather will capture a single image containing both the vegetable image and the reflected image from the mirror. But here the camera should also be placed in an angle to capture both these images.

Individual linear movement of vegetables

The vegetables should move in a linear way and not by a cluster, which makes difficult to detect the defects. Thus, a gear system is placed just before the conveyor belt. This gear system is designed in such a way that when the goods fall as a whole, it accepts only one vegetable per gap of a particular diameter. Now a motor is used to rotate the gear system slowly making the goods to move in a linear way. At first, the gear system was made up of steel which was difficult to rotate and the torque was also very low. Later we made it by plastic pipes and hence, now the rotation is smooth and efficient.

Complexity of better-quality images

A Camera with 2MP cannot be used as the images captured by it are a little blurry and calculates an approximate defect detection by Matlab. Thus, a 5MP Logitech C310 is used for better quality image so that even a small shrinkage or crack in the vegetable is detected and further processed via image processing. This camera should also consider the transparent belt while capturing the image as it is a long distant object. Hence a 5MP camera is more than enough.

Shadows of the vegetables

The vegetables lying on the conveyor needs to be recognized better by the camera for good image processing. Hence the light effect should be optimum. Low light effect makes the image less visible resulting in improper detection of defects. High light effect creates shadows of vegetables considering it as another object. This leads to the failure of defect detection as the keypoints extend more than the vegetables even though ROI (Region of Interest) algorithm is implemented.

Separator system after image processing

Once the images are processed via Matlab, the threshold values are updated by the keypoints. Thus, when total value crosses the given constant threshold value the separator system is moved to a 45-degree left so that the rotten goods fall aside. When the values are less than the constant threshold value, vice-versa takes places and finally the goods are separated.

Installing OpenCV to Raspberrypi

There were 2 methods of image processing. Either installing OpenCV or installing Matlab to Raspberrypi and then processing the input image. But we experienced problems in installing OpenCV as errors and malfunctioning occurred. So, we decided to process the image by Matlab through Raspberrypi.

3.2 OBJECTIVES

To automate the process of sorting agricultural goods, such as vegetables for an efficient mechanized system. This automation results in effectively classifying goods based on their quality and other aspects that determine fresh goods, with the help of image processing via Matlab. The classification results in fresh and rotten goods. The rotten goods are separated and thereby the separated goods are packed.

Based on the literature survey and problems identified, the objectives are refined as follows:

- To identify / develop an algorithm using Matlab instead of OpenCV to design a sorting system
- To obtain the top and bottom side images of the vegetable for better defect detection
- To establish a continuous individual linear movement of vegetables for accurate processing
- To get a better quality image of the vegetables by a camera and also have an optimum light effect

CHAPTER 4

PROPOSED SYSTEM DESCRIPTION

4.1 PROPOSED SYSTEM

The system proposed here is to mainly overcome the drawbacks of the existing system and it also has some added features. The sift algorithm is the appropriate and effective classification algorithm to be used in the Vegetables detection System. The system that has been developed is able to recognize all the test vegetable images which are being selected by user or system tester from the vegetable selection menu on the system. There are some future works to be implemented on the Vegetables Recognition System in order to improve and enhance the functionality and flexibility of the system for more widely usage. The system should be improved by extending its functions to process and recognize more variety of different vegetable images. In SIFT technique, which provides consistent, reasonably accurate, less time consuming and cost effective solution. The results of the system are accurate up to 90%.

The introduced technique depends on the SIFT algorithm to elicit solid features which enable it to specify if a region of an image was a copy-moved. The introduced technique decreases time complexity of SIFT using FCM clustering method. In the proposed algorithm, SIFT keypoints are clustered on the basis of their descriptors then, center keypoint and its neighbor are matched with other center keypoint and its neighbor clusters instead of identifying all keywords in the picture.

- When traditional SIFT algorithm for two images extracted respective feature points matching, firstly calculate each feature point of the first image nearest neighbor match in second image, that is the minimum Euclidean distance of key points between the descriptor vector.

- Lowe's SIFT algorithm uses conventional nearest neighbor and next nearest neighbor distance ratio to execute matching feature points, the experience value of ratio is 0.8. Because of the high-dimensional feature space, a similar distance may have a number of errors match.

- The traditional SIFT methods also do not consider duplicate matching, several-for-one matching and other false match; matching accuracy has greatly optimized space. While calculating the direction of SIFT key points, the same key point may have a main direction, one or more secondary direction. In SIFT algorithm we can be classified them into different feature points.

- All or part of the feature points that might occur corrects point, but they are actually the same point, this will generate repeat matching phenomenon. Exhaustive search for matches among SIFT feature may also have one-to-many, several-forgone match. In addition to duplicate matching, and many-to-one-matching, inevitably may also have other mismatch. Those need to be eliminated one by one; otherwise the subsequent image registration accuracy will have great impact.

4.2 PROPOSED SYSTEM BLOCK DIAGRAM

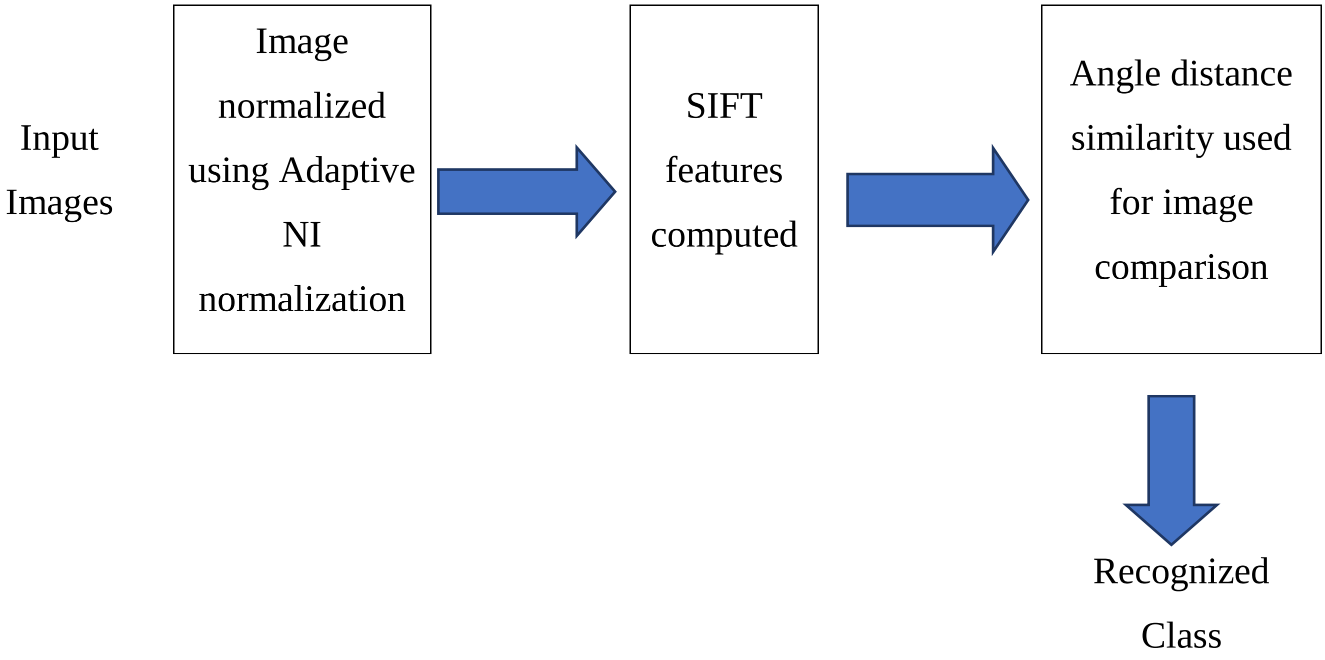


Fig 4.1 Block Diagram Of SIFT

4.3 BLOCK DIAGRAM EXPLANATION

The main purpose of this design is to provide an FPGA implementation of SIFT feature detection module for image processing applications. Feature detection is carried out by one of the most efficient feature detection algorithm, i.e., Scale Invariant Feature Transform algorithm. This design is based on two stages of processing. It includes scale space extreme detection and keypoint localization. Here we are considering the reduction in complexity of software implementation by using Xilinx system generator platform with Verilog HDL coding. The input image is driven from a digital camera for real time implementation and the pixel values are recorded for the key point detection purpose.

This is an optimized pure software implementation of feature detector using the Xilinx ISIM simulator with Verilog coding.

The process diagram of the proposed system is the process diagram of the proposed system. Here the modification is done in the Gaussian filter section by reducing the resource utilization and thus produce an optimized design for feature detection in SIFT algorithm.

- The introduced technique depends on the SIFT algorithm to elicit solid features which enable it to specify if a region of an image was a copy-moved. The introduced technique decreases time complexity of SIFT using FCM clustering method. In the proposed algorithm, SIFT keypoints are clustered on the basis of their descriptors then, center keypoint and its neighbor are matched with other center keypoint and its neighbor clusters instead of identifying all keywords in the picture.
- The proposed algorithm has the ability to reveal Copy Move forgery very fast without influencing the accuracy of matching process. Illustrates a block diagram of the proposed algorithm. First, SIFT algorithm from the image is used to elicit key points. Then, the feature descriptor is elicited from every key point on the image including 128 dimensional. The resemblance between the descriptors is calculated to specify the matching among the descriptors for specifying the potential forgery on the image. The basic obstacle in this algorithm is the computational complexity of the matching stage where it is very high as a result of the big number of key points elicited from the image and the matching process among them. Using clustering algorithm for clustering the keypoints depending on their descriptors can be a solution to this issue. Specifying data points for every cluster such that items in the same cluster as similar as possible, but items that belong to the diverse clusters are as various as possible.

CHAPTER 5

HARDWARE DESCRIPTION

5.1 CONVEYOR

5.1.1 Conveyor system

A **conveyor system** is a common piece of mechanical handling equipment that moves materials from one location to another. Conveyors are especially useful in applications involving the transportation of heavy or bulky materials. Conveyor systems allow quick and efficient transportation for a wide variety of materials, which make them very popular in the material handling and packaging industries.

5.1.2 Conveyor belt

A **conveyor belt** is the carrying medium of a **belt conveyor system** (often shortened to belt conveyor). A belt conveyor system is one of many types of conveyor systems. We are using a **Transparent belt** for acquiring the front and back images of the vegetable. A belt conveyor system consists of two or more pulleys (sometimes referred to as drums), with an endless loop of carrying medium—the conveyor belt—that rotates about them. One or both of the pulleys are powered, moving the belt and the material on the belt forward. The powered pulley is called the drive pulley while the unpowered pulley is called the idler pulley. There are two main industrial classes of belt conveyors; Those in general material handling such as those moving boxes along inside a factory and bulk material handling such as those used to transport large volumes of resources and agricultural materials.

Belt conveyors are the most commonly used powered conveyors because they are the most versatile and the least expensive.

Product is conveyed directly on the belt so both regular and irregular shaped objects, large or small, light and heavy, can be transported successfully. These conveyors should use only the highest quality premium belting products, which reduces belt stretch and results in less maintenance for tension adjustments.

Belt conveyors can be used to transport product in a straight line or through changes in elevation or direction. In certain applications they can also be used for static accumulation or cartons.

5.1.3 Conveyor pulley

A conveyor pulley is a mechanical device used to change the direction of the belt in a conveyor system, to drive the belt, and to tension the belt. Modern pulleys are made of rolled shells with flexible end disks and locking assemblies. Pulleys are made up of several components including the shell, end disk, hub, shaft and locking assembly. The end disk and hub may be on piece. The locking assembly may also be replaced with a hub and bushing on lower tension pulleys.

5.2 RASPBERRY PI VERSION 2

Raspberry pi is more than computer as it provides access to the on-chip hardware i.e. GPIOs for developing an application. By accessing GPIO, we can connect devices like LED, motors, sensors, etc and can control them too. It has ARM based Broadcom Processor SoC along with on-chip GPU (Graphics Processing Unit)The CPU speed of Raspberry Pi varies from 700 MHz to 1.2 GHz. Also, it has on-board SDRAM that ranges from 256 MB to 1 GB. Raspberry Pi also provides on-chip SPI, I2C, I2S and UART modules.

Fig 5.1. Raspberry pi version 2

While the Raspberry Pi uses an SD card for its main storage device is known as a boot device you may find that you run into space limitations quite quickly. Although large SD cards holding 32 GB, 64 GB or more are available, they are often prohibitively expensive. Thankfully, there are devices that provide an additional hard drive to any computer when connected via a USB cable. Known as USB Mass Storage (UMS) devices, these can be physical hard drives, solid-state drives (SSDs) or even portable pocket-sized flash drives.

A 10/100 USB Ethernet adapter with the numbers referring to its two-speed mode, 10 Mb/s and 100 Mb/s can be purchased from online retailers for very little money. When buying an Ethernet adapter, be sure to check that Linux is listed as a supported operating system. A few models only work with Microsoft Windows, and are incompatible with the Raspberry Pi.

5.2.1 POWER REQUIREMENT

The Raspberry Pi is powered by the small micro USB connector found on the lower left side of the circuit board. This connector is the same as found on the majority of smart phones and some tablet devices. Many chargers designed for smart phones will work with the Raspberry Pi, but not all.

The Pi is more power-hungry than most micro-USB devices, and requires up to 700mA in order operating. Some chargers can only supply up to 500mA, causing intermittent problems in the Pi's operation connecting the Pi to the USB port on a desktop or laptop computer is possible, but not recommended. As with smaller chargers, the USB ports on a computer can't provide the power required for the Pi to work properly.

Only connect the micro-USB power supply when you are ready to start using the Pi. With no power button on the device, it will start working the instant power is connected and can only be turned off again by physically removing the power cable.

Features:

- 800mA operating current
- 200µA during sleeping mode
- Processor: ARM V6 Single core
- Operating voltage :5.1V
- Processor speed:700 MHz

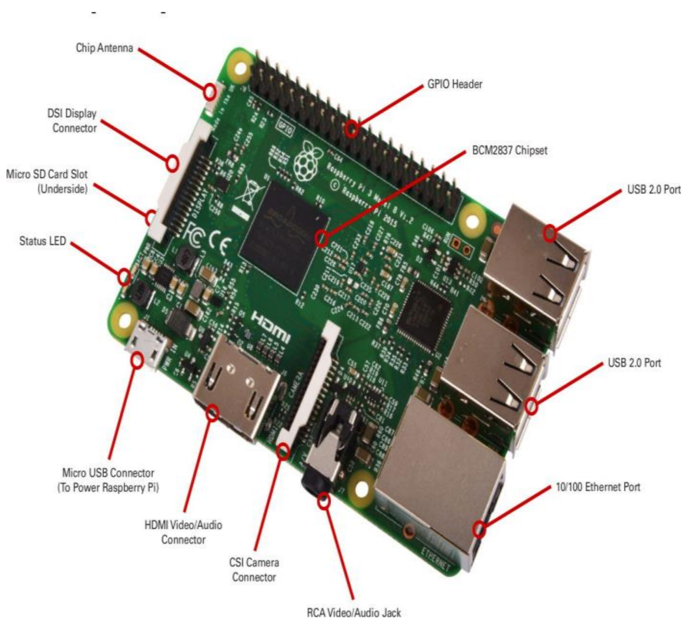


FIG 5.2 Power Requirement

5.2.2 Peripherals and I/O:

The Raspberry pi version 2 has total 40 GPIO pins broken out to the pin headers on both sides of the development board. If you want to reduce the number of power sockets in use, connect the Raspberry Pi's USB power lead to your powered USB hub. This way, the Pi can draw its power directly from the hub, rather than needing its own dedicated power socket and mains adapter. This will only work on hubs with a power supply capable of providing 700mA to the Pi's USB port, along with whatever power is required by other peripherals. The alternate functions are usually peripheral I/Os, and most peripherals appear twice to allow flexibility on the choice of I/O voltage.

Peripherals:

- 48x GPIO
- 2x I2C
- 2x SPI
- 2x UART
- 2x SD/SDIO
- 1x HDMI 1.3a
- 1x USB2 HOST/OTG
- 1x DPI (Parallel RGB Display)
- 1x NAND interface (SMI)
- 1x 4-lane CSI Camera Interface (up to 1Gbps per lane)
- 1x 2-lane CSI Camera Interface (up to 1Gbps per lane)
- 1x 4-lane DSI Display Interface (up to 1Gbps per lane)
- 1x 2-lane DSI Display Interface (up to 1Gbps per lane)

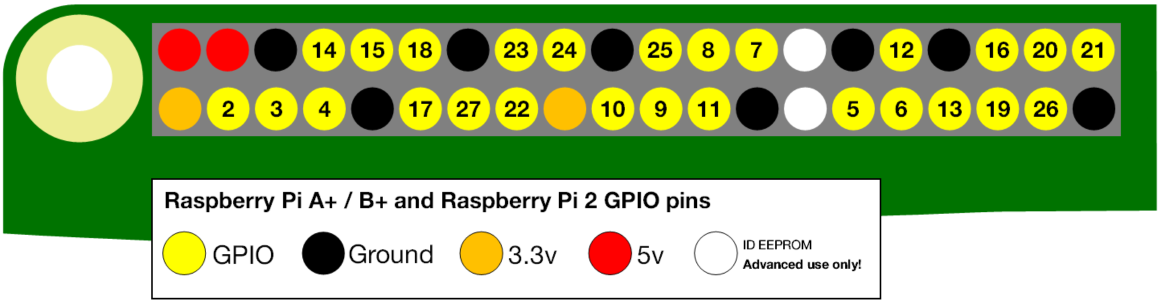


Fig 5.3. Raspberry pi GPIO pin out

5.2.3 ARCHITECTURE OF RASPBERRY PI:

Raspberrypi board is a miniature marvel, packing considerable computing power into a footprint no larger than a credit card. It’s capable of some amazing things, but there are a few things you’re going to need to know before you plunge head-first into the bramble patch.

The processor at the heart of the Raspberry Pi system is a Broadcom BCM2835 system-on-chip (SoC) multimedia processor. This means that the vast majority of the system's components, including its central and graphics processing units along with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB memory chip at the center of the board.

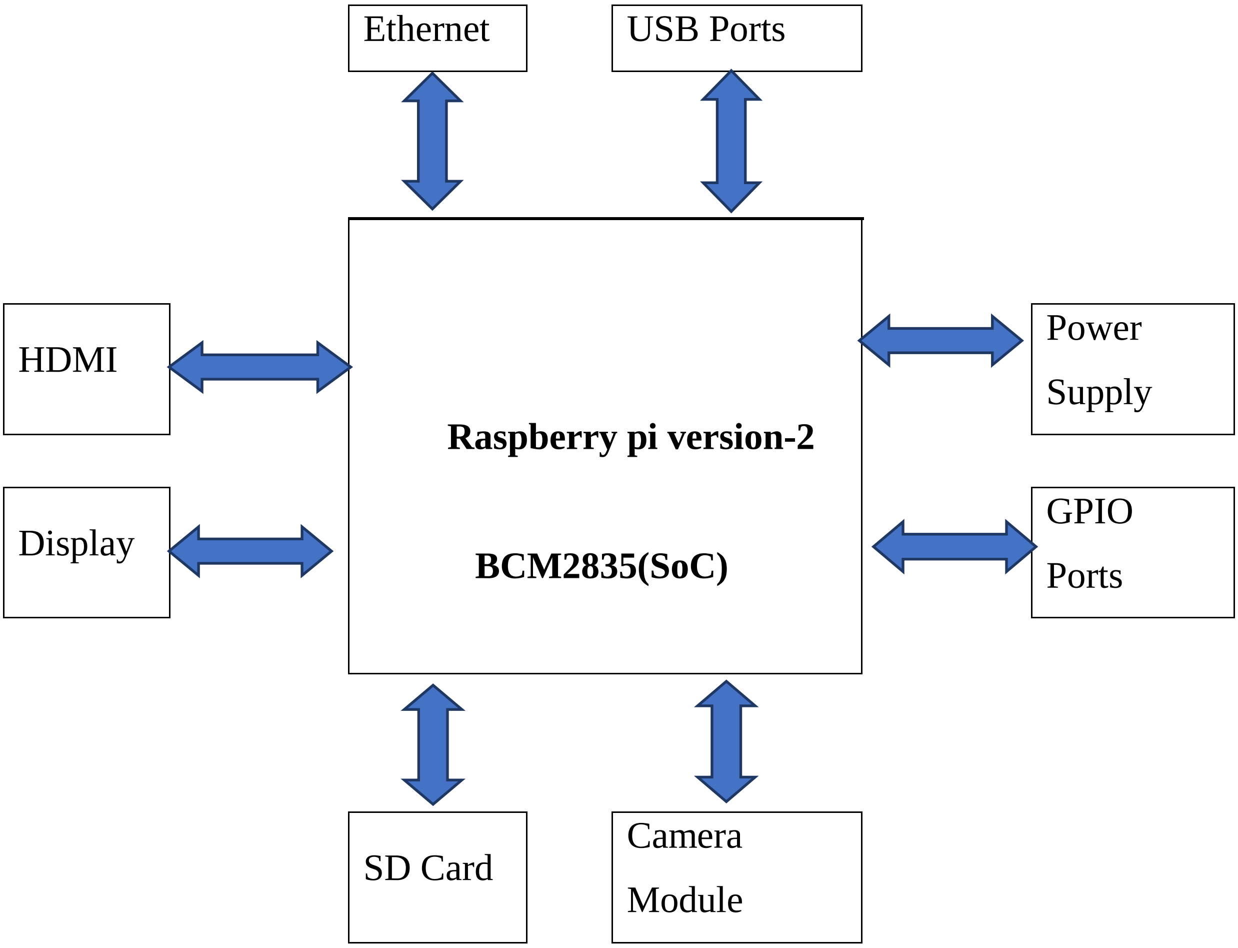


Fig 5.4. Architecture of Raspberry pi

The BCM2835 SoC, located beneath a Hynix memory chip Developed by Acorn Computers back in the late 1980s, the ARM architecture is a relatively uncommon sight in the desktop world. Where it excels, however, is in mobile devices: the phone in your pocket almost certainly has at least one ARM-based processing core hidden away inside.

Its combination of a simple reduced instruction set (RISC) architecture and low power draw make it the perfect choice over desktop chips with high power demands and complex instruction set (CISC) architectures.

5.3 CAMERA MODULE

A camera module is an image sensor integrated with a lens, control electronics, and an interface like CSI, Ethernet or plain raw low-voltage differential signaling.

An Internet Protocol camera, or IP camera, is a type of digital video camera that receives control data and sends image data via the Internet. They are commonly used for surveillance. Unlike analog closed-circuit television (CCTV) cameras, they require no local recording device, but only a local area network. Most IP cameras are webcams, but the term *IP camera* or netcam usually applies only to those used for surveillance that can be directly accessed over a network connection.

5.3.1. FUNCTIONS OF CAMERA MODULE

A webcam is a hardware camera and device that connects to a computer and the Internet and captures either still pictures or motion video of a user or another object. The picture of the Logitech Webcam C310 is an example of what a webcam may look.

Today, most webcams are either embedded into the display with laptop computers or connected to the USB or FireWire port on the computer.

A webcam is a video camera that feeds or streams its image in real time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks via systems such as the internet, and emailed as an attachment.

CHAPTER 6

SOFTWARE DESCRIPTION

6.1 MATLAB

6.1.1 MATLABIDE

MATLAB IDE(*matrix laboratory*) is a paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

6.1.2 Interfacing with other languages

MATLAB can call functions and subroutines written in the programming languages C or Fortran. A wrapper function is created allowing MATLAB data types to be passed and returned. MEX files (MATLAB executables) are the dynamically loadable object files created by compiling such functions. Since 2014 increasing two-way interfacing with Python was being added.

Libraries written in Perl, Java, ActiveX or .NET can be directly called from MATLAB, and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries.

Calling MATLAB from Java is more complicated, but can be done with a MATLAB toolbox which is sold separately by MathWorks, or using an undocumented mechanism called JMI (Java-to-MATLAB Interface), (which should not be confused with the unrelated Java Metadata Interface that is also called JMI). Official MATLAB API for Java was added in 2016.

As alternatives to the MuPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Maple or Mathematical. Libraries also exist to import and export MathML.

6.1.3 Image processing tool box:

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development. You can perform image segmentation, image enhancement, noise reduction, geometric transformations, image registration, and 3D image processing.

Image Processing Toolbox apps let you automate common image processing workflows. You can interactively segment image data, compare image registration techniques, and batch-process large data sets. Visualization functions and apps let you explore images, 3D volumes, and videos; adjust contrast; create histograms; and manipulate regions of interest (ROIs).

Image Enhancement

Increase the signal-to-noise ratio and accentuate image features by modifying the colors or intensities of an image. Perform convolution and correlation, remove noise, adjust contrast, and remap the dynamic range.

Image Deblurring

Correct blurring caused by out-of-focus optics, movement by the camera or the subject during image capture, atmospheric conditions, short exposure time, and other factors.

3D Visualization

Explore a 3D volume by using different visualization methods to explore the structure of the data. You can map the pixel intensity of a 3D volume to opacity to highlight a specific region within the volume.

3D Processing

Use many 3D-specific functions in addition to ND functions that enable complete image processing workflows with 3D data.

6.1.4 MATLAB Support Package for Raspberry Pi Hardware

MATLAB Support Package for Raspberry Pi Hardware enables to communicate with a Raspberry Pi remotely from a computer running MATLAB or through a web browser with MATLAB Online that can acquire data from sensors and imaging devices connected to the Raspberry Pi and process them in MATLAB. Matlab can also communicate with other hardware through the GPIO, serial, I2C, and SPI pins.
Ethernet cable was used communicate with raspberry pi

The support package functionality is extended if you have MATLAB Coder. With MATLAB Coder, you can take the same MATLAB code used to interactively control the Raspberry Pi from your computer and deploy it directly to the Raspberry Pi to run as a standalone executable.

```
mypi = raspi
```

```
        mypi =

  raspi with properties:

           DeviceAddress: 'raspberrypi-computername'
                    Port: 18725
               BoardName: 'Raspberry Pi Model B Rev 2'
           AvailableLEDs: {'led0'}
    AvailableDigitalPins: [4 7 8 9 10 11 14 15 17 18 22 23 24 25 27 30 31]
    AvailableSPIChannels: {}
      AvailableI2CBuses: {'i2c-0'  'i2c-1'}
             I2CBusSpeed: 100000
```

Fig 6.1 Connection of Matlab and Raspberrypi

6.1.5 Servo motor with raspberry pi:

For communicating the servo motor, we create object to indicate servo with pulse width modulation giving pulse duration different servo motor contain different pulse duration values Configure a servo object using the PWM pin 9 and set the minimum pulse duration to 1e-3 seconds and the maximum pulse durations to 2e-3 seconds. Write 1, to turn the motor by the maximum angle. The angle depends on device pulse duration. Refer to your device data sheet for valid pulse duration values.

6.2 INTRODUCTION TO RASPERRY PI IDE:

Raspberrypi has a wide range of IDEs that provide programmers with good interfaces to develop source code, applications and system programs. Let's explore the top IDEs for Raspberry Pi. Well there are other OS's available to the raspberry pi on the ROM site which require or can allow better use for other languages, but the main reason why python is preferred for use on the raspberry pi is because it is a lab-on-a-chip where it’s more for educational use than anything where we all know python.

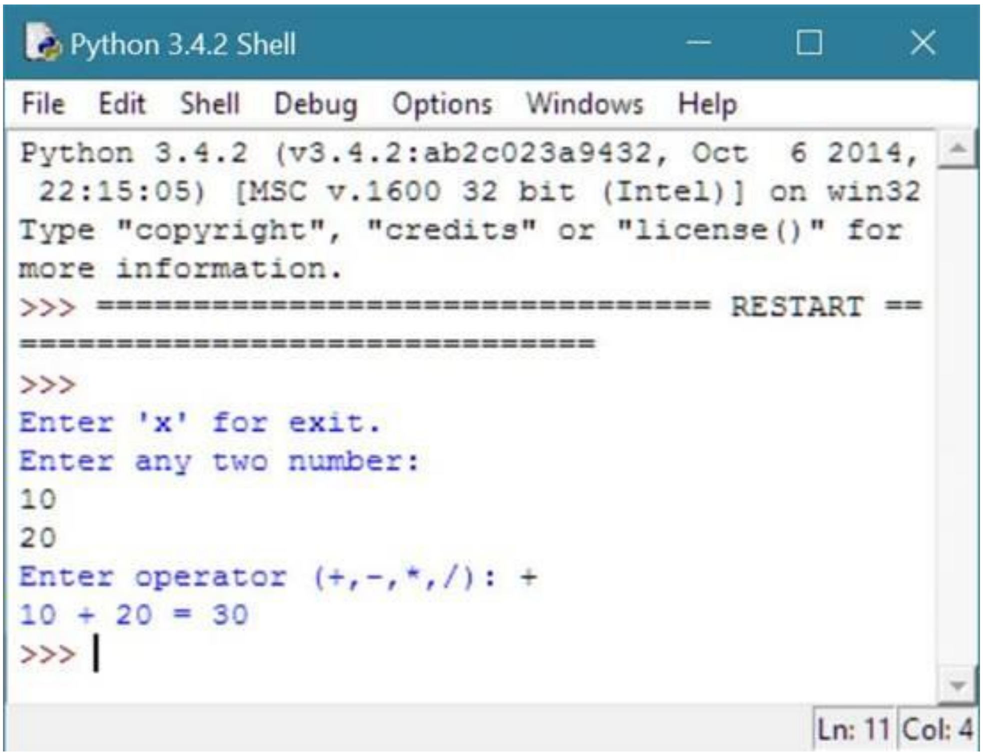


Fig 6.2. Raspberry PI IDE screenshot

Python is a very useful programming language that has an easy to read syntax, and allows programmers to use fewer lines of code than would be possible in languages such as assembly, C, or Java.

The Python programming language actually started as a scripting language for Linux. Python programs are similar to shell scripts in that the files contain a series of commands that the computer executes from top to bottom.

Raspberry PI IDE Features

Sketchbook unlike C programs, Python programs don't need to be compiled before running them. However, you will need to install the Python interpreter on your computer to run them. The Python interpreter is a program that reads Python files and executes the code.

It is possible to run Python programs without the Python interpreter installed though. Programs like Py2exe or Pyinstaller will package your Python code into stand-alone executable programs.

6.3 SIFT (Scale-Invariant Feature Transform)

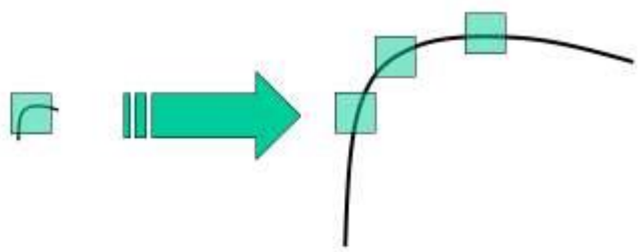
Goal

In this chapter,

- We will learn about the concepts of SIFT algorithm
- We will learn to find SIFT Key points and Descriptors.

Theory

In last couple of chapters, we saw some corner detectors like Harris etc. They are rotation-invariant, which means, even if the image is rotated, we can find the same corners. It is obvious because corners remain corners in rotated image also. But what about scaling? A corner may not be a corner if the image is scaled. For example, check a simple image below. A corner in a small image within a small window is flat when it is zoomed in the same window. So Harris corner is not scale invariant.



So, in 2004, D.Lowe, University of British Columbia, came up with a new algorithm, Scale Invariant Feature Transform (SIFT) in his paper Distinctive, Image Features from Scale-Invariant Key points, which extract Key points and compute its descriptors. (This paper is easy to understand and considered to be best material available on SIFT. So, this explanation is just a short summary of this paper).

There are mainly four steps involved in SIFT algorithm. We will see them one-by-one.

1. Scale-space Extrema Detection

From the image above, it is obvious that we can't use the same window to detect Key points with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values. LoG acts as a blob detector which detects blobs in various sizes due to change in σ. In short, σ acts as a scaling parameter. So, we can find the local maxima across the scale and space which gives us a list of (x, y, σ) values which means there is a potential keypoint at (x,y) at σ scale. But this LoG is a little costly, so SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different σ, let it be σ and $k\sigma$.

This process is done for different octaves of the image in Gaussian Pyramid.

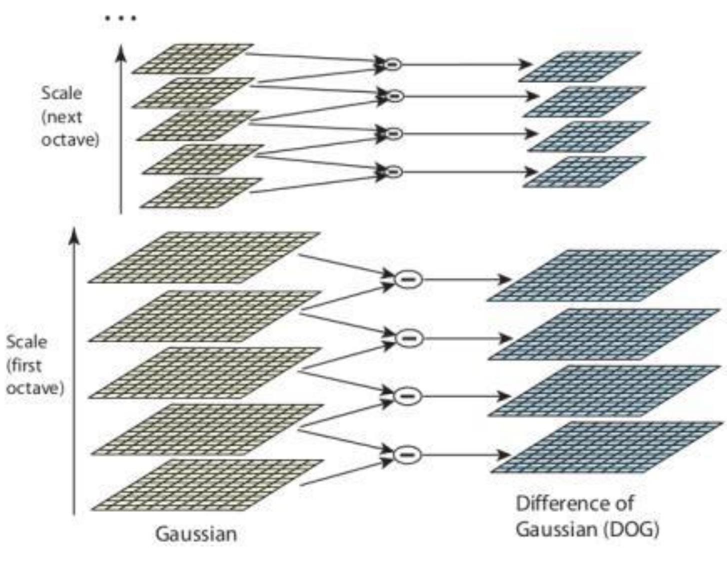


Fig 6.3 Image in Gaussian Pyramid

Once this DoG is found, images are searched for local extrema over scale and space. For e.g. one pixel in an image is compared with its 8 neighbors as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extremum, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in below image:

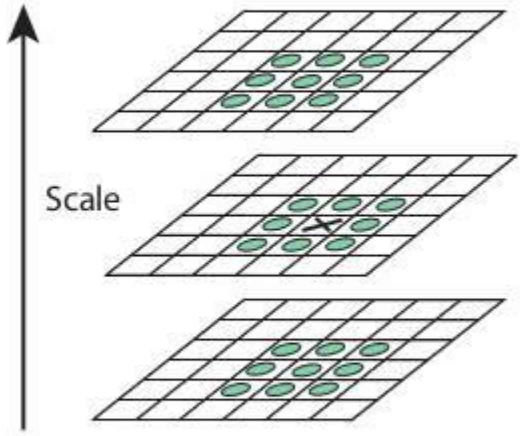


Fig 6.4 Keypoint Representation

Regarding different parameters, the paper gives some empirical data which can be summarized as, number of octaves = 4, number of scale levels = 5, initial $\sigma = 1.6$, $k = \sqrt{2}$ etc. as optimal values.

2. Keypoint Localization

Once potential Key points locations are found, they have to be refined to get more accurate results. They used Taylor series expansion of scale space to get more accurate location of extrema, and if the intensity at this extremum is less than a threshold value (0.03 as per the paper), it is rejected. This threshold is called contrast Threshold in Open CV

DoG has higher response for edges, so edges also needed to be removed. For this, a concept similar to Harris corner detector is used. They used a 2x2 Hessian matrix (H) to compute the principal curvature. We know from Harris corner detector that for edges, one Eigen value is larger than the other. So here they used a simple function, If this ratio is greater than a threshold, called edge Threshold in OpenCV, that keypoint is discarded. It is given as 10 in paper.

So, it eliminates any low-contrast Key points and edge Key points and what remain is strong interest points. Use the Key points scale to select the Gaussian smoothed image L, from above

Compute gradient magnitude, m

$$\mathbf{m(x,y) = \sqrt{(L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y-1))^2}}$$

- Form an orientation histogram from gradient orientations of sample points
- Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation
- Some points will be assigned multiple orientations
- Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position.

3. Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created. (It is weighted by gradient magnitude and gaussian-weighted circular window with σ equal to 1.5 times the scale of keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates Key points with same location and scale, but different directions. It contributes to stability of matching.

4. Keypoint Descriptor

Now keypoint descriptor is created. A 16x16 neighborhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 sizes. For each sub-block, 8 bin orientation histogram is created. So, a total of 128 bin values are available. It is represented as a vector to form key point descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc.

5. Keypoint Matching

Key points between two images are matched by identifying their nearest neighbors. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminators around 90% of false matches while discards only 5% correct matches, as per the paper.

So, this is a summary of SIFT algorithm. For more details and understanding, reading the original paper is highly recommended. Remember one thing, this algorithm is patented. So, this algorithm is included in Non-free module in Open CV.

6.3.1 Improved SIFT Algorithm:

A. Traditional SIFT Algorithm:

When traditional SIFT algorithm for two images extracted respective feature points matching, firstly calculate each feature point of the first image nearest neighbor match in second image, that is the minimum Euclidean distance of key points between the descriptor vector. By previously described, the key point SIFT descriptor is 128-dimensional feature vector. Lowe's SIFT algorithm uses conventional nearest neighbor and next nearest neighbor distance ratio to execute matching feature points, the experience value of ratio is 0.8.

Because of the high-dimensional feature space, a similar distance may have a number of errors match. The traditional SIFT methods also do not consider duplicate matching, several-for-one matching and other false match; matching accuracy has greatly optimized space. While calculating the direction of SIFT key points, the same key point may have a main direction, one or more secondary direction. In SIFT algorithm we can be classified them into different feature points. All or part of the feature points that might occur corrects point, but they are actually the same point, this will generate repeat matching phenomenon. Exhaustive search for matches among SIFT feature may also have one-to-many, several-forgone match. In addition to duplicate matching, and many-to-one-matching, inevitably may also have another mismatch. Those need to be eliminated one by one; otherwise the subsequent image registration accuracy will have great impact.

B. Improved Traditional SIFT Algorithm:

We base on the initial matching points obtained after near neighbors and secondary ratio registration, and then eliminate errors one by one. Among them, for repetitive match and several-for-one match, extract all of them by comparing with the index value of the pixel and coordinates of the corresponding point pairs, to guarantee the uniqueness and one-to-one correspondence.

The mismatch in other areas, we based on geometry relationship, excluding the coordinate difference of corresponding point from apparent greater or less than adjacent points coordinate difference. Owing to the adjacent point, its corresponding matching point in the geometric position could also be adjacent. If the difference between matching points with the coordinates of the adjacent match points difference is very obvious, it is likely to be false matches.

The above steps of matching points process can only be guaranteed the key point B in image 2(Two images that to be matched are image 1& 2) is a matching point of point A in image 1.

Conversely the key point B of image 2 is not necessarily the matching point to A in image 1,so there be false matching points. Thus, we introduce the idea of bilateral matching. After above processing steps, the rest matching points are basically correct. Finally, for the rest matching points, we use RANSAC algorithm improve matching accuracy.

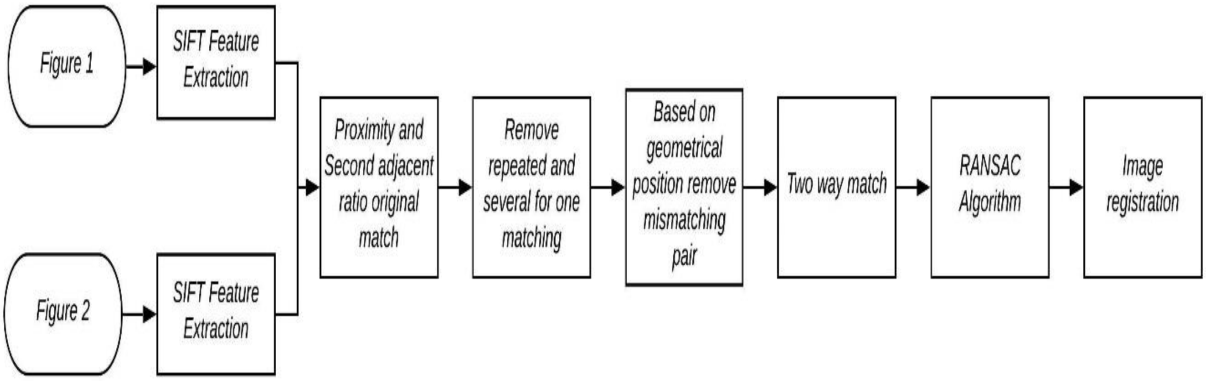


Fig 6.5 SIFT Algorithm block diagram

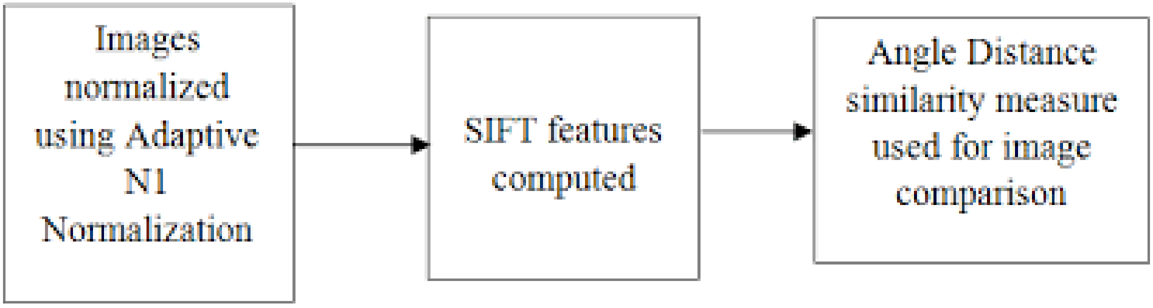


Fig 6.6 Improved SIFT Algorithm

6.4 COLOUR DETECTION FROM IMAGE:

An RGB image is a colorful image consisting of fixed values of color contents for each pixel. These color contents have different values ranging from 0 to 255.There are inbuilt functions and commands available in MATLAB to extract the required color content from an RGB image. we required extracting a particular color from an RGB image, there are no integral commands that we use directly to do so. For such type of operations, we required some algorithms. A simple algorithm is introduced having series of MATLAB commands and looping statements to extract a particular color from an RGB image. It is very helpful in image processing such as in pattern reorganization and mapping to find best equivalent used in many application fields. To extract a particular color from an RGB image or extract a particular area of interest for processing then we have no need to course the whole image. We have a smaller number of values for processing further. It becomes easier to process the image for some other errands. So, a simple algorithm or a simple method is introduced in this paper to extract a requisite area of interest and a particular color from an RGB image. That can show the red color objects in image. Tomato is having red color and given image having Tomato and conveyor so that we need to separate the conveyor background. Using red color detection algorithm, the tomato was shown in a bounding box. With the help of cropping code, we can crop the tomato only on the image so that we can get accurate keypoints value from SIFT algorithm

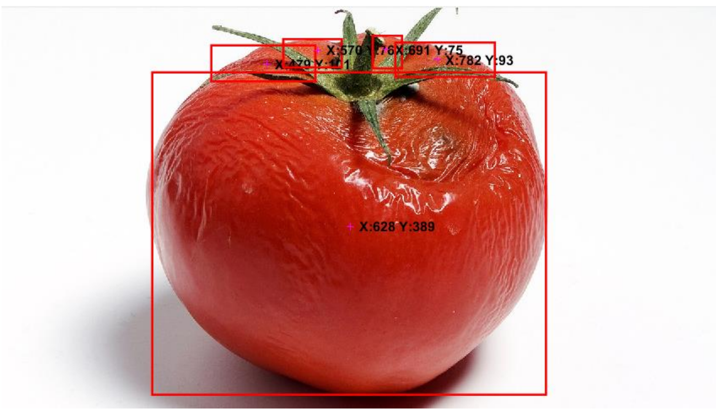


Fig 6.7 Red Color Detection

CHAPTER 7

RESULTS AND DISCUSSIONS

The number of the collected vegetable images was used to develop and train the system; Area and perimeter values for each type of vegetable that have been stored during system training. These stored shape roundness values, area and perimeter values are being used as standard feature values for comparison and classification of query or input vegetable image to the system. The test results are accurate for entire testing vegetable set.

ROI Image

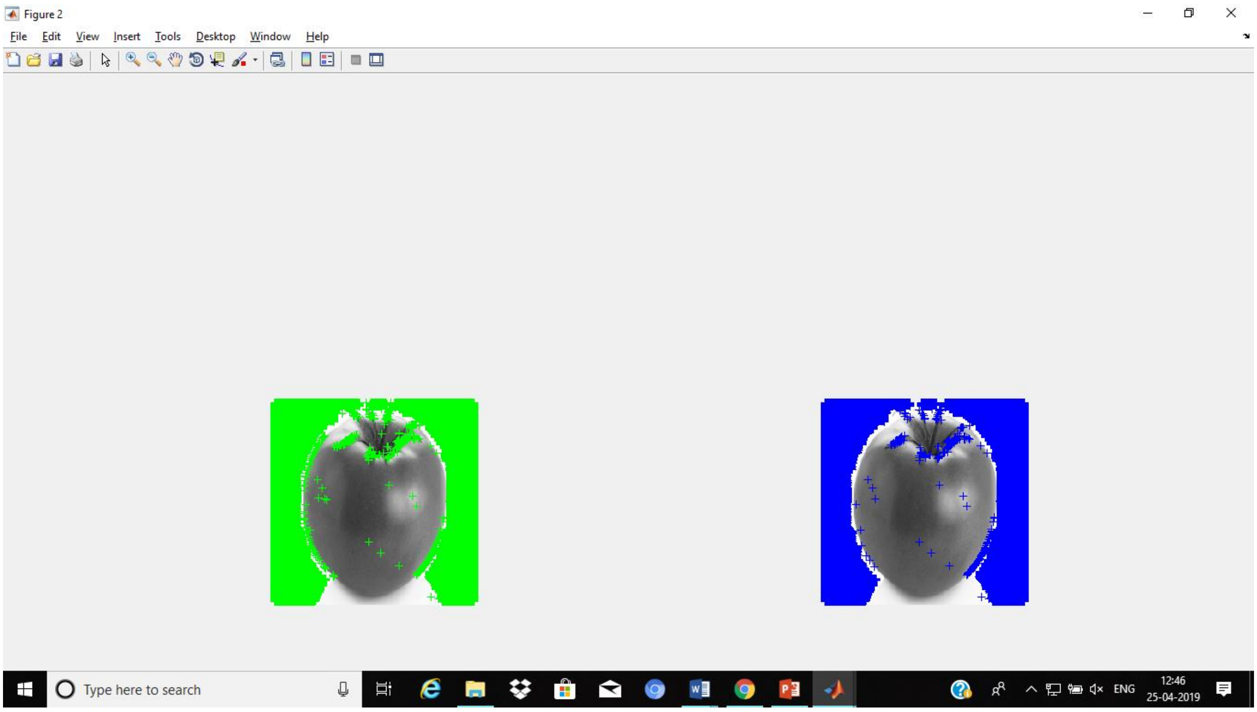


Fig 7.1 ROI Image

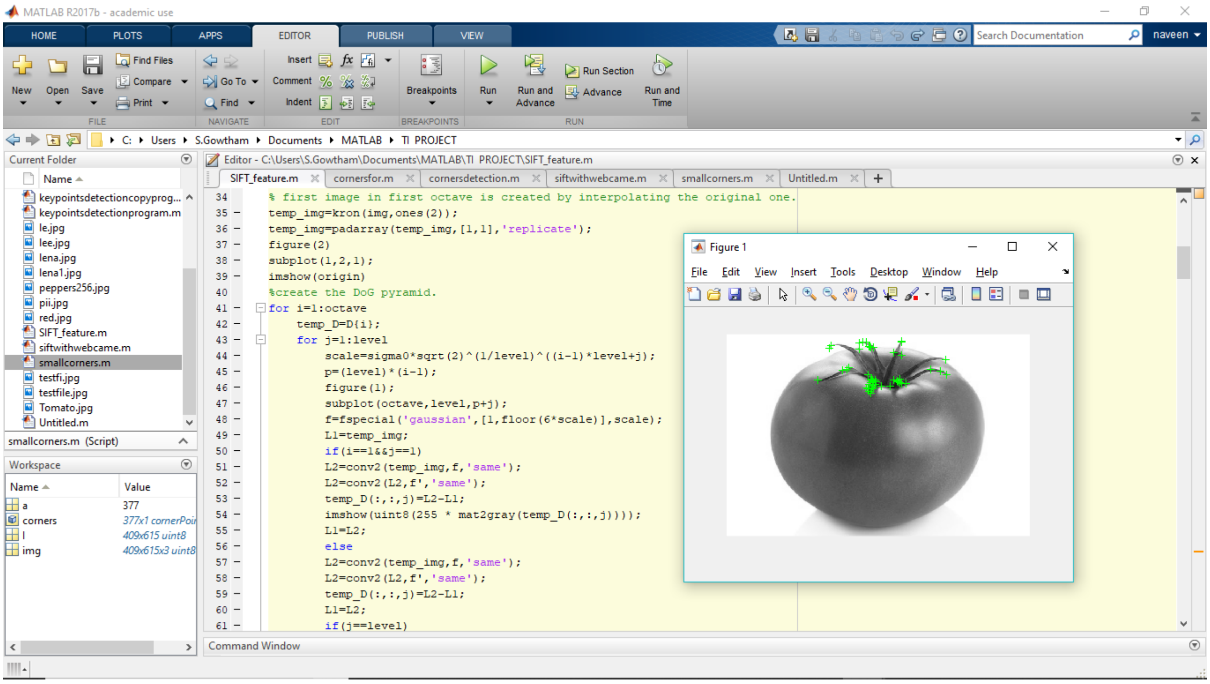


Fig 7.2 Result of good vegetable image.

As well, the system is able to recognize all the test vegetable images. The results of vegetables System on the vegetable images that are being sent in as input images during testing the system.

The images of vegetable are uploaded to the raspberry pi. Based on image processing coding the images are analysis the defected area. Detect the defected area in row and column of the pixel. After detecting any defected area, it will check the neighborhood pixel of row and column and then match the trained input. Finally, the defect will be identified.

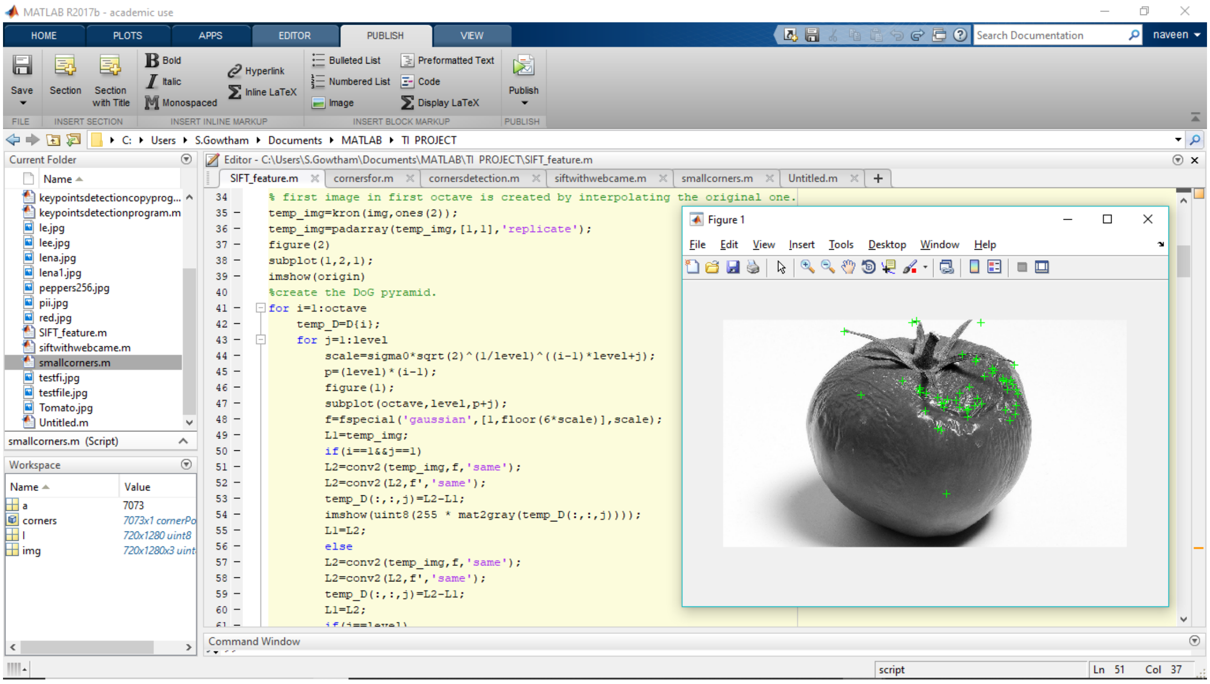


Fig 7.3 Result of spoiled vegetable image.

Separation is done based on the total number of keypoints. Rotten goods have keypoints more than the fresh goods because of the shrinkages and cracks on the surface. We provide a threshold value so that if the keypoint value is more than the threshold, the servo motor allows a 45-degree left deviation and vice-versa.

HARDWARE

Fig 7.4 Conveyor Setup

Conveyor setup is made up of a transparent belt carrying the tomatoes in a linear way by a gear system. When the tomatoes are made to run, it is stopped for a specific time and the camera captures the tomatoes. This image is processed via Matlab through Raspberrypi. Now after processing by necessary algorithms, a servo motor is used to separate the good and rotten tomatoes by a separator system.

CHAPTER 8

CONCLUSION

Most of the video analytical and computer vision applications, images play a fundamental role. It receives images as inputs and outputs may be either an image or some features of the image indexed as pixel values. A computationally efficient feature detector based on SIFT algorithm is implemented in this design. In this thesis work an FPGA based architecture for image feature detection in real time was designed. By this design a stable and reliable scale space extrema detection is achieved based on SIFT algorithm. This design is focused on the feature detection with invariance to image rotation, scaling, translation and illumination changes. The future work includes an embedded SoC architecture incorporating the SIFT detector and BRIEF feature descriptor and matching. In this an optimized feature descriptor invariant to scale and orientation is also planned to implement on a single FPGA chip with less resource utilization.

FUTURE WORKS

- Making the algorithms effective to obtain maximum accuracy.
- Fixing two cameras to capture the top and bottom side images.
- Acquiring more illumination to get the clear picture of the goods.
- Size based separation can be done further by image processing.

REFERENCES:

[1] FengyunWang, JiyeZheng, Xincheng Tian – “An automatic sorting system for fresh white button mushrooms based on image processing” in Elsevier Computers and Electronics in Agriculture, Volume 151, August 2018, Pages 416-425

[2] C. S. Nandi et al, B. Tudu and C. Koley, "A Machine Vision-Based Maturity Prediction System for Sorting of Harvested Mangoes," in *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, pp. 1722-1730, July 2014

[3]Chong-Yaw Wee, Raveendran Paramesran, Fumiaki Takeda, “Fast Computation of Zernike Moments for Rice Sorting System”, in Systems Engineering, Volume 61, pp. 165-168,2007.

[4]GhobadMoradi, MohammadH.Sedaghi, Mohammad R.Alsharif; Mousa Shamsi 2011 International Conference on Electronic Devices, system and applications(ICEDSA) “Vegetable Detection From color images using ACM and MFCM algorithms” year:2011.

[5]Qingsong Meng; ZhihuiLv 2015 Seventh International Conference on Advanced Communication and Networking(ACN) year:2015 “An Image Registration Method Based on Improved SIFT Algorithm”

[6] Ashwani Kumar; Yogesh, Dubey 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) year:2016 “Vegetable Defect Detection based on speeded up robust feature technique”.

[7] Hongshe Dang; Jinguo Song; Qin Guo 2010 Second International Conference on Intelligent Human-Machine System and Cybernetics “A Vegetable Size Detecting and Grading System Based on Image Processing; Year:2010.

[8] D. G. Lowe, “Distinctive image features from scale- invariant keypoints,” Int. J. Compute. Vision, vol. 60, no. 2, pp. 91–110, 2004.

[9] R. S. Lakshmi et al, “A Review on vegetable grading system for quality inspection,” International Journal of Computer Science and Mobile Computing, vol. 3, no.7, pp.615– 621, July 2014.

[10] Y.-C. Chen,F, -C. Huang, S.-Y. Huang and J.-W. Ker, “High performance SIFT hardware accelerator for real- time image feature extraction,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 3, pp. 340–351, Mar. 2012.

[11] M. R. Alsharif, G. Moradi, M. H. Sedaaghi and M. Shamsi, "Segmentation of pomegranate MR images using spatial fuzzy c-means (SFCM) algorithm, " presented at the International Conference on Graphic and Image Processing (ICGIP), 2010.

[12] H. Sardar, “Vegetable quality estimation by color for grading,” International Journal of Modelling and Optimization, vol. 4, no.1, pp. 38-42, Feb. 2014.

[13] Z. Cao, L. Kang, J. Wang, L. Yan and S. Zhong, “A real-time embedded architecture for SIFT,” J. Syst. Arch., vol. 59, no. 1, pp. 16–29, Jan. 2013.

[14] B. Jarimopas, Jaisin, N., 2008. An experimental machine vision system for sorting sweet tamarind. J. Food Eng. 89 (3), 291–297.

[15] Pourdarbani R, Ghassemzadeh, H.R., Seyedarabi, H., Nahandi, F.Z., Vahed, M.M., 2015,” Study on an automatic sorting system for date fruits”. J. Saudi Soc. Agric. Sci. 14 (1), 83–90