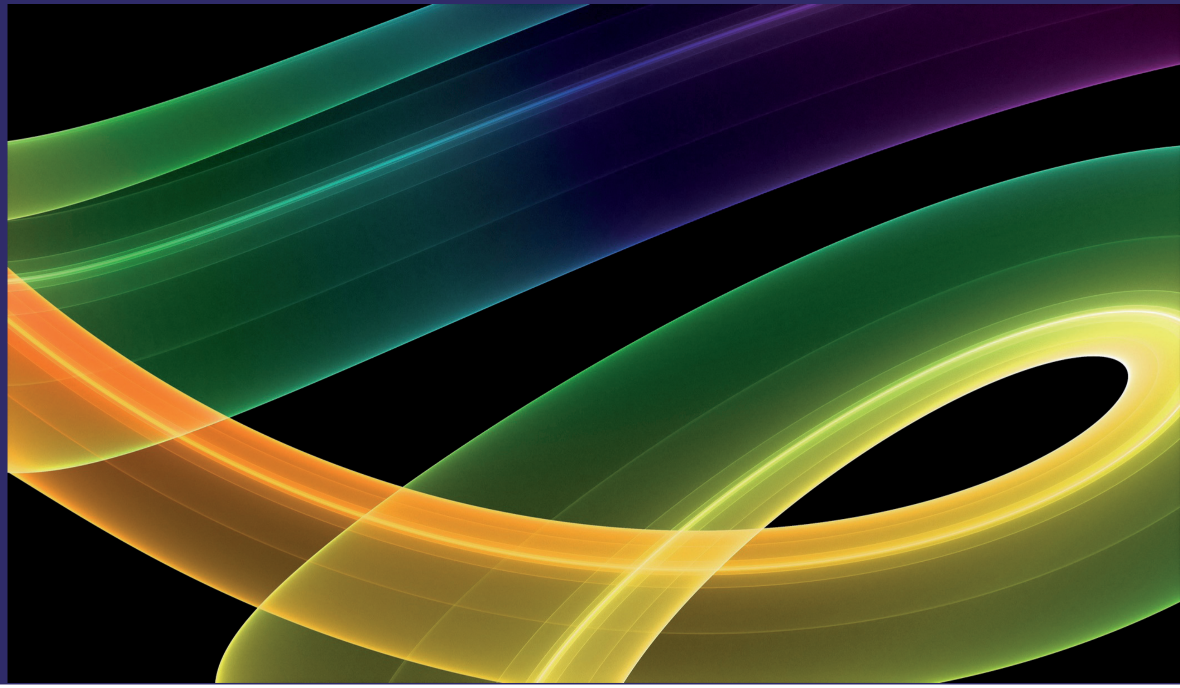
In the urban setting, high standards of living have encouraged automation to come right to the forefront and be an integral factor of any home design. Simultaneously, environmental concerns have ensured that energy efficient housing models and appliances are the preferred choice. We integrate these requirements into a smart and energy efficient model home which employs device control using a technique as intuitive as gestures. For decades now, remote controls have enabled us to dictate the working and functions of everyday appliances. However, consider a scenario where a person, Carol, is all tucked in her bed and realizes that she has forgotten to switch off the lights and fans of the living room or the heating system. In such a scenario, a centralized control unit located at a convenient location from where Carol could control all devices would be useful. This is where the need for centralized automation arises. Now imagine if Carol had to find the remote to control the centralized system, then the purpose of automation would be lost and hence, gestures being an intuitive means of expression are an effective way to control household appliances at reasonable efficiencies.

Aman Chadha

Aman Chadha is currently pursuing his graduate studies in Electrical and Computer Engineering from the University of Wisconsin-Madison, USA. Prior to this, he pursued his Bachelors in Electronics and Telecommunication Engineering from the University of Mumbai. His interests include Image Processing, Computer Vision and Processor Microarchitecture.

978-3-659-17615-9



Aman Chadha

Divyajyoti Rajdev

Shruti Nirantar

Smart and Energy-Efficient Home Automation using Gesture Recognition

Controlling home appliances and devices with the flick of your hand

LAP LAMBERT Academic Publishing

Aman Chadha
Divyajyoti Rajdev
Shruti Nirantar

Smart and Energy-Efficient Home Automation using Gesture Recognition

Aman Chadha
Divyajyoti Rajdev
Shruti Nirantar

Smart and Energy-Efficient Home Automation using Gesture Recognition

Controlling home appliances and devices with the flick of your hand

LAP LAMBERT Academic Publishing

Impressum/Imprint (nur für Deutschland/only for Germany)
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Coverbild: www.ingimage.com

Verlag: LAP LAMBERT Academic Publishing GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland
Telefon +49 681 3720-310, Telefax +49 681 3720-3109
Email: info@lap-publishing.com

Approved by: Mumbai, University of Mumbai, Diss., 2012

Herstellung in Deutschland (siehe letzte Seite)
ISBN: 978-3-659-17615-9

Imprint (only for USA, GB)
Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Cover image: www.ingimage.com

Publisher: LAP LAMBERT Academic Publishing GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Germany
Phone +49 681 3720-310, Fax +49 681 3720-3109
Email: info@lap-publishing.com

Printed in the U.S.A.
Printed in the U.K. by (see last page)
ISBN: 978-3-659-17615-9

*Dedicated to my parents**

*without whom all my aims, aspirations and dreams wouldn't exist

ACKNOWLEDGEMENTS

We, namely, Aman Chadha, Divyajyoti Rajdev and Shruti Nirantar, owe thanks to a great many people who helped and supported us during the conceptualization and implementation of the proposed work as well as during its documentation.

Our deepest thanks to Professor K.Y. Rajput, our guide and mentor, for leading us at every step and reviewing various documents of ours with utmost attention and care.

We express our thanks to the principal and staff of Thadomal Shahani Engineering College, especially our head of department, Dr. Uday Pandit Khot for providing us with the required infrastructure and facilities to accomplish the design and development of the proposed system.

Lastly, we would like to thank the non-teaching staff of the Electronic Devices and Circuit Lab and well as the B.E. Project Lab for their help and patience.

Without the support of the above mentioned, this work would have been a distant reality.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Symbol	**Explanation**
DCT	Discrete Cosine Transform
DST	Discrete Sine Transform
DWT	Discrete Wavelet Transform
HMM	Hidden Markov Models
ANN	Artificial Neural Networks
SIFT	Scale-Invariant Fourier Transform
PCA	Principle Component Analysis
EEG	Electroencephalography
MATLAB	Matrix Laboratory
MEX file	MATLAB Executable file
XML	eXtensible Markup Language
SQL	Structured Query Language
JPEG	Joint Photographic Experts Group
TIFF	Tagged Image File Format
EEPROM	Electrically Erasable Programmable Read-Only Memory
GUI	Graphics User Interface
UID	User Input Device
BW	Black and White
DDR	Data Direction Register
UART	Universal Asynchronous Receiver and Transmitter
UBRR	USART Baud Rate Register

1. INTRODUCTION

1.1 OBJECTIVE

Technology has advanced with leaps and bounds over the last decade thus giving rise to computerization of our everyday living environment. It is no longer sufficient for the technology to have only utility rather other factors such as smartness or intuitiveness, energy-efficiency and degree of automation need to be taken into account. As time progresses, the development of IT technology shall further add luxury to common devices that merely served utilitarian purposes in the past. Commonplace devices like the washing machine come with “sixth sense” that can inform the end user about what stage of operation the appliance is on. Further, as networking of the home appliances is being realized and they are become even more intelligent. Therefore, for such appliances frequently used in everyday life, intuitive operation is desirable for a user. This is where a non-contacting interface based on man's natural actions comes into play. Gestures, which we use frequently and intuitively in our everyday communication, are one of such man machine interfaces.

Our work proposes a functional automation system based on gesture recognition using the following:

- Video content analysis and multi-frame type of processing
- Image enrollment and keypoints-based matching
- Combination of classic and novel approach to Image processing
- Portable computational devices
- Communication with handheld hardware mainframe module

The vision behind our work was to put forth a practical, commercially feasible and easily implementable home automation system that puts the entire control in the hands of the end user, literally. The system can be better understood by judging the software and hardware modules on their individual merit and their performance on conjunction with each other.

1.2 PROBLEM STATEMENT AND FORMULATION

This work proposes the benefits of a smart house and the areas of usage of smart living systems. Details about the technical substructure and application of the designed home automation system through gesture recognition are described. This report presents the extension of existing vision based gesture recognition system using an algorithm that combines the classical and novel approach for gesture recognition. Under future work, several improvements and support structures may be implemented in order to increase the capabilities and the functionality of the system. These improvements include position independent recognition, rejection of unknown gestures, and continuous online recognition of spontaneous gestures. These gesture matches when obtained are used to control technologies deployed in an intelligent room. They may either be power or utility related devices and thus centralizing their control gives rise to a truly dynamic home automation system. Our aim is to pull the computer out into the real world of people rather than to pull people into the virtual world of the computer. The highest level application we aspire to achieve is to build application systems that provide occupants of the room with specialized services for command and control of ambient conditions. We have demonstrated two different gesture recognition systems paired with hardware, to drive devices. The first gesture recognition system is based on absolute hand gestures and the second one is based on a touchscreen approach of giving commands. The work further highlights the challenges faced, certain adjustments made and the features of the final product.

In the urban setting, high standards of living have encouraged automation to come right to the forefront and be an integral factor of any home design. At the same time, environmental concerns have ensured that energy efficient housing models and appliances be preferred for common home usage. We reckon that given the above concepts, it would make sense to integrate them and thereby give rise to a smart and energy efficient model home which employs device control using a technique as intuitive as gestures.

For decades now, remote controls have enabled us to dictate the working and immediate functions of everyday appliances like the AC, TV etc. However, consider a scenario where a person, Carol, is all tucked in her bed and she realizes that she has forgotten to switch off the lights and fans of the living room, or the heating system. In such a scenario and many more, it would indeed help to have a centralized control unit located at a convenient location from where Carol could command any and all of the electrical or electronic devices connected to the centralized controlling system. This is where the need for centralized automation comes into place. Now imagine if Carol needed to find the remote to control the centralized system that controls all the other devices, then the purpose of automation would be lost, and thus, it was concluded that gestures being an intuitive means of expression, are the best way to control household appliances at reasonably moderate efficiency levels.

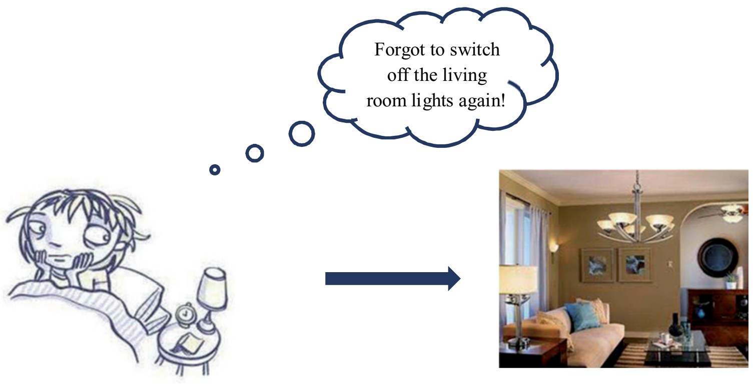


Fig. 1.1 Sample scenario depicting need for automation system

Most techniques systems currently employed for automation have the following drawbacks:

1. The one-time cost of installing the systems is ginormous
2. The maintenance costs further add on to overall expenses on devices

3. Due to extravagant prices such systems are affordable only to the upper class of society

4. Most automation systems cannot be ported or made compatible for all appliances of use in a house, thus very flimsy levels of automation are achieved

5. Human Computer Interface (HCI) based automated homes are designed for lavish spacious houses but are often found to work better for smaller spaces

The proposed work is targeted to reduce the implementation cost of a simple central automation system. Thus, the generic idea requires minimal hardware and any simple computational device like laptop, smart-phones etc. for its complete working and operation. At the same time, gesture recognition is a highly sought after and researched upon field. The report subsequently explains the algorithm and design considerations to realize the above defined target.

Thus, the generic idea requires:

Image Processing:

- Thresholding
- Sharpening & Smoothening
- Contrasting
- Selection
- Recognition Features
- Grayscale
- Feature Matching
- Feature enrolment

Hardware requirements:

- User input device
- Computational & Processing device
- Centralized control device or Driver Circuit

1.3 ORGANIZATION OF WORK

Chapter 1: Introduction

- Overview of the objective of the proposed system and outlines the functions to be performed by hardware and software modules

Chapter 2: Review of Literature

- Highlights reasons for the selection of specific software and tools.

Chapter 3: Mathematical Modeling, Analysis and Design

- Conceptualization based on problem formulation and the actual realization of the 2D and 3D gesture recognition schemes based on processing requirements. Implementation of the hardware module and interfacing with devices.

Chapter 4: Implementation

- The implementation of the entire system is explained as regards to an activity diagram of the same. We talk about the initial concepts, the first prototypes and the final product made. The testing of the proposed system is described with the help of a few test cases which prove the successful implementation of the system.

Chapter 5: Testing and Results

- We show just what happened when the proposed system was run and elaborate on the errors and limitations. Finally some areas of further work in the same field have been highlighted and conclusions have been drawn for the system's completion.

SUMMARY

- We have proposed a system to demonstrate an automation system based on gesture recognition using multi-frame processing and keypoint-based matching.
- The main aim behind the system was to put forth a practical, commercially feasible and easily implementable home automation system that puts the entire control in the hands of the end user.

Problem Statement

- Consider a scenario where a user, all tucked in their bed, realizes that they have forgotten to switch off the lights and fans of the living room, or the heating system.
- A centralized control unit located at a convenient location is hence sought from where a user could command any and all of the electrical or electronic devices connected to the mainframe.

Drawbacks of Current Techniques for Automation

- Enormous installation and maintenance cost.
- Affordable only to the upper class of society, due to it being expensive.
- Lack of portability limits the compatibility of the system and hence restricts its usage for all appliances.

2. REVIEW OF LITERATURE

2.1 WHAT ARE FEATURES?

A feature is defined as an "interesting" part of an image, and features are used as a starting point for many computer vision algorithms. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is repeatability: whether or not the same feature will be detected in two or more different images of the same scene. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

The following may be considered as feature points of an image:

- Edges and Corners
- Blobs within the same image
- Ridges or symmetry axis

2.2 FEATURE DETECTION & EXTRACTION

In order to successfully implement the software module, the techniques in use currently for image processing must be taken into account. This is because Video content analysis and Gesture recognition, when broken down to the very basics are nothing but Image processing. Even background dependent movement detection, employs simple difference vector principle to find relative displacement between two frames and calculate the motion trajectory.

2.2.1 TYPES OF FEATURE DETECTION ALGORITHMS

Discrete Cosine Transform: Discrete Cosine Transform (DCT) is a well-known signal analysis tool used in compression due to its compact representation power. DCT is a very useful tool for signal representation both in terms of information packing and in terms of computational complexity due to its data independent nature. DCT helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). DCT is conceptually similar to Discrete Fourier Transform (DFT), in the way that it transforms a signal or an image from the spatial domain to the frequency domain.

Discrete Wavelet Transform: Discrete Wavelet Transform (DWT) is a transform which provides the time-frequency representation. Often a particular spectral component occurring at any instant is of particular interest. In these cases it may be very beneficial to know the time intervals these particular spectral components occur. For example, in EEGs, the latency of an event-related potential is of particular interest. DWT is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal. In numerical analysis and functional analysis, DWT is any wavelet transform for which the wavelets are discretely sampled. In DWT, an image can be analyzed by passing it through an analysis filter bank followed by decimation operation. The analysis filter consists of a low pass and high pass filter at each decomposition stage. When the signal passes through filters, it splits into two bands. The low pass filter which corresponds to an averaging operation, extracts the coarse information of the signal. The high pass filter which corresponds to a differencing operation, extracts the detail information of the signal.

Discrete Sine Transform: Discrete Sine Transform (DST) is a Fourier-related transform similar to the Discrete Fourier Transform (DFT), but using a purely real matrix. It is equivalent to the imaginary parts of a DFT of roughly twice the length, operating on real data with odd symmetry (since the Fourier transform of a real and odd function is imaginary and odd), where in some variants the input and/or output data are shifted by half a sample.

Hidden Markov Model-Based: In the research area of dynamic gesture recognition, Hidden Markov Models are one of the mostly used methods. The movement of a person over a sequence of images is classified. The first approach for the recognition of human movements based on Hidden Markov Models has been described in a paper by Volder. It distinguishes between six different tennis strokes. This system divides the image into meshes and counts the number of pixels representing the person for each mesh. The numbers are composed to a feature vector that is converted into a discrete label by a vector quantizer. The labels are classified based on discrete HMMs. The system described in the paper by Rochelle is capable of recognizing 40 different connected person dependent gestures of the American Sign Language. This system uses colored gloves to track the hands of the user, but can also track the hands without the help of gloves. The position and orientation of the hands are used for the HMM based classification.

Neural Network-based: An Artificial Neural Network (ANN), usually called Neural Network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

2.2.2 SCALE INVARIANT FEATURE TRANSFORM

For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges.

Fig. 2.1 Study of feature points in the picture

Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

Fig. 2.2 Key point matching by SIFT algorithm

2.2.3 ADVANTAGES OF SIFT ALGORITHM

There has been an extensive study done on the performance evaluation of different local descriptors, including SIFT, using a range of detectors. The main results are summarized below:

- SIFT features exhibit the highest matching accuracies (recall rates) for an affine transformation of 50 degrees. After this transformation limit, results start to become unreliable.
- SIFT-based descriptors outperform other local descriptors on both textured and structured scenes, with the difference in performance larger on the textured scene.
- For scale changes in the range 2 to 2.5 and image rotations in the range 30 to 45 degrees, SIFT and SIFT-based descriptors again outperform other local descriptors with both textured and structured scene content.
- Introduction of blur affects all local descriptors, especially those based on edges, like shape context, because edges disappear in the case of a strong blur. But GLOH, PCA-SIFT and SIFT still performed better than the others. This is also true for evaluation in the case of illumination changes.

The evaluations carried out suggests strongly that SIFT-based descriptors, which are region-based, are the most robust and distinctive, and are therefore best suited for feature matching.

2.2.4 APPLICATIONS OF SIFT ALGORITHM

SIFT features can essentially be applied to any task that requires identification of matching locations between images. Work has been done on the following applications:

- Recognition of particular object categories in 2D images
- 3D reconstruction
- Motion tracking and Segmentation
- Robot localization
- Image panorama stitching
- Epipolar calibration

2.3 TOUCHSCREEN BASED APPROACH GESTURE RECOGNITION

A white screen is used as the touch plate to give an input in the form of a symbol. The user needs to create a database of the various symbols required. A camera placed below white screen catches the reflected shadow of the symbol created and interprets it as particular action that is required in home automation using micro-controllers. The following figure explains the 2-D gesture recognition concept:

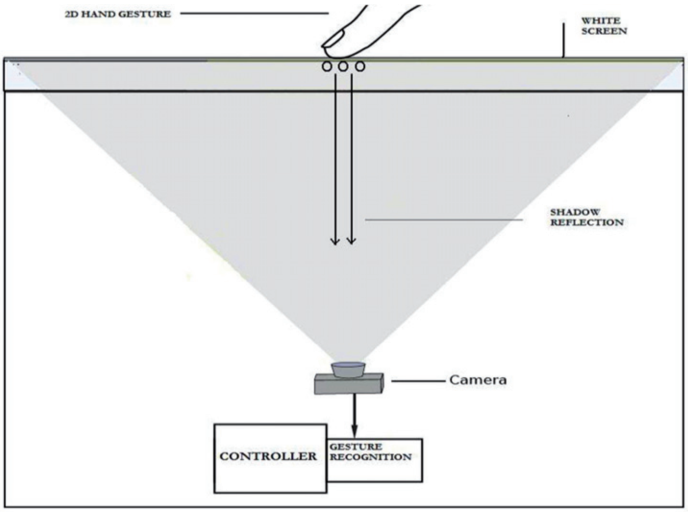


Fig. 2.3 Touchscreen-based gesture recognition block diagram

Apart from the above mentioned, there are techniques other techniques as well which are employed for gesture recognition. However, most of them are subject to conditions of surroundings. Thus for accurate and reliable results, a combination of the classic and novel approach with be used to finally implement gesture recognition. However, in this approach the biggest advantage it the economic and practical feasibility of the idea. The touchscreen can be readily built using the shadow arrangement and any standard smart phone may be used for the computational purpose.

SUMMARY

Proposed Home Automation Methods using Gesture Recognition

- 2-D gesture recognition
- 3-D gesture recognition

Feature Detection and Extraction

- Interesting points on the object in an image can be extracted to provide a feature description of the object.
- This is subsequently used for image matching
- To perform reliable recognition, it is important that the extracted features be detectable even after image scale, noise and illumination.
- Such points usually lie on high-contrast regions of the image, such as object edges.
- If the features extracted are carefully chosen then the input data can be replaced by this reduced representation.

Feature Detection Algorithms

- Discrete Cosine Transform
- Discrete Wavelet Transform
- Discrete Sine Transform
- Hidden Markov Model-Based
- Neural Network-Based

Scale Invariant Feature Transform (SIFT) Algorithm and its Advantages

- Highest matching accuracies are obtained within an affine transformation range of 50 degrees. Beyond this limit, results start to become unreliable.
- Better performance achieved by SIFT on textured and structured scenes.

- SIFT-based descriptors work best in comparison to other methods for scale changes in the range 2 to 2.5 and image rotations in the range 30 to 45 degrees.
- Upon introduction of blur and illumination, SIFT still outperforms other methods.
- Can be applied to any task that requires identification of matching locations between images. Generally deployed for use in recognition of particular object categories in 2D images, scene reconstruction etc.

Touch Screen-Based (2D) Gesture Recognition

- A white screen is used as a touch plate to give an input in the form of symbol.
- The camera place below the screen catches the reflected shadow of symbol.
- The symbol is then compared with the database created of various symbols.
- The accordingly the symbol is intercepted as the particular action that is required in home automation using microcontroller.

3. IDEA OF THE PROPOSED SOLUTION

3.1 SMART, ENERGY EFFICIENT AND AUTOMATION CONSIDERATIONS

The proposed system aims to integrate advanced Image Processing techniques with their plausible application to a real world scenario. In recent years the introduction of network enabled devices into the home environment has proceeded at an unprecedented rate. Moreover, there is the potential for the remote or centralized control and monitoring of such network enabled devices. Thus, the system undertakes the task of coupling common household devices with gesture recognition based decision unit. This will provide high levels of automation using next to none additional cost.

Elements of a Smart Home

Fig. 3.1 Smart home elements

A Smart home is usually centrally controls most of its devices. To realize such a case first consider some of the electronic and electric appliances which can be installed in a house or flat, such as light, heating, home entertainment, motorized window blinds, telephone system or security devices. Now imagine that a user of our system could control all of this functionality from her convenient spot at home i.e. her bed or even do it remotely while in the car. The concept of applications which facilitate the remote control of home appliances are the essence of smart homes. Apart from the above mentioned applications the above apparatus comprises of home control and automation hardware. In most cases these appear in the form of a small web server which is installed on the home server available for the particular system. The user can observe and control the installed devices via these web interfaces.

Elements of Energy Efficient Home

Often the brunt of the busy lifestyle in today's world is borne by energy consuming appliance. This may happen if the inhabitant forgets to switch off the light and fans of the entire house. Or if the refrigerator is left on normal mode even if the owners are out for the week end, etc. Such scenarios lead to needless energy wastage. To avoid this, a carefully connected system of relays may be implemented to centrally trip power supply to parts or whole of the house through a central gesture controlled server. Such an application scenario can be implemented based on profiles. For instance, imagine an "I am leaving now" gesture from the door which turns off all the lights and lowers the heating. Before going for dinner, the user could initiate a "coming back in 90 minutes" function which will switch heating back to comfort level, including the preparation of hot water necessary for a relaxing bath. Besides the obvious comfort enhancements, this promises significant energy savings by allowing centralized adjustment of resource intensive appliances.

Elements of Home Automation

Home automation can be achieved with embedded computing power and memory within dozens of pieces of domestic equipment, each of which can communicate with the user and with other equipment. The connected web of these devices forms a system that works as a smart home automation. Namely, it enables the user to control several home security and electrical devices by the concept of smart life system. This concept means the routines about house are realized automatically, the ideal comfort conditions, and probable malfunctions and danger warnings in a living area can be managed by the system. There are many classical definitions of home automation available in the literature and often home automation is described as the introduction of technology within the home to enhance the quality of life of its occupants, through the provision of different services such as tele-health, multimedia entertainment and energy conservation. The system is designed to achieve this automation through Gestures.

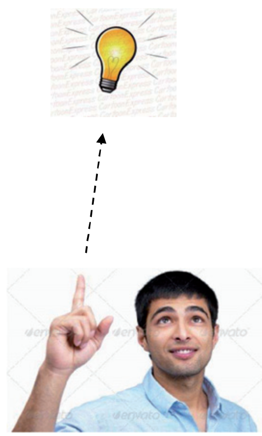
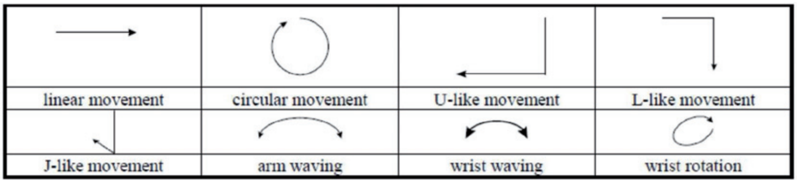


Fig. 3.2 Gesture controlled appliances

3.2 COMMONLY USED TECHNIQUES FOR GESTURE RECOGNITION

The actual gesture may comprise of several simple or complex motion trajectories. A combination of the simple wrist/hand movements gives rise to complex gestures. The following basic movements may be combined in any order to form a decipherable gesture:



The following are the sample 3D gesture recognition templates that give a quantitative idea of the function of each gesture:

				
Switch off	Switch on Light 1	Switch on Light 2	Switch on Light 3	Future Programming

Gesture 1 is used to switch off all the lights, while Gestures 2, 3 and 4 are used to switch on Light1, Light2 and Light 3. Gesture 5 is reserved for future programming for complicated functions like dimming the light, switching it on and off in succession etc.

			
Switch off	Switch on Fan 1	Switch on Fan 2	Future Programming

Gesture 1 is used to switch off all the fans, while Gestures 2 and 3 are used to switch on Fan1 and Fan2. Gesture 4 is reserved for future programming for complicated functions like adjusting the speed inclination (in case of a table fan) of the fan etc.

	
Gesture 1	Gesture 2
	
Gesture 1	Gesture 2

The above figures are general gestures which may be used to control auxiliary devices like water heaters, Refrigerators etc. Gesture 1 may be used switch off all the appliances while Gesture 2 may be used to switch on the appliance.

Gestures for the Touchscreen based Approach

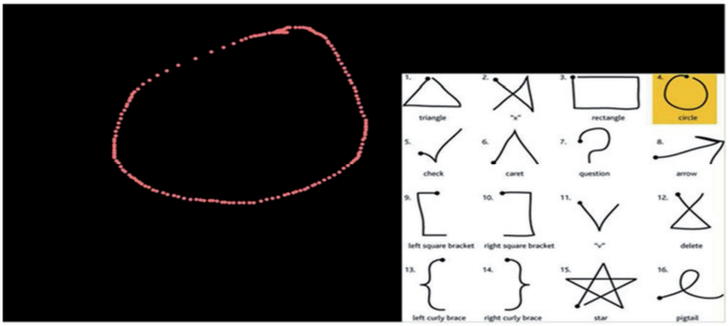


Fig. 3.3 Gesture templates

The above diagram shows templates for the gestures as registered on the screen surface. Here, a circle can be interpreted to switch on the fans. A square can be interpreted to turn off the fans. Similarly, other symbols can be used to operate on the other circuits in a home.

Also, symbols can be generated by just touching the finger on white screen. If one finger touch can be interpreted to switch on bulb, two fingers touch can be interpreted to switch on fan. Using the above concept 10 symbols can be created. The following figure gives the idea about this concept. For five fingers touch and four fingers touch.

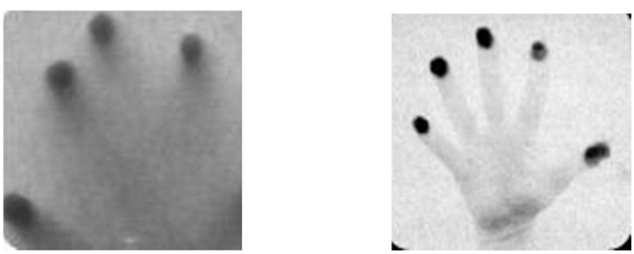


Fig. 3.4 Diagrams depicting the actual images generated on screen surface

SUMMARY

Proposed Solution

- We aim to integrate advanced Image Processing techniques with their plausible application to a real world scenario.

A "Smart" Home

- A smart home, essentially aimed at making residents' lives easier and more convenient, facilitates the remote control of home appliances. Central control of most of its devices is the essence of a smart home.

An Energy-Efficient Home

- This concept promises significant energy savings by allowing centralized adjustment of resource intensive appliances. This is achieved using a carefully connected system of relays that may be implemented to centrally trip power supply to parts or whole of the house through a central gesture controlled system.

Home Automation

- It can be achieved with embedded computing power and memory within dozens of pieces of domestic equipment, each of which can communicate with the user and with other equipment. The connected web of these devices forms a system that works as a smart home automation.

Techniques for Gesture Recognition

- A combination of the simple wrist/hand movements gives rise to complex gestures. Some gestures are assigned to switch on/off appliances while others are reserved for future programming.

Gestures for the Touchscreen based Approach

- In the touchscreen based approach for gestures, templates for the gestures are registered on the screen surface.

4. MATHEMATICAL MODELLING ANALYSIS AND DESIGN

4.1 SYSTEM OVERVIEW

The proposed system aims to integrate advanced Image Processing techniques with their plausible application to a real world scenario. In recent years the introduction of network enabled devices into the home environment has proceeded at an unprecedented rate. Moreover, there is the potential for the remote or centralized control and monitoring of such network enabled devices. Thus, the system undertakes the task of coupling common household devices with gesture recognition based decision unit. This will provide high levels of automation using next to none additional cost.

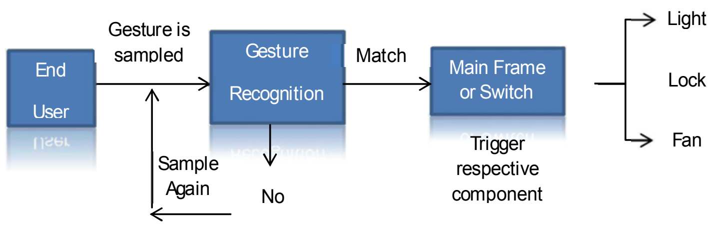


Fig. 4.1 Block diagram of the proposed system

The above figure shows the basic flow of control. It starts with image samples of the end user. These images may also be individual frames after splitting a video stream. This data is then sent to the software module where the matching takes place. The control signal generated by decision device is converted to suitable data packet format and sent to hardware module. Here, depending on the data packet received, high power appliances like refrigerators or low power appliances like CFLs may be switched on or off. Further, depending on desired complexity of the system, additional states like “dim the light”, “slow the fan” etc. may be added alongwith plain ON and OFF commands. The Hardware and software blocks are complimentary in function. The signals are generated on the software module and executed on the hardware module.

4.2 SOFTWARE SIDE IMPLEMENTATION

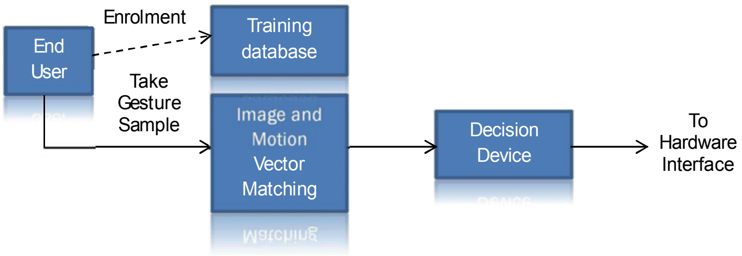


Fig. 4.2 Block diagram of the system's software end

The software side module breaks down the entire complicated recognition process into simple steps. The first of which is to enroll the user samples into the training database so that for all test gestures there exists a reference within the system. All test gestures will be matched with these training samples at a later stage and the output of this will be used to drive the decision device. The actual process of image matching starts with what is known as feature extraction; the feature vector may be in the form of Frequency domain components of the image, or difference image formed by subtracting two consecutive frames from each other or the motion trajectory of the entire gesture etc. Which method is used for feature extraction depends on the technique employed for image processing. Popular techniques and their corresponding software tools are discussed subsequently in the report.

4.3 HARDWARE SIDE IMPLEMENTATION

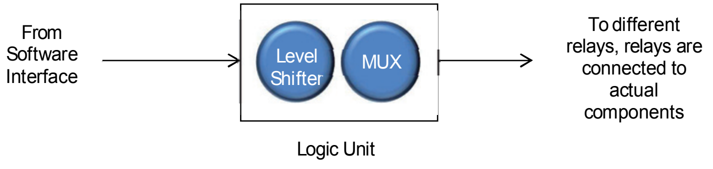


Fig. 4.3 Block diagram of the system's hardware end

The Hardware module accepts the control signal or data packet from the software module through an interface and drives the actual components to action. The data packet undergoes level shifter to make the output from computational device compatible with on board hardware i.e. controllers and drivers. Then it is fed to the multiplexer which selects the relay for which the data packet was intended. The bias of the relay switches the actual device circuit ON or OFF. The level shifter and multiplexer together are known as the logic unit of the Hardware as it is here that based on logic levels the operation is actually implemented. Once the above basic module is realized successfully it can be ported to a more convenient wearable unit.

4.4 REQUIREMENT ANALYSIS

The application requires certain hardware and software requirements to fulfill the needs of the system.
They are as follows:

4.4.1 HARDWARE REQUIREMENTS

- A processing unit with at least 10Kb programmable memory
- A camera of upto 2megapixels
- The appliances to be controlled

4.4.2 SOFTWARE REQUIREMENTS

- A computational device supporting MATLAB
- Compiler to convert MATLAB code to microcontroller code
- Interface between the camera, microcontroller and computational device

4.4.3 FUNCTIONAL REQUIREMENTS

- The primary function of the image processor is to be able to recognize the gestures made by the user under condition of changed lighting and orientation

- The function of the hardware module is to comprehend the incoming software command and accordingly switch on or off the device in question
- The gestures need to be specific but not user dependent such that the utility of the application is not hampered by incoherent gesturing

4.4.4 NON – FUNCTIONAL REQUIREMENTS

- **Portability:** The application code should be easily implementable on all necessary platforms, and the platforms on which it is expected to run.
- **Robustness:** The application should be able to handle error conditions gracefully, without critical failure.
- **Efficiency:** This is of utmost importance since the end-user must be satisfied by the editing work of the software.
- **Simplicity:** User-friendly environment will enable the user to navigate through the application with ease.

4.5 WHY MATLAB?

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces. However, in terms of globalization of code, MATLAB can call functions and subroutines written in the C programming language or Fortran. A wrapper function is created allowing MATLAB data types to be passed and returned. The dynamically loadable object files created by compiling such functions are termed "MEX-files" (for MATLAB executable).

Libraries written in 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. Microsoft Office Excel Worksheets, Textpad and Notepad documents can all be imported for manipulation in MATLAB.

Thus, broadly speaking, MATLAB code can be used in most other programming languages with little change. If the MATLAB library functions are predefined or explicitly user defined in C, C++, Java etc. then the program code can be ported to all these software easily. This is an important aspect to consider since it is desired that the application be implemented globally for all home automation systems.

Considering the current application, MATLAB provides an edge over other software because a serial port type object can be created in MATLAB which can support both synchronous and asynchronous type of communication through the Serial Com port. As well as this, MATLAB has excellent graphics capabilities, and its own powerful programming language. One of the reasons that MATLAB has become such an important tool is through the use of sets of MATLAB programs designed to support a particular task. These sets of programs are called toolboxes, and the particular toolbox of interest to us is the image processing toolbox. MATLAB supports a range of image formats including BMP, HDF, JPEG, PCX, TIFF, XWB etc.

4.6 ASPECTS OF IMAGE PROCESSING

It is convenient to subdivide different image processing algorithms into broad subclasses. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand.

Image enhancement:

This refers to processing an image so that the result is more suitable for a particular application.

Examples include:

- Sharpening or de-blurring an out of focus image
- Highlighting edges
- Improving image contrast, or brightening an image

- Removing noise

Image restoration:

This may be considered as reversing the damage done to an image by a known cause, For example:

- Removing of blur caused by linear motion
- Removal of optical distortions
- Removing periodic interference

Image segmentation:

This involves subdividing an image into constituent parts, or isolating certain aspects of an image:

- Finding lines, circles, or particular shapes in an image, in an aerial photograph
- Identifying cars, trees, buildings, or roads

These classes are not disjoint; a given algorithm may be used for both image enhancement or for image restoration. However, we should be able to decide what it is that we are trying to do with our image: simply make it look better (enhancement), or removing damage (restoration).

4.7 IMAGE PROCESSING USING MATLAB

MATLAB is a data analysis and visualization tool which has been designed with powerful support for matrices and matrix operations. As well as this, MATLAB has excellent graphics capabilities, and its own powerful programming language. One of the reasons that MATLAB has become such an important tool is through the use of sets of MATLAB programs designed to support a particular task. These sets of programs are called toolboxes, and the particular toolbox of interest to us is the image processing toolbox. MATLAB supports a range of image formats including BMP, HDF, JPEG, PCX, TIFF, XWB etc.

When you start up MATLAB, you have a blank window called the Command Window in which you enter commands. A command line style interface is used and the prompt consists of two right arrows:

>>

MATLAB supports several image types like Grayscale images, RGB images, L*a*b* space images etc.

Basic Image Commands in MATLAB

a. <variable name>=imread('filename'): Reads the image into a variable\
b. imshow(g): displays the matrix g as an image.

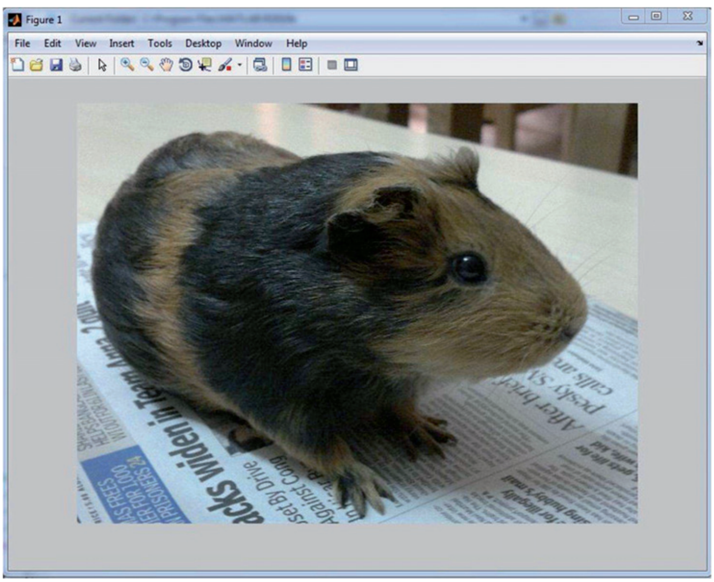


Fig. 4.4 MATLAB's imshow()

c. size(a): returns three values: the number of rows, columns, and _pages_ of a, which is a three dimensional matrix, also called a multidimensional array.

d. impixel(a,200,100): returns the red, green, and blue values of the pixel at column 200, row 100.

e. imfinfo('emu.tif'): Returns several information fields of the image

ans =

Filename: 'emu.tif'

FileModDate: '26-Nov-2002 14:23:01'

FileSize: 119804

Format: 'tif'

FormatVersion: []

Width: 331

Height: 384

BitDepth: 8

ColorType: 'indexed'

FormatSignature: [73 73 42 0]

ByteOrder: 'little-endian'

NewSubfileType: 0

BitsPerSample: 8

Compression: 'PackBits'

PhotometricInterpretation: 'RGB Palette'

StripOffsets: [16x1 double]

SamplesPerPixel: 1

RowsPerStrip: 24

StripByteCounts: [16x1 double]

XResolution: 72

YResolution: 72

ResolutionUnit: 'Inch'

Colormap: [256x3 double]

PlanarConfiguration: 'Chunky'

TileWidth: []

TileLength: []

TileOffsets: []

TileByteCounts: []

Orientation: 1

FillOrder: 1

GrayResponseUnit: 0.0100

MaxSampleValue: 255

MinSampleValue: 0

Thresholding: 1

Data Types and Conversions

Elements in MATLAB matrices may have a number of different numeric data types; the most common are listed as below.

Data type	Description	Range
int8	8-bit integer	-128 to 127
uint8	8-bit unsigned integer	0 to 255
intl6	16-bit integer	32768 to 32767
uintl6	16-bit unsigned integer	0 to 65535
double	Double precision real number	Machine specific

We can convert images from one image type to another. The table below lists all of MATLAB's functions for converting between different image types. Note that the gray2rgb function does not create a colour image, but an image all of whose pixel colors were the same as before. This is done by simply replicating the grey values of each pixel.

Image Type Conversions

Function	Use	Format
ind2gray	Indexed to Greyscale	y=ind2gray(x,map);
gray2ind	Greyscale to indexed	[y,map]=gray2ind(x);
rgb2gray	RGB to greyscale	y=rgb2gray(x);

gray2rgb	Greyscale to RGB	y=gray2rgb(x);
rgb2ind	RGB to indexed	[y,map]=rgb2ind;
ind2rgb	Indexed to RGB	y=ind2rgb(x,map);

It is important to make the distinction between the two functions double and im2double: double changes the data type but does not change the numeric values; im2double changes both the numeric data type and the values. The exception of course is if the original image is of type double, in which case im2double does nothing. Although the command double is not of much use for direct image display, it can be very useful for image arithmetic. We have seen examples of this above with scaling.

4.8 WHY ATMEGA?

ATmega16 is an Atmel product. It belongs to the megaAVR series and has 16Kb Flash memory. Like other AVR Microcontrollers they are ISP (in system programmable) i.e. they can be programmed without being taken out of the system. AVRs have been used in various automotive applications such as security, safety, powertrain and entertainment systems. With respect to the application ATmega16 is used as a control device to power on the appliance indicated by software code.

Features of ATmega16:

- 16 Kb In-System Self-Programmable Flash
- 1024 Bytes EEPROM
- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes. One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes
- Four PWM Channels
- Programmable Serial USART
- Internal Calibrated RC Oscillator

- Power-on Reset and Programmable Brown-out Detection
- 32 Programmable I/O Lines

The most important aspect of ATmega16 in question is the serial port. In computing, a serial port is a serial communication physical interface through which information transfers in or out one bit at a time. Throughout most of the history of personal computers, data transfer was accomplished through serial ports which connected the computer to devices such as terminals and various other peripherals. The program uses MATLAB to initiate, initialize and control block data transfer with the microcontroller through the serial port interface.

SUMMARY

Block Diagram Overview

- The input is given in the form of gestures by the user.
- It is compared with the database of gestures to produce a match.
- Then, as per the recognized gesture, the appropriate action on the home appliance is performed.

Non – Functional Requirements

- Portability
- Robustness
- Efficiency
- Simplicity

Use of MATLAB

- MATLAB can call functions and subroutines written in the C programming language or Fortran.
- Libraries written in Java, ActiveX or .NET can be directly called from MATLAB.
- Considering the current application, MATLAB provides an edge over other software because a serial port type object can be created in MATLAB which can support both synchronous and asynchronous type of communication through the serial communication port.

Aspects of Image Processing

- Image enhancement
- Image restoration

- Image segmentation

Use of ATMEGA16

- Due to features such as 16 Kb flash memory and the microcontroller being ISP (in system programmable), this particular microcontroller serves our purpose best.

5. IMPLEMENTATION

5.1 CONCEPTUALIZATION

We decided to implement the SIFT algorithm for feature detection and a Euclidean distance calculation algorithm for matching. However, both the above methods have their own substeps which need to be taken into account while writing the program. Thus starting from the SIFT algorithm the factor that garnered maximum emphasis is that SIFT image features provide a set of features of an object that are not affected by many of the complications experienced in other methods, such as object scaling and rotation.

While allowing for an object to be recognized in a larger image SIFT image features also allow for objects in multiple images of the same location, taken from different positions within the environment, to be recognized. SIFT features are also very resilient to the effects of "noise" in the image. All these play a vital role during real time implementation and testing of the application.

The features taken through the SIFT algorithm constitute the white points in the image i.e. the most distinctive points which will stand out or stay invariant in the image even after a heavy amount of blurring and difference calculation for distance is carried. These white points may be edges or distinct regions within the same image etc. The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors" (From "Object Recognition from Local Scale-Invariant Features", David G. Lowe). Each of these feature vectors is invariant to any scaling, rotation or translation of the image. This approach shares many features with neuron responses in primate vision. To aid the extraction of these features the SIFT algorithm applies a 4 stage filtering approach:

1. Scale-Space Extrema Detection

This stage of the filtering attempts to identify those locations and scales which are identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function. Further it has been shown under reasonable assumptions it must be based on the Gaussian function. The scale space is defined by the function:

$$L(x,y,\ \sigma)\ =\ G(x,y,\ \sigma)\ *\ I(x,y)$$

Where, * is the convolution operator,

G(x, y, σ) is a variable-scale Gaussian and

I(x, y) is the input image.

Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, D(x, y, σ) by computing the difference between two images, one with scale k times the other. D(x, y, σ) is then given by:

$$D(x,y,\ \sigma)\ =\ L(x,y,\ k\sigma)\ -\ L(x,y,\ \sigma)$$

To detect the local maxima and minima of D(x, y, σ) each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

2. Keypoint Localization

This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localized on an edge. This is achieved by calculating the Laplacian value for each keypoint found in stage 1. The location of extremum z is given by:

$$z\ =\ -\frac{\partial^2 D}{\partial x^2}^{-1}\frac{\partial D}{\partial x}$$

If the function value at z is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localisation it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the defference of Gaussian function.

If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the keypoint, the keypoint is rejected.

3. Orientation Assignment

This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:

- Use the keypoints scale to select the Gaussian smoothed image L, from above.
- Compute gradient magnitude, m.

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

- Compute orientation, θ.

  $$\theta(x,y) = \tan^{-1}\left((L(x,y+1) - L(x,y-1)) / (L(x+1,y) - L(x-1,y))\right)$$

- 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.

4. Keypoint Descriptor

The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centered on the keypoint.

Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This results in a feature vector containing 128 elements.

These resulting vectors are known as SIFT keys and are used in a nearest-neigbors approach to identify possible objects in an image. Collections of keys that agree on a possible model are identified, when 3 or more keys agree on the model parameters this model is evident in the image with high probability. Due to the large number of SIFT keys in an image of an object, typically a 500x500 pixel image will generate in the region of 2000 features, substantial levels of occlusion are possible while the image is still recognized by this technique.

The actual matching algorithm, as we said earlier uses the distance calculator as per the flow diagram given below.

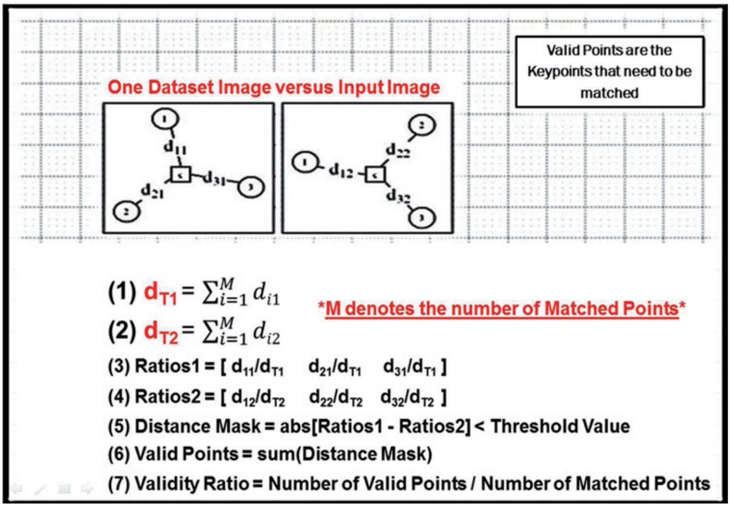


Fig. 5.1 Obtaining the distance ratio and validity ratio

Initially the SIFT threshold is taken as 0.65 and the matching MK-RoD algorithm is taken as 0.035. As the number of matches increase, the validity ratio increases thus we reduce the MK-RoD threshold by 0.005 and the SIFT algorithm by 0.05 and check again for matching points. This way we are precisely tuning the algorithm to obtain the optimum match from the database. This process continues till only one result is obtained after tuning and that will be the gesture which the user has displayed. That result is sent to the microcontroller.

System Overview

As discussed earlier, the system entails two software modules. The first is 3D gesture recognition and the second is 2D Gesture Recognition. And the Algorithm for both is distinct and needs to be discussed as such. The result obtained is then sent to the hardware module where it is used to turn on/off the appliances.

5.2 SEQUENCE FLOW FOR 2D GESTURE RECOGNITION

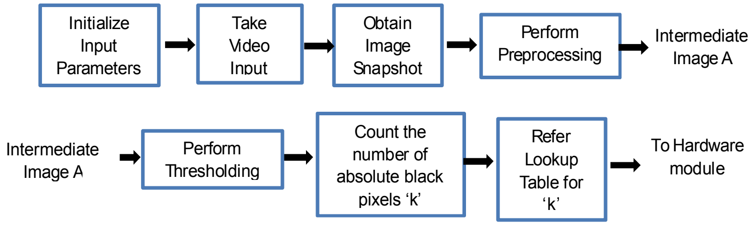


Fig. 5.2 Flow diagram for 2D gesture recognition

2D Gesture Recognition functions on the concept of defining a point threshold for the shadow image obtained on the touchscreen. However, first the test image must be obtained real-time. This is accomplished by taking a video feed input and pause it after a few seconds have lapsed. This ensures that the user does not move their hand and the gesture remains without any blurs. This paused video is then converted to a snapshot which gives us the test image. The preprocessing Techniques involve converting the image from color to gray level and cropping the image to focus only on the area of the screen when the gesture is positioned. After this preprocessing is performed we are left with the intermediate image. Let us call it, A. Thresholding is performed on this image A in order to convert grayscale image to purely black and white. Once the BW image is obtained, the number of absolute black pixels is counted using repeated iterations for position. This value 'k' is compared with the standard look up table to define which gesture was displayed by user. For Demonstration purposes, Gestures different by means of no. of fingers displayed have been considered.

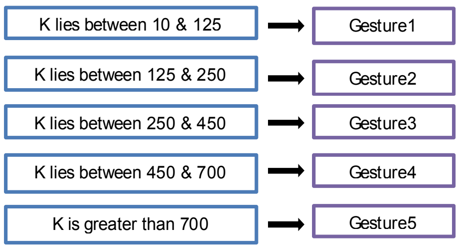
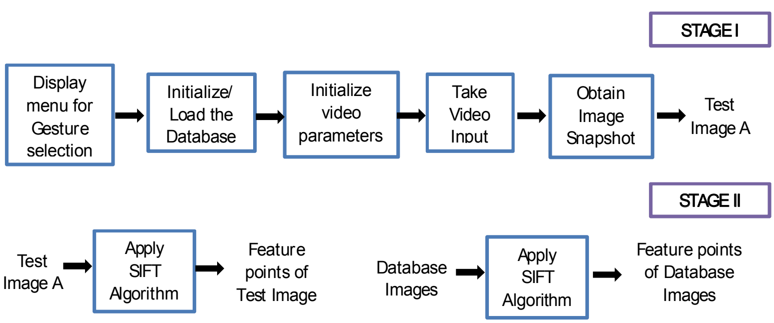


Fig. 5.3 Displaying of gesture on the basis of value of k

Once the gesture has been identified the respective command is issued to the microcontroller which in turn controls the switching on and off of the attached appliance.

5.3 SEQUENCE FLOW FOR 3D GESTURE RECOGNITION



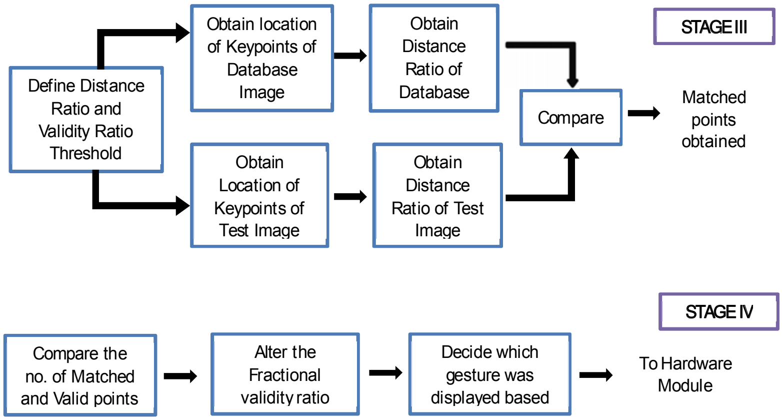


Fig. 5.4 Flow diagram for 3D gesture recognition

The sequence flow for 3D gesture recognition is broken down into four stages and must be discussed as such.

Stage I: Initialization and Image Acquisition

Here, the test image is obtained from the user. The system is initially assumed to be at idle or OFF stage. Thereafter, once the system is activated, the database containing sample image for the user to choose his gesture from is initialized or is loaded if it already is initialized on the mainframe. This database is then displayed as a menu for the convenience of the end user. The user is asked to input the required gesture real time and a video feed is taken. After giving it a few seconds for the user to form the gesture and hold it stable, a snapshot from the video screen is taken and this is treated as the test image for the entire program flow hence forth.

Stage II: Obtaining Key Points

As discussed earlier, the SIFT algorithm is used to obtain keypoints i.e. interesting points used for subsequent matching from the test images as well as database images. This is accomplished by running the SIFT algorithm on both sets of images. SIFT transform basically applies a Gaussian kind of blurring filter on the image with changing magnitude of blur. These images are grouped in octaves or user defined sets. For the sake of the program, sets of 4 are taken. Later on when comparative matching is to be carried out, these groups of strings of keypoints contain the information on the basis of which matching is carried out. Thus, the sets of Keypoints for the test image as well as database images are stored for future reference.

Stage III and Stage IV: Obtaining Matched Points

SIFT keypoints of objects are first extracted from a set of reference images using the 4step filtering discussed in Conceptualization. These are stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features (taken as 4 for the sake of the application in question) that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

The matched points thus pointed distinctly point towards the gesture made by the end user and thus can be used as a command and sent to the controller to manage the appliances connected to it.

5.4 SEQUENCE FLOW FOR HARDWARE IMPLEMENTATION

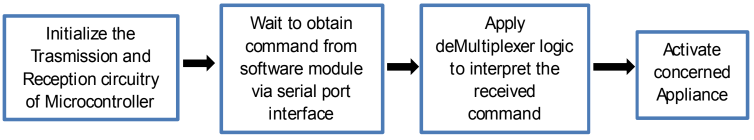


Fig. 5.5 Flow diagram for hardware module

We have seen earlier that MATLAB supports communication with a serial port by enabling the creation of a serial object type. Thus, the result of the image processing is then sent to the hardware interface which is a USB to serial converter connected to the serial port at the microcontroller. Once the device is activated, it is prepared for receiving data by waiting or 'listening in' to the input line. Here asynchronous type of communication is used. After receiving the command byte microcontroller converts it to executable action and performs corresponding device manipulation.

The following components within the microcontroller are used:

1. DDRX (Data Direction Register)

First of all we need to set whether we want a pin to act as output or input. DDRX register sets this. Every bit corresponds to one pin of PORTX. Let us have a look on DDRA register.

Bit	7	6	5	4	3	2	1
PIN	PA7	PA6	PA5	PA4	PA3	PA2	PA1

Table 5.1 Depicting bits in the DDRA register

Now to make a pin act as I/O we set its corresponding bit in its DDR register.

- To make Input set bit 0
- To make Output set bit 1

If we write **DDRA = 0xFF** (0x for Hexadecimal number system) that is setting all the bits of DDRA to be 1, will make all the pins of PORTA as Output

2. USCRB & USCRC

The Atmel AVR ATMega16 has a single Universal Asynchronous Receiver and Transmitter (UART) which can be used to connect the AVR to a device which has a serial port, typically a PC or a modem. The ATMega16 UART can have 8 or 9 data bits and supports baud rates of 2400 to 115200. The UART is initialized and used by accessing four specialised IO registers dedicated to the UART. They are the UART I/O Data Register (UDR), the UART Status Control Register A (USCRA), the UART Status Control Register B (USCRB), the UART Status Control Register C (USCRC), and the two UART BAUD Rate Register (UBRRH and UBRRL). Note that the UBRRH and USCRC share the same memory location

To use the UART to receive and transmit data, it is first necessary to program the UBRRL/H and the USCRB/C registers to set the UART up with the correct number of data bits, to enable the receiver, to enable the transmitter and to choose the baud rate. The UART can either be interrupt driven or polled. Here it is only shown how to use polled I/O. To poll the UART to check whether a character has been received, the USCRA register has to be checked to see whether the RXC bit is set. To poll the UART to see whether to see whether it is possible to send a character, the UDRE bit has to be set by the UART hardware.

3. UBRR & UBRL

The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable pre-scaler or baud rate generator. The down-counter, running at system clock is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= f_{osc}/(UBRR+1)). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the receiver's clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSEL, U2X and DDR_XCK bits.

SUMMARY

Implementation using SIFT

SIFT algorithm makes use of the following steps to extract useful features:

- **Scale-Space Extrema Detection:** This stage of the filtering attempts to identify those locations and scales which are identifiable from different views of the same object.
- **Keypoint Localization:** This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localized on an edge. This is achieved by calculating the Laplacian value for each keypoint found in stage 1
- **Orientation Assignment:** This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation.
- **Keypoint Descriptor:** The data used above, is also used to create keypoint descriptors. This data is then used to create a set of histograms over a window centered on the keypoint.
- Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This results in a feature vector containing 128 elements.
- These resulting vectors are known as SIFT keys and are used in a nearest-neighbours approach to identify possible objects in an image. Collections of keys that agree on a possible model are identified, when 3 or more keys agree on the model parameters this model is evident in the image with high probability.

Distance Ratio and Validity Ratio

- Initially the ratio distance is taken as 0.65 and the threshold is taken as 0.035.
- With an increase in the number of matches, the validity ratio increases thus we reduce the threshold by 0.005 and the increase ratio distance by 0.05 and check again for matching points.

- This way we are precisely tuning the algorithm to obtain the optimum match from the database.
- This process continues till only one result is obtained after tuning and that will be the gesture which the user has displayed. That result is sent to the microcontroller.

Software Module for 3D Gesture Recognition

- The samples are enrolled in a training database for future reference.
- All test gestures are matched the training samples.
- Feature extraction is used to detect and isolate the desired gesture.
- According to the gesture, the input is fed to the hardware module.

Sequence for 2D Gesture Recognition

- The test image is obtained real-time.
- The image is obtained by taking video feed input and then taking a snapshot after a pause of few seconds.
- The test image is YUV2 image which is then converted into grayscale image.
- Image is then cropped to focus only on the area of the screen when gesture is positioned.
- Thresholding is performed on the image to convert it into purely black and white image.
- Once the binary image is obtained number of absolute black pixels is calculated.
- This value 'k' is compared with the standard look up table to define the gesture Accuracy.

Hardware Implementation

- We have seen earlier that MATLAB supports communication with a serial port by enabling the creation of a serial object type.
- The result of the image processing is then sent to the hardware interface via the serial port.

- After receiving the command microcontroller converts it to executable action and performs corresponding device manipulation.

- **DDRX (Data Direction Register)**
  - Used to set whether we want a pin to act as output or input. Every bit corresponds to one pin of PORTX.
  - Now to make a pin act as I/O we set its corresponding bit in its DDR register.
    - To make Input set bit 0
    - To make Output set bit 1
  - If we write DDRA = 0xFF (0x for Hexadecimal number system) that is setting all the bits of DDRA to be 1, will make all the pins of PORTA as Output.

- **UART Status Control Register X (USCRX) and USART Baud Rate Register (UBRR)**
  - Powered by a Universal Asynchronous Receiver and Transmitter (UART), used to connect the AVR to a device which has a serial port, typically a PC or a modem.
    - Baud Rates: 2400 to 115200
  - To use the UART to receive and transmit data, it is first necessary to program the UBRRL/H and the USCRB/C registers to set the UART up with the correct number of data bits, to enable the receiver, to enable the transmitter and to choose the baud rate.

6. TESTING AND RESULTS

The tests are essentially walkthroughs for the entire application. We worked on it and used it as a normal user would and found instances which gave us errors and those which did not. The ones that did were corrected. However, it also helped augur certain problems which would not be that easily fixed and in some cases, required the dropping of crucial elements.

6.1 2D GESTURE RECOGNITION TEST

Fig. 6.1 The user input device

The procedure adopted here was to place the fingers on the shadow screen of the UID shown above and process the screen image for absolute black points which corresponded to the Gesture displayed. Figure below shows the test images.

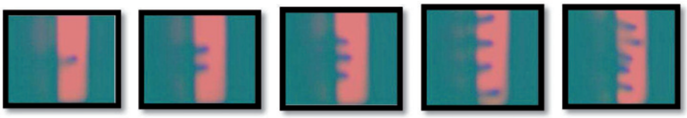


Fig. 6.2 The YUV2 Screen image obtained for 2D gesture recognition

The above images undergo preprocessing in the form of conversion to gray scale and region cropping and the following are set of images are obtained.

Fig. 6.3 The grayscale preprocessed image

The above intermediate images are further processed and converted to binary images and it is here that the actual algorithm to detect gesture kicks in. From the following pictures, the number of absolute black points is found and the accuracy of Gesture Recognition measured.

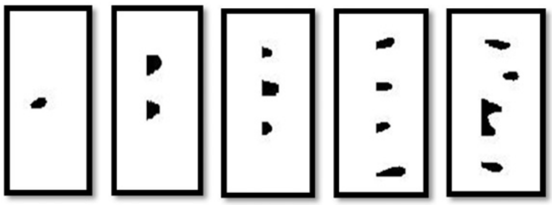


Fig. 6.4 The binary image showing finger imprints

Observation Tables

Results for one, two, three, four and five finger 2D gestures are as below:

Value of K	Gesture recognition
75	Successful
113	Successful
57	Successful
64	Successful
25	Successful

Table 6.1 Results for gesture 1 (one finger 2D gesture)

Value of K	Gesture recognition
221	Successful
155	Successful
140	Successful
124	Failed
158	Successful

Table 6.2 Results for gesture 2 (two fingers 2D gesture)

Value of K	Gesture recognition
360	Successful
290	Successful
268	Successful
433	Successful
482	Failed

Table 6.3 Results for gesture 3 (three fingers 2D gesture)

Value of K	Gesture recognition
521	Successful
198	Failed
652	Successful
796	Failed
473	Successful

Table 6.4 Results for gesture 4 (four fingers 2D gesture)

Value of K	Gesture recognition
1231	Successful
1486	Successful
885	Successful
773	Successful
621	Failed

Table 6.5 Results for gesture 5 (five fingers 2D gesture)

6.2 ACCURACY CALCULATION FOR 2D GESTURE RECOGNITION

$$\text{Accuracy} = \frac{\text{Total Number of Gestures recognized correctly}}{\text{Total number of Gestures}}$$

$$= (20/25)*100$$

$$= 80\%$$

6.3 3D GESTURE RECOGNITION TEST

After the software module has been initialized the video feed is taken which provides snapshots for the sample test images. However the database has to be built beforehand and figure below shows the sample entries of the image database with which all comparisons shall be made.

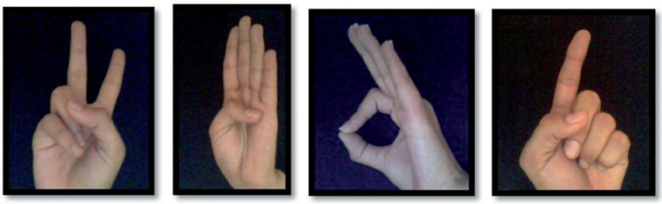


Fig. 6.5 Database images used as reference gestures

Each of these gestures represents a distinct application and as can be seen from the images, the change in skin tone, lighting, rotation and orientation do not significantly impact the results. Infact the images have been auto resized for representational purposes only and in actual experimentation the scale of the image i.e. the distance from the camera factor did not impact the accuracy of the feature detection, extraction and matching algorithms. The above is one of the most important aspect accomplished during implementation of the proposed system.

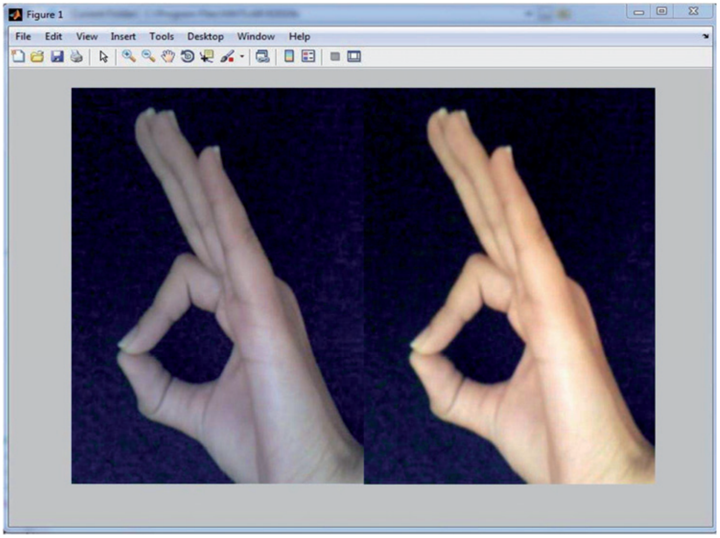


Fig. 6.6 Query image and matched database image

Above, the output of image matching in MATLAB has been shown, the gesture has been correctly identified as the image to the left is the query image or the test image and image to the right is the corresponding database image matched to it. Thus, 3D gesture recognition was accomplished successfully.

SUMMARY

3D Gesture Recognition Test

- Each of these gestures represents a distinct application and as can be seen from the images, the change in skin tone, lighting, rotation and orientation do not significantly impact the results.
- In fact the images have been auto resized for representational purposes only and in actual experimentation the scale of the image i.e. the distance from the camera factor did not impact the accuracy of the feature detection, extraction and matching algorithms.
- The above is one of the most important aspect accomplished during implementation of the proposed system.

Results

- Accuracy of touch screen based 2D gesture recognition is around 80%.
- It is economic and touchscreen can be easily built using shadow arrangement.
- It gives the better result for image of extremely low quality provided the threshold is decided judiciously.
- It is much faster hence can be implemented on small scale control.
- But the biggest disadvantage here is, it is affected by minor changes in lighting and thus 3D recognition has an edge in this area of comparison.

7. CONCLUSION AND FUTURE SCOPE

The aim of the proposed system was to accomplish Smart and Energy Efficient Home Automation by means of designing and building a man-machine interface using a video input to interpret the various gestures. The system is designed to be cost efficient as well and another advantage is that the user not only can communicate from a distance, but need have no physical contact with the computer. A visual system was chosen instead of an audio based system as the latter would fail during real time implementation in noisy environments or in situations where sound would cause a disturbance.

The above implementation was achieved by employing two distinct systems for image processing of the gesture i.e. 3D gestures and 2D touchscreen based gesture recognition. The commands obtained once the gesture is successfully recognized were used to drive a microcontroller which in turn switched the device on or off. At any point if an error occurs, the system can cause the wrong appliance to be worked upon. However, this is not considered as a case of critical failure as this does not cause damage under ordinary circumstances to both the program logic as well as appliance in consideration. Yet, as part of further work on the system we propose that a manual override be added in order to incorporate appliances which may cause a situation of critical failure.

A comparative analysis of 2D and 3D gesture recognition brought us to the following observations:

- 3D gesture recognition based on SIFT feature extraction algorithm is relatively unaffected by minor change in lighting, orientation and scaling thus making it an ideal choice for real time gesture recognition systems
- 2D gesture recognition quantitatively gives better accuracy of upto 80%
- the gesture set i.e. image database for gestures is significantly limited for 2D gesture recognition schemes and thus 3D recognition has an edge in this area of comparison

- 2D gesture recognition performs better when the video input device provides images of extremely low quality, provided that the gray threshold is decided judiciously
- 2D gesture recognition is much faster and thus may be implemented on a small scale to control devices. For instance, it could successfully replace electrical switched with 4-5 switches
- Both recognition schemes can be ported to control all kinds of appliances on a minimal budget or capital investment

Thus, we conclude our work by giving a brief of the Smart, Energy-Efficient and Automation aspects of the system.

Smart: A Smart home is usually centrally controls most of its devices. A user of our system can control all of this functionality from her convenient spot at. This, concept of applications which facilitate the remote control of home appliances are the essence of Smart Homes.

Energy Efficient: Often the brunt of the busy lifestyle in today's world is borne by energy consuming appliance. This may happen if the inhabitant in a hurry and cannot switch off the light and fans of the entire house. Such scenarios lead to needless energy wastage. To avoid this, a carefully connected system of relays may be implemented as part of future work to centrally trip power supply to parts or whole of the house through a central gesture controlled server. The code for the above functionality would not differ from the code implemented as part of the proposed system.

Automation: Home automation can be achieved with embedded computing power and memory within dozens of pieces of domestic equipment, each of which can communicate with the user and with other equipment. The connected web of these devices forms a system that works as a smart home automation. This concept has been demonstrated by the LEDs used to represent different devices.

Hence, it can be rightly said that the proposed system has great significance with respect to technological advancement in a moderate budget developing country like India.

SUMMARY

Limitations

- At any point if an error occurs, the system can cause the wrong appliance to be worked upon.
- However, this is not considered as a case of critical failure as this does not cause damage under ordinary circumstances to both the program logic as well as appliance in consideration.
- Accuracy and time response of the system can be improved.
- System can be made independent of background.

Conclusion

Comparative analysis of 2D and 3D gesture recognition:

- **Processing Time:** Processing time of 3D system is more than 2D system.
- **Effect of atmospheric changes:** 3D test is unaffected by minor changes in lighting, orientation and scaling.
- **Quality of image:** 2D gives better results with the image of extremely low quality.
- **Accuracy:** 2D systems gives accuracy around 80%, 3D system has better accuracy rates in comparison.
- During implementation, manual control must be incorporated to support appliances which may cause critical situations.

REFERENCES

[1] D. G. Lowe, **“**Distinctive image features from scale-invariant keypoints,**”** *International Journal of Computer Vision,* Vol. 60, No. 2, 2004, pp. 91-110

[2] K. Irie, K. Umeda and N. Wakamura, “Construction of an Intelligent Room Based on Gesture Recognition,” *2004 IEEE International Conference on Intelligent Robots and Systems*, Sandai, Japan, Oct 2004, pp. 193-198

[3] D. G. Lowe, "Local feature view clustering for 3D object recognition," *IEEE Conference on Computer Vision and Pattern Recognition,* Kauai, Hawaii, Dec. 2001, pp. 682-688

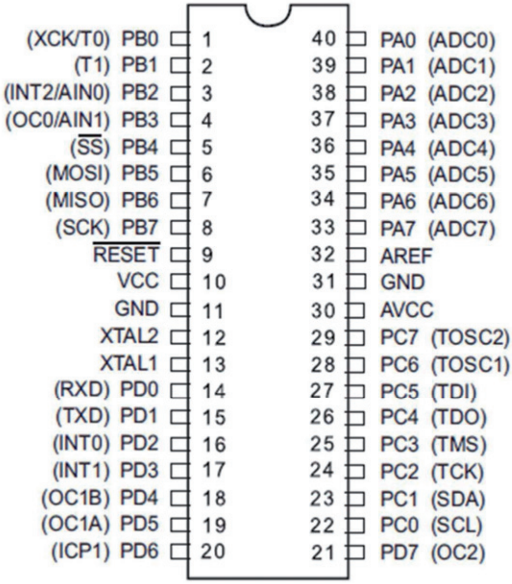
[4] A. R. Al-Ali and M. Al-Rousan, "Java-based home automation system", *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 2, 2004, pp. 498-504

[5] B. Bauer and H. Hienz, “Relevant feature for video-based continuous sign language recognition” Department of Technical Computer Science, Aachen University of Technology, Aachen, Germany, 2000.

[6] J. Davis and M. Shah, “Visual gesture recognition”, *Proceedings of IEE Vision, Image and Signal Processing*, 1994, Vol. 141, No. 2, pp. 101-106

APPENDIX

PIN Diagram of ATmega16




MoreBooks!
publishing
mb!
yes
i want morebooks!