Discrete & Continuous Mouse Motion Using Vocal & Non Vocal Characteristics of Human Voice.

Vipul Sharma
School of Computer Science & Engineering
Bahra University, Shimla Hills, India
vipul01sharma@gmail.com

Pratibha Sharma
School of Computer Science & Engineering
Bahra University, Shimla Hills, India
pratibhasharma80@gmail.com

Abstract- Mouse control today has become an important tool to interact with computers. It is quite easy for those who are physically fit and mentally sound, but for those who are suffering from physical disabilities, It is quite difficult to use mouse for interacting with computers. Keeping this thing in mind, we in this research paper present a system known as "speech based mouse". This speech based mouse will allow its users to control mouse pointer both in a discrete and continuous fashion. The application that we have developed will take input both in the form of words and non-vocal sounds. The algorithms that we have used to implement this system are MFCC technique for feature extraction and vector quantization with K-means clustering for speech recognition. Further we have proposed a new technique of phonetic feature extraction to cause continuous mouse motion possible.

Keywords – Training phase; Testing phase; Speech based mouse; Vector quantization; MFCC.

I. INTRODUCTION.

In recent years, there has been a lot of advancement in speech recognition technology, but still this field possesses a vast potential. This is because of the fact that human voice largely remains unexplored and unexploited. Voice based devices find their applications in day-to-day life, and have huge potential benefits especially for those people who are suffering from physical impairments and have some kind of disabilities.[5] Gone are the days, when it was considered that those who have such problems cannot make an efficient use of computers and are surpassed to show their creativity and talent, because they were unable to use traditional input devices like keyboard and mouse. In the United States alone, there are nearly 700,000 people suffering from disabilities of the spinal cord and 70% of them are unemployed.[5]

As, they have some kind of physical disabilities, so making an efficient use of computers is just a dream for them. In the nut shell we can say that such people are restricted to show their hidden talents and creativity. The only option they are available with is to utilize their voice as a tool to make this interaction with computers possible.[5]

It is not only the case with the people who are physically disabled but also for those who are absolutely fine physically but sometimes find themselves in impairing situations. For example "Say a pilot is flying a plane and suddenly a prompt message appears on his wall-sized display that requires his immediate response". So, in such kind of impairing situations voice based control is much better than traditional input devices which otherwise are manual in operation.

So, it has now become immensely important to enhance the quality of voice-based interactions owing to the above said reasons. The theoretical foundation of the "speech based mouse" is the "voice recognition technology" that is having two main phases/stages.

First one is the "Training phase". In this we record the speech signals, convert them into a suitable form that, can further be used for extraction of the features. The extracted features are then stored in a kind of repository that would be referenced later.

The second stage is called the "Testing phase". In this stage, features of the sound wave to be tested are extracted using the same method used earlier in the training session. The extracted features of the sound to be tested are then compared with the features stored in the repository to recognize the sounds exactly. Different algorithms can be used for extraction and recognition purposes. Here in this research paper, we would be using MFCC algorithm for feature extraction and Vector Quantization for comparison using Euclidean distance criterion.[3][4]

The most prominent limitation of existing voice based systems is that they take inputs in the form of words given as "commands" only. Owing to this, such systems cannot be used by people having vocal impairments. Further this would cause a discrete type of mouse cursor movement, because words are processed at the word level only. So, keeping this thing in mind we, in this paper have developed two different interfaces, one which takes words as input and causes a discrete type of cursor movement in four cardinal directions. And other which instead of words, take vowel sound utterances of upto 6 seconds as input to cause continuous mouse cursor movement in four cardinal directions depending upon the length of the input signal uttered.

Thus we have successfully overcome the above said limitation by developing a speech- based mouse that efficiently works for both vocal and non vocal speech signals, hence making both discrete and continuous mouse pointer movement possible. The application that we have developed is build in Matlab. Further we would be proposing a new technique to cause continuous mouse motion using non vocal characteristics or vowel sounds.

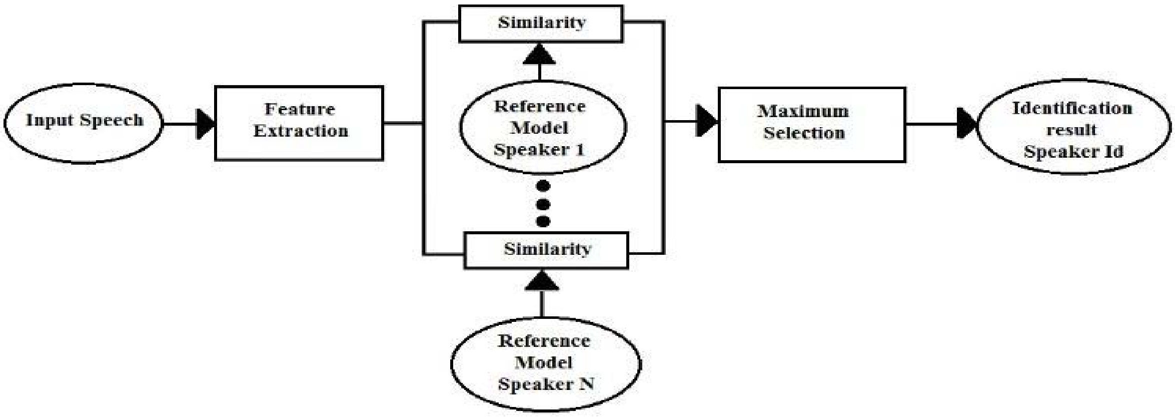

Input Speech
Feature Extraction
Similarity
Reference Model Speaker 1
Similarity
Reference Model Speaker N
Maximum Selection
Identification result Speaker Id

Fig. 1: General voice recognition technique.

The rest of the paper is organized as follows: section II will describe the objectives of this research along with advantages of the speech based mouse. Section III covers the underlying methodology along with an explanation of MFCC feature extraction and vector quantization algorithms. Section IV covers the implementation details of the discrete mouse motion and continuous mouse motion using MFCC & vector quantization approaches. Results along with discussions are discussed in section V. Conclusion is derived in section VI. And finally we would be presenting the future scope in section VII.

II. THE SPEECH BASED MOUSE.

We in this research paper will describe the work that has been done as part of speech-based mouse project which causes both discrete and continuous mouse cursor movement using vocal & non vocal characteristics of human voice respectively.

A. Objectives Framed for Speech Based Mouse.

The primary objective of the speech based mouse is to cause both discrete and continuous mouse cursor movement possible.

The second objective is to make interfaces particularly for people who have some physical disabilities, so that they can interact with the computer system easily.[1]

The third objective is to define a robust and new technique for continuously controlling the mouse pointer by varying the length of speech signal using MFCC & vector quantization approaches. This new approach would be explained later with details.

B. Functionality of Speech Based Mouse.

As we have already mentioned that the speech based mouse we have designed will work for both vocal and non vocal characteristics of human voice and would be able to track the speech features including volume, pitch and accent in real time environment [2]. The application we have created in matlab will enable its users to control mouse cursor movements by spoken words given as commands or by uttering different vowel sounds particularly for moving mouse pointer in four cardinal directions.

In case of discrete motion control interface, we have used four words i.e. 'Left', 'Right', 'Up' & 'Down' to control mouse pointer movement in four cardinal directions respectively. The features corresponding to these four words are extracted and stored in a kind of repository during the training phase. Now, for causing the motion possible, user needs to speak up a word. The underlying voice recognition algorithm finds the best possible match and further invokes motion control function that causes the mouse to move some pixel positions in the desired direction. Here, since the signals are processed at the word level, the motion is discrete in nature.

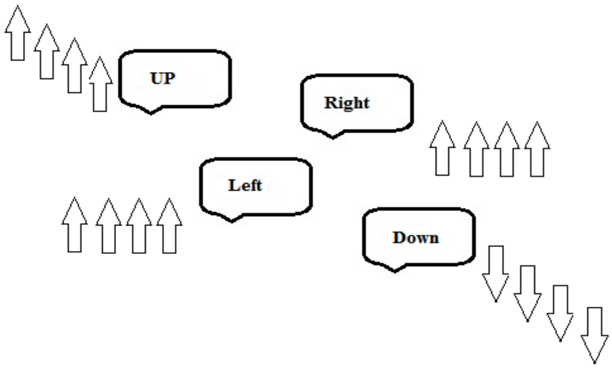


Fig. 2: Mouse cursor movement using English words

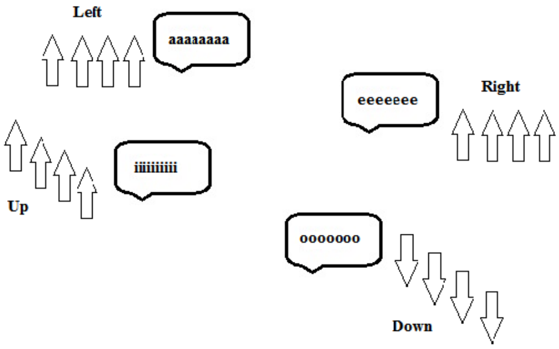


Fig. 3: Mouse cursor movement using non vocal speech sounds.

Now considering the case of continuous mouse motion, we are using vocal sounds instead of words, in which different vowel sounds are mapped to different cardinal directions as given below: -

aaaaaaaaa -------------------------> Left.

eeeeeeeee -------------------------> Right.

iiiiiiiiiiiiiii -------------------------> Up.

ooooooooo -------------------------> Down.

This continuous motion technique is explained later. User here, can control the mouse pointer movement and speed by making alterations in vowel sounds and volume.

C. Advantages of Speech Based Mouse.

- Speech based mouse by-passes the use of traditional input devices like keyboard and mouse.
- It can provide numerous benefits to people particularly suffering from physical disabilities while interacting with computers.
- It does not require any additional resources. The only thing that is requires is a good quality microphone.
- This technology working behind speech based mouse can further be extended to control electronic appliances.
- Speech based mouse can be used in real world environment for developing new applications and appliances with thrilling and extraordinary features.

So, these are some of the potential benefits of speech based mouse.

III. SPEECH BASED MOUSE UNDERLYING METHODOLOGY.

The underlying methodology working behind speech based mouse system actually comprises of different modules. The output of one module is acting as an input for the next module. Each module is using different techniques and algorithms to complete its operation. For example: - for feature extraction MFCC algorithm is used, these features are stored in a kind of repository that is later referenced during the pattern recognition phase which uses vector quantization approach based on Euclidean distance criterion.

The end result would be a combined effort of all the different modules. In our case the end result is the corresponding mouse motion

We have divided our system into three modules, these are as: -

- Feature extraction using MFCC.
- Pattern recognition using vector quantization.
- Mouse cursor motion control.

Figure depicting the overall flowchart of various modules of the proposed system.

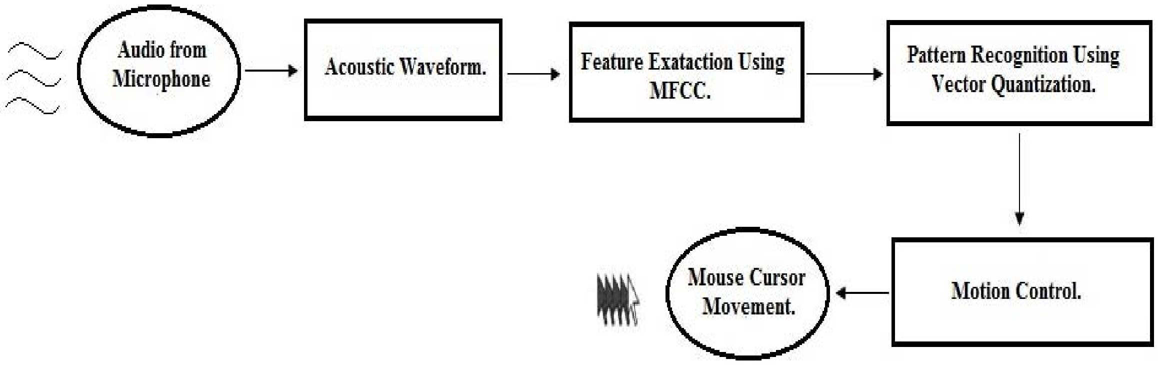

Audio from Microphone
Acoustic Waveform.
Feature Exataction Using MFCC.
Pattern Recognition Using Vector Quantization.
Motion Control.
Mouse Cursor Movement.

Fig. 4: Flowchart showing working of different modules of speech based mouse.

A. Speech Processing.

Feature Extraction using MFCC: - This module is also known as acoustic signal processing. The primary objective of this module is to extract the acoustic features using MFCC.

This phase is very much important, because the efficiency of the next phases directly depend on this. MFCC algorithm is based on the perception of human hearing which cannot perceive frequencies over 1KHz. In other words, MFCC is based on known variation of the human ear’s critical bandwidth with frequency.[3],[4]

The overall process of MFCC is as given below: -

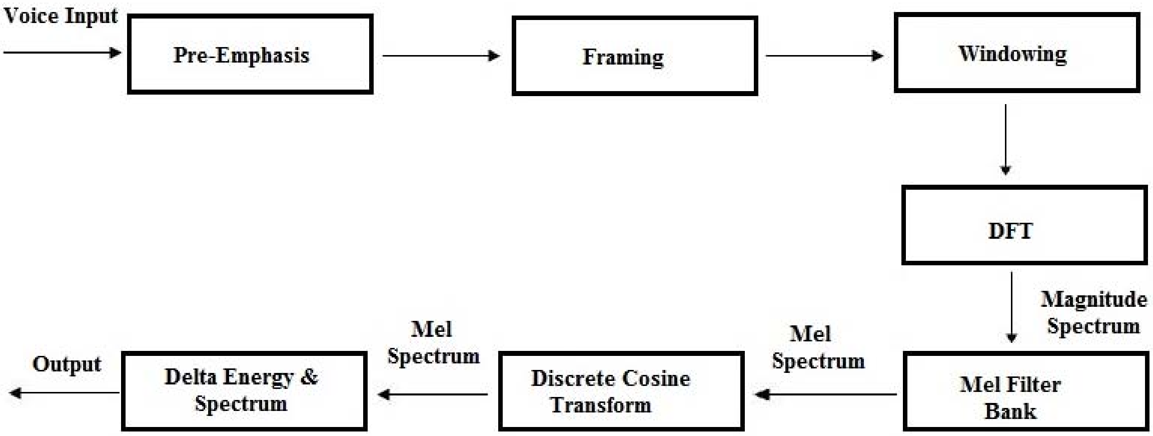


Fig. 5: MFCC block diagram.[3][4]

There are seven different computational steps in MFCC these are as explained below: -

Step 1: - Pre-emphasis: In this step the speech signal is passed through a filter which emphasizes higher frequencies. [6][3][4]

$$Y[n] = X[n] – 0.95\ X[n-1] \qquad (1)$$

Here a = 0.95, which means that the new sample is originated from the previous one by a ratio of 95%. Pre-emphasis is performed to flatten the spectrum of speech signal.

Step 2: - Framing: In framing the speech signal is usually broken down into small duration blocks known as frames. Adjacent frames are being separated by M (M<N) where N is the frame of samples. After that spectral and cepstral analysis is then performed on these frames.[6][3][4]

Step 3: - Windowing: Due to framing process, some discontinuity arises in the signal. So, in order to reduce that discontinuity each of the above frames is then multiplied with a window function. [6][3][4]

Y(n) = X(n) * W(n) (2)

where W(n) is the window function.

Step 4: - Fast Fourier Transform: Fast fourier transform is performed to convert th N samples obtained above from time domain into frequency domain. [6][3][4]

Y(w) = FFT [H(t) * X(t)] (3)

= H(w) * X(w)

Where X(w) and H(w) are fourier transforms of X(t) and H(t) respectively.

Step 5: - Mel-filter bank processing: In this step, the spectrum of speech signals obtained after FFT is then filtered by a group of triangular bandpass filters which simulates human ear's characteristics. [6][3][4]

Bank of filters is as shown below: -

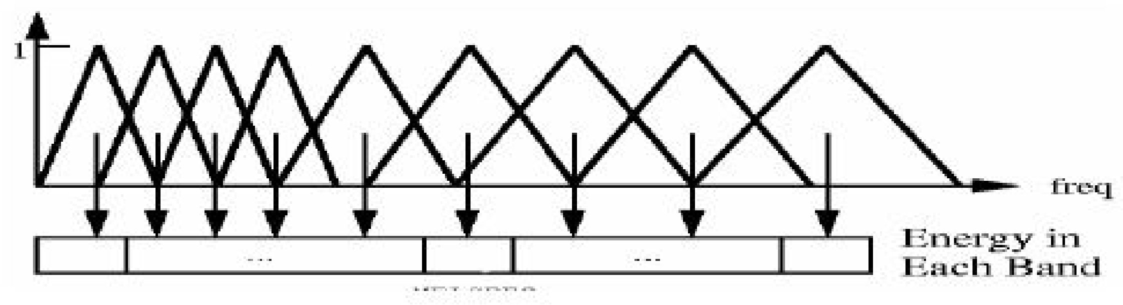


Fig. 6: Mel scale filter bank, from (young et al,1997) [6]

After this, the following equation is used to compute the Mel for given frequency f in Hz: -

F(Mel) = [2595 * log10 [1+f] 700] (4)

Step 6: - Discrete Cosine Transform: The log Mel spectrum that we obtained above is converted into time domain using discrete cosine transform. The result that we get after this transformation is called Mel-Frequency Cepstrum Coefficient.[6][3][4]

Step 7: - Delta Energy and Delta Spectrum: Frames and voice signal changes, such as the slope of the formant at its transition. There we need to add certain features that relate change in cepstral features over time. This is done using delta energy and delta spectrum.[6][3][4]

The MFCC features we obtain after this step are stored in a form of repository which is later referred for pattern recognition.

B. Pattern Recognition Using Vector Quantization Approach.

Vector Quantization: - For pattern recognition, we are using vector quantization method with K-means clustering technique. It is defined as the process of mapping vectors present in a large space of vectors to a limited number of regions in that very space. Each region obtained above is known as a cluster and each cluster is identified by its center which is known as its centroid. These centroids collectively contribute to the formation of a codebook.[8][9]

As, we are making clusters from large a very large vector space, the resulting amount of data obtained is significantly low because the number of centroids is at least 10 times smaller than the original vector space. This will further reduce the amount of computations needed for comparison and pattern matching.

Generation of codebooks: - Codebooks can be generated by using different techniques and algorithms. Since, we need to enhance the recognition accuracy of our system. We have been looking for the best algorithm and technique. One such technique is the LBG algorithm. It is also known as binary split algorithm.[8],[9],[10],[11]

Following recursive technique is used to implement it. The steps are as illustrated below: -

1. The first step is to design a single-vector codebook. This initial codebook is the center of all the vectors used for the training purpose.
2. In second step, we need to double the size of the codebooks by splitting each codebook Yn according to the following rule.

$$Y_n^{+} = Y_n (1 + E). \quad (5)$$

$$Y_n^{-} = Y_n (1 – E). \quad (6)$$

Where n varies from 1 to current codebook size and e is known as the splitting factor. For our system e =.0.02

3. Next step is the nearest neighbour search, which is implemented using K-means clustering algorithm. Here we need to find the centroid present in the current codebook and need to assign that particular vector to the corresponding cell.
4. Next step is to update the centroids. In this step we need to update the centroid of each cell using the centroid of the trained vectors we have assigned to that cell.
5. Repeat steps 3 & 4 until the average distance falls below a present threshold.
6. Repeat steps 2,3 & 4 until we reach a codebook of size 'M'.

Pattern matching: - In this step we need to extract the features of the unknown word, that has to be recognized. The extracted features is then represented using a sequence of feature vector X, {x1,x2,x3,………………xn}.

The sequence of feature vector extracted above is then compared with all the stored codewords in the codebook and the codeword that has the minimum feature distance is selected as the recognized word.

Minimum distance is calculated using the Euclidean Distance Technique.

$$D = (\sum (xi – yj)^2)^{1/2} \quad (7)$$

C. Mouse Cursor Motion Control.

This is the final module of our system. This module receives the information from the second module regarding the recognized word or sound pattern (in case of vowel sounds for causing continuous motion.). The module then invokes the underlying motion control function to cause corresponding mouse cursor movement on the screen. For example if the word or sound recognized is left or 'aaaaaaa' respectively, then action would be corresponding mouse cursor movement in left direction. For implementing mouse movement we have used java.robot package.

IV. IMPLEMENTATION.

The proposed systems for causing discrete & continuous mouse cursor movements are implemented in MATLAB programming language. Here we have designed user interfaces for both the above said systems separately. To work with speech recognition you need not to buy any bulky software, what you need is just a good quality microphone and voice recognition matlab routines.

A. Implementation of Discrete Mouse Cursor Motion.

Algorithm for causing discrete mouse motion using vector quantization approach: -

The process begins with the training of four words left, right up & down for causing mouse motion in four cardinal directions.

The steps of the algorithm are as illustrated below.

Start.

1. Invoke the recording function to record 1 second sound samples of word left, right, up & down.
2. The second step is to remove all background noise from the recorded word samples.
3. Invoke feature extraction module and extract acoustic features of each recorded word using MFCC technique. Store the extracted features in a repository for future references.

   Training process is now over.

   Testing process is as.
4. Again invoke the recording function, to record 1 second sound sample.
5. Remove the background noise and invoke the feature extraction module to extract the features of the word to be tested using MFCC technique.
6. Now invoke pattern matching module to recognize the word with minimum feature distance, this is performed using vector quantization approach.
7. The last step is to invoke motion control to cause corresponding mouse cursor movement on the screen.

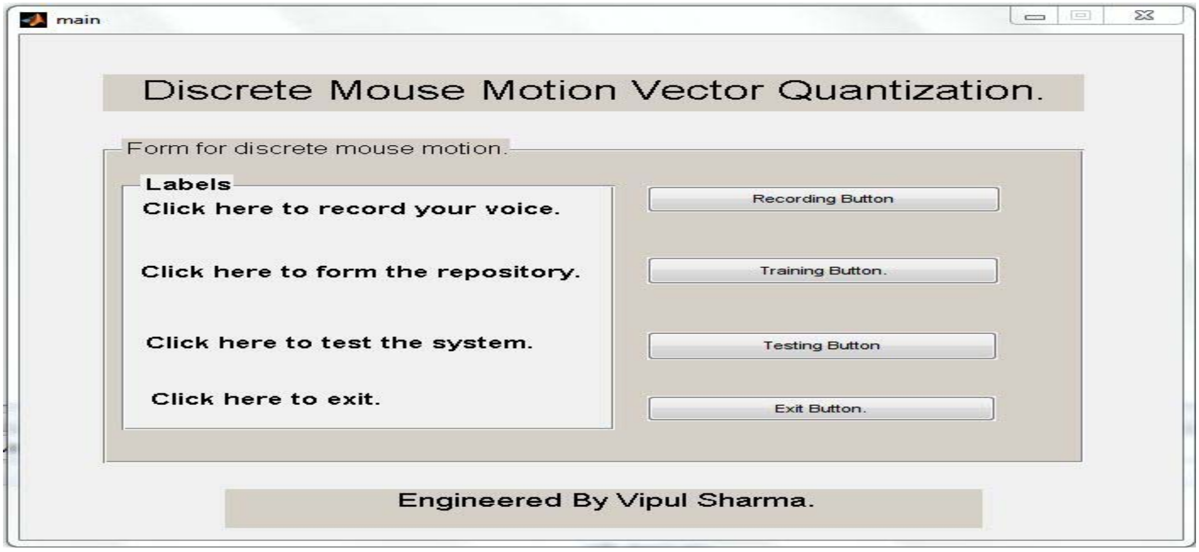
End of algorithm.

B. Details of the System for Discrete Mouse Motion Using Vector Quantization.

The main layout of our system comprises of four basic modules these are, the recoding module, the training which involves recording sound samples corresponding to four words. The second phase is the training module where the system extracts 20 acoustic features using MFCC technique corresponding to each word sample and stores them in a kind of repository that is referenced later during the testing phase.

The third phase is the testing module. Here user needs to record the word sample he wants, the system to recognize. The system after recording will extract the features corresponding to the recorded word and invokes the recognition function to recognize the word. After the recognition process is over, the system automatically invokes the motion module, that takes corresponding action.

C. Screen Shots of the System for Discrete Mouse Motion.



main

Discrete Mouse Motion Vector Quantization.

Form for discrete mouse motion.

Labels

Click here to record your voice.

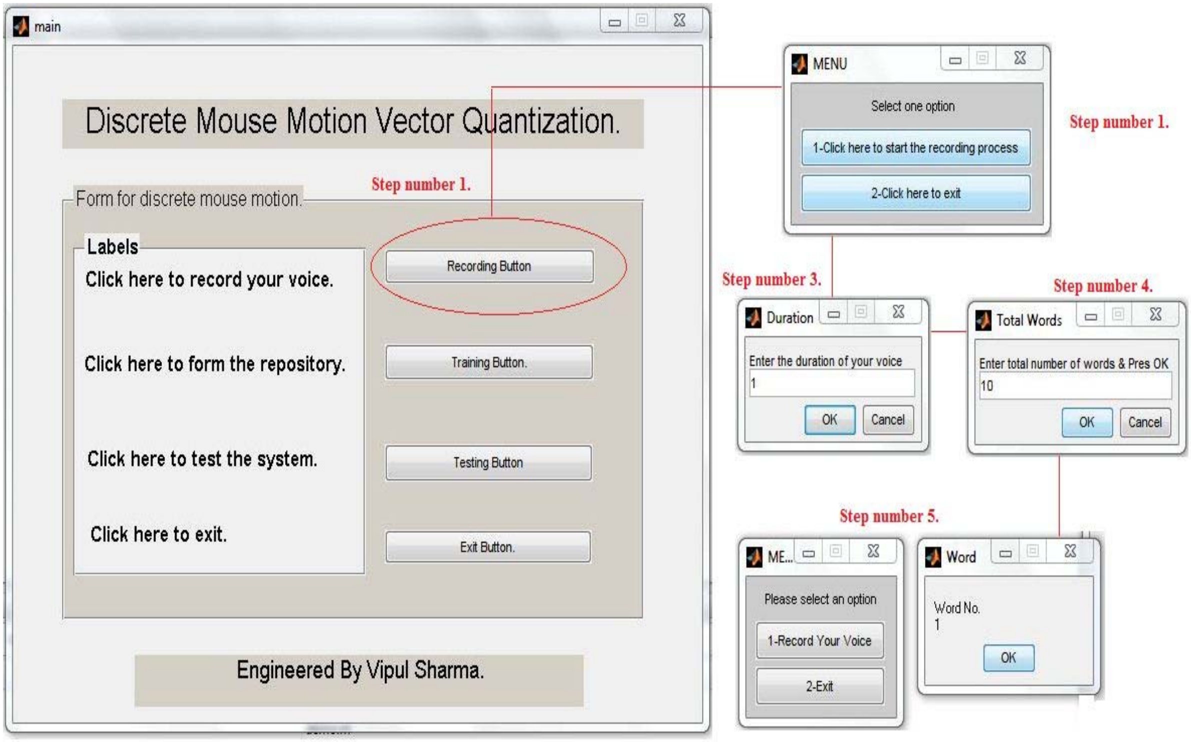
Click here to form the repository.

Click here to test the system.

Click here to exit.

Engineered By Vipul Sharma.

Fig. 7: Main GUI of discrete mouse motion control system.



main

Discrete Mouse Motion Vector Quantization.

Form for discrete mouse motion.

Step number 1.

Labels

Click here to record your voice.

Click here to form the repository.

Click here to test the system.

Click here to exit.

Engineered By Vipul Sharma.

MENU

Select one option

Step number 1.

Step number 3.

Duration

Enter the duration of your voice

1

Step number 4.

Total Words

Enter total number of words & Pres OK

10

Step number 5.

ME...

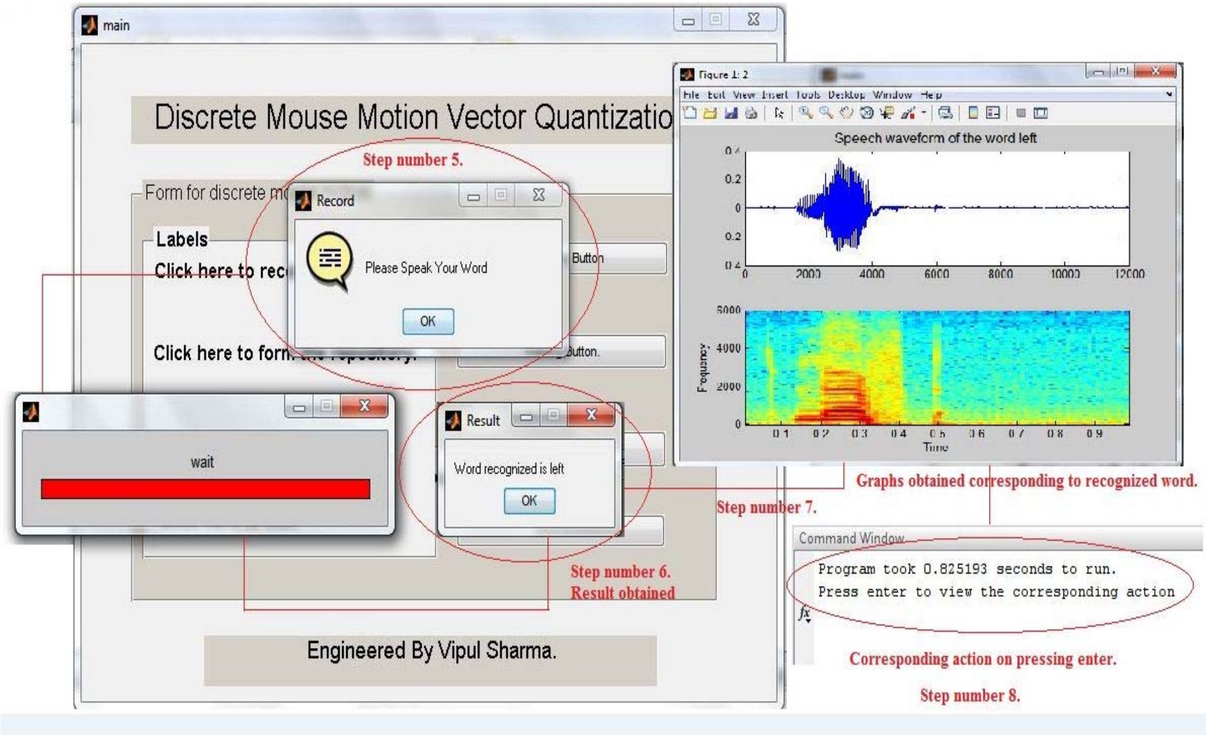
Please select an option

Word

Word No.

1

Fig. 8: Recording process of discrete mouse motion control system.



Step number 5.

Step number 6.
Result obtained

Step number 7.

Graphs obtained corresponding to recognized word.

```
Program took 0.825193 seconds to run.
Press enter to view the corresponding action
```

Corresponding action on pressing enter.

Step number 8.

Fig. 9: Testing process of discrete mouse motion control system.

D. *Implementation of Continuous Mouse Cursor Motion.*

Underlying approach: - For making continuous mouse cursor movement possible, we have used non-vocal vowel sounds, where sound ‘a’ corresponds to leftward motion, sound ‘e’ corresponds to rightward motion, sound ‘i’ corresponds to upward and sound ‘o’ corresponds to downward motion.

Here in implementation of this system we have used the sound phonetics technique. We have recorded 1 second sound samples of four vowel sounds, then after noise reduction, we have divided each 1 second sound sample into multiple frames of 20 ms duration each. The next step is to take the first frame and after extracting the phonetic features we need to save them in matrix form. This procedure is repeated for each vowel sound.

For testing purpose, user need to call the recording function and start uttering the vowel sound of desired duration this desired duration value is saved in a variable which is later used in the motion control phase. After recording the vowel sound, the next step is to invoke the phonetic feature extraction phase, which breaks the recorded sample into frames of 20 ms each. Similar procedure is followed to extract the phonetic features as we have followed in the training phase. The next step is to invoke the pattern matching module which recognizes the underlying non vocal sound by using minimum feature distance technique.

As soon as the system is having the recognized sound, mouse motion control module is invoked which causes mouse cursor to move for the desired amount of duration.

The algorithm framed for this system is as: -

Start.

1. Invoke the recording function to record 1 second sound samples of vowel sound ‘a’, ‘e’, ‘i’ and ‘o’.
2. Remove the background noise from the samples and divide each sample into multiple frames of 20 ms each.
3. Call the feature extraction module that takes the first frame of each sample and extracts the phonetic features corresponding to each vowel sound. These extracted features are stored in the form of a matrix.

   Training process is now over.

   Testing process is as.
4. Again invoke the recording function, to record vowel sound sample of desired duration for testing purpose. The duration of sound is saved in a variable which later would be referenced during the mouse motion phase.
5. After removing the background noise recorded sample is broken down into multiple frames of 20 ms duration and first frame is taken for extracting the phonetic features.
6. Pattern matching module is then invoked to recognize the underlying vowel sound.

7. The last step is the desired mouse cursor movement which is reflected on the screen for the desired amount of time after invoking the motion control module.

End of algorithm.

The user interface we have designed for continuous motion system is similar to that of the discrete motion, except that instead of regular words we are recording vowel sounds.

E. Screen Shots of the System for Continuous Mouse Motion.

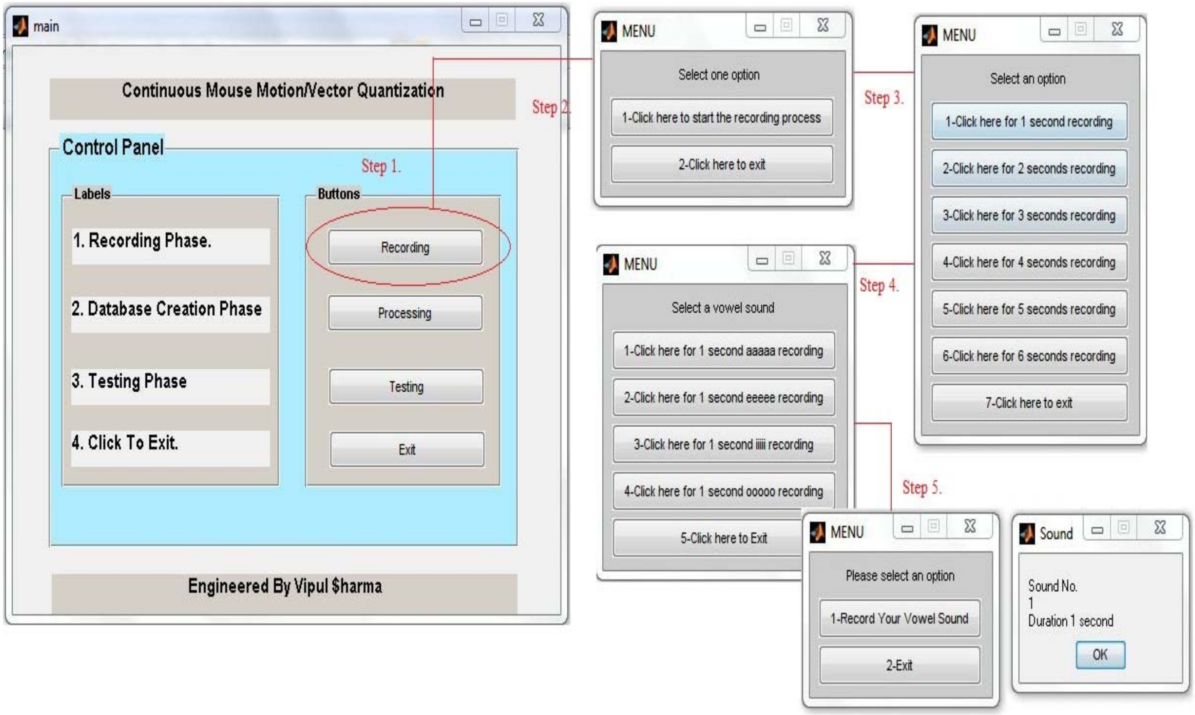
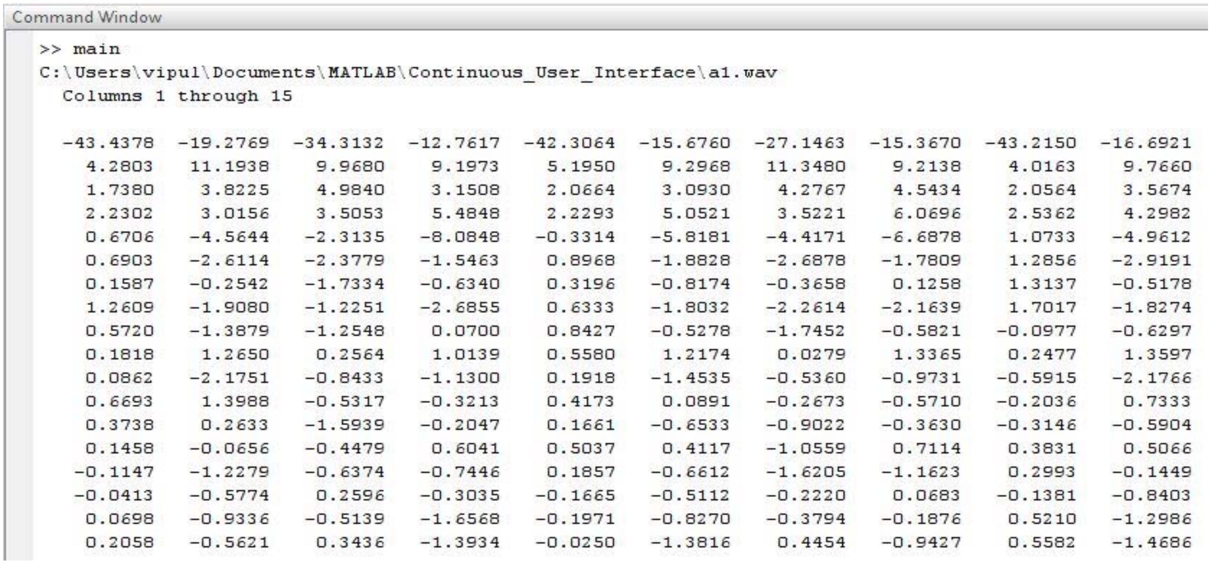

main
Continuous Mouse Motion/Vector Quantization
Control Panel
Labels
1. Recording Phase.
2. Database Creation Phase
3. Testing Phase
4. Click To Exit.
Buttons
Recording
Processing
Testing
Exit
Engineered By Vipul Sharma
Step 1.
Step 2.
MENU
Select one option
1-Click here to start the recording process
2-Click here to exit
Step 3.
MENU
Select an option
1-Click here for 1 second recording
2-Click here for 2 seconds recording
3-Click here for 3 seconds recording
4-Click here for 4 seconds recording
5-Click here for 5 seconds recording
6-Click here for 6 seconds recording
7-Click here to exit
Step 4.
MENU
Select a vowel sound
1-Click here for 1 second aaaaa recording
2-Click here for 1 second eeeee recording
3-Click here for 1 second iiiii recording
4-Click here for 1 second ooooo recording
5-Click here to Exit
Step 5.
MENU
Please select an option
1-Record Your Vowel Sound
2-Exit
Sound
Sound No.
1
Duration 1 second
OK

Fig. 10: Recording process for continuous mouse motion control system.

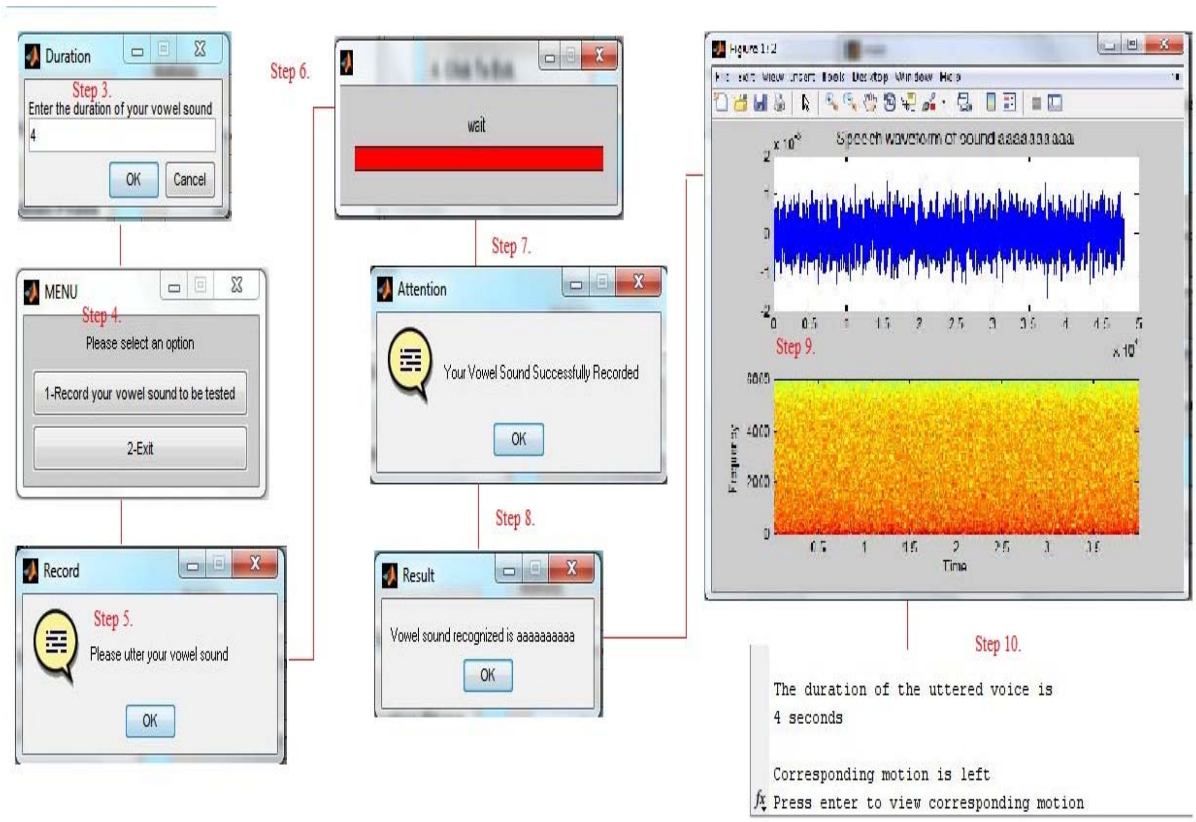


Command Window

```
>> main
C:\Users\vipul\Documents\MATLAB\Continuous_User_Interface\a1.wav
  Columns 1 through 15

  -43.4378  -19.2769  -34.3132  -12.7617  -42.3064  -15.6760  -27.1463  -15.3670  -43.2150  -16.6921
    4.2803   11.1938    9.9680    9.1973    5.1950    9.2968   11.3480    9.2138    4.0163    9.7660
    1.7380    3.8225    4.9840    3.1508    2.0664    3.0930    4.2767    4.5434    2.0564    3.5674
    2.2302    3.0156    3.5053    5.4848    2.2293    5.0521    3.5221    6.0696    2.5362    4.2982
    0.6706   -4.5644   -2.3135   -8.0848   -0.3314   -5.8181   -4.4171   -6.6878    1.0733   -4.9612
    0.6903   -2.6114   -2.3779   -1.5463    0.8968   -1.8828   -2.6878   -1.7809    1.2856   -2.9191
    0.1587   -0.2542   -1.7334   -0.6340    0.3196   -0.8174   -0.3658    0.1258    1.3137   -0.5178
    1.2609   -1.9080   -1.2251   -2.6855    0.6333   -1.8032   -2.2614   -2.1639    1.7017   -1.8274
    0.5720   -1.3879   -1.2548    0.0700    0.8427   -0.5278   -1.7452   -0.5821   -0.0977   -0.6297
    0.1818    1.2650    0.2564    1.0139    0.5580    1.2174    0.0279    1.3365    0.2477    1.3597
    0.0862   -2.1751   -0.8433   -1.1300    0.1918   -1.4535   -0.5360   -0.9731   -0.5915   -2.1766
    0.6693    1.3988   -0.5317   -0.3213    0.4173    0.0891   -0.2673   -0.5710   -0.2036    0.7333
    0.3738    0.2633   -1.5939   -0.2047    0.1661   -0.6533   -0.9022   -0.3630   -0.3146   -0.5904
    0.1458   -0.0656   -0.4479    0.6041    0.5037    0.4117   -1.0559    0.7114    0.3831    0.5066
   -0.1147   -1.2279   -0.6374   -0.7446    0.1857   -0.6612   -1.6205   -1.1623    0.2993   -0.1449
   -0.0413   -0.5774    0.2596   -0.3035   -0.1665   -0.5112   -0.2220    0.0683   -0.1381   -0.8403
    0.0698   -0.9336   -0.5139   -1.6568   -0.1971   -0.8270   -0.3794   -0.1876    0.5210   -1.2986
    0.2058   -0.5621    0.3436   -1.3934   -0.0250   -1.3816    0.4454   -0.9427    0.5582   -1.4686
```

Fig. 11: Extracted features using phonetic technique for continuous motion control system.



Step 3.

Enter the duration of your vowel sound

4

Step 4.

Please select an option

Step 5.

Please utter your vowel sound

Step 6.

wait

Step 7.

Your Vowel Sound Successfully Recorded

Step 8.

Vowel sound recognized is aaaaaaaaaa

Step 9.

Speech waveform of sound aaaaaaaaaa

Step 10.

```
The duration of the uttered voice is
4 seconds

Corresponding motion is left
Press enter to view corresponding motion
```

Duration
OK
Cancel
MENU
1-Record your vowel sound to be tested
2-Exit
Record
OK
Attention
OK
Result
OK
Figure 1:2
File Edit View Insert Tools Desktop Window Help
x 10^4
Frequency
Time

Fig. 12: Testing process of continuous mouse motion control system.

V. RESULTS & DISCUSSION.

A. *Results for Discrete Mouse Motion.*

We have tested the system using 50 samples of each of the following words.

Left, right, up & down. The results that we have obtained are as given below in tabular form.

TABLE 1: Discrete mouse cursor movement result using vector quantization for 1 second word utterance.

Vector Quantization. Number of words: - 4 & Sound Duration: - 1 second. Number of samples for each word: - 50			
Word.	**Accuracy of Recognition.**	**Average Time of Recognition.**	**Motion**
Left	100%	0.939315 secs.	True
Right	80%	0.8316498 secs.	True
Up	100%	0.8311384 secs.	True
Down	100%	0.83433548 secs.	True
Overall Accuracy: - 95%			
Average Time Taken: - 0.854037308 secs.			

From the above results we conclude that system gives 95% accurate results and motion of the cursor is smooth.

B. *Results for Continuous Mouse Motion.*

Here again the system is tested using 50 samples of each vowel sound. We have further tested the system for 1 to 6 seconds duration of vowel sounds.

The results we have obtained are as given below in tabular form.

TABLE 2: Continuous mouse cursor movement result using vector quantization for 1 second sound utterance.

Algorithm	1 sec /a/ left motion	1 sec /e/ right motion	1 sec /i/ up motion	1 sec /o/ down motion
VQ	True	True	True	True

TABLE 3: Continuous mouse cursor movement result using vector quantization for 2 seconds sound utterance.

Algorithm	2 sec /a/ left motion	2 sec /e/ right motion	2 sec /i/ up motion	2 sec /o/ down motion
VQ	True	True	True	True

TABLE 4: Continuous mouse cursor movement result using vector quantization for 3 seconds sound utterance.

Algorithm	3 sec /a/ left motion	3 sec /e/ right motion	3 sec /i/ up motion	3 sec /o/ down motion
VQ	True	True	True	True

TABLE 5: Continuous mouse cursor movement result using vector quantization for 4 seconds sound utterance.

Algorithm	4 sec /a/ left motion	4 sec /e/ right motion	4 sec /i/ up motion	4 sec /o/ down motion
VQ	True	True	True	True

TABLE 6: Continuous mouse cursor movement result using vector quantization for 5 seconds sound utterance.

Algorithm	5 sec /a/ left motion	5 sec /e/ right motion	5 sec /i/ up motion	5 sec /o/ down motion
VQ	True	True	True	True

TABLE 7: Continuous mouse cursor movement result using vector quantization for 6 seconds sound utterance.

Algorithm	6 sec /a/ left motion	6 sec /e/right motion	6 sec /i/ up motion	6 sec /o/down motion
VQ	True	True	True	True

From the above tables it is clear that we are getting 100% accuracy for continuous mouse motion using phonetic feature extraction technique.

VI. CONCLUSION.

In this paper we have implemented both discrete and continuous mouse cursor movements. The discrete mouse cursor movement is implemented using Vector quantization technique and gives an accuracy rate of about 95% when tested. For implementing continuous muse cursor movement we have proposed a new technique of phonetic feature extraction using vector quantization. The accuracy rate we are obtaining for sound recognition and corresponding mouse cursor movement is 100% using this new technique. Hence we can conclude that this new technique is robust and very efficient for causing continuous mouse cursor movement.

Further we conclude, that the systems we have developed are: -

More interactive and easy to use: - The speech based mouse we have developed enables a user to control mouse pointer both in discrete and continuous fashion.

Robust: - For continuous motion we are getting an accuracy rate of 100% and for discrete it is 95%, hence we can conclude that the systems we have developed are robust.

Training required is minimum: - Anyone can use our system. The GUI’s we have designed are self explanatory and interactive in nature. New users need a maximum of 6 to 10 minutes of training to use the system.

VII. FUTURE SCOPE.

We have used only vector quantization to implement our systems. Apart from vector quantization technique there are many algorithms like Hidden markov model, Neural Networks etc that can be used for implementing the same system. Also we can use phonetic feature extraction technique with all these algorithms and can analyse the results obtained on parameters like accuracy rate, time taken for recognition and space occupied by feature matrices.

VIII. REFERENCES.

[1] R Norma Conn and Michael McTear, “Speech Technology: A Solution for People with Disabilities”, Savoy Place, London WCPR OBL, UK: IEE, 2000.

[2] Susumu Harada, Jacob O Wobbrock, Jonathan Malkin, Jeff A Bilmes and James A Landay ,“Longitudinal Study of People Learning to Use Continuous Voice-Based Cursor Control”, Boston, MA: Conf. on Human Factors in Computing Systems

[3] Zaidi Razak,Noor Jamilah Ibrahim, emran mohd tamil,mohd Yamani Idna Idris, Mohd yaakob Yusoff,Quranic verse recition feature extraction using mel frequency ceostral coefficient (MFCC),Universiti Malaya.

[4] http://www.cse.unsw.edu.au/~waleed/phd/html/node38.html, downloaded on 15th April 2013.

[5] Susumu Harada, “Harnessing the Capacity of the Human Voice for Fluidly Controlling Computer Interface”, University of Washington, 2010

[6] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques” JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010, ISSN 2151-9617.

[7] Mahdi Shaneh, and Azizollah Taheri “Voice Command Recognition System Based on MFCC and VQ Algorithms” World Academy of Science, Engineering and Technology 33 2009.

[8] Rabiner, L. R. and Juang, B.-H. (1993), Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cli_s, NJ.

[9] Christian Spanner 2005, Speech codec identification for Error Correction of Across-Channel effects in speech coded environments.

[10] B. Richard, january, 2001, "Text-independent speaker recognition using source based features", Master of philosophy, Wildermoth Griffith University Australia.

[11] Tejaswini Hebalkar, Spring 2000 Voice Recognition and Identification System Final Report 18-551 Digital Communications and Signal Processing Systems Design.