1. Introduction

The brief overview of human speechreading in the previous chapter suggests the significant gains that may be made by integrating auditory and visual cues for automatic speech recognition and audio-visual speech recognition. There are two major problems for an audio-visual speech recognizer. The first problem is to identify robust visual features for recognition and similarly the salient visual speech features as compared to acoustic speech recognition, this problem is even more difficult due to the much higher visual data rates. Therefore two clear paradigms have been emerged for visual feature extraction. The high-level or model based approach asserts prior decision about what visual features are important. The low-level or pixel based, approach ignores all prior knowledge and applies a direct statistical analysis to the data. Provided that the correct model is used then model-based system focuses on specific image features, that are considered to be important, might be expected to be more robust.

The second problem is how and when to integrate the audio and visual speech features. Similar to the discussion of human integration, these can be described as either early (pre-classification) or late (post-classification). The extremes of model-based vs. pixel based and early integration vs. late integration is a continuum of implementation possibilities [01, 02]. This is further expanded by considering the complexity of the speech recognition tasks; that is how many talker, size of vocabulary, type of speech (discrete or continuous) dictations etc. following section
briefly describes the experimental framework used in this research work and visual analysis.

2. Experimental Framework

As discussed above, the first main problem in the area of speechreading is the question of appropriate visual speech representation in terms of small number of informative features. The various set of features have been proposed for this purpose in the literature over last 20 years. The System takes the input in the form of video (moving picture) which is comprised of visual and audio data as shown in Figure 3.1; this was act as an input to the audio visual speech recognition. The samples from the subjects having devnagari language as mother tongue; been collected. These samples are for the isolated words of city names.

![Figure 3.1: Proposed Model](image)

Each speaker or subject is requested to begin and end the each utterance for isolated city names with their mouth in ‘closed-open-closed’ position. No head movement is allowed and speakers are provided with close up view of their mouth and urged to do not move face out of the frame. With these constraints the dataset is prepared. This video input was acquired in acquisition phase. In sampling phase the acquired video is sampled in to frames. This sampling of frame was done with the standard rate of approximately 32 frames per second. Normally the Video input of 2 seconds was recorded for each subject. The sampler produces 64 images for face and was considered for a vector ‘I’ of size 64 images. The frame vector enhancements were carried out by the enhancement phase and are discussed in detail in the section 4 of this chapter. The frame vector was analyzed for feature extraction with following two approaches of visual analysis:

a. Low-Level Visual Analysis

b. High-Level Visual Analysis
The details of visual feature extraction, matching and recognition are discussed in section 5.

3. Audio Visual database

The statistical models for classification require a representative database of examples if they are to be reliably trained and tested. In the audio-visual speech recognition community there is a currently need of large standard database on which statically well trained models can tested and comparative results can be made. This was well addressed by the audio-visual speech recognition community but the problem of storing and distributing large audio-visual database is still significant issue. Therefore, in absence of a standard database each research group has collected own database. There are some of the audio-visual databases are exits and they are developed as per the research interest and problem taken by the researchers. These databases are available with limited set of vocabulary; some of the popular databases are as follows:

3.1 Tulips1 Database

Tulips database [03] of a subset of isolated digits was made freely available from the University of California at San Diego. The Tulips database [03] was recorded by Javier Movellan, of the department of cognitive science at university of California at San Diego. It contains only two repetitions of isolated digits 1-4 by each of 12 talkers, 9 male and 3 females, and total 96 utterances. This was recorded with office ceiling lights with an additional incandescent lamp at the side to simulate office working conditions. Talkers were not restrained but could view their mouths and were asked not to move out of shot. All talkers participated in development of this database were students of cognitive science school at the time of recording and are of mixed ethnic origin.

The database was digitized at 100 x 75 resolutions at 30fps (NTSC frame rate) using Macintosh Quadra 840 AV in ITU-R BT 601-4 8-bit headroom grayscale. Audio was simultaneously recorded at 11 KHz and 8 bit resolution. This database is available to download from http://cogsci.ucsd.edu/~movellan/. Each utterance was hand segmented so that the video and audio channels extended to one frame either side of interval containing significant audio energy. If the lips were clearly moving before or after this time to an additional three extra frames were also included. This database was extensively studied and recognition results obtained using several extraction methods was published by Movellan et.al [03-06] and Luettin [07-11].
3.2 AV Database (Audio-Visual Database by IBM)

IBM has developed AV database at Thomas J Watson Research center. The database consist of full-face frontal video and audio of 290 subjects, uttering ViaVoice™ training scripts that is continuous read speech with mostly verbalized punctuation (dictation style) and vocabulary size of approximately 10,500 words. Transcriptions of all 24,325 database utterances, as well as a pronunciation dictionary are provided. The database video is size 704x480 pixels, interlaced, captured in color at rate of 30 Hz (that is 60 fields per second are available at a resolution of 240 lines) and it is MPEG2 encoded at the relatively high compression ratio about 50:1. High quality wideband audio is synchronously collected with the video at a rate of 16 kHz and at a relatively clean audio environment (quite office, with some background computer noise). The duration of entire database is approximately 50 hrs. it is mentioned that to this date this is the largest audio visual database collected and it constitutes the only one suitable task of continuous large vocabulary, speaker independent audio-visual speech recognition, as all other existing databases are limited to small number of subjects and/or small vocabulary tasks. [12, 13, 14, 15, 16, 17, 05, 18, 19].

3.3 CUAVE database

This database project has been initiated at Clemson University, for audio-visual experiments. The main interest is audio-visual processing where visual technique such as lipreading is combined with more traditional audio methods. The multimodal speech processing can be much more robust to noise and has other benefits as well. This was beneficial to researchers working in this area. CUAVE is a speech corpus of over 7000 utterances of continuous, connected and isolated digits. It includes both individual speakers and speaker pairs. The speech is fully labeled and all files are available on DVD. There are 36 individuals in the database that is 17 female and 19 male speakers. All these speakers have different accents, skin, tone, facial hairs, hats and glasses. There are also 20 groups of speakers that may be used for multi-speaker research. The video is compressed at 5000 kbps at MPEG-2 encoding. Audio is included with 44 kHz stereo and 16 kHz mono. The speakers are recorded either moving or standing still while saying connected and continuous digit sequences [20].

3.4 Indian Isolated City name Database

The work described in this thesis was started with an assumption as there were no audio-visual speech database available in accordance with the speakers with devnagari accents therefore we have recorded our own aligned visual database of
isolated words of city names. As the nature of the database is different therefore the results reported in this research work are solely based for database of isolated city names.

The database consists of ten repetitions by each of ten talkers for each of isolated name of city. The recording of the samples have been done with female talkers with different age groups uttering city names as \{AURANGABAD, MUMBAI, CHENNAI, PARBHANI, OSMANABAD, NANTED, SATARA, BEED, PARBHANI, NAGPUR\}. All recordings was of the full face video is of size 100k, interlaced, captured in the color at a rate of 15Hz (as capturing camera source is INTEX-IT305WC with night vision mode and speed of capturing video @ the rate of 32fps) in relatively clean background. The Video format used for recording is NTSC instead of PAL; with 720 resolutions which is maximum resolution provided by the capturing device similarly the audio is captured with PCM audio compressor.

The duration of video is depending upon the utterance time required for the set of defined words and as per speakers. Each utterance was digitized the camera in color mode. The video sample of each talker has to be sampled in order to obtain the frame vector using sampler. The sampler used here sample the video of utterance with the rate of 32 frames per second. All samples were collected from talkers in ‘Close-Open-Close’ mode. Therefore this variation (Close-Open-Close) can easily been seen in the vector of sampled frame vector of video. Usually the utterance of 2 seconds was recorded for each talker. This duration may be varying depending upon utterance of talker with respective to city name. After sampling the frame vector of almost 64 image frames was formed. The full face image as shown in Figure 3.2 were further cropped to region of 90x135 pixels by locating the center of mouth and all other remaining frames of vector was cropped with the same size. An example utterance sequence Figure 3.3

![Figure 3.2](a) Color (b) Grayscale

**Figure 3.2** (A)-(B) Full face frame image after sampling of video
This database was made available on CDROM and be available to researchers on request. For understanding the significance of this database, it is compared with other existing database with attributes like tasks, repetitions, number of talkers, total utterances, total frame, frame sampling rate, area of mouth size etc., the significant difference is observed when it is compared with tulips1 database and with other are as shown in Table 3.1

**Table 3.1: Comparison between Isolated City Name databases and existing database**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Isolated City Name database</th>
<th>CUAVE Database</th>
<th>AV Database</th>
<th>Tulips database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>10 City Names</td>
<td>Connected Contiguous Digit Sequences</td>
<td>10,500 Words</td>
<td>Digits '1'-4'</td>
</tr>
<tr>
<td>Repetitions</td>
<td>10</td>
<td>-</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Number of Talkers</td>
<td>10</td>
<td>37</td>
<td>290</td>
<td>12</td>
</tr>
<tr>
<td>Utterances</td>
<td>1000</td>
<td>7000</td>
<td>24325</td>
<td>96</td>
</tr>
<tr>
<td>Frames</td>
<td>62000</td>
<td>-</td>
<td>1459500</td>
<td>934</td>
</tr>
<tr>
<td>Frame rate</td>
<td>32fps</td>
<td>MPEG-2 format</td>
<td>60fps</td>
<td>30fps</td>
</tr>
<tr>
<td>Image size (Mouth area)</td>
<td>95x135</td>
<td>-</td>
<td>Image size 704x480</td>
<td>100x75</td>
</tr>
<tr>
<td>Audio</td>
<td>16 bit</td>
<td>44 Khz Stereo and 16 Khz Mono</td>
<td>-</td>
<td>8 bit</td>
</tr>
</tbody>
</table>
4. Frame Vector Enhancement

All sampled image frame are stored in the image vector ‘I’. This image vector has to be enhanced because images in vector ‘I’ are depend on lighting conditions, head positions etc. The vector ‘I’ was checked with the subjective and objective image enhancement quality parameters. There are many subjective and objective quality measuring methods have been developed for image quality evaluation in last few years and they are particularly based on numerical measures of image quality and computable distortion measures [21, 22]. They can be divided into two categories 1) Image differencing where it uses usually matrix based operations to derive important parameters of image that is the output are processed on image matrix and 2) Feature extraction where extracting the feature is important for image quality. Therefore most of the methods are based on Human Visual System (HVS), which allows to a better correlation with the response of human observer. This research work, we have chosen some of the common objective quality measures which are evaluated in [22, 23, 24, 25]. They are discrete and provide some degree of closeness between the digital images by exploiting the differences in the statistical distributions of pixel.

4.1 Statistical Quality Measures for Image enhancements:

The quantitative distortion of enhanced image is commonly measured by:

4.1.1 MSE (Mean Square Error): the mean square error or MSE of an estimator is one of many ways to quantify the difference between an estimator and the true value of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the square of the "error." The error is the amount by which the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

\[
MSE = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} (x_{j,k} - x'_{j,k})^2 
\]

4.1.2 PSNR (Peak Signal to Noise Ratio): is the term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many
signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

\[ PSNR = 10 \log(255^2 / MSE) \]  

### 4.1.3 NAE (Normalized Absolute Error):

\[
NAE = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} |x_{j,k} - \hat{x}_{j,k}|}{\sum_{j=1}^{M} \sum_{k=1}^{N} |x_{j,k}|}
\]

### 4.1.4 MD (Maximum Difference)

\[
MD = \max \left( |x_{j,k} - \hat{x}_{j,k}| \right)
\]

### 4.1.5 SC (Structural Content)

\[
SC = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k}^2}{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k}^2}
\]

### 4.1.6 AD (Average Difference)

\[
AD = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} (x_{j,k} - \hat{x}_{j,k})}{MN}
\]

### 4.1.7 NK (Normalized Cross-correlation)

\[
NK = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k} - \hat{x}_{j,k}}{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k}^2}
\]

### 4.1.8 CQ (Correlation Quality)

\[
CQ = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k} \cdot \hat{x}_{j,k}}{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k}^2}
\]

These metrics for measuring image quality were recorded before and after enhancements. The registration of images are not required because the sample collected from subject is in constrained environment and subject is not allowed to move head during utterances of isolated words.
4.2 Frame Enhancement

Image vector $I$ was processed for color to gray and gray to binary representations by following mechanism:

4.2.1 **Histogram equalization:** is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive.

4.2.2 **Image Morphology:**

Image morphology is mathematical morphology and is the nonlinear analysis of signals by shape. The work by Matheron in 1975[26] and was extended by Serra [27-29]. This is used for extracting image components that are useful in representation and description of region shape, such as boundaries, skeleton and the convex hull. This section is a basic introduction to the morphological processors. The basic morphological operation is a probing of the signal, $X$, using some structuring element, $B$, with

4.2.2.1 **Dilation:** It is fundamental operation of image morphology. Dilation is the operation that “grows” or “thickens” objects in binary image. The specific manner and the extent of this thickening is controlled by shape referred as structuring element. Computationally, structuring element typically is represented by
matrix of 0s and 1s; sometime it is convenient to show with 1s; so that the origin of the structuring element must be identified.

Mathematically, dilation is defined in terms of set operation. The dilation of A by B denoted by \( A \oplus B \), is defined as

\[
A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}
\]

Where, \( \emptyset \) is the empty set and B is the structuring element, in other words, the dilation of A by B is the set of consisting of all the structuring element origin locations where the reflected and translated B overlaps at least some portion of A. The translation of structuring element in dilation is similar to the mechanics of spatial convolution. The dilation is commutative that is \( A \oplus B = B \oplus A \). It is the convention of image processing to let the first operand of \( A \oplus B \) to be the image and the second operand to be the structuring element, which usually is much smaller than the image.

4.2.2.2 Erosion: this term refer to “shrinks” or “thins” objects in binary image. As in dilation, the manner and extent of shrinking is controlled by structuring element. The mathematical definition of erosion is similar to that of dilation, the erosion of A by B, denoted by \( A \ominus B \) is defined as

\[
A \ominus B = \{ z \mid (B)_z \cap A^c \neq \emptyset \}
\]

In other words, erosion of A by B is the set of all structuring element origin locations where the translated B has no overlap with the background of A.

4.2.2.3 Combination on Dilation and Erosion: In image processing applications, dilation and erosion are used most often at various combinations. An image undergoes a series of dilation and/or erosion using the same or sometimes different, structuring elements. There are three most combinations of dilation and erosion: opening, closing and the hit-or-miss transformation.
4.2.2.3.1 Opening and Closing: The morphological opening of A by B, denoted \( A \circ B \), is simply erosion of A by B followed by dilation of the result by B:
\[
A \circ B = (A \ominus B) \oplus B
\]  
\text{...(11)}

An alternative mathematical formulation of opening is
\[
A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}
\]  
\text{...(12)}

Where \( \bigcup \{ \bullet \} \) denotes the union of all sets inside the braces and the notation \( C \subseteq D \) means that C is a subset of D. This formulation has a simple geometric interpretation that \( A \circ B \) is the union of all translations of B that fit entirely within A. The morphological opening removes completely a region of an object that cannot contain the structuring element, smoothes object contours, breaks thin connections and removes thin protrusions.

The morphological closing of A by B denoted by \( A \bullet B \), is the dilation followed by an erosion:
\[
A \bullet B = (A \oplus B) \ominus B
\]  
\text{...(13)}

Geometrically, \( A \bullet B \) is the complement of the union of all translations of B that do not overlap A. Like opening, morphological closing tends to smooth the contours of objects. Unlike opening, however, it generally joins narrow breaks, fills long thin gulls and fills holes smaller than the structuring element.

4.2.2.3.2 The Hit-or-Miss transformation: It is useful to identify specified configurations of pixels such as isolated foreground pixels or the pixels that are end points of line segments. The hit-or-miss transformation is useful for application such as these. The hit-or-miss transformation of A by B denoted by \( A \otimes B \). Here B is a structuring element pair \( B = (B_1, B_2) \), rather than single element as before. The hit-or-miss transformation is defined in the terms of these two structuring elements as
\[ A \odot B = (A \ominus B_1) \cap (A^C \ominus B_2) \quad \ldots(14) \]

**4.2.2.4 Structuring Element Decomposition**: Dilation is associative that is \( A \ominus (B \ominus C) = (A \ominus B) \ominus C \), suppose that, the structuring element B can be represented as dilation of two structuring element \( B_1 \) and \( B_2 \) as:

\[ B = B_1 \oplus B_2 \quad \ldots(15) \]

Then, \( A \ominus B = A \ominus (B_1 \ominus B_2) = (A \ominus B_1) \ominus B_2 \), in other words, dilating A with B is the same as first dilating A with \( B_1 \), and then dilating the result with \( B_2 \). Therefore we can say that B can be decomposed into structuring elements \( B_1 \) and \( B_2 \).

The associative property is important because the time required to compute dilation is proportional to the number of nonzero pixels in the structuring element. Consider for example dilation with 5 x 5 arrays of 1s:

\[
\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

This structuring element can be decomposed into five-element of row of 1s and a five-element column of 1s:

\[
[1 \ 1 \ 1 \ 1 \ 1] \oplus \begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
1 \\
\end{bmatrix}
\]

The number of element in the original structuring element is 25, but the total number of element in the row-column decomposition is only 10. This means the dilation with the row structuring element first, followed by dilation with the column element, can be performed 2.5 times faster than dilation with the 5x5 arrays of 1s. In practice the speedup was somewhat less because there is usually some overhead associated with each dilation operation and at least two dilation operations are required when using the
decomposed form. However, the gain in speed with decomposed implementation is still significant.

While performing experiments of this research work, combination of opening and closing morphological operations were considered and the frame vector is binarized as shown in figure 3.4 so that the features can be extracted for the training and recognition of the utterances of speaker.

![Color, Grayscale, Binary Images](image)

**Figure 3.4** Full face frame image after performing morphological operations, a) Color, b) Grayscale and c) Binary Image

![Full face frame vector](image)

**Figure 3.5** Full face frame vector of images obtained after performing morphological operations.
(B) Visual Feature Extraction

As discussed earlier, the first main problem to isolate the area for speechreading which was the question of appropriate visual speech representation in terms of a small number of informative features. Various sets of visual features have been proposed for this purpose in the literature. In general, they can be grouped into High-level analysis using contour based feature, low-level analysis using pixel based and the combination of both.

In first approach, the speaker’s inner and (or) outer lip contours are extracted from image sequences using model based visual features, alternatively lip contour geometric features are used, such as mouth height and width. In second approach the entire image containing speaker’s mouth is considered as informative for lipreading (region of interest –ROI), and appropriate transformation of its pixel values are used as visual features. The full face frame image after performing morphological operation over entire image vector, the vector is processed for visual feature extraction method by low level feature extraction.

5.1 Face Feature Estimation

In order to extract useful features from images for lipreading is some way of condensing this information in to relevant feature space. The work presented in this section concerns face feature extraction from face vector by automated way. The results obtained with the process are presented in Chapter 04. The problem during face feature extraction is to identify the Region of Interest (ROI) containing mouth from the full face image vector. If ‘F’ be the first full face image of from the set of vector ‘I’ and be represented by two dimensional light intensity function \( F(x, y) \) which returns amplitude at an coordinate \( x, y \). This was made represented for the isolated word ‘AURANGABAD’ uttered by the speaker. By computing scan line projection of the face image in row wise as \( R(x) \), columns as \( C(y) \)

\[
R(x) = \sum_{x=1}^{m} \sum_{y=1}^{n} F(x, y) \quad \text{....(16)}
\]

\[
C(y) = \sum_{y=1}^{n} \sum_{x=1}^{m} F(y, x) \quad \text{....(17)}
\]

This process suggests the area which can be considered for segmentation of eyes, nose & mouth from the every image of vector. This was also helpful in
classifying open-close-open mouth of the subject as well as some geometrical features such as height, width of mouth in every frame corresponding to the utterance of isolated word of the speaker.

**Figure 3.6 Full face image of subject from vector**

**Figure 3.7 2D Histogram of Row wise scan line projection**

**Figure 3.8 2D Histogram of Column wise scan line projection**

**Figure 3.9 Histogram of Full face of subject**
5.1.1 Curve Fitting: In order to approximate the geometrical feature such as size, height and width from the scan-line projection plot the curve fitting methods can be applied by performing data smoothing on $R(x)$ by

5.1.1.1 Moving Averages

In statistics, a moving average, also called rolling average, rolling mean or running average, is a type of finite impulse response filter used to analyze a set of data points by creating a series of averages of different subsets of the full data set. Given a series of numbers and a fixed subset size, the moving average can be obtained by first taking the average of the first subset. The fixed subset size is then shifted forward, creating a new subset of numbers, which is averaged. This process is repeated over the entire data series. The plot line connecting all the (fixed) averages is the moving average. Thus, a moving average is not a single number, but it is a set of numbers, each of which is the average of the corresponding subset of a larger set of
A moving average may also use unequal weights for each data value in the subset to emphasize particular values in the subset. A moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles. The threshold between short-term and long-term depends on the application, and the parameters of the moving average were set accordingly. For example, it is often used in technical analysis of financial data, like stock prices, returns or trading volumes. It is also used in economics to examine gross domestic product, employment or other macroeconomic time series. Mathematically, a moving average is a type of convolution and so it is also similar to the low-pass filter used in signal processing. When used with non-time series data, a moving average simply acts as a generic smoothing operation without any specific connection to time, although typically some kind of ordering is implied.

5.1.1.2 **LOESS or LOWESS** (locally weighted scatter plot smoothing): is one of many "modern" modeling methods that build on "classical" methods, such as linear and nonlinear least squares regression. Modern regression methods are designed to address situations in which the classical procedures do not perform well or cannot be effectively applied without undue labor. LOESS combines much of the simplicity of linear least squares regression with the flexibility of nonlinear regression. It does this by fitting simple models to localized subsets of the data to build up a function that describes the deterministic part of the variation in the data, point by point. In fact, one of the chief attractions of this method is that the data analyst is not required to specify a global function of any form to fit a model to the data, only to fit segments of the data.

The trade-off for these features is increased computation. Because it is so computationally intensive, LOESS would have been practically impossible to use in the era when least squares regression was being developed. Most other modern methods for process modeling are similar to LOESS in this respect. These methods have been consciously designed to use our current computational ability to the
fullest possible advantage to achieve goals not easily achieved by
traditional approaches.
Plotting a smooth curve through a set of data points using this
statistical technique is called a Loess Curve, particularly when each
smoothed value is given by a weighted quadratic least squares
regression over the span of values of the y-axis scatter gram criterion
variable. When each smoothed value is given by a weighted linear least
squares regression over the span, this is known as a Lowess curve;
however, some authorities treat Lowess and Loess as synonyms.
The smoothed R(x) can be fitted with the Gaussian Model to find out the
fitting parameters for Eyes, Nose and Mouth respectively therefore the
function F(x) were computed with three parameters individually, that was for
Eyes, Nose and Mouth.

**Function used for fitting curve**

General model Gauss3:

\[ F(x) = a_1 \exp\left(-\frac{(x-b_1)}{c_1}\right) + a_2 \exp\left(-\frac{(x-b_2)}{c_2}\right) + a_3 \exp\left(-\frac{(x-b_3)}{c_3}\right) \]

and the results for one typical example is as follows

Coefficients (with 95% confidence bounds):

- \(a_1 = 148.3\) (145.3, 151.2)
- \(b_1 = 48.68\) (48.49, 48.88)
- \(c_1 = 10.52\) (10.24, 10.8)
- \(a_2 = 129.3\) (126.8, 131.8)
- \(b_2 = 81.06\) (80.8, 81.33)
- \(c_2 = 15.22\) (14.81, 15.63)
- \(a_3 = 74.13\) (70.6, 77.65)
- \(b_3 = 203.6\) (203.3, 203.8)
- \(c_3 = 7.374\) (6.969, 7.779)

Goodness of fit:

- SSE: 4974
- R-square: 0.9896
- Adjusted R-square: 0.9893
- RMSE: 4.443
5.2 Duplicate frame detection using PCA

Principal Component Analysis is a way that identifies the patterns in data and expressing the data in such a way that it highlights their similarities and differences [9] and for recognition of patterns a smaller feature space is desired as this allows less complex statistical models that are easier to train. PCA is such statistical model used for indentifying orthogonal directions ordered by their relative variance contribution to multidimensional data. PCA is linear transform and each direction is linear combination of original data. The result is an orthogonal rotation of axes to align in the first in the direction of the most variance and so on by decreasing variance contribution. If data is correlated PCA transforms into linearly decorrelated space. If there is no correlation PCA will simply order the axes by variance. The values of original data transformed along the top N directions can be used as decorrelated features in a reduced N-dimensional transformed feature space. The process of applying PCA on masked images from image vector is as follows

1. A 2-D frame can be represented in to 1-D vector by Concatenating each row or column into a long thin vector as

   \[ X_i = [p_1, \ldots, p_N]^T, i = 1 \ldots N \quad \ldots (18) \]

2. These frame images are mean centered by subtracting the mean image from each image vector and calculated as m

   \[ m = \frac{1}{M} \sum_{i=1}^{M} X_i \quad \ldots (19) \]

   and \( W_i \) be defined as mean centered image

   \[ w_i = X_i - m \quad \ldots (20) \]

3. The \( e_i \) and \( \lambda_i \) are given by the eigenvector and eigenvalues of the covariance matrix

   \[ C = WW^T \quad \ldots (21) \]
Where \( W \) is a matrix composed of column vector \( W_i \) placed side by side

4. A image can be projected onto \( M' \) (Eigenspace)

\[
\Omega = [v_1 v_2 \ldots v_m]^T
\]

\[
\text{....(22)}
\]

Where, \( v_i = e_i^T w_i \), \( v_i \) is the \( i^{th} \) coordinate of the image in eigenspace, which came to be principal component.

5. Compute the Euclidean distance as

\[
\epsilon_k = ||(\Omega - \Omega_k)||
\]

\[
\text{....(23)}
\]

Where \( \Omega_k \) is a vector describing \( k^{th} \) class, if \( \epsilon_k \) is less than some predefined threshold as ’\( t \)’, the image is classified as it belongs to \( k^{th} \) class

**5.3 Visual Feature extraction using Incremental Difference**

![Figure 3.12 HRL and VRL Reference Lines](image)

The Horizontal Reference Line (HRL) and Vertical Reference Line (VRL) for the lip was plotted. The points for HRL and VRL are chosen from scan line projection vectors that is \( R(x) \) and \( C(y) \). The initial values for \( P_1 \) which was the midpoint for HRL, be calculated as \( x_p = (x_2 - x_1)/2 \) and \( y_p = (y_2 - y_1)/2 \) where \( x_2, x_1 \) are the co-ordinates obtained from \( R(x) \) and \( y_2, y_1 \) are obtained for \( C(y) \). The initial values for \( P_2 \) which was the midpoint for VRL and be calculated as that of \( P_1 \). Therefore to obtain exact midpoint of HRL with reference to \( P_1 \) at \( (x_p, y_p) \) and VRL with reference to \( P_2 \) at \( (x_p, y_p) \), the HRL & VRL are represented by the line implicit function with coefficient \( a, b \) and \( c : F(x, y) = ax + by + c = 0 \) (the \( b \) coefficient of \( y \) is unrelated to the \( y \) intercept \( B \) in the slope intercept form). If \( dy = y_2 - y_1 \) and \( dx = x_2 - x_1 \) the slope intercept form can be written as
\[ y = \frac{dy}{dx} x + B \]

Therefore

\[ F(x, y) = dy \cdot x - dx \cdot y + B \cdot dx = 0 \]

Here \( a = dy, b = -dx \) and \( c = B \cdot dx \) in the implicit form as it is important for the proper functioning of the midpoint of HRL to choose ‘a’ to be positive; so that it meets this criterion if \( dy \) is positive, since \( y_2 > y_1 \).

To calculate midpoint criterion for HRL and VRL as for the pixel point \( P_1 \) and \( P_2 \) as we need to compute \( F_{HRL}(M) \) and \( F_{VRL}(M) \) as

\[ F_{HRL}(M) = F(x_p + 1, y_p + \frac{1}{2}) \] ....(24)

\[ F_{VRL}(M) = F(x_p + 1, y_p + \frac{1}{2}) \] ....(25)

The decision is based on the value of the function at \( (x_p + 1, y_p + \frac{1}{2}) \) it is necessary to define decision variable for HRL and VRL respectively

\[ d = F(x_p + 1, y_p + \frac{1}{2}) \] ....(26)

Therefore by definition

\[ d = a(x_p + 1) + b(y_p + \frac{1}{2}) + c \] ....(27)

Conditions

1. If \( d > 0 \) then we choose pixel NE (North East)
2. If \( d < 0 \) then we choose pixel E (East)
3. If \( d = 0 \) then we can choose either, recommended to choose E

The location of M is on whether we chose E or NE, if E is chosen, and then M is incremented by one step in x direction then
3.22 | Page

\[ d_{\text{new}} = F(x_p + 2, y_p + \frac{1}{2}) = a(x_p + 2) + b(y_p + \frac{1}{2}) + c \quad \ldots(28) \]

But,

\[ d_{\text{old}} = a(x_p + 1) + b(y_p + \frac{1}{2}) + c \quad \ldots(29) \]

If NE is chosen, M is incremented by one step each in both x and y direction then

\[ d_{\text{new}} = F(x_p + 2, y_p + \frac{3}{2}) = a(x_p + 2) + b(y_p + \frac{3}{2}) + c \quad \ldots(30) \]

By equation (24) and (25) [23] with support of decision variable new coordinate for \( P_1(x,y) \) for HRL and \( P_2(x,y) \) for VRL is computed. The pixel \( P_1 \) lie at the middle of HRL and pixel \( P_2 \) lie at the middle of VRL. The difference between the \( P_1 \) and \( P_2 \) is considered as incremental difference feature and was unique feature for the frame. This feature is invariant to scale, rotation and scaling. This difference is computed for all frames for utterance of word and stored in vector; this vector was referred as feature vector for the word. The feature vector contains the information of all samples of word such as 

{AURANGABAD, MUMBAI, CHENNAI, PARBHANI, OSMANABAD, NANTED, SATARA, BEED, PARBHANI, and NAGPUR}. 

![Image of features](image-url)
The difference between $P_1$ and $P_2$ is recorded as feature of the frame and similar difference with respect to all frames of vector have been computed and stored in the feature vector. The feature vector corresponding to all utterance of the isolated word is formed; their mean feature vector is calculated. The results obtained by this process will be discussed in Chapter 04.

5.4 Audio feature extraction

The extracted visual features and the features of speech signal were collected and recognition was done whereas, visually extracted features for the known set of class can also be sufficient for the recognition of utterances. In accordance to the audio visual integration the speech signal is need to be understood so that features from signal can be obtained. These features are computed using MFCC (Mel-Frequency Cepstral Coefficients).

As discussed earlier, the audio-visual samples were collected from speakers and samples were provided to sampler to sample the data into visual image frame vector for visual processing and into speech signal for speech processing task of signal. The speech signal of speaker ‘PURVA’ for the utterance ‘AURANGABAD’ was recorded in silence-utterance-silence mode and is as shown in figure 3.14.

![Spectrogram for utterance ‘AURANGABAD’](image)
The signal is containing noise as shown in figure 3.14, the small blue line indicate some speech which was considered to be noise in silence and not part of the utterance of ‘Aurangabad’. This noise can be removed and signal for the said utterance is shown in figure 3.15

![Spectrogram for utterance ‘AURANGABAD’ after removal of noise](image)

The spectrogram is a mechanism which provides a time-varying spectral representation by forming an image that shows how the spectral density of a signal varies with time. This can also be called as voiceprints or voicegrams. The spectrograms are used to identify phonetic sounds. Spectrograms are usually created in one of two ways: approximated as a filter bank that results from a series of bandpass filters (this was the only way before the advent of modern digital signal processing), or calculated from the time signal using the short-time Fourier transform (STFT).

The mean-energy intensity in speech signal was measured as 90.29 dB, the maximum intensity in the utterance (speech signal) was measured as 92.68 dB. The parameters considered for the analyzing spectrogram is shown in table 3.2
Table 3.2 Standard parameters for spectrogram analysis of signal

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Standard Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A) Time and Frequency resolution</strong></td>
<td></td>
</tr>
<tr>
<td>Number of Time Steps</td>
<td>1000</td>
</tr>
<tr>
<td>Number of Frequency steps</td>
<td>250</td>
</tr>
<tr>
<td><strong>B) Spectrogram analysis settings</strong></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Fourier</td>
</tr>
<tr>
<td>Window shape</td>
<td>Gaussian</td>
</tr>
<tr>
<td><strong>C) Spectrogram view settings</strong></td>
<td></td>
</tr>
<tr>
<td>Maximum (dB/Hz)</td>
<td>100</td>
</tr>
<tr>
<td>Dynamic compression (0-1)</td>
<td>0</td>
</tr>
</tbody>
</table>

In sampling phase the video was converted into image frame vector and audio signal was analyzed for MFCC features.

### 5.4.1 Extraction of MFCC

Our goal in this section is to describe how we transform the input waveform into a sequence of acoustic feature vectors, each vector representing the information in a small time window of the signal. Now that we have a digitized, quantized representation of the waveform, we are ready to extract MFCC features. The seven steps of this process are shown in Figure 3.16 (and a-b) and individually described in each of the following sections.

**Figure 3.16-a) Extracting a sequence of 39-dimensional MFCC feature vectors from a quantized digitized waveform**
5.4.2 Pre-emphasis
The first stage in MFCC feature extraction is to boost the amount of energy in the high frequencies. It turns out that if we look at the spectrum for voiced segments like vowels, there is more energy at the lower frequencies than the higher frequencies. This drop in energy across frequencies (which is called spectral tilt) is caused by the nature of the glottal pulse. Boosting the high frequency energy makes information from these higher formants more available to the acoustic model and improves phone detection accuracy.

This pre-emphasis is done by using a filter. Figure 3.17 shows an example of a spectral slice from the pronunciation of the single vowel [aa] before and after pre-emphasis. This pre-emphasis filter is a first-order high-pass filter. In the time domain, with input $x[n]$ and $0.9 \leq \alpha \leq 1.0$, the filter equation is $y[n] = x[n] - \alpha x[n-1]$.

5.4.3 Windowing
The goal of feature extraction is to provide spectral features that can help us to build phone or sub-phone classifiers. We therefore don’t want to extract spectral features from an entire utterance or conversation, because the spectrum changes very quickly. Technically, we say that speech is a non-stationary signal, meaning that its statistical properties are not constant across time. Instead, we want to extract spectral features from a small window of speech that characterizes a particular sub-phone and for which we can make the (rough) assumption that the signal is stationary (i.e. its statistical properties are constant within this region). We’ll do this by using a window which is
non-zero inside some region and zero elsewhere, running this window across the speech signal, and extracting the waveform inside this window.

We can characterize such a windowing process by three parameters: how wide is the window (in milliseconds), what is the offset between successive windows, and what is the shape of the window. We call the speech extracted from each window a frame, and we call the number of milliseconds in the frame the frame size and the number of milli-seconds between the left edges of successive windows the frame shift. The extraction of the signal takes place by multiplying the value of the signal at time n, $s[n]$, with the value of the window at time n, $w[n]$:

$$y[n] = w[n]s[n]$$

Figure 3.18 suggests that these window shapes are rectangular, since the extracted windowed signal looks just like the original signal. Indeed the simplest window is the rectangular window. The rectangular window can cause problems, however, because it abruptly cuts off the signal at its boundaries. These discontinuities create problems when we do Fourier analysis. For this reason, a more common window used in feature extraction is the Hamming window, which shrinks the values of the signal toward zero at the window boundaries, avoiding discontinuities.

**Figure 3.18** The windowing process, showing the frame shift and frame size, assuming a frame shift of 10 ms, a frame size of 25 ms, and a rectangular window.
Figure 3.19 shows both of these windows; the equations are as follows (assuming a window that is L frames long):

**rectangular**

\[ w[n] = \begin{cases} 
1 & 0 \leq n \leq L-1 \\ 
0; & \text{otherwise} 
\end{cases} \]

**hamming**

\[ w[n] = \begin{cases} 
0.54 - 0.46 \cos\left(\frac{2\pi n}{L}\right) & 0 \leq n \leq L-1 \\ 
0; & \text{otherwise} 
\end{cases} \]

### 5.5 Recognition system

The designed system was checked for recognition. The process of recognition of utterances is done by computing visual and audio features separately. In case of visual recognition of utterances the visual feature metrics have been prepared in accordance with utterance of isolated city name and can be considered for training. The test sample was checked with the training sample and Euclidean distance metrics was prepared for recognition. For MFCC feature same process has been adopted and 12 features have been computed for the utterance of the speaker. The details related to the experiment and experimental results will be discussed in chapter 04.
6 Summary

This chapter deals with the methodology used in order to analyze patterns generated by processing the utterances of known dataset. The methodology clearly describes method of acquiring samples, converting samples into discrete set of image frame vectors. The preprocessing of frame vector was discussed prior to feature extraction using subjective and objective quality metrics as well as use of image enhancement operations for improving the quality of image for further processing. There might be chances of having duplicate frame or the image frame with statistical characteristics and needs to be identified in order to minimize overhead on frame analysis for the utterance. This was done by implementing Principal component analysis based feature extraction of frame and detecting duplicate frame from image vector. The visual feature extraction was carried out using incremental difference procedure, preparing and representing contours for analyzing of shape of the mouth. Audio-visual integration is an area for research. This chapter focused on an informative approach towards integration of audio and visual integration using MFCC components based recognitions. The details of the experiments designed as per the methodology will be discussed in chapter 04.
7 References


[02] A J Goldchen, O S Garcia and E D Petjan, Continuous automatic speech recognition by lipreading, chapter 14, pages 321-343, Volume 9 of Shah and Jain [01], 1997


[05] J R Movellan and G Chadderdon, Channel seperability in audio visual integration of speech: A Bayesian approach, In Stork and Hennecke [11], pages 473-487

[06] J R Movellan and P Mineiro, Modularity and catastrophic fusion: A Bayesian approach with application to audio visual speech recognition, A Technical Report 97.01, University of California, San Diego, Department of Cognitive science, Jan 1996


[20] CUAVE Database, Clemson university database for Audio visual experiments http://www.ece.clemson.edu/speech/cuave.htm