Shot Boundary Detection and High-level Features Extraction for the TREC Video Evaluation 2003

Xin Huang, Gang Wei, and Valery A. Petrushin
Accenture Technology Labs
161 N. Clark St.
Chicago, IL 60601, USA

Abstract

The paper describes approaches to shot boundary detection and high-level features extraction from video that have been developed at the Accenture Technology Labs for the TREC Video Evaluation 2003. For shot boundary detection an approach which uses the chi-square test for the intensity histograms of adjacent I-frames has been applied. Of seventeen features that have been suggested for the TREC Video Evaluation, three features were selected: “People”, “Weather news” and “Female speech”. For detecting the “people” feature an approach that is based on multiple skin-tone face detection has been used. The “weather news” feature has been detected using a sequence of simple filters that pass only segments of proper length with specific color distribution and having video text in specific locations. For detecting “female speech” feature an approach that combines speaker gender recognition using fundamental frequency distributions, skin tone based face detection, and moving lips detection using optical flow has been implemented.

1. Shot Boundary Detection Task

Segmenting video clips into continuous camera shots is the prerequisite step for many video processing and analysis applications. With the video compressed format MPEG dominating today, we developed a cut detection agent that works in compressed domain, i.e., it does not require the fully decompression of the video data, which significantly reduces the computation overhead. The cut detection is based on the method described in [1], which uses the Chi-square test for the three histograms (global intensity histogram, row intensity histogram and column intensity histogram) to evaluate the similarity between frames and find possible scene cuts.

Despite the variety of methods proposed for shot boundary detection, using histogram comparison is the most common approach. However, it is observed that sometimes scene cuts occur without causing significant changes in global intensity histograms between consecutive frames. To address this problem, in [1] two additional histograms has been introduced, namely row (horizontal) and the column (vertical) histograms. And the three histograms are used to further distinguish two categories of scene changes, namely abrupt cut and gradual transition.

As mentioned above, the algorithm works in compressed domain. In [1], only I-frames in the MPEG video are used to find the approximate location of the shot boundaries. Since I-frames are independently encoded and directly accessible from the MPEG data, using I-frames only can reduce the computations overhead. However as I-frame usually occurs every 12 or 15 frames, the algorithm [1] doesn’t give the exact frame number where the scene changes take place. We refined the algorithm as described below.

I-frames are encoded in the same format as JPEG specification, which is based on Discrete Cosine Transform (DCT). To compute the three histograms, the first coefficients of the 8 x 8 DCT encoded blocks are used. These coefficients represent the average block intensities. As in [1], the row and the column histograms of an I-frame with *MxN* DCT blocks are defined as:

$$X_i = \frac{1}{M}\sum_{j}^{M} b_{0,0}(i,j)$$

and

$$Y_j = \frac{1}{N}\sum_{i}^{N} b_{0,0}(i,j)$$

respectively, where $b_{0;0}\,(i;\,j)$ refers to the DC coefficient of a DCT block in row *i* and column *j*. The row and column histograms reflect the intensity distributions in the vertical and horizontal directions, and thus by combining these three measures, we can get more robust results, with higher detection rate and fewer false alarms.

The three histograms of the current I-frame are compared with those of the previous I-frame. The comparison is based the chi-square test, which is the most accepted test for determining whether two binned distributions are from the same source or not. Let HP_j and HC_j, respectively, represent the *j*-th bin of a histogram pair being compared, then the chi-square statistic is given as

$$x^2 = \sum_j \frac{(HPj - HCj)^2}{(HPj + HCj)^2}$$

Applying the chi-square test to the three pairs of histograms generates three distance values, which are then used to generate two comparison decisions. First, each value is compared against a threshold. When the value is greater than the threshold, the result is 1, and otherwise the result is 0. This produces three binary decisions, for the row, column and global histograms, respectively. If two or more decisions are 1, the first comparison result is 1, and otherwise 0. Let it be denoted by d_{maj}. The second comparison result is obtained by checking the average of the three distance values against another threshold. Let this result be denoted by d_{avg}.

When both d_{maj} and d_{avg} are 1, a hard cut is triggered. A gradual cut is detected whenever d_{maj} is 0, but d_{avg} is 1. When both d_{maj} and d_{avg} are 0, no shot boundary is indicated between the current I-frame and the previous one.

The method described above can tell the approximate location of shot boundaries as only I-frames are considered, which is not accurate enough for some applications. To find the exact frame number of the scene cuts, we applied post-processing. As the above method gives the range within which the scene cut occurs, we only need to search for within the range instead of go through all frames.

In this refinement step, abrupt cuts and gradual transitions are treated differently. The former always takes place between two adjacent I-frames. Therefore, the global histograms of all frames between these two I-frames are calculated, and we calculate the distance values between successive frames. The frame pair with the largest distance value is considered to be the location of shot boundary. Optical transitions could last several I-frames. We calculate the distance values of successive frames between the starting I-frame and ending I-frame. We observed that such distance values would increase when the gradual transition begins, reaching a plateau, and then drop as the transition finishes. Therefore, the frame pair where the distance value exceeds a threshold is considered to be the starting location of the gradual transition, and when the distance value gets below the threshold, the transition is complete.

The novelty of the system is a multi-resolution approach to location the exact frame where the scene cuts or gradual transitions take place. In the first step, only the similarity values between consecutive I-frames are calculated to find the approximate shot boundaries. Then, frames between a certain range are inspected one-by-one to find where exactly the shot boundaries are. By doing this we can achieve both high resolution of shot boundary locations and high processing speed.

The results of the system are checked with the video database from Trecvid and the result is very promising. Unfortunately, for some CSPAN videos, our video decoder inserted extra frames which introduce many errors. For the video clips where the decoder works correctly, the agent achieves high recall and precision. We plan to rerun the algorithm with another decoder to get its “real” performance.

2. Trecvid Feature extraction task

In a video database, various high-level semantic features such as “Indoor/Outdoor", "People", "Female Speech" etc., occur frequently. The semantic feature extraction will enable efficient multimedia query, multimedia content presentation and multimedia database management.

The basic unit of semantic feature extraction defined in Trecvid is video shot. Given a test video collection and the associated shot boundary reference, semantic features are extracted for each video shot. Trecvid provides a list of feature definitions for the feature extraction task and each feature is assumed to be binary, i.e., it is either present or absent in the given reference shot.

In our participation of Trecvid feature extraction task, we designed and implemented feature extraction algorithms for three semantic features, namely “People”, “Female Speech” and “Weather News”. For each feature extraction, there is a common preprocessing step to perform the temporal sampling. Because a video shot usually contains large number of video frames and there exists big redundancy among consecutive frames, temporal sampling can reduce the data size to be processed and thus make the algorithm more efficient both in time and space with little cost in feature extraction accuracy. Since we are dealing with MPEG 1/2 videos, the temporal sampling is achieved by taking into consideration only I-Frames of the MPEG video stream.

2.1 “People” Feature

The semantics of the “People” feature in Trecvid is defined as “segment contains at least THREE humans”. Detection and localization of a human in a specific environment can be achieved with high accuracy with help of predefined assumptions and specific knowledge. However, human detection in an unknown environment is much more difficult. For human detection, there are several cues we can use: shape information, skin color, human face and motion. In this paper, we proposed a people detection approach based on human face detection.

2.1.1 People feature extraction approach

For each I-frame of a given video segment, the face detection is performed to find human faces (including both front-view and side-view). If the number of detected face in an I-frame is no less then three, the feature “People” is triggered on this frame. After doing the feature extraction on all the I-frames, the ratio of the number of I-frames with feature “People” and the total number of I-frames of the given video segment is calculated. If the ratio is greater than a predefined threshold, we conclude that the feature “People” is detected on the given video segment. The ratio can serve as the confidence measure and can be used to rank all the video segments with the “People” feature found (Figure 1).

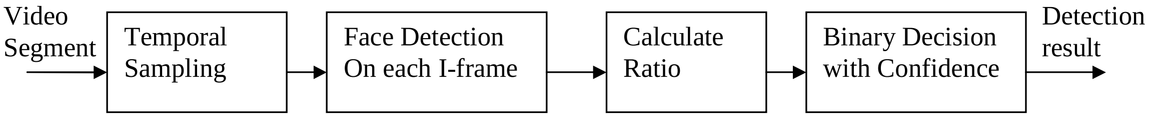
Video Segment
Temporal Sampling
Face Detection On each I-frame
Calculate Ratio
Binary Decision with Confidence
Detection result

Figure 1. Block diagram of the “People” feature detection

2.1.2 Face detection

A variety of face detection methods has been reported in the literature. The face detection methods can be assigned into one of the two categories: (i) *feature-based method*; and (ii) *classification-based method*. The feature-based methods search for different facial features and use their spatial relationship to locate faces [2, 3, 4, 5, 6, 7]. The classification-based methods detect faces by classifying all possible sub-images of a given image as face or non-face sub-images [8, 9, 10]. A more detailed survey of face detection systems can be found in [11].

We use the omni-face detection method proposed by Wei and Sethi [12]. A block diagram of the omni-face detection system is shown on Figure 2. It consists of a skin-tone filter followed by two largely independent processing modules, the frontal face module and the side-view face module, working in parallel. The frontal face module is responsible for detecting front view faces by analyzing regions of skin-tone pixels. The side-view face module operates on edge segments of the skin-tone pixel regions and is responsible for detecting side view faces.

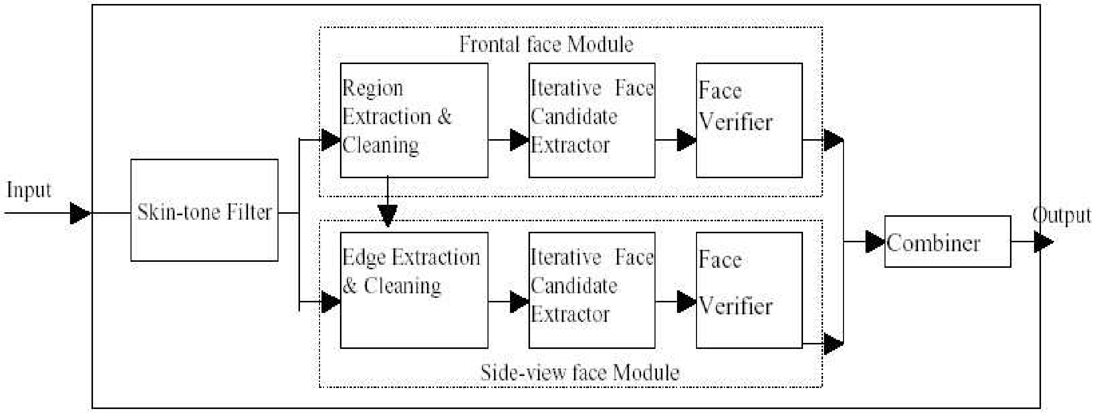
Frontal face Module
Region Extraction & Cleaning
Iterative Face Candidate Extractor
Face Verifier
Input
Skin-tone Filter
Combiner
Output
Edge Extraction & Cleaning
Iterative Face Candidate Extractor
Face Verifier
Side-view face Module

Figure 2. Block diagram of face detection

- **Skin-tone filtering**

It is well known that the skin tone color is distributed over a very small area in the chrominance space. The skin-tone filtering is to eliminate all those image pixels which are unlikely to present a human face. A half-ellipse skin-tone model in Y-I subspace of the YIQ color space is trained from a training set of images from different sources and serves as the classification model of skin-tone detection. The output of the skin-tone filtering is a binary image wherein the white pixels denote skin-tone in the original image.

- **Region and edge extraction and cleaning**

The purpose of this stage is to minimize the effect of noise, shading and illumination variations as well as to deal with the presence of skin-tone background and objects. The frontal face module applies morphology operations to eliminate all small isolated regions, fills small holes, breaks small region bridges, and erases thin protrusions. Since the side-view module operates on edge segments, it first performs edge detection on skin-tone regions.

- **Front-view and side-view face candidates selection**

Besides the skin-tone, a front-view face candidate also presents some other face characteristics such as certain region size and shape. An iterative approach is applied to determine frontal face candidates. It first analyzes the size and shape information for each region. If a region exceeds certain preset size and has a gross oval shape, it is retained as a candidate for further verification. Otherwise, it is partitioned into several sub-regions through an iterative region partitioning scheme based on *k-means clustering*. The partitioned sub-regions are once again subjected to size and oval shape test and the iterative partitioning is repeated, if necessary.

The most distinguishable features of a face viewed from a side are the protrusion of the nose and two minor dips to correspond to eyes and lips in the face profile. The side-view face module uses similarity measurement based on Normalized Similarity Value (NSV) to find groups of edge segments that resemble a predefined side-view face profile. Similar to the frontal face module, the side-view face module also does not discard the rejected regions but splits them for further testing to locate side-view faces.

- **Face verification**

The function of this stage is to look for more supporting evidence for regions or edge segments labeled as face candidates. The frontal face module looks for facial features like eyes, eyebrows, and lips within the candidate regions. A histogram-based thresholding is first applied to extract possible locations of facial features. Then, the frontal face candidate is either accepted or rejected based on knowledge of spatial relationship of facial organs and hole analysis.

For side-view faces, since facial feature patterns are not so salient due to the invisibility of both eyes, the approach for front-view face verification is no longer feasible. Instead, a classification model based on hidden Markov models (HMM) is used to do the side-view face verification. Two of hidden Markov models are trained from training set containing both real side-view faces and non-faces and then used to decide whether to accept or reject side-view face candidates.

2.1.3 Ratio calculation and binary decision with confidence

After performing the face detection on each I-frames with a given video segment, each I-frame is labeled with feature "People" or without feature "People" according to the number of faces detected. The ratio of the number of I-frames with feature "People" and the total number of I-frames within the video segment is calculated. If the ratio is greater than a predefined threshold, we conclude that the feature "People" is detected and the ratio can serve as the confidence which in turn can used to rank all the video segments with feature "People".

2.2 "Weather news" Feature

Weather news is usually highly artificially edited following some special patterns. Let's take CNN weather news for example. According to our observation, the CNN news video segments have the following patterns:

- **Color distribution**

The video frames of CNN weathers news have specific color distributions. Four most representative frames of CNN weather news video are shown on Figure 3.

- **Video text**

Each video frame contains some video text on specific location such as the text "Saturday" and "Extended Forecast" in the first figure as shown in Figure 3.

- **Motion**

The most common scenario of CNN weather news is as follows: a weather map is continuously displayed on the screen while a meteorologist is doing report on the background and there are only slight changes on the weather maps within a weather news segment. Therefore, the motion activity of a weather news segment is very small.

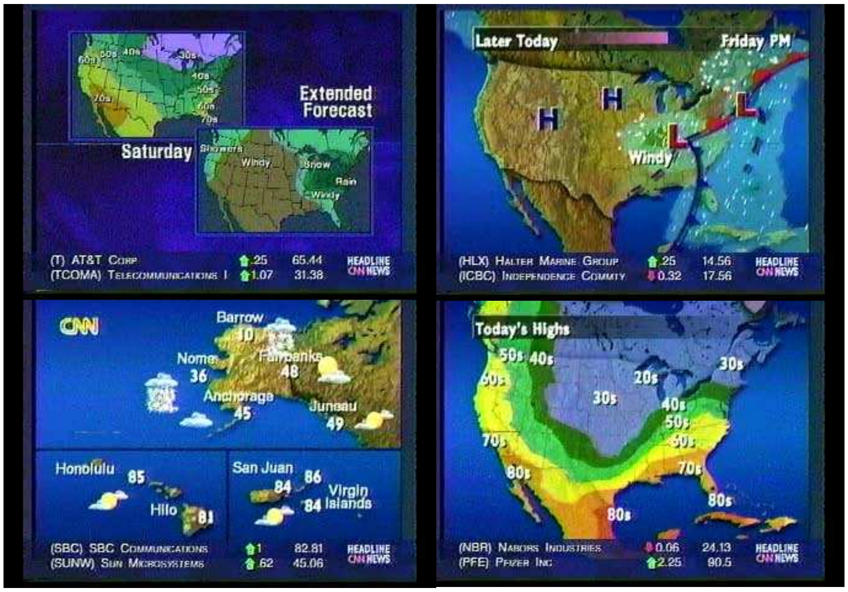

Figure 3. Four representative frames of CNN weather news video

- **Video segment length**

The length of a weather news segment is usually within a certain range. According to statistic on the Trecvid development videos, the range of the length of a CNN weather news segment is from 200 to 1000 frames.

2.2.1 CNN Weather news detection

As discussed above, CNN weather news contains certain patterns in color distribution, video text, motion and video segment length. We proposed a sequential pattern matching (filtering) approach for CNN weather news detection. Figure 4 shows the block-diagram of the proposed CNN weather news detection approach.

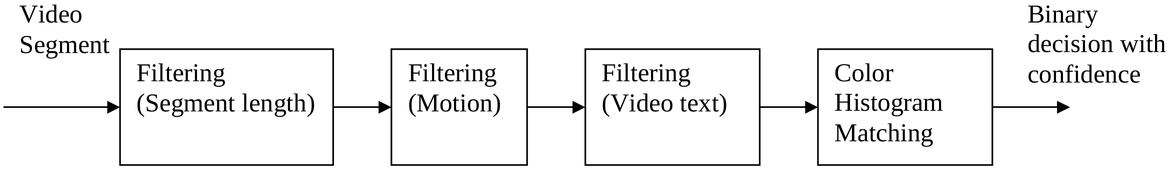

Figure 4. Block diagram of weather news detection

One important issue in our design of the sequential filtering structure is how to order those four difference filters. The guideline here is: 1) place the filter with less computation complexity in early stage and 2) place the stricter filter in early stage. By doing this, we can filter out the negative video segment as early as possible with least computation cost and thus make the sequential filtering approach more efficient in term of computation cost. On the other hand, since those four filters are independent, different ordering of them will not make any difference in terms of detection results.

The filtering based on the length of a given video segment is computationally cheapest and thus it serves as the first filter. Since the CNN weather news video segment has very small motion activity, we set very strict threshold for motion filtering. Therefore, it will eliminate most of non-weather-news video segments from further filtering. As to the video text filter, because our purpose here is not to precisely detect and recognize the video text, we only use a very simple thresholding method on the pixel intensity. Hence we put it before “Color Histogram Matching”, which is the most time-consuming one.

The output of the sequential filtering is a binary decision on whether the feature “Weather news” is detected on the given video segment. It also produces a pseudo-confidence along with the positive decision. The pseudo-confidence is actually the average distance between the I-frame and the template frames to be matched. In the following sections, each filter is described.

2.2.1.1 Filtering based on video segment length

As mentioned before, the length of a CNN weather news segment is usually within the range from 200 to 1000. If the length of a given video segment is outside of this range, it is rejected; otherwise, it is fed into the next filter.

2.2.1.2 Filtering based on motion

As mentioned before, the motion activity of a weather news segment is very small. Therefore, we can capture the motion information of a given video segment and test whether it is above a predefined threshold for making decision to reject this video segment or go on with next filtering step.

We use the difference between two consecutive I-frames to measure the motion. The difference of each pair of consecutive I-frames is measured as the summation of the magnitude of intensity difference of corresponding pixels. The average difference is calculated on the whole video segment. If the average difference is higher than a predefined threshold, the video segment is rejected; otherwise it fed into the next filter.

2.2.1.3 Filtering based on video text

As mentioned before, frames of a CNN weather news segment contains some video text at specific locations and hence video text can be used to detect potential CNN weather news. However, for the purpose of weather news detection, we do not need to precisely detect and recognize the video text and we also care about some specific locations. In CNN weather news, the video text is very bright. In other words, it has high values of R, G and B components. Therefore, a simple approach based on the pixel values is sufficient for our purpose. For each I-Frame, we detect those pixels with high RGB values at each specific location. If the ratio of pixels with high RGB values is among certain range, we label this frame as a weather news frame in term of the video text. For a given video segment, if the ratio of I-frames which are labeled as weather news frame is beyond a predefined threshold, then it is considered as a potential weather news segment to be processed with the next filtering step; otherwise, it is rejected.

2.2.1.4 Color histogram matching

The video frames of CNN weathers news have specific color distributions as shown in Figure 3. We can come up with a number of template weather news frames and match them with the video frames in a given video segment.

The color histogram in YUV color space is used to represent the color distribution. A video frame is first partitioned into four (2x2) sub-frames. For each sub-frame, its color histogram is calculated and the four color histograms are concatenated as the color histogram for the whole frame.

One issue need to address is how to generate the template frames (or we can say template color histograms). A number of most representative frames from weather news segments in Trecvid development video collections are selected to serve as the template frames. First, a number of CNN weather news segments are selected from the development video collection as training set. Then we use c-mean clustering method to do clustering on all the I-frames in the training set based on the color histograms. For each cluster, the average histogram (namely the cluster's representative color histogram) is calculated and serves as one template color histogram.

For each I-frame of a given video segment, its color histogram is first calculated and then its distance to each template color histogram is measured. The minimum distance among all the distances to different template color histograms is defined as that I-frame's distance to weather news template. The average distance of all the I-frame's distance to weather news template is calculated. If the average distance is beyond certain predefined threshold, the given video segment is rejected; otherwise, we detect the feature "Weather news" on the given video segment and the inverse of the average distance can serve as a pseudo-confidence to rank all the video segments with detected "Weather news" feature.

2.3 "Female speech" Feature

The feature "Female speech" is defined as a segment that "contains a female human voice uttering words during and the speaker is visible". The feature "Female speech" contains both female speech and visible talking face. To detect this feature, we have to combine both audio cue and visual cue; namely, we need to use multi-model approach. The proposed multi-modal approach is shown in Figure 5. It contains an audio module and a visual module which are running in parallel and the outputs are fused to produce "Female speech" detection results. Given a video segment, its audio signal is first extracted and speaker gender detection is used to identify female voice. At the same time, we can do face detection on the video frames to locate the face and extract facial features such the mouth location. But the face detection alone is not enough. We also need do detect whether the face is talking or not. Having the female voice information and talking face information, we can combine them to make decision.

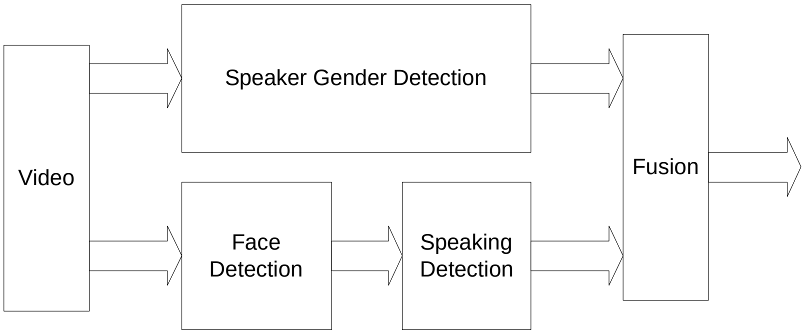

Figure 5. Block diagram of "Female speech" feature detection

2.3.1 Speaker gender detection

Fundamental frequency or pitch is the most obvious difference between the male and female voice. The average fundamental frequency for male voices varies from 90 to 140 Hz while the average for female voices lays in the range 130-250 Hz. A simple classifier can be build taking into account this difference in average pitch.

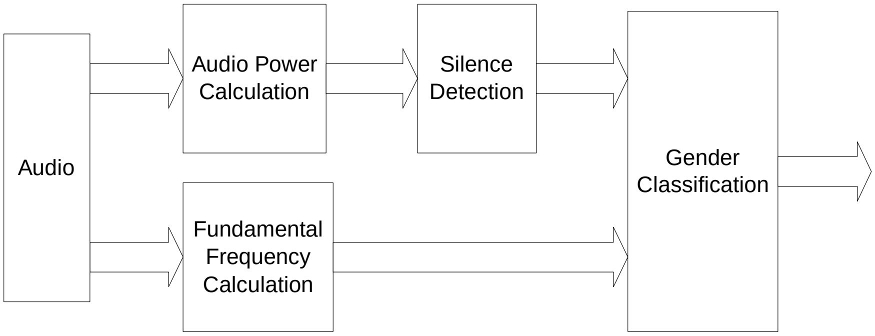

Figure 6. Block diagram of speaker detection

Figure 6 presents the block diagram of speaker gender detection. The audio signal is partitioned into a number of overlapping audio segment with certain window size which serves as the basic unit of the audio feature extraction. The audio power for each audio segment is calculated and to be used by the silence detection module. Since silent segment has low value of audio power, a predefined threshold can be used to identify silent and non-silent segments. For each non-silent segment, we also calculate its fundamental frequency. Since male voice and female voice have different ranges of fundamental frequency, it can be used to identify speaker gender. Instead of using a simple threshold method which gives hard binary decisions, the gender classification module produces soft decisions. The gender classification has two configuration parameters: *T* and *W*. If the fundamental frequency is less than *T-W*, the speaker gender is classified as male and if the fundamental frequency is larger than *T+W*, the speaker gender is classified as female. The confidence of female or male voice is linear with respect to the fundamental frequency as shown on Figure 7. The output of the whole speaker detection system is a confidence of an audio segment being female speech.

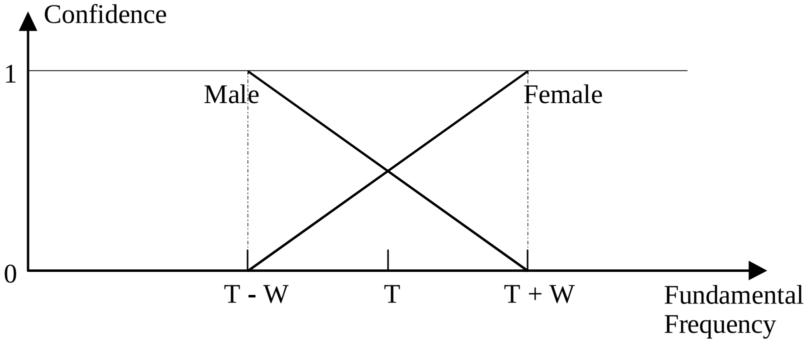

Figure 7. Confidence calculation for gender recognition

2.3.2 Talking face detection

As mentioned before, face detection is not enough for feature “Female speech” extraction, we also need do detect whether the face is talking or not. This can be done by mouth motion analysis. When human is talking, his upper lip and lower lip moves in opposite vertical directions. This specific mouth movement pattern can be used to detect whether a person is talking.

We can use optical flow to analyze the mouth movement. Optical flow [13] is the velocity field which warps one image to another as show in Figure 8. The leftmost picture and the rightmost picture show two consecutive frames of a video clip of a rotating sphere. And the picture in between visualizes the optical-flow field computed from these two frames. In optical flow field, each pixel has a motion vector which indicates the direction and magnitude of the pixel's movement.

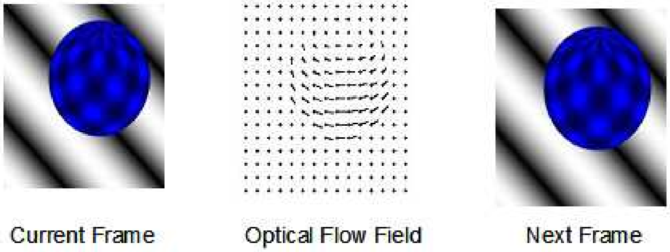

Figure 8. Optical flow illustration

Given a video segment, face detection as described above is performed on each I-frame. If a human face is detected in an I-frame with associated facial information such as mouth region, the optical flow within the mouth region is calculated on this I-frame and the next I-frame. Since our purpose is to detect opposite vertical movement of upper lip and lower lip, only the vertical component of calculated optical flow is taken into consideration. Because people usually move their heads while talking, which causes global vertical motion, we have to reduce the effect of global vertical motion by subtracting the average vertical component from each individual vertical component. Then the pixels with small vertical movement are filtering out by using a threshold. The mouth region is equally divided into two parts in vertical axis. Upper part represents upper lip and lower part represents lower lip. For each part, the ratio of moving up pixels is calculated. The same calculations are done for moving down pixels. Let us assume that R_{uu} and R_{ud} are the ratios of moving up/down pixels in the upper lip. Similar definitions apply to R_{lu} and R_{ld} for the lower lip. If $R_{uu} + R_{ld} > threshold$ or $R_{ud} + R_{lu} > threshold$, then the classifier concludes that a talking mouth is detected and $(R_{uu} + R_{ld})/2$ or $(R_{ud} + R_{lu})/2$ serves as the decision confidence. Otherwise, the talking mouth classifier outputs zero, which means no talking face has been detected.

2.3.3 Synchronization and Fusion

Because the speaker gender detection is based on audio signal and talking face detection makes use of visual information, we first need to synchronize the results of speaker gender detection and those of talking face detection and then fuse both of them to make final decision.

As mentioned before, the basic unit of audio signal is audio segment. Each audio segment is associated with a time interval. The basic unit of talking face detection is I-frame. Each I-frame is associated with a time-stamp. To do synchronization, we associate each I-frame with an audio segment based on the relationship between audio segment's time interval and I-frame's time stamp. If the time stamp of I-frame *A* is only within the time interval of one audio segment *B*, I-frame *A* is associated with audio segment *B*. If the time stamp of I-frame *A* is within more than one audio segments' intervals, it is associated with the audio segment whose central point of its interval is closest to I-frame *A*'s time stamp.

After synchronization, each I-frame has an associated audio segment. Then we need to fuse the talking face detection output on an I-frame with the speaker detection result on its associated audio segment. Since the outputs of both detection modules are confidence values, one possible fusion method is to do the weighed linear combination. However, it is not trivial how to set the optimal relative weights. To avoid the weight selection problem, we use the multiplication combination. In other words, we multiple the confidence from face detection and that from talking face detection. If the multiplication is less than a predefined threshold, we conclude that no "Female speech" is detected on an I-frame; otherwise, the decision is positive.

For each I-frame, we can do the fusion as described above. Then, the ratio of I-frames with detected feature "Female speech" is calculated. If the ratio is less than a predefined threshold, we conclude that no "Female speech" is detected on a given video segment; otherwise, the decision is positive and the ratio can be used as confidence to rank all the video segment with the "Female speech" feature.

References

1. Nilesh V. Patel and I.K. Sethi, "Video Shot Detection and Characterization for Video Databases", *Pattern Recognition*, Special Issue on Multimedia, Vol. 30, pp. 583-592, April 1997
2. S-H. Jeng, H.Y.M. Liao, C.C. Han, M.Y.Chen and Y.T. Liu, "Facial feature detection using geometrical face model: an efficient approach", *Pattern Recognition*, Vol. 31, No. 3, pp. 273-282, 1998.
3. . T. K. Leung, M.C. Burl and P.Perona, "Finding faces in cluttered scenes using random labeled graph matching", *Proceedings Fifth Intl. Conference on Computer Vision*, pp. 637-644, Cambridge, MA, June 1995.
4. A. Tankus, Y. Yeshurun and N. Intrator, "Face detection by convexity estimation", *Pattern Recognition Letters*, Vol. 18, pp. 913-922, 1997.
5. G. Wei and I.K. Sethi, "Face Detection for Image Annotation", *Pattern Recognition Letters*, Vol. 20, pp. 13-21, 1999
6. G.Z. Yang and T.S. Huang, "Human face detection in a complex background", *Pattern Recognition*, Vol. 27, No. 1,pp. 53-63, 1994.
7. K.C. Yow and R. Cipolla, "Feature-based human face detection", *Image and Vision Computing*, Vol. 15, pp. 713-735, 1997.
8. B. Moghaddam and A. Pentland, "A subspace method for maximum likelihood target detection", *Proceedings IEEE International Conference on Image Processing*, Washington DC, October 1995. 735, 1997.
9. H. A. Rowley, S. Baluja and T. Kanade, "Neural networkbased face detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 20, No. 1, p23-37, 1998.
10. K-K. Sung and T. Poggio, "Example-based learning for view-based human face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 39-50, 1998.
11. Rama Chellappa R. Chellappa, C.L. Wilson, S. Sirohey (1995). Human and machine recognition of faces: A Survey. *Proceedings of the IEEE*, VOL. 83, No. 5, p705-740
12. Gang Wei and Ishwar K. Sethi. "Omni-Face Detection for Video/Image Content Description", *International Workshop on Multimedia Information Retrieval, in conjunction with ACM Multimedia Conference 2000*, (MIR2000), Los Angeles, November 2000
13. Berthold Klaus Paul Horn. *Robot Vision*. The MIT Press, Cambridge, MA, 1986.