

Real-Time Automated Video and Audio Capture With Multiple Cameras and Microphones

Ce Wang (wangc@hrl.harvard.edu)

*Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138*

Scott Griebel (griebel@fas.harvard.edu)

*Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138*

Bo-June (Paul) Hsu (bohsu@microsoft.com)

Microsoft Corporation, Redmond, WA 98052

Michael Brandstein (msb@hrl.harvard.edu)

*Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138*

Abstract. This work presents the acoustic and visual-based tracking system functioning at the Harvard Intelligent Multi-Media Environments Laboratory (HIMMEL). The environment is populated with a number of microphones and steerable video cameras. Acoustic source localization, video-based face tracking and pose estimation, and multi-channel speech enhancement methods are applied in combination to detect and track individuals in a practical environment while also providing an improved audio signal to accompany the video stream. The video portion of the system tracks talkers by utilizing source motion, contour geometry, color data, and simple facial features. Decisions involving which camera to use are based on an estimate of the head's gazing angle. This head pose estimation is achieved using a very general head model which employs hairline features and a learned network classification procedure. Finally, a beamforming and postfiltering microphone array technique is used to create an enhanced speech waveform to accompany the recorded video signal. The system presented in this paper is robust to both visual clutter (e.g. ovals in the scene of interest which are not faces) and audible noise (e.g. reverberations and background noise).

Keywords: person tracking, talker tracking, acoustic localization, head pose estimation, speech enhancement, audio enhancement video conferencing

1. Introduction

The overall goal of the work at the Harvard Intelligent Multi-Media Environments Laboratory (HIMMEL) is to produce automated and relevant high-quality signal capture in environments populated with microphones and video cameras without the active supervision or distraction of its human users. In this work, we present the details of



© 2000 Kluwer Academic Publishers. Printed in the Netherlands.

our system for acoustic and visual-based tracking and audio signal acquisition of individuals in a practical room.

The ability to do real-time talker tracking and audio enhancement is an important requisite of many developing technologies, including video-conferencing, interactive human-computer communications, automated surveillance systems, and virtual reality interfaces. As an example application of this technology, consider video-conferencing. Until recently, multi-participant systems required a human camera operator to track speakers and select shots. Some of our earlier work involved the development of acoustically-based talker tracking which provided rudimentary automated camera control [1, 2]. A number of commercial video-conferencing systems now employ these and similar technologies. However, the reliance on acoustic data alone limits performance. The system described here fuses the video and audio data to dramatically improve source tracking capability, allowing for the effective isolation of one or more desired talkers within the video frame. The quality of the associated audio signal is also important. In addition to ease of use and information conveyance considerations, the quality of the speech signal plays a key role in the performance of related analysis systems such as speech recognition and speaker identification. To this end, microphone array methods are employed to reduce the effects of ambient noise, interfering sources, and room acoustics in the resulting audio signal.

The overall system is outlined in Figure 1. A microphone array detects active acoustic sources in the environment and provides initial talker locations to the video portion of the system. Video processing methods are then employed to allow the computer-controlled cameras to isolate the image of one or more individuals within a frame, adjust for head orientation and range, and to compensate for any source motion. Source tracking information is fed to the acoustic processing element to allow for enhancement of the appropriate audio signal. The system is designed to be robust over a wide range of acoustic and visual environmental conditions. The algorithms are computationally efficient, allowing for real-time operation on standard personal computer systems.

The remainder of this paper describes the algorithms employed in the current hybrid system and illustrates their functionality. Sections 2 and 3 discuss details of the acoustic and visual components, respectively. Section 4 presents some of the results achieved. Finally, Section 5 offers some conclusions and avenues for future work.

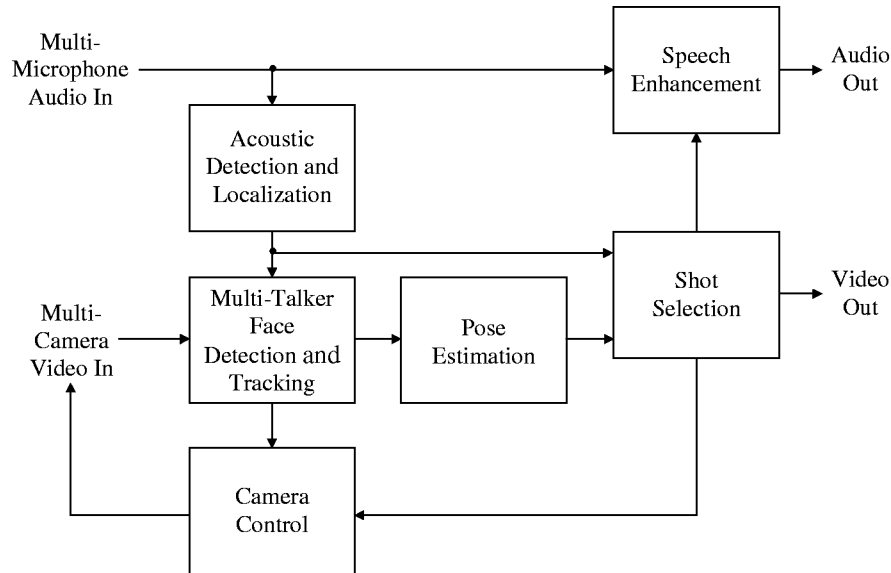


Figure 1. Outline of the overall system

2. Audio Processing

The goal of the multi-channel audio system is twofold:

1. to detect acoustic activity and determine the position of a talker based on the signals observed at microphones scattered throughout the room
2. to use the multi-channel audio data to produce an enhanced audio signal.

The positional information is used to determine the initial selection and steering of the appropriate video camera. The camera then takes over and tracks the active talker using the video techniques to be discussed in the following section. As time progresses, the source locator is used to detect the presence of new dominant speech sources, which lead to image reinitialization and possibly the selection of a new camera. An enhanced speech signal is estimated for the talker being tracked and accompanies the video signal. Figure 2 presents an outline of the acoustic processing used in the current HIMMEL system. The following sections discuss the source locator and speech enhancement techniques.

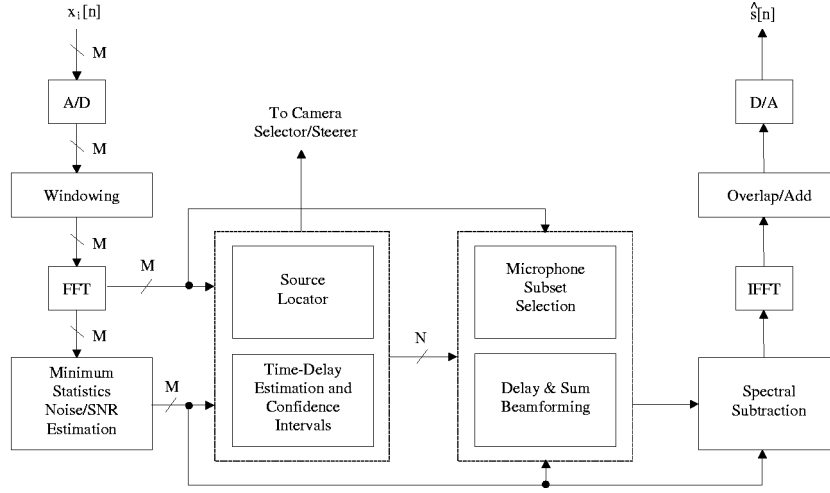


Figure 2. Audio system block diagram. M = number of microphones, $N = M/2$.

2.1. SPEECH SOURCE LOCATOR

In a room with M microphones, after analog-to-digital conversion, the audio signal observed at the i^{th} microphone, $x_i[n]$, is given by:

$$x_i[n] = h_i[n] * s[n] + u_i[n] \quad i = 1, 2, \dots, M \quad (1)$$

where $s[n]$ is the original speech utterance, $h_i[n]$ is the room impulse response from the speech source to the i^{th} microphone, and $u_i[n]$ is ambient noise present at the i^{th} microphone. In the current HIMMEL system, the microphone array consists of 16 microphones ($M = 16$) arranged in pairs positioned along the walls and in the ceiling panels of our lab room. See Figure 3 for details of the room layout. The acoustic conditions in this room are far from ideal. Computer fans and air ducts contribute to an ambient noise factor over 70 dBm. The floors are tiled and the walls are made of a non-absorbing material. Reverberation levels are on the order of 300 ms. Illumination is provided by fluorescent lighting mounted in the 9.5 ft ceiling.

Note that in a noise-free, non-reverberant room, the room impulse response can be written as $h_i[n] = g_i \delta[n - \tau_i]$ where τ_i is the time it takes for the signal to travel from the sound source to the i^{th} microphone via the direct path and g_i is the attenuation factor due to propagation in air. In this case, the observed audio signal can be written as:

$$x_i[n] = g_i \cdot s[n - \tau_i] \quad i = 1, 2, \dots, M. \quad (2)$$

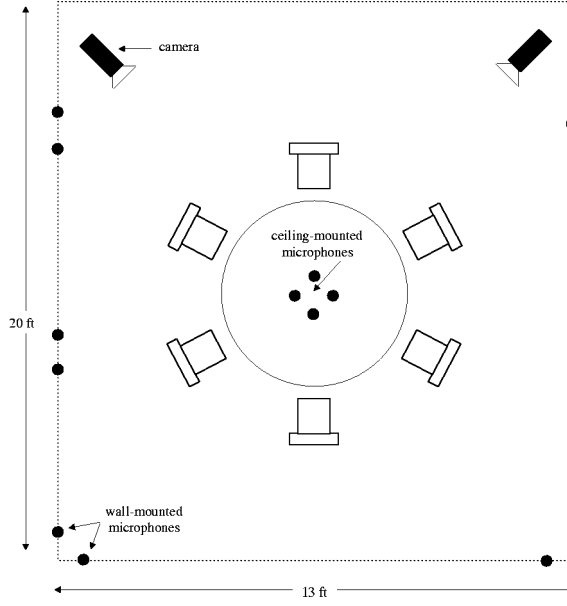


Figure 3. Overhead view of the room showing microphone and camera positions

However, in a real room, both convolutional noise (e.g. reverberations) and additive noise (e.g. background and fan noise) are present. This makes estimating the time delays, τ_i , a much more challenging problem.

The source localization procedure employed here is a two step process. A set of relative time delays is estimated for a number of spatially diverse pairs of microphones. These delays are then used to determine the source location. These two distinct steps are outlined below. Further details on source localization strategies in acoustic environments may be found in [1].

2.1.1. Time Delay Estimation

The relative time delay between a pair of microphones, i and j , is defined as $\tau_{ij} = \tau_i - \tau_j$. The traditional approach to estimating the relative time delays between two microphones is the generalized cross-correlation (GCC) technique [3]. This method forms a frequency-weighted cross-correlation between audio frames from two different microphones:

$$\psi_{x_i x_j}(\tau) = \int_{-\infty}^{\infty} W(\omega) S_{x_i x_j}(\omega) e^{j\omega\tau} d\omega \quad (3)$$

where $W(\omega)$ is a frequency-dependent weighting function and $S_{x_i x_j}(\omega)$ is the cross-power spectrum of the two observed signals. The resulting

time delay estimate is given by:

$$\hat{\tau}_{ij} = \arg \max_{\tau} \psi_{x_i x_j}(\tau). \quad (4)$$

When $W(\omega) = 1$, this leads to simply picking the lag associated with the cross-correlation's global maximum, the effectiveness of which is highly compromised by convolutional and additive noise. When $W(\omega) = 1/|X_i(\omega)X_j(\omega)|$, the resulting time delay estimator is called the phase transform. This method has proven to be more robust to real room conditions, particularly reverberations, than the simple cross-correlation method and has been used in several practical applications [4, 5]. However, the phase transform has proven to be of limited utility when reverberation and noise levels are sufficiently high [6, 7].

A number of more sophisticated time delay estimators have recently been developed. The current system makes use of the eigenvalue decomposition-based estimator presented by Benesty in [8]. Benesty noted that in the absence of additive noise,

$$x_i[n] * h_j[n] = s[n] * h_i[n] * h_j[n] = x_j[n] * h_i[n]. \quad (5)$$

The observation that $x_i[n] * h_j[n] = x_j[n] * h_i[n]$ allows one to estimate the relative time delay, τ_{ij} , by searching for the eigenvector of the covariance matrix for the two signals which corresponds to an eigenvalue of 0. An unconstrained frequency-domain least-mean-square (UFLMS) adaptive filter algorithm [9] was proposed for the simple calculation of the relative time delay.

In the current system, the discrete-time channel impulse response estimates of the Benesty algorithm are parabolically interpolated to increase the precision of the resulting time delay estimates. Moreover, a confidence measure is computed as the ratio between the values of the largest and second largest peaks. To minimize the effect of background noise, the instantaneous SNR is computed for each frequency band using a minimum statistics approach to determine the noise floor [10]. When implementing Benesty's algorithm for time delay estimation, our system only allows UFLMS adaptation to occur for frequencies where the SNR exceeds a minimum threshold. More generally, the adaptation convergence factor for each frequency band can be computed as a function of the frequency's SNR, frequency power, and the normalized convergence factor. This modification has been found to improve the overall time delay estimator performance as well as to prevent the source locator from focusing on noise sources in the absence of a dominating speech signal.

2.1.2. Position Estimation

Given the set of relative time delay estimates associated with a number of microphone pairs, the source location is calculated via the minimization of a least squares error criterion. Specifically, we denote the $N = M/2$ pairs of the M microphones as m_{i1} and m_{i2} for $i \in [1, N]$ and the ordered triplet (x, y, z) of their spatial coordinates as $\mathbf{m}_{i1}$ and $\mathbf{m}_{i2}$, respectively. In the current system the 16 microphones are positioned in $N = 8$ pairs spaced 42 cm apart. This corresponds to a maximum delay of approximately 10 samples using an 8 kHz sampling rate.

For each microphone pair, the relative time delay, τ_i , for a signal source located at the unknown position $\mathbf{s}$ has been estimated as described above. The true time delay associated with a source, $\mathbf{s}$, and the i^{th} microphone-pair is given by:

$$T(\{\mathbf{m}_{i1}, \mathbf{m}_{i2}\}, \mathbf{s}) = \frac{|\mathbf{s} - \mathbf{m}_{i1}| - |\mathbf{s} - \mathbf{m}_{i2}|}{c} \quad (6)$$

where c is the speed of propagation in air (≈ 342 m/s). In practice, τ_i is corrupted by noise and in general, $\tau_i \neq T(\{\mathbf{m}_{i1}, \mathbf{m}_{i2}\}, \mathbf{s})$.

The source location estimate, $\hat{\mathbf{s}}$, is found through the minimization of a least-squares error criterion:

$$J(\mathbf{s}) = \sum_{i=1}^N w_i \cdot [\tau_i - T(\{\mathbf{m}_{i1}, \mathbf{m}_{i2}\}, \mathbf{s})]^2 \quad (7)$$

where w_i is a weighting term derived from the confidence interval associated with τ_i . In practice this weighting term is used to discount potentially inaccurate delay estimates. Since $T(\{\mathbf{m}_{i1}, \mathbf{m}_{i2}\}, \mathbf{s})$ is a non-linear function of $\mathbf{s}$, the minimizer of J does not possess a closed-form solution and must be solved via an optimization method such as a gradient descent algorithm.

To resolve the changing speaker locations, the current algorithm resets all time estimates when a valid source position with low error has been determined. If the locator is unable to determine a low error position after a fixed amount of time, it assumes that the channel transfer functions associated with the time delay estimation procedure are most likely corrupt and again resets all time delays. Using this approach, the source position is only updated when a valid source position has been determined. If no position with low error is found, no new position is reported.

2.2. SPEECH ENHANCEMENT

Given a tracked source, the remaining function of the system's acoustic component is to produce an enhanced speech signal with reduced ambient noise and attenuation of interfering signals to accompany the video stream. A number of sophisticated beamforming methods are available for this endeavor, for example [11, 12, 13]. However, we have found that because of the disparate and sparse arrangement of our microphones within the enclosure and the directionality of the talkers, there is a large variance in signal quality between individual microphones. Rather than using the entire array, we found that better results are obtained by using only a small subset (in this case two) of the 'best' microphones to derive the enhanced signal. Only the microphone pair with the highest average instantaneous SNR is chosen. These two audio signals are first delay-and-sum beamformed to produce one channel of audio data which is then postfiltered using Spectral Subtraction [14] to further reduce the remaining additive noise.

Without loss of generality, we assume that microphones 1 and 2 have the highest SNR and are to be used for audio enhancement. Assuming no reverberation, uncorrelated ambient noise across channels, and equal propagation attenuation at the two closely-spaced microphones (i.e. $g_1 = g_2 = g$), the resulting beamformed signal, $x_b[n]$ is given by:

$$\begin{aligned}
 x_b[n] &= 0.5 \{x_1[n] + x_2[n - \tau_{12}]\} \\
 &= 0.5 \{g \cdot s[n - \tau_1] + u_1[n] \\
 &\quad + g \cdot s[n - \tau_2 - (\tau_1 - \tau_2)] + u_2[n - (\tau_1 - \tau_2)]\} \\
 &= g \cdot s[n - \tau_1] + 0.5 \{u_1[n] + u_2[n - (\tau_1 - \tau_2)]\} \\
 &\triangleq \tilde{s}[n] + \tilde{u}[n].
 \end{aligned} \tag{8}$$

The effect of this operation is to preserve the desired signal component while attenuating the background noise by a factor of two. The Fourier transform, $X_b(\omega)$, of the beamformed signal, $x_b[n]$, is:

$$X_b(\omega) = \tilde{S}(\omega) + \tilde{U}(\omega). \tag{9}$$

The Spectral Subtraction procedure involves modification of the magnitude component while leaving the phase term unaltered. Specifically:

$$\hat{S}(\omega) = [|X_b(\omega)| - \hat{U}(\omega)] e^{j\theta_{X_b}(\omega)} \tag{10}$$

where $\hat{S}(\omega)$ is the estimate of the clean speech spectrum, $\hat{U}(\omega)$ is the estimate of the background noise spectral magnitude, and $\theta_{X_b}(\omega)$ is the phase response of $x_b[n]$. Normally, one tries to estimate the noise

spectral magnitude during silence intervals, but this requires the implementation of a voice-activity detector - a non-trivial task. However, our system already has the noise power spectrum estimate available from the minimum statistics procedure incorporated into the time delay estimator. This data may also be used for Spectral Subtraction purposes. Additional steps were taken to reduce the tonal artifacts that are characteristic of spectral subtraction algorithms. A number of such steps are outlined in [14].

In general, Spectral Subtraction produces unnatural sounding artifacts in the enhanced speech for situations where the background noise level is comparable to that of the original speech signal. Beamforming prior to the Spectral Subtraction operation serves two purposes. It improves the received signal SNR and mitigates many of these audible effects. Additionally, the averaging process associated with the delay-and-sum reduces the effects of room reverberation, particularly those long-term echoes which tend to be uncorrelated across channels. Ideally, incorporating all 16 microphones into the beamforming procedure would be beneficial for both of these considerations. However, we have found that because of the large discrepancies in signal quality across the individual microphones, using only the two 'best' array elements produces a better sounding enhanced signal than that which is achieved by beamforming with the entire set of microphones. This last result is primarily due to the sparse placement of the microphones within the room. In this case, the array elements were positioned with localization considerations in mind. As would be intuitively suggested (and quantified in [15]), localization accuracy is improved by having a general sampling of the environment, rather than placing all or the majority of the microphones in one location.

2.3. AUDIO HARDWARE OVERVIEW

The current system consists of an array of 16 electret condenser omnidirectional microphones (Panasonic WM-034BY), preamplifiers and microphone powering circuits, A/D and D/A converters, a fiber-optic I/O sound card, and a personal computer. For the analog-to-digital conversion, we used Frontier Design Group's Tango 24, an 8-channel A/D, 8-channel D/A converter sampling at 44.1 or 48 kHz. Alternatively, the sampling rate can be determined by an external word clock input via a RG-59 coaxial BNC cable. For our application, 16-bit quantization is sufficient, although the Tango can also support 24-bit quantization.

The Tango takes input via 1/4" stereo audio jacks (TRS) for balanced input. The output is transmitted via fiber-optic cables (light pipes) to the computer in the Alesis ADAT format. By performing

the A/D away from the computer, we can minimize the amount of noise in the signal, as the computer is a significant noise source, both acoustically and electrically.

To interface the computer with the fiber-optic cables, we use the Sonorus StudI/O PCI card. With its chameleon feature, we configured the StudI/O to behave as a 16-channel audio input device in order to capture 16 synchronous channels of audio.

The StudI/O can lock onto the signal in the fiber-optic cable or use its internal 44.1 or 48 kHz clock. Unlike typical PC sound cards, it does not support lower sampling rates. Thus, in our application which requires only 8 kHz sampling, we decimate from the 48 kHz sampling rate with software.

The personal computer used was a Pentium III 550MHz computer with 128MB RAM running Windows 98 and the latest StudI/O driver. With this setup, the software system can process all 16 channels of audio input and determine the source location in real-time without any frame loss.

3. Video Processing

The visual part of the system consists of a set of mobile communication video cameras (Canon VC-C3), each with three degrees of freedom (pan, tilt, and zoom) which may be controlled remotely via a computer interface. The current system consists of two cameras positioned as shown in Figure 3. Given the initial camera steering provided by the acoustic localization process, the goals of the video component of the system are to:

1. detect and isolate the talker's head within the acquired image
2. track individuals as they move
3. evaluate the head orientations

The first two items require a feedback control mechanism that allows the cameras to be continually steered to keep the individual in the center of the frame with the appropriate zoom. The head orientation information is used in conjunction with the acoustic localization data to select the camera with the best view of the active talker. The following sections will discuss details of the face detection, source tracking, pose estimation, and camera control procedures.

3.1. FACE DETECTION

Face detection and tracking algorithms may be roughly categorized by their processing requirements. A variety of approaches are available for offline processing, such as neural networks [16], template matching [17], and statistical shape analysis [18]. While successful, these methods tend to be of a computational complexity that precludes real-time application or requires some form of interactive initialization which limits their practical utility. Thus, real-time face trackers have relied primarily on skin color information to limit their computational complexity [19, 20, 21]. However, face detection based on color regions alone is sensitive to many factors, such as variations in lighting conditions, skin color, and the background image.

As an alternative to these methods, we employ a hierarchical procedure which uses a variety of features that are easily extracted from the image stream. The result is a multi-person face detection procedure that is robust to a range of environmental conditions while only requiring a limited amount of computation. In order of application, the utilized features are: (1) source motion, (2) contour geometry, (3) color statistics, and (4) facial analysis. Source motion and contour geometry identify potential face regions while color and facial analysis are used as validation criteria. Each of these elements are discussed below.

3.1.1. *Stage 1: Source Motion*

Humans in their natural state are rarely motionless. While the degree of movement can vary significantly, our experiments show that even under relatively sedentary conditions, source motion is a sufficient indicator for distinguishing possible human bodies from the background environment. This observation is used as the initial step of an effective and computationally simple face detection algorithm.

Motion detection is achieved through the process of inter-frame differencing, the goal of which is to detect image differences in two consecutive frames of a visual stream. It is assumed that the camera is motionless during the analysis period. However, in what follows it would be possible to compensate for known camera movements or not process images acquired while the camera is moving. A color transformation is first performed on the image data to reduce the color space dimension while preserving the intensity information. Given a color pixel triplet (R, G, B) , the transformed color pair (r, g) is found from:

$$\begin{aligned} r &= \frac{R}{R + G + B} \\ g &= \frac{G}{R + G + B}. \end{aligned} \tag{11}$$

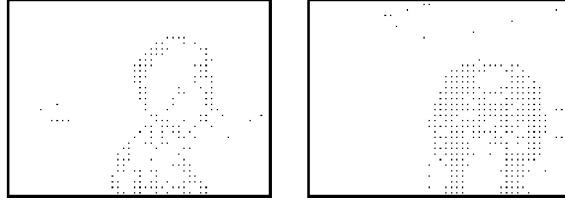


Figure 4. Raw motion plots for individuals in profile and head-on orientations

These transformed pixels are then used to generate a motion plot for frame n , $\mathcal{S}_n$, via inter-frame subtraction followed by thresholding:

$$\mathcal{S}_n(X) = \begin{cases} 1, & (|r_n(X) - r_{n-1}(X)| + |g_n(X) - g_{n-1}(X)|) > T_m \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where X is the set of points within the image, T_m is the motion threshold, and the subscripts n and $n - 1$ are used to denote the current and previous frames respectively. Essentially, image samples that are constant across frames are assigned a value of 0, while image samples that change across frames are set to 1. Figure 4 shows a pair of raw motion plots derived from this inter-frame differencing process. The following algorithms will mainly rely on these motion plots. Unless otherwise specified all computations are conducted on them.

As would be expected, despite the threshold applied at the inter-frame differencing, there are a number of noise samples in the motion plot $\mathcal{S}_n(X)$. These may be attributed in part to high light reflections, movements in the background, or fluorescent lamp or video monitor flicker. Thus, before further processing, a neighborhood checking algorithm is applied to remove isolated noise samples. This denoising procedure is conducted as follows:

For each point X , s.t. $\mathcal{S}_n(X) = 1$, look at its neighborhood $\mathcal{N}(X)$ for all the points $\{Y_i\}$ such that $Y_i \in \mathcal{N}(X)$ and $\mathcal{S}_n(Y_i) = 1$. If $\|\{Y_i\}\|$ is below a threshold, where $\|\{Y\}\|$ is the cardinality of set $\{Y\}$, then discard this point, i.e. set $\mathcal{S}_n(X) = 0$. Otherwise, keep the point.

This method is efficient for isolated noise spot removal and is faster than many traditional techniques, like morphological erosion [22].

The denoised motion plot is then segmented into a number of potential source regions each of which may contain an individual. The denoised motion plot, $\mathcal{S}_n$, is first projected vertically onto the x -axis yielding a 1-D function $P_{\mathcal{S}_n}(x)$. By clustering $P_{\mathcal{S}_n}(x)$, the motion plot $\mathcal{S}_n$ is segmented into several vertical bands. Within each band, $\mathcal{S}_n$ is further projected horizontally onto the y -axis giving $P_i(y)$, where i

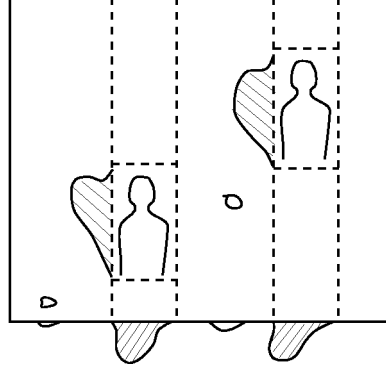


Figure 5. Segmentation of multiple moving people.

is the band index. Again clustering on $P_i(y)$, the individual motion regions of $\mathcal{S}_n$ are discriminated. The segmentation procedure is illustrated in Figure 5. At this initialization stage, it is assumed that each region may contain a person. The remaining stages are applied to each detected region to determine its content.

3.1.2. Stage 2: Contour Geometry

Once the motion plots have been denoised and clustered, an adaptive edge tracking algorithm is employed to delineate the outline of the objects in each region and perform a simple analysis of their general shape. The following procedure describes the procedure for detection of the left side of the potential body contour. Refer to Figures 6 and 7. The analysis for the right side is similar.

1. In $\mathcal{S}_n$, compare the leftmost and rightmost points across horizontal scan lines. A region of consecutive lines with consistent leftmost and rightmost pairs is used to detect a starting position. The horizontal line at the boundary of the consistency region is denoted as the starting line, y .
2. Save the leftmost point of line y as $EdgeLeft[y]$.
3. Continue scanning downward across horizontal lines searching for the point on $y + 1$ nearest to $EdgeLeft[y]$. Save this point as $EdgeLeft[y + 1]$. Identify the nearest left neighbor to $EdgeLeft[y + 1]$. If this neighbor exists and its separation from $EdgeLeft[y + 1]$ is within a threshold T_l then update $EdgeLeft[y + 1]$ to be the neighboring point. Continue this process until no other neighboring points within the threshold, T_l , are found. If the current scan line is empty or has only one point, go on to the next line, and then interpolate the above empty lines.

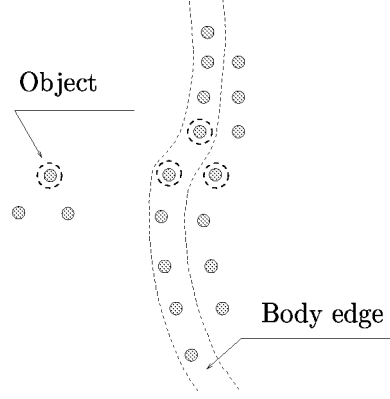


Figure 6. Finding edges

4. Track the values $\Delta x_l[y] = EdgeLeft[y + 1] - EdgeLeft[y]$. To improve stability, additional terms may be employed to calculate $\Delta x_l[y]$. For example, $\Delta x_l[y] = a_1 EdgeLeft[y+1] + a_2 EdgeLeft[y] - a_3 EdgeLeft[y-1] - a_4 EdgeLeft[y-2]$. A sudden deep valley followed by a sharp peak in $\Delta x_l[y]$ is indicative of an excessive T_l value. In such a case, decrease T_l , go back to the valley, and trace the edge again. Similarly, a sharp upward sloping in $\Delta x_l[y]$ is due to an overly small T_l value. Go back to that position, increase T_l , and trace again. This process is illustrated in Figure 7.
5. After completing the lower portion of the contour, return to the starting line and apply the same procedure to the region above the starting line.

The above adaptive edge detection method, in addition to detecting the desired body contour, is capable of removing noise and any objects reasonably separated from the object of interest. Figure 8 illustrates the edge plots derived from individuals in head-on and profile orientations.

The analysis of the derived edge plots is motivated by the observation that the neckline is a prominent feature of the body contour. As Figure 9 illustrates, the neckline is present across a variety of talker positions, orientations, and gestures, and may be roughly identified from the points on the body contour exhibiting the greatest concavity. The relative ease of estimating the neckline provides a straightforward method for detecting the presence of a body in the region of interest and for identifying the separation between the head and torso, thereby localizing the facial region.

The relation between the neckline and the nearby shoulder and head is shown in Figure 10. The points of peak concavity and convexity are found from the extrema of the second order derivative of the edge

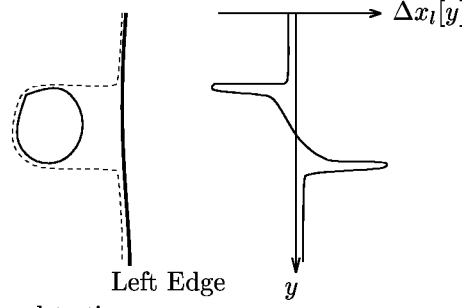


Figure 7. Adaptive edge detection

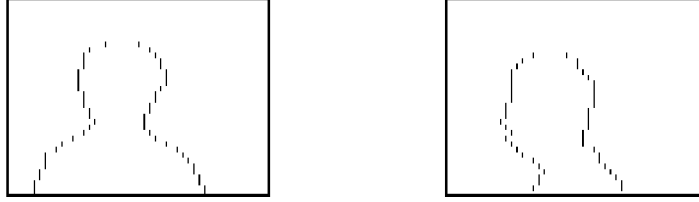


Figure 8. Edge plots constructed for individuals in head-on and profile orientations

contour. Because the second order derivative is sensitive to noise, the contour curve is first smoothed with a low pass filter. To improve the estimate reliability, the distances and heights of the extrema, as defined in Figure 10, are incorporated into the following criterion for estimation of the neckline:

$$y_{neck\ point} = \arg \max \{ (h_l h_r)^{w_1} (d_l d_r)^{w_2} h_m^{w_3} \}. \quad (13)$$

This criterion represents the geometry character of the edge curve near the peak point. The three weights (w_1, w_2 , and w_3) determine the influence of the geometric parameters. Through experimentation, the weight values were selected as: $w_1 = 1$, $w_2 = 2$, $w_3 = 1.8$. The distances between the extrema tend to be more consistent than the other statistics and are given higher significance. The curvature at the hypothesized neck point is weighted more highly than the variations due to the shoulder and head which tend to be more dependent on the specific orientation of the individual. The proximity of the potential neckline points for both the left and right contours is evaluated. If they are sufficiently close, the candidate region is maintained for additional analysis. Otherwise the region is discarded.

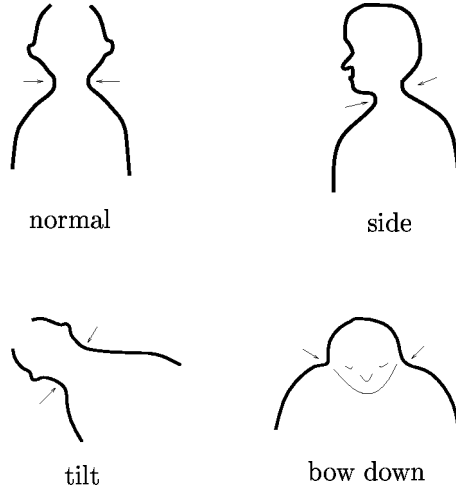


Figure 9. Neck lines for various positions

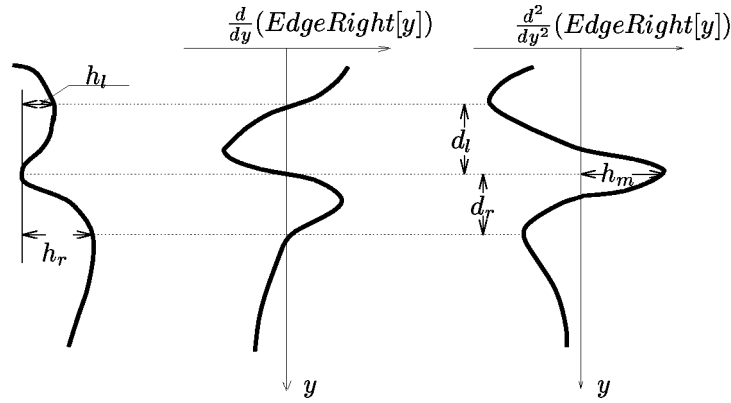


Figure 10. Illustration of neck line detection by second order derivatives

3.1.3. Stage 3: Color Statistics

The motion and contour analyses described above are insufficient on their own to differentiate a human body from a moving object of similar shape. The color content of the potential head region is now employed as a means of validating the presence of a face.

Color-based tracking systems typically rely on the assumption that skin colors are isolated to a specific region of some color space. While a number of color space mappings for skin/non-skin discrimination are available in the literature [19], a further study of appropriate color transformations was performed here. The results below show the prospects of distinguishing skin colors using only a single dimension of the transformed color space.

The new color space, $YI'Q'$, represents a transformation of the standard NTSC transmission system. Y , I' , and Q' are a set of orthogonal axes with the Y axis identical to the luminance (or color intensity) component of the NTSC system. The $I'Q'$ plane is perpendicular to the Y axis and represents the color hue and saturation (chrominances). The I' (or Q') axis is defined by its rotation angle with respect to Y . This angle is determined by projecting the color vectors of a training picture set onto the $I'Q'$ plane, such that the projected skin colors may be discriminated from non-skin colors to the greatest extent.

Our experiments show that with a properly defined $I'Q'$ plane, the projected skin colors can be constrained to a limited interval in the I' dimension. The decision interval boundary is purposely made general, allowing for a number of false positive skin color decisions. This is reasonable given that the color discrimination feature employed at this stage of the algorithm is used only to confirm that an identified potential face region contains the appropriate color statistics. Although this boundary may be broken under unusual lighting conditions, our experiments have shown that it can capture skin colors in most cases. The color validation test requires developing histogram data in a one-dimensional space and is therefore computationally straightforward to implement.

Figure 11 illustrates an example of the color confirmation scheme. Plot (A) shows the original image of a face surrounded by a complex background. Plot (C) gives the color distribution as projected onto the $I'Q'$ plane. Skin colors in the face region are projected as o's while all other pixels are represented as +s. Although the lighting on the face is poor, the face color is clearly distinguishable from the background in the I' (vertical) dimension. Plot (B) shows the image filtered with a simple threshold of $14 < I' < 40$. A significant portion of the head region is identified as containing skin color.

This color statistics test is applied only to those regions which have been validated by the prior stages. Based on this test, any candidate regions which do not possess a sufficient fraction of skin-like pixels are discarded.

3.1.4. *Stage 4: Facial Analysis*

Even with the contour geometry analysis employed in stage 2 and the color confirmation test of stage 3, it is still possible to confuse a face with a similarly shaped, skin colored object (e.g. a waving hand). Therefore, as a final validation procedure, a facial feature analysis is performed on the candidate region. A variety of feature detection techniques of differing complexity are available for this purpose. To avoid excessive computation, a simple but prominent facial feature is

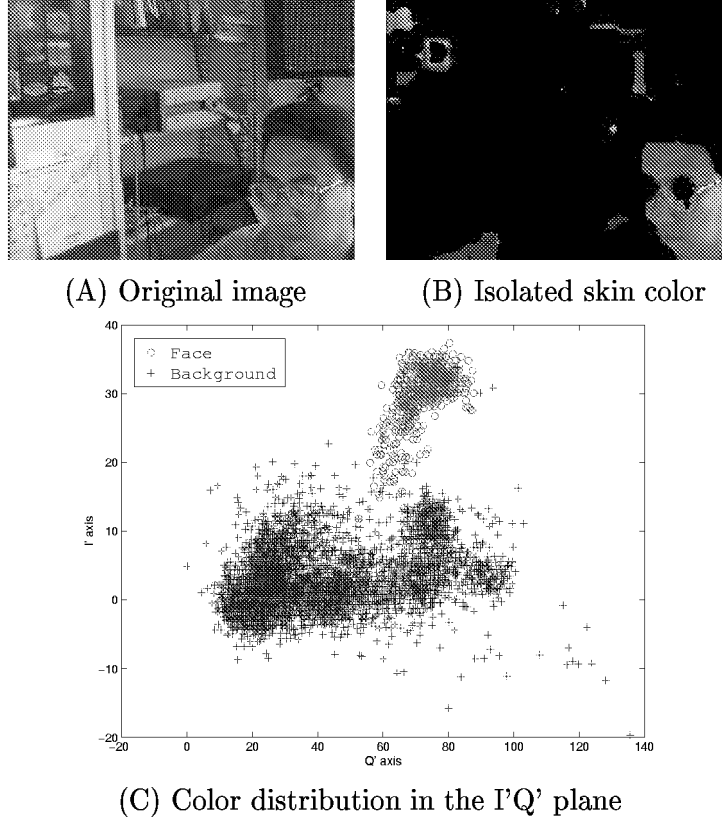


Figure 11. Color discrimination test

exploited in this work. Observation of Plot (B) in Figure 11 reveals that the eyes, eyebrows, and possibly eyeglasses are not identified as possessing skin colors while the forehead above and the cheek area below present integrated, skin-colored regions. This observation is the basis of a simple facial feature detection scheme. By employing a spatial histogram analysis of the candidate head region, it is possible to detect the presence of an eye region appropriately located within the face. This procedure is general enough to be robust to a large variety of orientation angles, and therefore does not require a front view in order to detect the faces.

An example of the entire face detection procedure applied to a multi-person scenario is provided in the results section to follow.

3.2. SOURCE TRACKING

Once a face has been detected, a tracking algorithm is activated to predict the position and velocity of the target in the subsequent frames. The predicted position estimate is used to reduce the tracking search region, thereby decreasing processing requirements and providing a further resistance to noise. A very general model of source motion is given below, followed by a description of the visual tracking procedures.

3.2.1. Mathematical Model of Source Motion

A second order dynamic model is used to predict the source motion. The process noise is assumed to be symmetrically distributed. Model accuracy is evaluated through a simple statistical analysis.

Denote the measured position, velocity, and acceleration of image samples of the current frame, t_n , to be s_n , $\dot{s}_n$, and $\ddot{s}_n$, respectively. Note that these equations must be estimated for motion in both the x and y directions. These values may all be calculated directly from image samples from the current and past frames. Assuming knowledge of the past n frames of image data $(t_0, \dots, t_{n-1})$, the current state values of the motion features ($\tilde{s}_n$, $\tilde{\dot{s}}_n$, and $\tilde{\ddot{s}}_n$) can be calculated from ensemble statistics as follows:

$$\begin{aligned}\tilde{\ddot{s}}_n &= \sum_{i=0}^n k_i \ddot{s}_i \\ \tilde{\dot{s}}_n &= \sum_{i=0}^n k_i \dot{s}_i + \tilde{\ddot{s}}_n \sum_{i=0}^n k_i \Delta t_i \\ \tilde{s}_n &= -\frac{1}{2} \tilde{\ddot{s}}_n \sum_{i=0}^n k_i \Delta t_i^2 + \tilde{\dot{s}}_n \sum_{i=0}^n k_i \Delta t_i + \sum_{i=0}^n k_i s_i\end{aligned}\quad (14)$$

where $\sum_{i=0}^{n-1} k_i = 1$, $k_0 \geq k_1 \geq \dots \geq k_{n-1}$, and $\Delta t_i = t_n - t_i$.

The sample position in the following frame is predicted from:

$$s'_n(t) = \frac{1}{2} \tilde{\ddot{s}}_n (t - t_n)^2 + \tilde{\dot{s}}_n (t - t_n) + \tilde{s}_n \quad (15)$$

where t it is the time measured when the next frame is available.

The confidence of the prediction is evaluated from the mean square error between the measured sample positions and the state value posi-

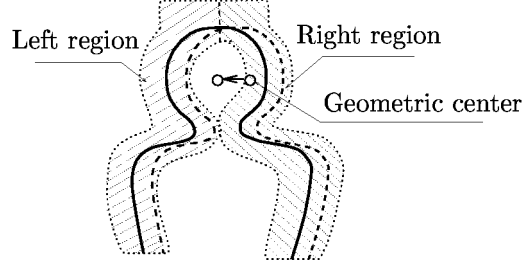


Figure 12. Illustration of edge searching region

tions for the current and past n frames:

$$\begin{aligned}
 \mathcal{E}_{\tilde{s}} &= \left(\sum_{i=0}^n k_i (s_i - \tilde{s}_i)^2 \right)^{\frac{1}{2}} \\
 &= \left(\sum_{i=0}^n k_i \left(s_i - \tilde{s}_i + \tilde{s}_i \Delta t_i + \frac{1}{2} \tilde{s}_i \Delta t_i^2 \right)^2 \right)^{\frac{1}{2}} \\
 &= \left(\sum_{i=0}^n k_i \left(s_i - \tilde{s}_i + \tilde{s}_i \Delta t_i - \frac{1}{2} \tilde{s}_i \Delta t_i^2 \right)^2 \right)^{\frac{1}{2}}. \quad (16)
 \end{aligned}$$

3.2.2. Adaptive Face Tracking

Typically, head motion in space is very nonlinear, involving sudden accelerations, rotations, and changes in orientation. The corresponding image plane projection tends to be extremely complex. However, if the frame rate is sufficient, the shape differences between frames are small. This enables the use of inter-frame coherence and allows for the simplification of head image movement to a 2-D displacement according to the mathematical model described above.

The head edges are predicted to be in two regions as shown in Figure 12. The medial axis of each search region is estimated from the position prediction equation (15). The width of the region, W , is determined adaptively from the edge contour prediction accuracy statistic (16) by:

$$W = A\mathcal{E}_{\tilde{s}} + T_W \quad (17)$$

where T_W is a constant value designed to maintain a minimum width in the search regions and A is an adaptive factor which is updated according to the success or failure of the edge detector in the predicted regions.

3.2.3. *Motion/Rest Decisions*

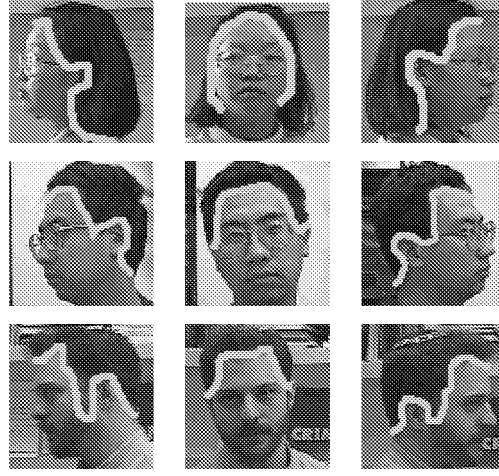
In situations where the source possesses only minimal motion, the edge detection and contour analyses detailed above are unreliable and susceptible to noise. A motion/rest decision is therefore required to detect this condition. The motion test employed here relies on edge continuity as the major criterion. A solid piece of edge is searched along the detected contour from the top. Here the term “solid edge” means that a piece of edge is continuous without big jumps or empty lines before interpolation. If a given length of solid edge is not detected within the predicted search region and the number of points in the motion plot is below a preset threshold, then the source’s motion is minimal and a rest decision is made. Neckline detection is deferred and the state from the previous frame is maintained without updating the tracking search regions.

3.3. HEAD POSE ESTIMATION

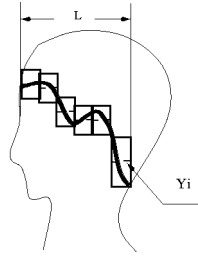
Given the data provided by the face detection and tracking procedures outlined above, we now address the problem of automatically estimating an individual’s head orientation. This information is useful for determining shot relevance and deciding which camera best captures the scene of interest. A number of methods for facial orientation or head pose estimation have been proposed in the literature [23, 24, 25]. The general approach involves estimating the positions of specific facial features in the image and fitting this data to a head model. Many of these methods still require manually selecting feature points and assume near-frontal views and high-quality images. For our applications, such conditions are usually not satisfied. Poor lighting conditions and hidden or unclear facial features are common problems. This has necessitated the use of only facial features that may be robustly estimated given the video quality available. Details of the feature extraction and orientation estimation procedures are given below.

3.3.1. *Feature Selection and Detection*

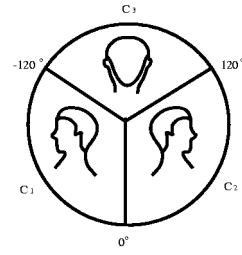
In situations where most of the fine facial details (e.g. eyes, lips, and mouth) are difficult to estimate reliably, we focus on a more overt facial property, namely the border between an individual’s hair and skin, as the major feature for estimating head orientation. The hair-skin border is straightforward to estimate across a wide range of viewing angles and is relatively insensitive to lighting conditions and poor image quality. As will be shown, the curve shape of the hairline is sufficient for estimating head orientation to a satisfactory degree and is a robust and consistent statistic across a variety of people and hairstyles. Figure 13(A)



(A)



(B)



(C)

Figure 13. (A) Hairline patterns, (B) hairlines as a feature, and (C) subset classification

illustrates the hairline contour derived from three people in different orientations.

Estimation of the hairline contour is performed in conjunction with the head tracker. For the detected head regions, hairlines are generated by clustering brightness, color, and connectivity into hair and skin regions. Having the pre-segmented head region available significantly simplifies and improves this process. Once estimated, the hairline is parameterized into a simple feature vector. Referring to Figure 13(B), denote the extent of the hairline along the horizontal axis as L . Along L , the hairline is evenly segmented into 6 pieces. The average vertical position of the contour segment for each piece, Y_i , is normalized to form the 6-point feature vector, $\vec{y}$, whose elements are given by $y_i = [Y_i - (Y_3 + Y_4)/2] / L$, ($i = 1, \dots, 6$). The purpose of this normal-

ization is to make the feature vector invariant to the head size and relative frame position.

3.3.2. Orientation Estimation by Support Vector Machine Regression

The hairline contours shown in Figure 13 display specific patterns for various facial orientations. The problem of orientation estimation becomes one of finding the mapping function $f : \vec{y} \rightarrow \theta, \theta \in [-\pi, +\pi]$, where θ is the facial orientation. Given a training data set consisting of P vector-orientation pairs $\{\vec{y}_i, \theta_i\}$, the mapping function is locally specified from linear projections associated with specific training data subsets. Specifically, the training samples are first partitioned into three classes C_1, C_2 and C_3 , based upon their coarse orientation angles as in Figure 13(C). These classes roughly correspond to right, left, and rear facing head orientations. For each class and its corresponding subset of training vector indices, V_i , the local orientation estimate is found from $\hat{\theta} = \vec{w}_i^T \vec{y} + \beta_i$ which is the mapping function for linear Support Vector Machine (SVM) regression. The mapping function is obtained by solving the following optimization problem:

$$\min_{\vec{w}_i, \xi, \xi^*} \left[\frac{1}{2} \|\vec{w}_i\|^2 + K_i \left(\sum_{j \in V_i} \xi_j + \sum_{j \in V_i} \xi_j^* \right) \right] \quad (18)$$

subject to the constraints:

$$\begin{aligned} \theta_j - \vec{w}_i^T \vec{y}_j - \beta_i &\leq \epsilon_i + \xi_j, & j \in V_i \\ \vec{w}_i^T \vec{y}_j + \beta_i - \theta_j &\leq \epsilon_i + \xi_j^*, & j \in V_i \\ \xi_j &\geq 0, & j \in V_i \\ \xi_j^* &\geq 0, & j \in V_i \end{aligned} \quad (19)$$

where $\|\vec{w}_i\|^{-1}$ is relevant to the lower bound on the distance between the points $\vec{y}_j$ and the hyperplane $(\vec{w}_i, \beta_i)$, and is referred to as the margin. ϵ_i and K_i are predefined parameters. ϵ_i determines the deviation tolerance of the training data from the regression, and K_i controls the trade off between the margin and the empirical risk which is proportional to $\sum \xi_j + \sum \xi_j^*$. The idea underlying SVM regression is to minimize the empirical risk while enlarging the margin at the same time. Details of finding the optimal set of $(\vec{w}_i, \beta_i)$ parameters and thus the local linear projections are available in [26].

Figure 14 illustrates the whole process. In practice, an observed feature vector $\vec{y}$ is first classified into one of the three subsets using a Bayesian classifier, i.e., finding the class which maximizes the *a posteriori* probability given by: $p(C_j|\vec{y}) \propto p(\vec{y}|C_j)p(C_j)$, $j = 1, 2, 3$. The

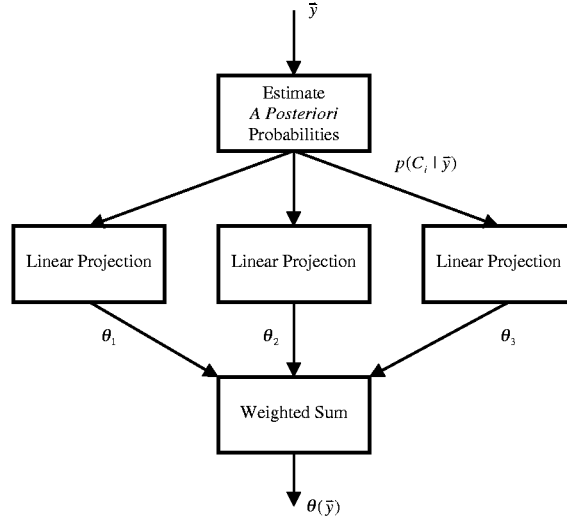


Figure 14. Pose estimation outline

a priori term $p(C_j)$ is calculated from the ratio of the subset cardinality, $|V_j|$, to the total number of training pairs, i.e.

$$p(C_j) = \frac{|V_j|}{P} \quad (20)$$

and the class-conditional density is found from

$$p(\vec{y}|C_j) = \frac{1}{|V_j|} \sum_{i \in V_j} \Phi(\vec{y}, \vec{y}_i, \Sigma). \quad (21)$$

where Φ is the Gaussian distribution with the variance matrix, Σ , predefined.

Without loss of generality, assume C_1 is the class which maximizes $p(C_j|\vec{y})$ and $C_k, k \in \{2, 3\}$ is the class which is second most likely. If the likelihood associated with class 1 is significantly greater than that of the C_k , then the pose orientation is estimated directly from the class 1 projection. Specifically, given a predefined threshold, δ , if

$$|p(C_1|\vec{y}) - p(C_k|\vec{y})| > \delta \quad (22)$$

then the estimated orientation is :

$$\theta(\vec{y}) = \hat{\theta}_1 = \vec{w}_1^T \vec{y} + \beta_1. \quad (23)$$

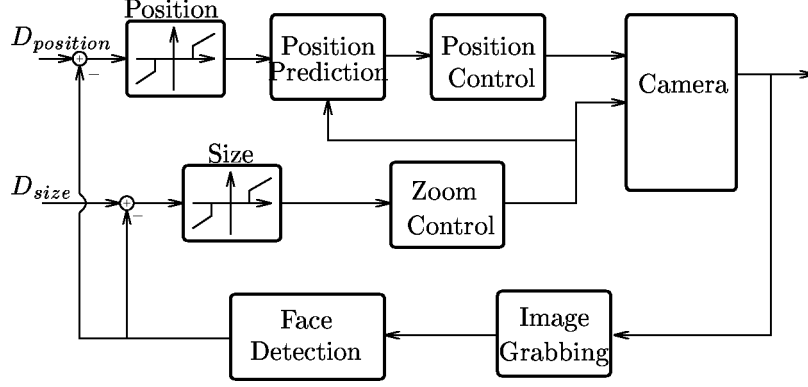


Figure 15. Camera control diagram

Otherwise the orientation angle is found using a weighted sum of the individual subset orientation angles according to:

$$\begin{aligned} \theta(\vec{y}) = & \left(\vec{w}_1^T \vec{y} + \beta_1 \right) \left(\frac{p(C_1|\vec{y})}{p(C_1|\vec{y}) + p(C_k|\vec{y})} \right) \\ & + \left(\vec{w}_k^T \vec{y} + \beta_k \right) \left(\frac{p(C_k|\vec{y})}{p(C_1|\vec{y}) + p(C_k|\vec{y})} \right). \end{aligned} \quad (24)$$

Note that the number of rough classes into which the continuous range of orientations are segmented is subject to the balance between the efficiency of the local prediction and the class separability. Increasing the number of classes improves prediction accuracy of the linear projection, but at the cost of decreased subset classification performance. The use of three subset classes was found to provide a reasonable compromise between these effects.

An example of the pose estimation procedure applied to an individual turning through a full 360° of rotation is provided in the results section.

3.4. CAMERA CONTROL

The Canon VC-C3 camera has three degrees of freedom (pan, tilt and zoom). The pan and tilt parameters are derived from the source position estimate and the camera motion model. Zoom control is evaluated from knowledge of the facial contour size and may be adjusted to provide consistent framing of the subject. The motion of the camera is approximated by a constant rotation rate plus a time delay. In situations where the camera motion is too slow to track the target, the system defaults to the acoustic localization procedure in an effort to relocate the source.

The camera control procedure is outlined in Figure 15, where $D_{position}$ and D_{size} are the desired position and size respectively. The image

grabbing and face detection portions have been discussed in the previous sections. As shown, there are two nonlinear blocks in the system, the position control and the zoom control. In an effort to avoid excessive and undesirable camera movement, camera rotations and zooming are limited. This is achieved by the two clipping blocks inserted in the feedback system. If the size and position of the face are within the regions specified by the corresponding thresholds, no changes are imposed on the camera's status. These measures have been found to produce greater image stability.

When the size of the detected face is above or below the size clipping threshold, the zoom is adjusted appropriately to bring the face size to within a tolerance of the desired size, D_{size} . The updating of the zoom control is based on the pin-hole camera model.

Similarly, when the face position is out of the tolerance range, the rotation angles of the camera (pan and tilt) are determined by predicting the face position based on the camera motion model and the current face information $\tilde{x}$, $\tilde{\dot{x}}$, $\tilde{y}$, $\tilde{\dot{y}}$, and $\tilde{\ddot{y}}$, as well as the input focus length from the zoom control block.

4. Results

The results achieved for various aspects of the system are now presented. Specifically, still images derived from the face tracker and pose estimator are given below, as well as a time series representation of the enhanced speech signal. The website <http://himmel.hrl.harvard.edu> contains audio and video demos of the entire tracking system.

4.1. FACE TRACKING

Computation for the visual component of the system is implemented on a Pentium Pro 200MHz PC running Windows 98. The image sizes are 144 by 192 pixels. Figure 16 displays an image derived from the 4-stage face detection process. The outer boxes delineate the segmented regions derived from the motion information (stage 1). In this case, three candidate locations have been identified. The ovals are the result of the geometrical analysis (stage 2) which generates a two-sided contour boundary for each region and then estimates a neckline location. For this example each of the three candidates (two faces and a hand) were found to possess a headlike geometry. The color confirmation test (stage 3) is then applied within each of the ovals. The detected skin color regions are indicated by a white box. Up to this point, the hand is still misclassified as a face candidate due to its similar shape and

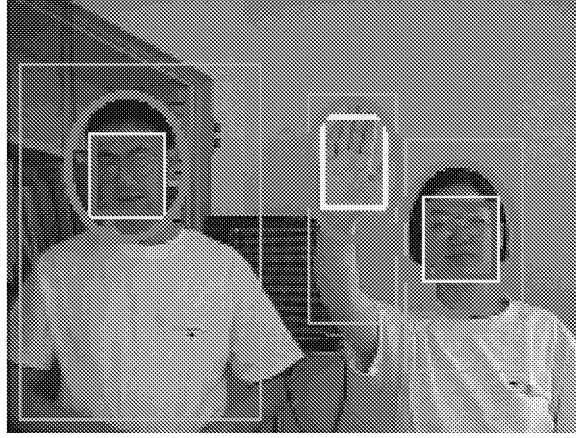


Figure 16. Multi-person face detection example

color. Lastly, the facial analysis (stage 4), with positive results shown as a gray box within the white box, correctly identifies the two faces and discards the hand.

4.2. POSE ESTIMATION

Supervised training for the estimation algorithm was achieved using a total of 300 hand-labeled vector-orientation pairs derived from video sequences of several individuals across the full range of potential orientation angles. The individuals were selected to provide a variety of hairstyles. Figure 17 illustrates the pose estimation results achieved for one individual. In each picture, a single image frame is shown along with a clock indicating the estimated facial orientation (6 o'clock corresponds to forward-facing, 12 o'clock is backward-facing). Also shown is a plot illustrating the segmented head region along with the detected hair and skin regions used to extract the feature vector.

The pose estimation results are very accurate across the complete range of possible orientation angles despite the presence of object shadows and a complex image background. This illustrates the appropriateness of the extracted image features and the effectiveness of the proposed pose learning and estimation procedures. Our experiments indicate that the hairline contour can be reliably estimated from facial images encompassing a range of lighting conditions and source-camera distances using only relative intensity statistics, color disparity, and region connectedness. In situations where the skin or hair regions are partially misidentified, the pose estimator still provides reasonable results. Figure 18(A) illustrates an example where the head has been

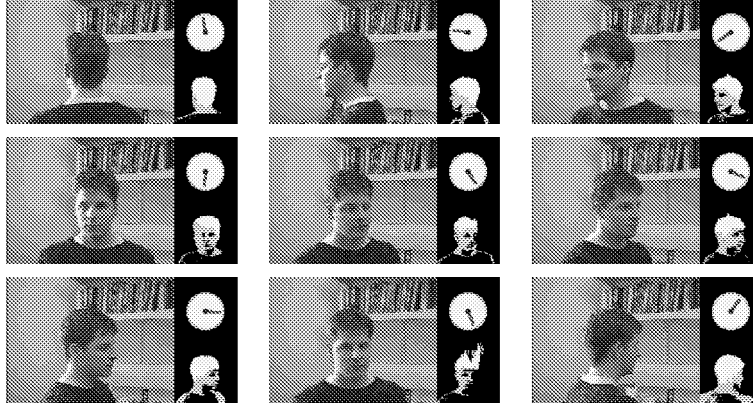


Figure 17. Examples of face orientation estimation

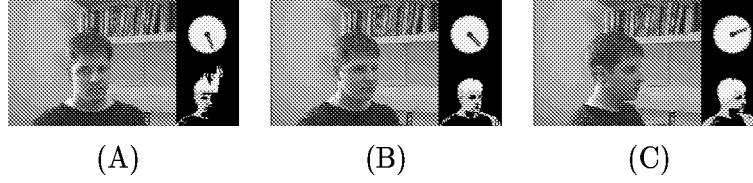


Figure 18. (A) Pose estimate for a misidentified hair contour; (B, C) estimation of a class 2 pose angle using a class 1 projection

incorrectly segmented from the background and the hair region takes on a peculiar shape. In this case the orientation is still correctly estimated.

Besides accurately estimating orientation angles appropriate for their specific class, the local linear projections extrapolate well to pose angles outside their defined range. In Figure 18(B) and (C), a pair of images which would fall in the class 2 angle region are estimated entirely from the projection associated with class 1, i.e. $\theta(\vec{y}) = \hat{\theta}_1$. In both instances the estimated pose angles are very reasonable.

4.3. AUDIO ENHANCEMENT

Figure 19 shows an example of the output of the speech enhancement procedure described in Section 2. The top plots shows a segment of speech received at a single microphone within the room. In this case the talker was standing 5 ft from a microphone belonging to the pair evaluated with the highest SNR. The bottom plot shows the enhanced signal. As is evident from the graph, the background noise has been reduced considerably, approximately by a factor of five. A portion of this attenuation (roughly a factor of two) may be attributed to the two-microphone

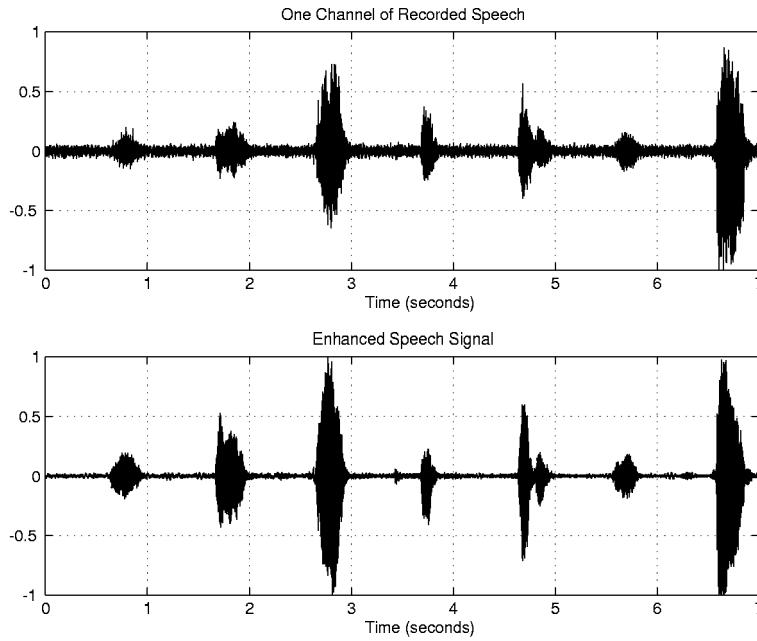


Figure 19. Speech enhancement results

delay-and-sum beamforming procedure. The remaining noise reduction is provided by the Spectral Subtraction process.

5. Conclusion

This work has presented a real-time face tracker based on both visual and acoustic information. While not demonstrated completely here, our experiments have shown that the system is (1) able to track multiple people in an unconstrained (but stable) environment, (2) relatively insensitive to facial orientation, (3) able to estimate complete head contours and adjust the camera zoom appropriately to achieve facial image size normalization, (4) capable of a video analysis rate of 15-20 frames per second using off-the-shelf PC hardware, (5) capable of accurately evaluating head orientations over a complete 360° interval, and (6) able to provide an accompanying enhanced audio signal with reduced reverberation and additive noise. The algorithms are designed to be simple enough for real-time applications and have proven robust to both the acoustic and visual environments encountered in practical video conferencing or surveillance scenarios.

References

1. M. Brandstein and H. Silverman. A practical methodology for speech source localization with microphone arrays. *Computer, Speech, and Language*, 11(2):91–126, April 1997.
2. M. Brandstein, J. Adcock, and H. Silverman. A closed-form location estimator for use with room environment microphone arrays. *IEEE Trans. Speech Audio Proc.*, 5(1):45–50, January 1997.
3. G. Carter, editor. *Coherence and Time delay Estimation*. IEEE Press, first edition, 1993.
4. M. Omologo and P. Svaizer. Acoustic event localization using a crosspower-spectrum phase based technique. In *Proceedings of ICASSP94*, pages II–273–II–276. IEEE, 1994.
5. H. Wang and P. Chu. Voice source localization for automatic camera pointing system in videoconferencing. In *Proceedings of ICASSP97*, pages 187–190, Munich, Germany, April 20–24 1997. IEEE.
6. B. Champagne, S. Bédard, and A. Stéphenne. Performance of time-delay estimation in the presence of room reverberation. *IEEE Trans. Speech Audio Proc.*, 4(2):148–152, March 1996.
7. M. Brandstein. Time-delay estimation of reverberated speech exploiting harmonic structure. *J. Acoust. Soc. Am.*, 105(5):2914–2919, 1999.
8. J. Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *J. Acoust. Soc. Am.*, 107(1):384–391, 2000.
9. D. Mansour and A. Gray, Jr. Unconstrained frequency-domain adaptive filter. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-29(3):561–571, June 1981.
10. R. Martin. An efficient algorithm to estimate the instantaneous SNR of speech signals. In *Proceedings of EUROSPEECH-93*, pages 1093–1096, Berlin, Germany, September 1993.
11. D. Ward and R. Williamson. Beamforming for a source located in the interior of a sensor array. In *Proc. IEEE Int. Symp. Sig. Process. Applicat. (ISSPA-99)*, pages 873–876, Brisbane, Australia, August 1999. IEEE.
12. B. Van Veen and K. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Mag.*, 5(2):4–24, April 1988.
13. D. Ward and M. Brandstein. Grid-based beamformer design for room-environment microphone arrays. In *1999 Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, October 17–20 1997.
14. S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-27(2):113–120, April 1979.
15. M. Brandstein, J. Adcock, and H. Silverman. Microphone array localization error estimation with application to sensor placement. *J. Acoust. Soc. Am.*, Vol.99(6):3807–3816, 1996.
16. H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE CVPR'96*, 1996.
17. A. Blake and A. Yuille. *Active Vision*. MIT Press, 1992.
18. M. C. Burl, T. K. Leung, and P. Perona. Face localization via shape statistics. In *Intl. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switz., 1995.
19. J. Yang and A. Waibel. A real-time face tracker. In *Proceedings of the third IEEE Workshop on Applications of Computer Vision*, pages 142–147, Sarasota, Florida, Dec. 1996.

20. N. Oliver, A. P. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *IEEE CVPR*, pages 123–129, 1997.
21. T. Jebara and A. Pentland. Parameterized structure from motion for 3D adaptive feedback tracking of faces. In *IEEE CVPR*, pages 144–150, 1997.
22. A. K. Jain. *Fundamentals of digital image processing*. Prentice Hall, 1989.
23. R. Lopez and T. S. Huang. Head pose computation for very low bit-rate video coding. In *6th International Conference on Computer Analysis of Images and Patterns*, pages 440–447. Springer-Verlag Berlin Heidelberg, 1995.
24. N. Kruger, M. Potzsch, and C. Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. *Image and Vision Computing*, 15(8):665–673, August 1997.
25. I. Shimizu, Z. Zhang, S. Akamatsu, and K. Deguchi. Head pose determination from one image using a generic model. In *3rd IEEE International Conference On Automatic Face and Gesture Recognition*, pages 100–105, Nara, Japan, April 1998.
26. V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

6. Biographies

Ce Wang was born in Beijing, China. He received his B.S. degree from the Department of Automation of Tsinghua University in 1992, his M.S. degree from the National Laboratory of Pattern Recognition in the Institute of Automation at the Chinese Academy of Sciences in 1995, and his M.S. degree in Computer Science from Harvard University in 1996. He is currently a Ph.D. candidate in Computer Science at Harvard University. His research interests include computer vision, pattern recognition, image processing and multi-media applications. He is a student member of IEEE.

Scott M. Griebel was born in Erie, PA, in 1972. He received his B.S. in Computer Engineering from Penn State University in 1994, his M.S. in Electrical Engineering from Northeastern University in 1996, and his S.M. in Engineering Sciences from Harvard University in 1998. He is currently a Ph.D. candidate in Engineering Sciences at Harvard University. His research interests include speech signal processing, multi-channel signal enhancement, automated video-conferencing, and biomedical signal and image processing.

Bo-June (Paul) Hsu was born in Taiwan in 1978. He received both his A.B. in electrical engineering and S.M. in Computer Science from Harvard University in 2000. He is currently a software design engineer in Speech.Net at Microsoft working on the acoustic front-end for speech recognition engines.

Michael S. Brandstein was born in Willimantic, CT, in 1967. He received his Sc.B. from Brown University in 1988, his S.M. from Massachusetts Institute of Technology in 1990, and his Ph.D. from Brown University in 1995, all in electrical engineering. He is an Associate Professor of Electrical Engineering at Harvard University. His primary research interest is in the area of multi-media, specifically multi-channel methods for speech, audio, and image processing. His work has featured developments in source localization, microphone array design, time-delay estimation, multi-channel speech enhancement, and acoustic and visual-based tracking.

Dr. Brandstein is currently an associate editor for the IEEE Transactions on Speech and Audio Processing and a member of the IEEE Signal Processing Society's Audio and Electroacoustics, Education, and Multimedia Signal Processing Technical Committees.

Address for Offprints:

33 Oxford Street

341 Maxwell-Dworkin Building

Cambridge, MA 02138

USA

