Multi-Source Human Identification

by

Brian A. Kim

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[June 2003]
May 23, 2003

Author .. ..
Department of Electrical Engineering and Computer Science
May 23, 2003

Certified by..... .....
Tomaso Poggio
Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Multi-Source Human Identification

by

Brian A. Kim

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2003, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

In this thesis, a multi-source system for human identification is developed. The system uses three sources: face classifier, height classifier, and color classifier. In the process of developing this system, classifier combination and the integration of classifer outputs over sequences of data points were studied in detail. The method of classifier combination used relies on weighing classifiers based on the *Maximum Likelihood* estimation of class probabilities. The integration of classifer outputs, which is termed "temporal integration" in this thesis, has been developed to take advantage of the information implicitly contained in data correlated through time. In all experiments performed, temporal integration has improved classification, up to 40% in some cases. Meanwhile, the method of temporally integrating the outputs of multiple classifiers fused using our classifier weighting method outperforms all individual classifiers in the system.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor
Brain Sciences Department
McGovern Institute and Artificial Intelligence Lab

Acknowledgments

I would like to thank Professor Poggio, Yuri Ivanov and Bernd Heisele for giving me the opportunity to work on this thesis. I would also like to thank my parents and friends for all the support they've given me throughout the year.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human Identification is a very popular topic in computer vision. Its application to areas such as surveillance and robotics are obvious. While popular media often portrays systems that can quickly identify people, this task is not as simple as Hollywood directors would have people believe. Current state of the art systems can be very accurate but are designed to operate in highly constrained environments where several factors such as lighting and view are held constant. In more realistic settings, input data is more plentiful but of lower quality. However, most systems do not take this into account, preferring to view each data point independently while significantly greater amounts of information may be gained by viewing each data point as part of a larger whole.

1.1 Problem Statement

The goal of this thesis is to build a robust human identification system that analyzes a stream of data using multiple experts. In this case, the stream of data is a sequence of images taken from a video which are related temporally in that all images are 1/30 of a second apart. The multiple experts are different classifiers which analyze each image to prodict the identity of the person seen. The emphasis of this work is to study methods of combining classifiers over a set of data points rather than individually. In doing so, it is hoped that greater system robustness can be achieved

such that accurate identification results can be achieved using videos that are typical of a standard surveillance camera.

1.2 Previous Work

Research in audio-visual speech recognition [5, 8, 9] shows that classification results can be improved by integrating evidence from multiple sources of observation. Similar results have been observed and demonstrated in verification systems [16], as well as in multi-modal trackers [13].

In Nakajima *et al*, [7, 6], a real-time person identification system using SVM-based ([17]) multi-class classification is presented. The goal of the system is to identify lab members using the espresso machine based on four sets of features which include color color and shape histograms. Since the system operated under the assumption that a person's clothing would stay the same during the day, single day results were very good. However, results for multi-day settings were poor since the system did not have provisions for longer term identity models.

Yang *et al* [11] developed a multi-modal identification system for identifying people participating in a meeting for enhanced meeting records. The system used face recognition, speaker identification, color appearance, and sound source direction to determine who was speaking at a particular time. However, only the results from a single experiment containing three people were presented. In that experiment, combining classifiers produced an error rate two percent less than using the classifiers individually.

In [14], an automotive pedestrian detection system that took advantage of temporal information was developed. Using the *a priori* knowledge that a person in one frame will appear in a similar position in the next frame, spurious false positives could easily be rejected. Implementation of this heuristic resulted in 55% fewer false positives.

In [1], a system for person recognition using multiple classifiers was presented. The system used composed of two cameras, one for profile views and the other for frontal

views. Using classifier combination, the system was able to achieve significantly better performance over each classifier individually. However, the system relied on high resolution images (512x342 pixels) taken under controlled lighting conditions.

In [10], Kittler *et al* developed a theoretical framework for combining classifiers that uses distinct pattern representations. They show that many existing schemes can be seen as compound classification, where all classifiers are used to make a decision. In their findings, they state that the sum rule and its derivatives (max rule, majority vote rule, and median rule) consistently outperform other classifier combination schemes. In [12], Kittler extends this work to state that sum outperforms majority vote when all classifiers are of equal strength and estimation errors are conditionally independent and identically distributed. However, for estimation errors modeled by heavy tail distributions, Kittler finds that voting may outperform sum.

The work in this thesis continues in the general direction of classifier combination and sequence analysis. Chapter 2 discusses our methods of combining classifiers and combining classification results across sequences of data. Chapter 3 describes our Human Identification system in detail. Results from all experiments are presented in Chapter 4. Chapter 5 contains a discussion and conclusions based on the work thus far.

Chapter 2

Strategies for Classifier Combination

In the simplest classification system, there is a single classifier that classifies each data point independently. In this system, there are multiple classifiers that operate over an entire set of related data points rather than each data point individually. In designing such a system, the issue of how to combine the classifier outputs to form a decision must be addressed. The problem is well studied in the setting of a homogeneous set of classifiers. However, when the classifiers in the ensemble apply to different features sets and, therefore, produce outputs of different strengthes this task is not as straightforwards. In addition, the issue of how to combine the classifier outputs over several data points must also be investigated.

2.1 Temporal Integration

In this document, we are closely exploring the problem of integration of outputs of a classifier over a number of data points. Temporal integration addresses how the outputs from a classifier should be combined over data points that are related. Given that all data analyzed in this system is temporally related, we have chosen to call this "temporal integration."

Temporal streams of data occur naturally almost everywhere. A video of some-

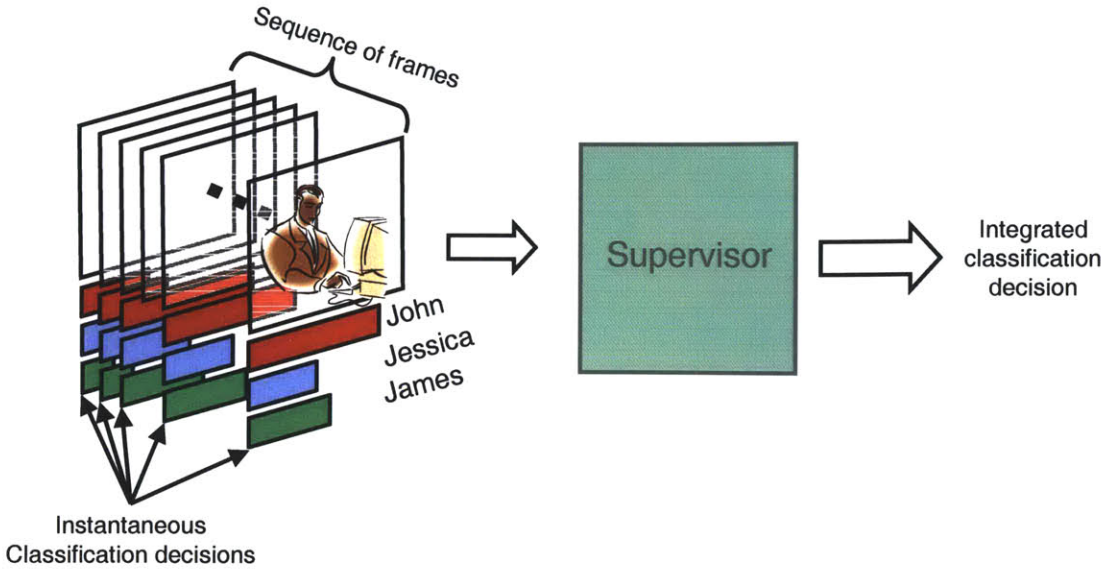

Figure 2-1: Given a sequence of images that are temporally related, such as the frames from a video, each frame is classified. Rather than make a single classification decision at every frame, all classifier scores for the entire sequence are given to a "supervisor" that performs temporal integration to produce an integrated classification decision.

one crossing the street contains several frames of that person in different positions and locations. While analyzing each frame statically may yield poor results, taking advantage of the *a priori* knowledge that the person appears in a similar position in the next frame can dramatically improve results as seen in [14].

In the scenario above, it is assumed that only one person is seen crossing the street and that the identity of that person remains constant throughout all frames. Unfortunately, data such as this is difficult to collect in large quantities since it requires editing or labeling of videos by hand. Assuming that the videos can be constrained to contain only one person at a time, the problem of determining when we stop seeing one person and start seeing a new person remains. This issue of object tracking is well known and lies beyond the scope of this thesis.

2.1.1 Sequence Based Temporal Integration Strategies

Given that the sequence of data being classified has a singular, consistent label and that it is known when this label changes (without actually knowing the true label itself), several strategies for performing temporal integration can be devised. As-

Sequence Based Temporal Integration Strategies	
Strategy	*Formula*
maximum of classifier values	$c^m = \arg\max_c \left(\max_t (s^c_{t=1}, s^c_{t=2} \ldots s^c_{t=n}) \right)$
minimum of classifier values	$c^m = \arg\max_c \left(\min_t (s^c_{t=1}, s^c_{t=2} \ldots s^c_{t=n}) \right)$
median of classifier values	$c^m = \arg\max_c \left(\underset{t}{med} (s^c_{t=1}, s^c_{t=2} \ldots s^c_{t=n}) \right)$
average of classifier values	$c^m = \arg\max_c \left(\frac{1}{n} \sum_{t=1}^{n} s^c_t \right)$
majority vote over sequence	$c^m = \arg\max_c \left(\sum_{t=1}^{n} I(s^c_t > 0) \right)$

Table 2.1: Given a set of one-vs-all classifiers C which is used to classify a sequence of data points ranging from $t = 1$ to $t = n$. The output of a particular classifier in C is s^c_t and the label assigned to the entire sequence is m.

suming that our classifier is actually a set of m one-vs-all classifiers for performing multi-class classification, for each data point there is an output from each one-vs-all classifier giving m outputs in total.

The first rule we can devise is to take the maximum of the classifier values for each one-vs-all classifier, and then take the maximum over all classifiers to determine the label. Similarly, the minimum, median, and average rules would take the minimum, median, and average of the classifier values and choose the the classifier with the highest score to set the label. Lastly, there is the vote rule in which the number of times a one-vs-all classifier has positive output on the sequence of data points is counted and considered a "vote". The classifer with the most votes is then used to determine the label.

A more mathematical statement of these rules can be found in Table 2.1. There we have a set of C one-vs-all classifiers which is used to classify a sequence of data points ranging from $t = 1$ to $t = n$. The output of a particular classifier in C is s^c_t and the label assigned to the entire sequence is m.

2.1.2 Non-Sequence Based Temporal Integration Strategies

If we remove the assumption that it is known when the class label changes, slightly different temporal integration strategies must be devised. Since we no longer know

Non-Sequence Based Strategies	
Strategy	*Formula*
running maximum	$c_k = \arg\max_c \left(\max_t (s^c_{t=k-w+1}, s^c_{t=k-w+2} \cdots s^c_{t=k}) \right)$
running median	$c_k = \arg\max_c \left(\underset{t}{med} (s^c_{t=k-w+1}, s^c_{t=k-w+2} \cdots s^c_{t=k}) \right)$
running minimum	$c_k = \arg\max_c \left(\min_t (s^c_{t=k-w+1}, s^c_{t=k-w+2} \cdots s^c_{t=k}) \right)$
running average	$c_k = \arg\max_c \left(\frac{1}{w} \sum_{t=k-w+1}^{k} s^c_t \right)$
majority vote over window	$c_k = \arg\max_c \left(\sum_{t=k-w+1}^{n} I(s^c_t > 0) \right)$
decaying average	$c_k = \arg\max_c \left(\overline{s}_k = \overline{s}_{k-1} + \alpha(s_k - \overline{s}_{k-1}) \right)$

Table 2.2: Given a set of one-vs-all classifiers C which is used to classify a set of sequences whose boundaries are unknown, the classification decision for point k is based on the previous w classifier outputs.

what the sequence boundaries are that separate one sequence of points from the next, all data looks like one long sequence. Using any of the sequence based strategies, in this case, would yield poor results since the same label would be set across all sequences. To remedy this, a simple change can be made. Rather than looking at all classifier outputs from $t = 1$ to $t = n$, consider the classification outputs of w data points before the current point to make a classification decision. Although this means performing point-by-point classification, hopefully improvements can occur by looking beyond a single data point.

These modified rules can be seen in Table 2.2 where for each position k in set of data being classified, the previous w classifier outputs are considered to set the label for that position.

2.2 Combining Classifiers

While temporal integration answers the question of how to combine the outputs of a single classifier, the issue of combining multiple classifiers remains. Any method of combination that uses only raw classification scores fails to take into account other factors such as how similar the classes are. In our system, we have chosen to weigh the set of classifier outputs for each data point by the class probability measured

on the training set before combining. Using this method of "weighing" classifiers makes the outputs of different classifiers easily comparable and therefore easily combinable. Weights can then be placed on the classifiers themselves for a more complete probabilistic measure of identity.

2.2.1 Computing the class probability

To compute the class probability for a particular classifier, a confusion matrix is generated from classifying the training set. The confusion matrix, which we denote by C, is basically an empirical evaluation of how alike people are from the point of view of a particular classifier. After counting the mistakes and correct hits for each person in the data set, we collect the results in a matrix which has each row corresponding to a different person and each column - the outcome of the classification. Denoting the true person identity by ω and the outcome of the classification by $\tilde{\omega}$,we can compute the joint density, $P(\omega, \tilde{\omega})$, which describes a probability with which the events $\omega = i$ and $\tilde{\omega} = j$ co-occur. A *Maximum Likelihood* estimate of this probability is simply a normalized version of the confusion matrix, C:

$$P(\omega = j, \tilde{\omega} = j) = \frac{C_{j,i}}{\sum_n \sum_m C_{m,n}} \tag{2.1}$$

The goal of classification is to determine the probability of the true class given an observation, $P(\omega|x)$:

$$\begin{aligned} P(\omega|x) &= \sum_{\tilde{\omega}} P(\omega, \tilde{\omega}|x) = \sum_{\tilde{\omega}} P(\omega|\tilde{\omega}) P(\tilde{\omega}|x) \\ &= \sum_{\tilde{\omega}} \frac{P(\omega, \tilde{\omega})}{P(\tilde{\omega})} P(\tilde{\omega}|x) = \sum_{\tilde{\omega}} \frac{P(\omega, \tilde{\omega})}{\sum_{\omega} P(\omega, \tilde{\omega})} P(\tilde{\omega}|x) \end{aligned} \tag{2.2}$$

Note that in the end of the first line of this equation ω no longer depends on x, as all the information about it is contained in the estimated class label, $\tilde{\omega}$. The final equation is just a matrix-vector multiplication between the column-normalized

confusion matrix and a vector of classifier scores, assuming that they behave like probabilities.

2.2.2 Combining classifier outputs

A set of the confusion matrices for all classifiers in the ensemble represents the distribution $P(\omega, \tilde{\omega}|\lambda)$, where λ is the type of classifier (height, sound, etc.). Again, the goal of the final classification is to produce the value of $P(\omega|x)$, computed from each individual type of classifier:

$$P(\omega|x) = \sum_{\lambda} P(\omega|x, \lambda)P(\lambda|x) \tag{2.3}$$

At this point we assume that $P(\lambda|x)$ is uniform [1]. The first term of the sum can be calculated from the confusion matrices:

$$\begin{aligned} P(\omega|x, \lambda) &= \sum_{\tilde{\omega}} P(\omega, \tilde{\omega}|x, \lambda) = \sum_{\tilde{\omega}} P(\omega|\tilde{\omega}, \lambda)P(\tilde{\omega}|x, \lambda) \\ &= \sum_{\tilde{\omega}} \frac{P(\omega, \tilde{\omega}|\lambda)}{P(\tilde{\omega}|\lambda)} P(\tilde{\omega}|x, \lambda) = \sum_{\tilde{\omega}} \frac{P(\omega, \tilde{\omega}|\lambda)}{\sum_{\omega} P(\omega, \tilde{\omega}|\lambda)} P(\tilde{\omega}|x, \lambda) \end{aligned} \tag{2.4}$$

This is basically the eqn. 2.2, formally conditioned on the type of the classifier, λ. Putting equations 2.3 and 2.4 together, we get the answer:

$$P(\omega|x) = \sum_{\lambda} P(\lambda|x) \sum_{\tilde{\omega}} \frac{P(\omega, \tilde{\omega}|\lambda)}{\sum_{\omega} P(\omega, \tilde{\omega}|\lambda)} P(\tilde{\omega}|x, \lambda) \tag{2.5}$$

Here the fraction inside the sum is the column-normalized confusion matrix for a classifier λ, $P(\tilde{\omega}|x, \lambda)$ is the vector of probability-like scores of the observation x for classifier λ, and $P(\lambda|x)$ is the probability-like weight.

Use of empirical prior to combine classifiers and temporal integration to account for sequence information lays the framework for a multi-source classification system

[1] Setting $P(\lambda|x)$ dynamically based on factors external to the classifier is an area to be explored in future research.

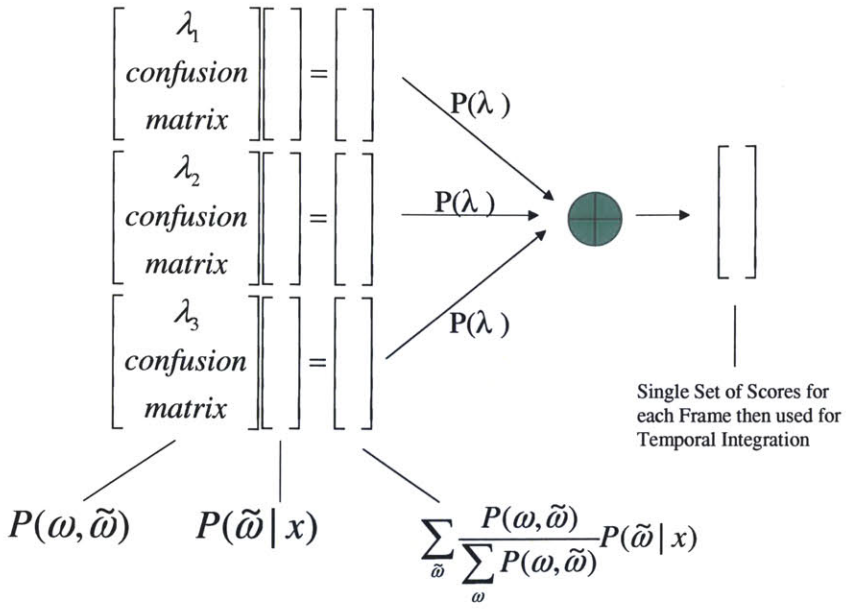

Figure 2-2: In this illustration, there are three classifiers, denoted by λ, that distinguish between n people. Each classifier $lambda_i$ produces a nx1 vector of outputs from it one-vs-all classifier members. In order to combine the outputs of all three classifiers, the vector of outputs from each classifier is multiplied by its confusion matrix ($P(\omega|\tilde{\omega})$) and weighted by $P(\lambda|x)$ before being combined. The end result is a single set of scores that has appropriately weighted classifier decisions from all classifiers which can then be used to make an integrated decision or used for temporal integration.

which we develop in the next chapter.

Chapter 3

Multi-source Human ID

This chapter covers the design, implementation, and performance of a multi-source human identification system that runs in close to real-time. The system is composed of a camera focused on a fixed scene and a computer for storing and processing data. Identification is performed by temporally integrating the outputs of several classifiers and combining the results at the end of the sequence to produce a final decision.

3.1 Data Acquisition

An application has been developed for the automatic collection and storage of video data. Given that the camera monitors a fixed scene, data is recorded when a sufficient amount of motion is detected. Video data is written to disk using Mpeg-4 encoding for efficient file storage. A background image is immediately stored after recording has finished to simply later processing. All files are saved in a hierarchical directory tree for easy lookup. Attributes that are necessary for labeling and generated test and training sets are stored in a relational database.

3.2 Pre-Processing

Before images from the camera or videos can be classified, a certain amount of pre-processing must occur. All classifiers require background subtraction to be performed

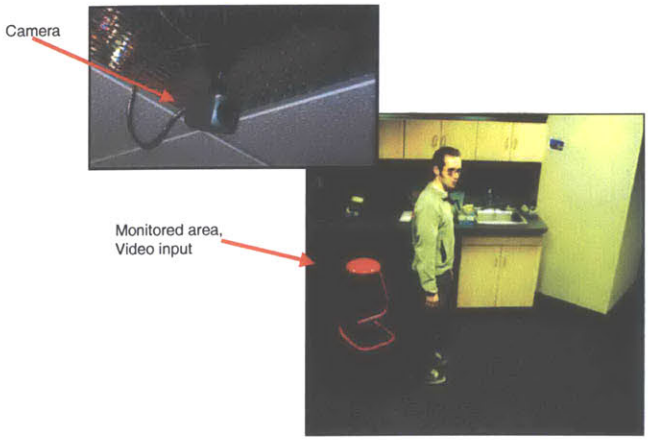

Figure 3-1: The system is composed of a fixed camera monitoring a scene such as the coffee machine area in our office. When sufficient motion is detected in the scene, data is recorded and/or analyzed to produce a classification decision.

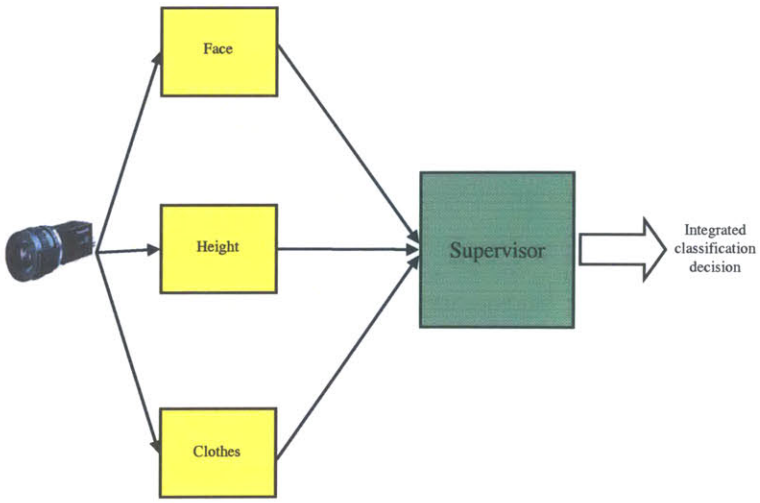

Figure 3-2: Data from the camera or input video is given to three classifiers: face classifier, height classifier, and adaptive color modeling (clothes) which produce outputs that are then combined by a Supervisor module to produce an integrated classification decision over a sequence of frames.

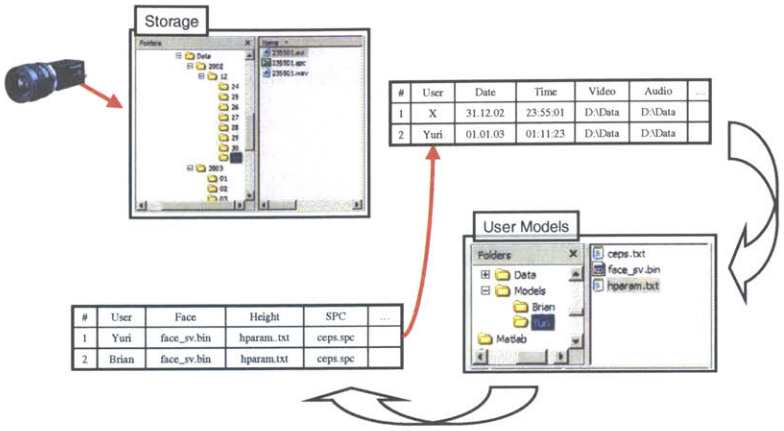

Figure 3-3: Outline of Data Acquisition Software. Data from the camera is stored in a hierarchical directory structure which denotes the date. The actual time of recording and length in seconds is used to generate the actual filename. After data is labeled, attributes are stored in a relational database to simplify generation of test and training data.

to segment the person from the background. Face detection must also be performed for the face recognition classifiers.

3.2.1 Adaptive Background Modeling

Adaptive background modeling is used to account for variations in illumination which occur throughout the day and changes in scenery due to objects being moved. The model is initialized with a single image. For each successive image, the mean value of each pixel x at time t is calculated according to the formula

$$\overline{x}_t = \overline{x}_{t-1} + \alpha(x_t - \overline{x}_{t-1})$$

where value of α determines the rate at which a still object becomes part of the background. When performing image subtraction, the absolute difference between the new image and the background model is performed. The result is then thresholded to rule out slight changes due to noise. This process produces a binary image where pixels are marked by 1's and 0's depending on whether the pixel belongs to foreground or background. Further noise reduction is performed through the use of morphological operations to eliminate objects that are too small to be people.

3.2.2 Face Detection

The component based system developed by Heisele *et al* in [3] is used to detect faces. Component based detection has been shown to be more robust against out-of-plane rotations. This face detection scheme consists of a two level hierarchy of SVM classifiers. In the first level, components such as the eyes, nose, and mouth are detected. In the second level, a single classifier is used to determine if the detected components from the first level are in the correct configuration for a face. Faces of different sizes are searched for by rescaling image according to minimum and maximum expected face size parameters.

3.3 Classifiers

The system uses three classifiers to determine the identity of a person: face classifier, height classifier, and color classifier. Face recognition is an important aspect of developing long term identity models since people's faces do not dramatically change over significant periods of time. Although height classification is not the most foolproof method of determining the identity, it can act as a strong "supporting" classifier. Suppose the face recognition classifier is confused between two people that look very similar but have different heights. In this case, height classifier can be used to gain significantly higher accuracy in these situations [1]. Color classification are known to be accurate in classifying people over a short period of time such as a day or less.

3.3.1 Face Classifier

Face recognition is performed in a one-vs-all approach using n SVMs, where n is the number of people the system is trained to identify. Each SVM is trained using the SvmFu package [15] on a labeled set of positive and negative samples. The positive set contains images of the person that should be identified and the negative set is composed of images of everyone else. The set of SVMs is seen as a single classifier for the purposes of this system and the scores of all the SVMs for a given face are the classification result[2].

3.3.2 Height Classifier

The height classifier is actually composed of two parts: height approximation and height classification. Height approximation is a general purpose method of measuring the height of objects in the scene. Height classification then uses the measured height to return a set of probabilities that measure the likelihood of that height being from

[1] Similarly, another use for the height classifier would be to rule out certain people from other classifiers. When the height of the person is seen to be h, only consider people with heights in the range of $h - \epsilon$ and $h + \epsilon$.

[2] To obtain a recognition result for a particular frame, you take the maximum score from all classifiers.

a particular person.

Height Approximation

Measuring height of a person is possible with a calibrated video system. Typically, camera calibration algorithms estimate *intrinsic* and *extrinsic* parameters of the optical camera system. Intrinsics refer to the internal parameters of the camera system that define the projective relation between the 3D scene and a 2D image and include *focal length, principal point coordinates* and *lens distortion parameters.* Extrinsics define the 3D camera position and orientation (rotation and translation) with respect to some fixed origin.

Having a calibrated camera, estimation of the height of the person is relatively easy. We simply need to invert the projective transformation from the camera image to the 3D scene for two points defining the person's height - top of the head and bottom of the feet. Unfortunately the inversion of this relationship is only up to scale, so it cannot be done without additional constraints. To solve that we can use two additional constraints - people tend to touch the ground plane and to stay vertical. These assumptions allow us to solve a system of equations to find a height of the person from an image taken from a calibrated camera.

Alternatively, when camera calibration is difficult to perform due to physical constraints, we can approximate the parameters of the camera system. The approach we chose for this purpose is to approximate the function that maps the apparent hight and position of the person to the true height.

In order to estimate the height of the person in the camera image we use a 2-nd degree polynomial of 3 variables - apparent height, h and x- and y- positions. The mapping function is the weighted sum of products of these variables taken to different powers up to[3] 2:

$$f(x, y, h) = \sum_{p=0}^{2} \sum_{q=0}^{2} \sum_{r=0}^{2} a_{4p+2q+r} x^p y^q h^r \tag{3.1}$$

[3]In practice we do not exhaustively enumerate all powers between 0 and 2 but omit some cross terms to reduce the number of terms in the expression.

where coefficients a_i need to be found. In order to solve for the coefficients we collect training data from people of known heights, H_c, walking in the scene, while we measure the apparent height, h_i and the lowest point of the person blob in the image, x_i and y_i. With this data we form two matrices:

$$\mathcal{P} = \begin{bmatrix} x_1^2 y_1^2 h_1^2 & x_1^2 y_1^2 h_1 & \cdots & 1 \\ x_2^2 y_2^2 h_2^2 & x_2^2 y_2^2 h_2 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ x_N^2 y_N^2 h_N^2 & x_N^2 y_N^2 h_N & \cdots & 1 \end{bmatrix} \tag{3.2}$$

which is the matrix of the powers of the collected data points, and the matrix of true heights:

$$\mathcal{H} = \begin{bmatrix} H_1 \\ H_1 \\ H_1 \\ \cdots \\ H_C \end{bmatrix} \tag{3.3}$$

These two matrices are related by the linear equation:

$$\mathcal{H} = \mathcal{P}\mathbf{a} \tag{3.4}$$

where $\mathbf{a}$ is the vector of coefficients that we need to find. We find a least squares solution for $\mathbf{a}$ by taking a pseudoinverse of $\mathcal{P}$:

$$\mathbf{a} = \mathcal{P}^{\dagger}\mathcal{H} \tag{3.5}$$

Now for a new observation vector, $(x, y, h)^T$ we can find an approximation to the true hight of the person from equation 3.1:

$$H = f(x, y, h) \tag{3.6}$$

Classification

To account for errors in the approximation as well as for noisy estimates of position and apparent height we model height of each person with a one dimensional Gaussian density. After regressing the polynomial on the training data we estimate heights of the people in the data set and compute their means and variances:

$$\mu_c = \frac{1}{N_c} \sum_{n=1}^{N_c} f(x_c^n, y_c^n, h_c^n) \tag{3.7}$$

and

$$\sigma_c = \frac{1}{N_c - 1} \sum_{n=1}^{N_c} (f(x_c^n, y_c^n, h_c^n) - \mu_c)^2 \tag{3.8}$$

A likelihood for a new observation vector $(x, y, h)^T$ can now be found by plugging it into a corresponding Gaussian:

$$p(\mathbf{x}|\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{x} - \mu_\mathbf{c})^2}{2\sigma^2}\right) \tag{3.9}$$

The *a posteriori* of the identity of the person is found from the Bayes rule:

$$P(\omega|x) = \frac{p(\mathbf{x}|\omega)P(\omega)}{p(\mathbf{x})} \tag{3.10}$$

where we assume a uniform distribution for $P(\omega)$. And, finally, the classification decision is made from the maximum *a posteriori*:

$$\omega = \arg\max_c P(\omega = c|\mathbf{x}) \tag{3.11}$$

3.3.3 Adaptive Color Models

Color appearance is known to be robust in a complex scene and is fairly invariant to distance, pose, and occlusion. To obtain the identity of a person in a frame using their color appearance, the following steps are taken:

-segment person from the background

-extract color feature of the person for that frame

-compare against known color features for each person and produce probabilities that measure the likelihood of that color feature being from a particular person

Extracting Frame Color Features

To generate the color feature for a given frame, we first obtain the segmented person using background subtraction and adaptive background modeling. Then, a color histogram is computed in hue-saturation space. Use of the hue-saturation space greatly reduces variations in color due to illumination.

Generating Color Models

During training, a set of color models is generated based on the labeled training videos. These color models are generated in a fashion similar to frame color features. However in the case of color models, the histograms are accumulated over all images and then normalized. This ensures that the range of values for the color models and frame color features match so that they can be readily compared.

Classifying Color Features

To determine the identity of the person being seen, we compare the color feature for the frame against all known color models. This comparison is performed using the normalized correlation given by the equation:

$$d(H_1, H_2) = \frac{\sum_I (H_1'(I) H_2'(I))}{\sqrt{\sum_I H_1'(I)^2 \sum_I H_2'(I)^2}} \tag{3.12}$$

where

$$H_k'(I) = H_k(I) - \frac{1}{N \sum_J H_K(J)}. \tag{3.13}$$

and N is equal to the number of histogram bins.[4] These correlation coefficients are normalized to sum to 1 and returned.

[4] N = 360 since 30 bins are used for hue and 32 bins are used for saturation.

The "adaptiveness" of this classifier comes from learning new color features as people are classified. While frame color features are generated, a cumulative color model is simultaneously generated for later addition to the set of known color features. Once a final classification decision is made using all classifiers and frames, the cumulative color feature assigned a label and used for future color classification. However, this requires certain conditions to be met in terms of the confidence of the identity assigned. This is detailed more in the discussion of the Supervisor module in Section 3.4.

3.4 Supervisor

The Supervisor is a module for determining the final classification result. The inputs to the Supervisor are the classification outputs from each classifier on every frame. Classifier combination and temporal integration is then performed to fuse the classifier decisions and take advantage of sequence information as outlined in Chapter 2.

3.4.1 Setting Classifier Weights

To recall from Chapter 2, a probability-like weight denoted by $P(\lambda|x)$ is assigned to each classifer before combining. The Supervisor is of great importance because of its responsibility in determining $P(\lambda|x)$ for each classifier at every frame. A low value of $P(\lambda|x)$ means that the classifier is very unreliable whereas a high value means good reliability. To accurately set $P(\lambda|x)$, the Supervisor must account for other factors not seen by the classifier.

In the current system, $P(\lambda|x)$s are dynamically set depending on which classifiers are considered reliable. $P(\lambda|x)$ is set to 0 when the classifier is considered unreliable. When multiple classifiers are present, the value $P(\lambda|x)$ is equal to the accuracy of the classifier divided by the total accuracy of all classifiers deemed reliable for that frame.

Reliability is determined using simple heuristics. For face recognition, $P(\lambda|x)$ depends on the face detection score which is a measure of how likely that the face

found is actually a face. For the height classifier, the heuristic used simply checks to see that the bounding box of the object is fully contained in the image. A person that is only partially seen in the image would fail this test. Color modeling is much less restrictive than height in terms of object location or size, however time is an important issue. When multiple known color models are shown to be highly correlated with the frame color feature, a high $P(\lambda|x)$ is assigned to the known color model that is from the same day. Through additional experimentation, additional heuristics can be developed which is also the subject of future research.

3.4.2 Labeling Color Features

An additional task of the Supervisor is to keep the color models current. This is necessary considering that people tend to change clothes daily. After a person is observed in a video, the color model for that person is appended to the list of known color models if an integrated classification decision has been made with high confidence. High confidence is determined by the value of the scores used to make the final classification.

Chapter 4

Experiments

To test our ideas on classifier combination and temporal integration, three sets of experiments were conducted. In the first set, synthetic data was used to study temporal integration of a single classifier in more detail. In the next set of experiments, face and audio data were used as data sets to see how the results from the synthetic data experiments generalized to real data. The final set of experiments used data from the multi-source system described in Chapter 3 to test classifier combination and temporal integration.

4.1 Synthetic Data Experiments

In order to examine temporal integration in a more controlled environment, synthetic data was studied. By using synthetic data, it is simple to generate datasets with differing degrees of separability. Since temporal integration relies on smoothing the classifier outputs from correlated data, sequences of data points were generated. In these experiments, each class is modeled by a gaussian distribution in $\mathbf{R}^3$. To model a sequence, a random walk with fixed step size is generated from a starting point drawn from the distribution.

To generate data for each class, the following procedure was used:

1) select a class label

2) select a starting point using the gaussian distribution for that class which has

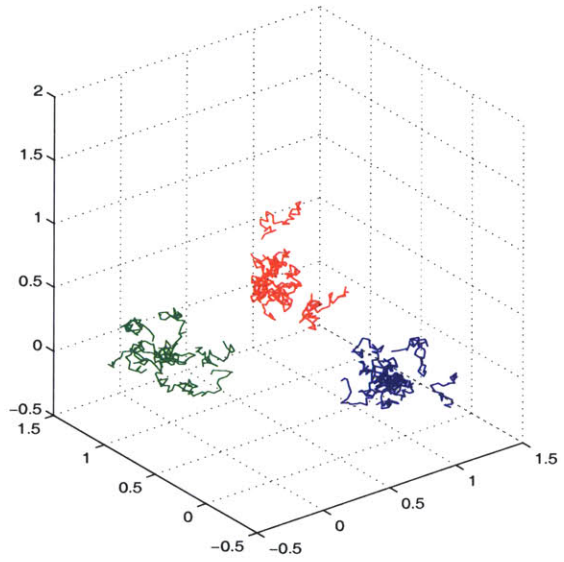

Figure 4-1: Plot of synthetic data where each class has a variance of 0.01 and a step size of 0.05. In this data set, classes are completely separated allowing no room for classification improvement.

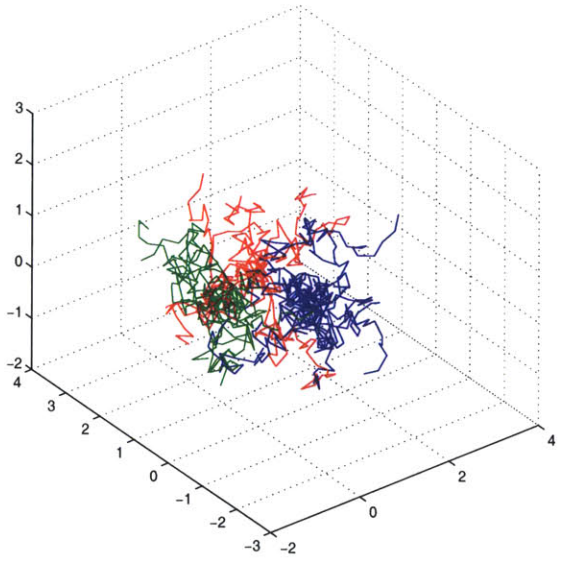

Figure 4-2: Plot of synthetic data where each class has a variance of 0.01 and a step size of 0.30. Here the classes are difficult to separate when classified point-by-point. However, when temporal integration is used, an accuracy of 90% can be achieved over baseline performance of 72% accuracy.

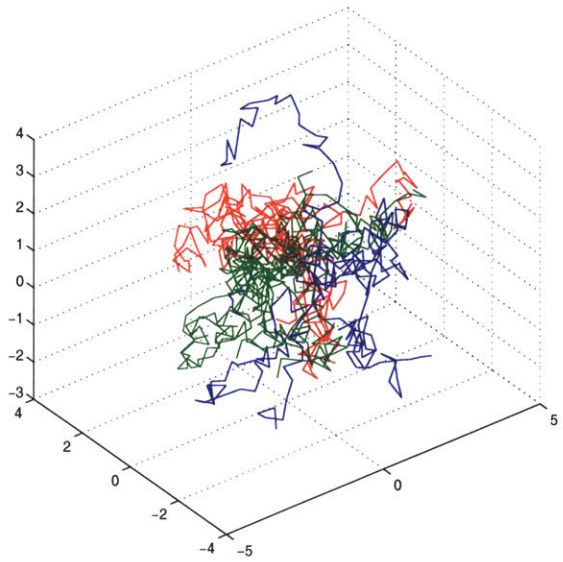

Figure 4-3: Plot of synthetic data where each class has a variance of 0.10 and a step size of 0.50. In this dataset, classes are even harder to separate. However, temporal integration still improves classification.

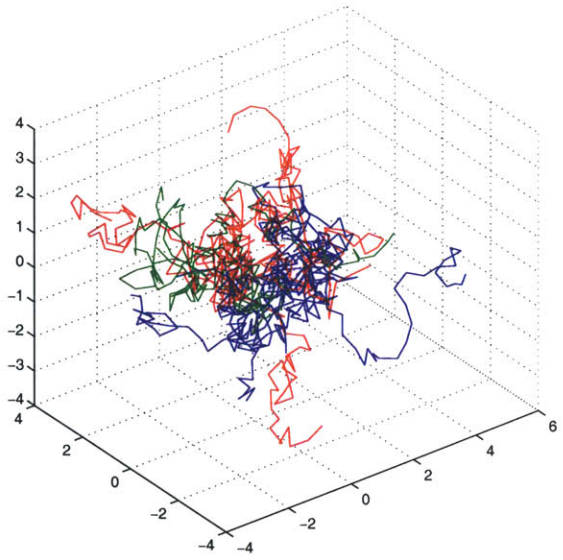

Figure 4-4: Plot of synthetic data where each class has a variance of 0.30 and a step size of 0.50. This is the most inseparable dataset studied. All strategies yield little or no improvement.

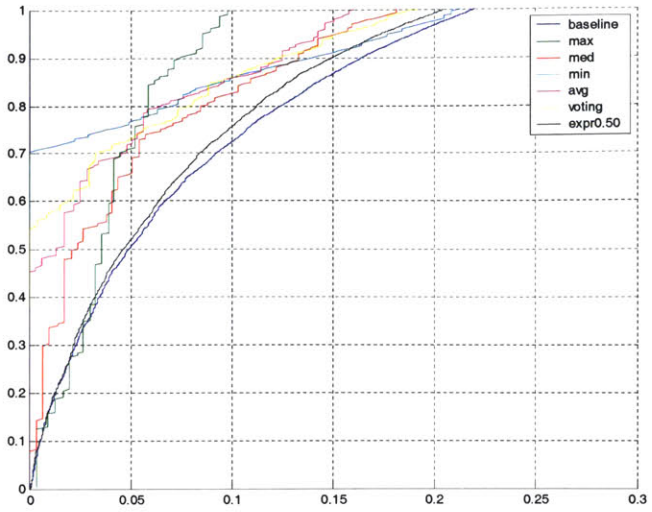

Figure 4-5: ROC Curves of Temporal Integration Strategies on dataset with variance of 0.01 and a step size of 0.30. The minimum strategy yields highest acceptance rate at 1% error rate while maximum yields the smallest error rate at full acceptance rate.

variance σ

3) generate a random walk from that point with step size d with 20-40 steps

In total, 16 datasets containing three classes were generated. The variances of the distributions used ranged from 0.01 to 0.3 and step sizes ranged from 0.05 to 0.5. Each class contained 300 sequences which ranged between 20 and 40 points in length. Plots of a few of these datasets can be seen in Figures 4-1 to 4-4.

4.1.1 Training and Classification

Each data set was split roughly in half to form training and test data. One-vs-all classifiers were then trained using SVMs with Gaussian kernels to distinguish between the classes. Each test set was then classified and the output from the SVMs were "smoothed" using the temporal integration strategies from Tables 2.1 and 2.2.

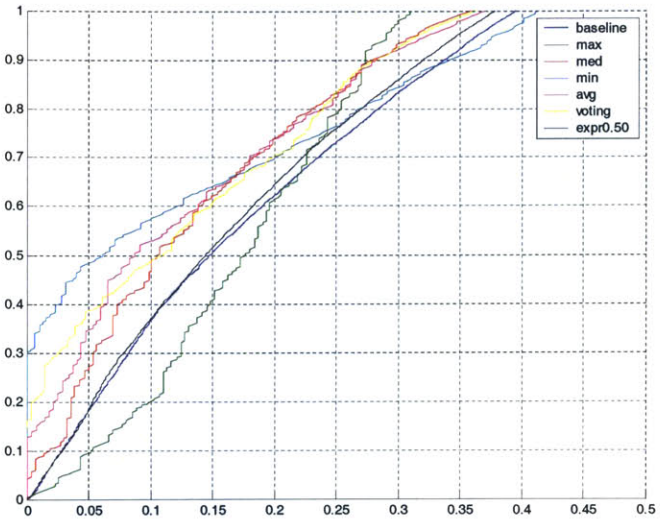

Figure 4-6: ROC Curves of Temporal Integration Strategies on dataset with variance of 0.10 and a step size of 0.50. While the effects are less noticeable, temporal integration still improves classification. The minimum strategy yields highest acceptance rate at 1% error rate while maximum yields the smallest error rate at full acceptance rate.

4.1.2 Performance

The performance of the sequence based strategies can be seen in Figures 4-5 and 4-6 in the form of combined ROC curves[1]. In these figures, we can see that the relative ordering of the ROC curves is the same in both datasets. This was characteristic of all data sets where temporal integration actually benefitted classification. Figure 4-5 shows the dataset in which the performance benefit from temporal integration is the most dramatic. Figure 4-6 shows the most inseparable dataset that still benefits from temporal integration. In data sets where temporal integration was of no use, the classes were already well separated or too inseparable for any difference to be made.

Figures 4-7 and 4-8 display the relative performance of the sequence based temporal integration strategies. Given that many of the ROC curves cross at various points in the graph, it is difficult to simply "eyeball" the best strategy. In these two figures, the relative performance of temporal integration strategies is measured for two different modes of operation: minimum false positive rate and maximum classification rate. The relative performance is determined by taking the difference between the temporal integration strategy and the baseline classification at that for that particular mode. Another point to notice in those two charts is the set of variances (denoted by σ) and step sizes (denoted by d) in which temporal integration benefits classification.

Figure 4-9 displays the relative performance benefits of the non-sequence based strategies. In all cases where these strategies improved performance, max rule yielded the best results. However, these improvements are significantly less than when the sequences are known. In many cases, performance from using non-sequence based strategies actually decreased.

4.2 Face Data and Audio Data Experiments

Sequences of face images for nine people were also studied. All images were 60x60 pixels in size and contained only the extracted facial region in grayscale. Each class

[1] A slightly modified type of ROC curve is used. To make the curves more visually comparable, acceptance rate is plotted against error rate.

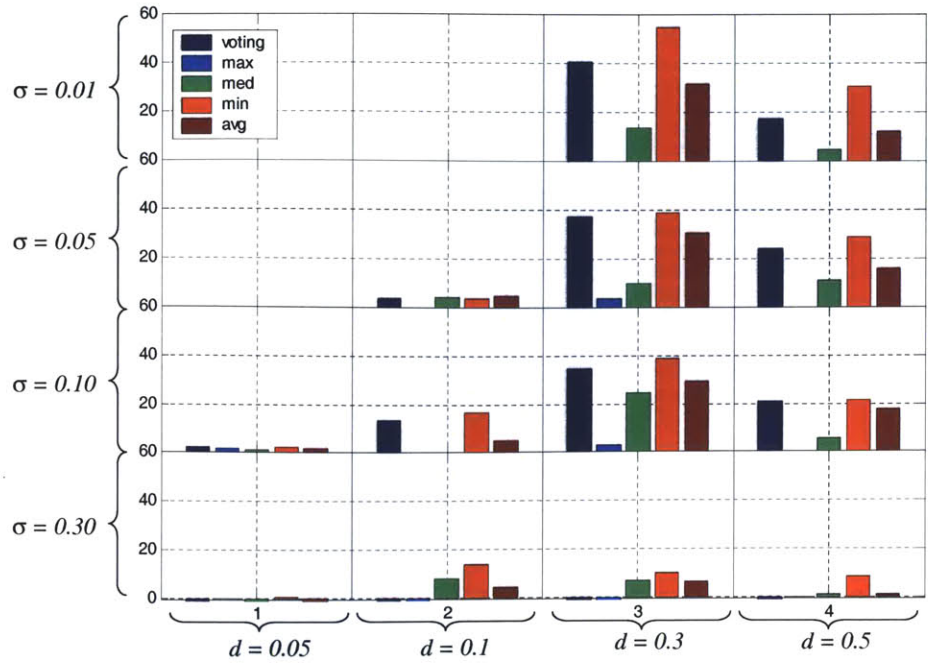

Figure 4-7: Relative Performance of Sequence Based strategies at 1% error rate. Across all datasets where temporal integration improves classification, the relative or

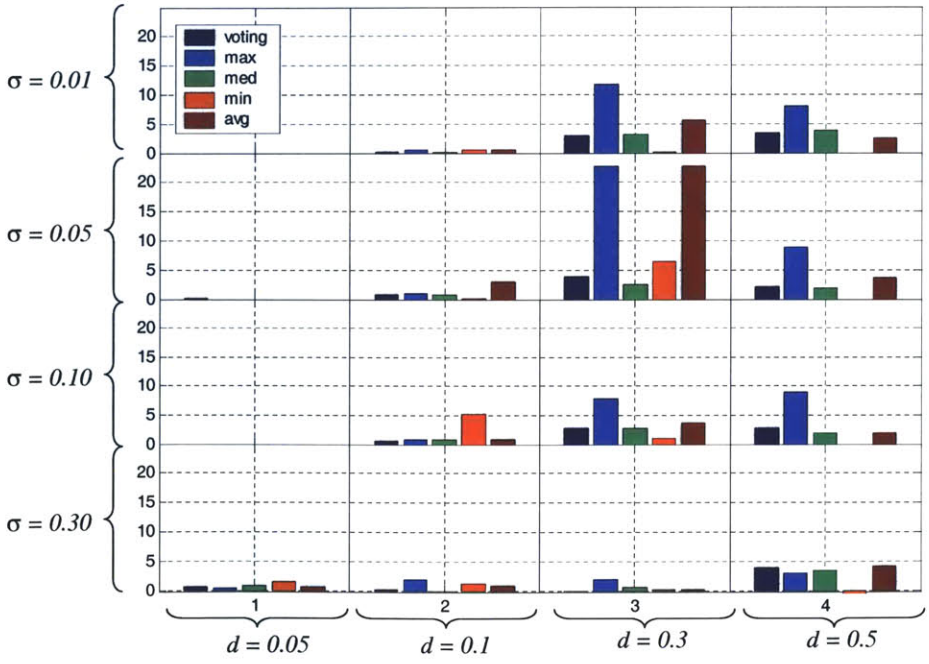

Figure 4-8: Relative Performance of Sequence Based strategies at 100% acceptance rate. Across all datasets where temporal integration improves classification, the relative ordering of the strategies remains the same.

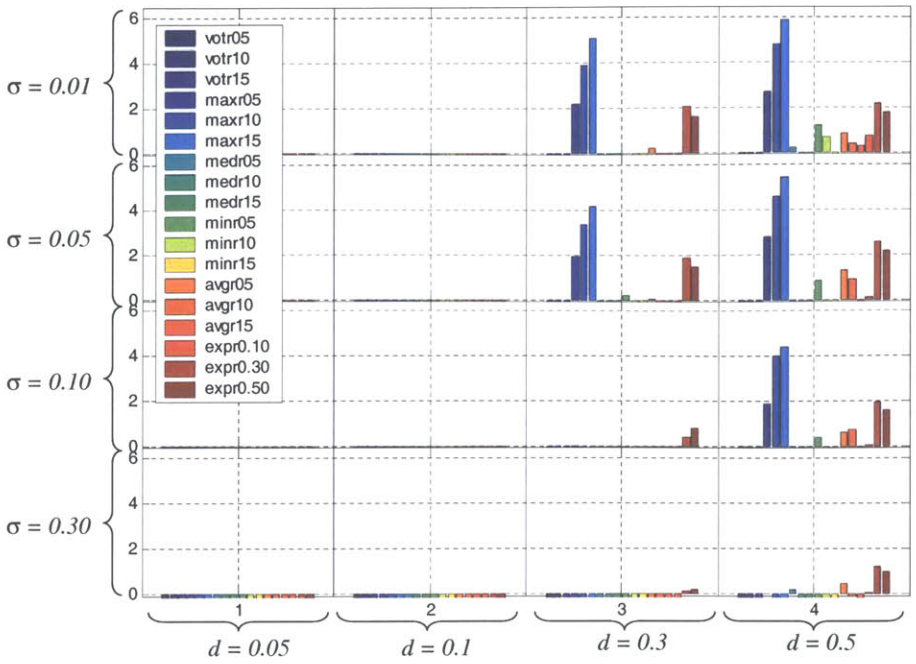

Figure 4-9: Relative Performance of Non-Sequence Based strategies at 100% acceptance rate

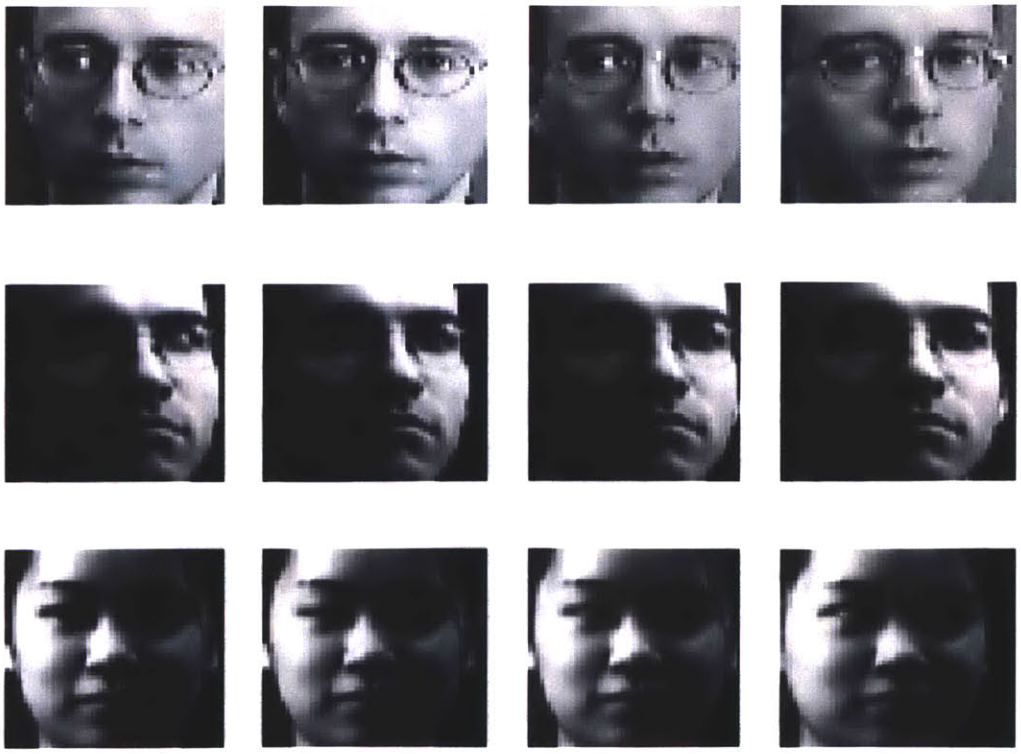

Figure 4-10: Sample images from Face Dataset

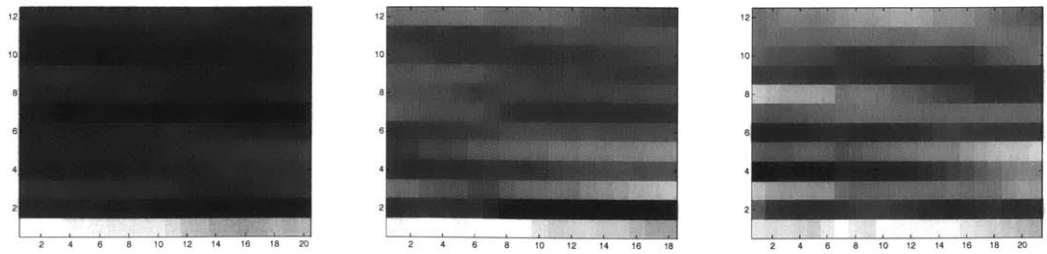

Figure 4-11: Plots of cepstral coefficients calculated from Audio Data

had 10 to 20 sequences and each sequence contained between 20 and 100 images. Sample images from this data set can be seen in Figure 4-10.

We also tested temporal integration strategies on auditory data using Japanese vowel data from the UCI Machine Learning Repository [4]. This dataset is composed of samples from nine people uttering the same vowel. For each utterance, a series of cepstral coefficients were calculated to perform speech feature extraction. The samples generated from each utterance were then treated as a sequence. Each class contained roughly 600 sequences which ranged from 10 to 40 samples in length.

Training and classification for face and audio data were conducted similarly to the training and classification of synthetic data.

4.2.1 Performance

ROC curves for face and audio data can be seen in Figures 4-12 and 4-13. In both cases, baseline classification performance is very high. However, using sequence based temporal integration strategies yielded improved results. For faces, the baseline classifier is able to achieve 88% acceptance rate at 1% error rate. However, the best temporal integration strategy is able to achieve 99% acceptance rate at 1% error rate.

For auditory data, the improvement can be seen at both 1% error rate as well as at 100% acceptance rate. The baseline classifier for auditory data has an acceptance rate of 20% at 1% error rate while using temporal integration has 86% acceptance at 1% error rate. Looking at total classification rates, the baseline classier has an error rate of 5% when all points are classified whereas temporal integration has can error rate of less than 1.5%.

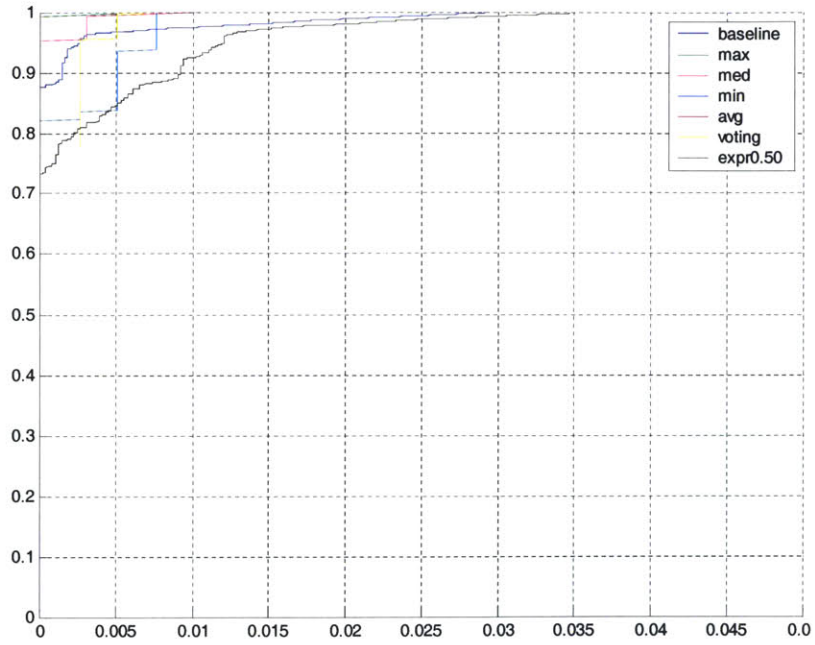

Figure 4-12: ROC Curves of Temporal Integration Strategies on Face Data. For face data we see that temporal integration still improves results. The general trends observed from the synthetic data experiments are followed here as well, although less pronounced.

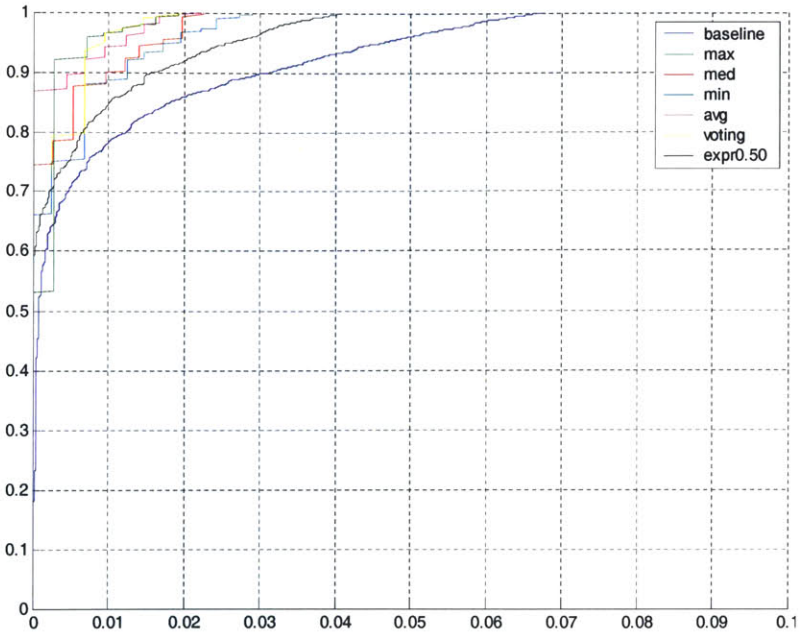

Figure 4-13: ROC Curves of Temporal Integration Strategies on Audio Data. The general trends observed from the synthetic data experiments are followed here as well.

4.3 System Experiments

In this section, data from the multi-source system developed in Chapter 2 was analyzed. The system was used to identify eight people in our office when using the coffee machine. A set of 65 videos were collected over the course of a week. Each video is 30 seconds in length on average and was recorded at 30 frames per second. Furthermore, only one person is seen per video.

4.3.1 Training and Classification

To form training and test data, one day of videos is set aside for training while the remaining videos are used for testing.

To train the face recognition classifiers, a separate set of videos was collected for each person where more frontal views of the face are seen. A set of face images for each person was generated by cropping faces detected by the face detection component. These images were then hand checked to make sure no false positives were contained. All images were then resized to 50x50 pixels and histogram normalized. Eight one-vs-all SVMs with polynomial kernels[2] were trained according to [2]. To generate the confusion matrix for this classifier, all videos in the training set are classified to count mistakes and correct hits for each person. The matrix is then column normalized and can be seen in Figure 4-14.

Training the height classifier requires one video per person. For each video, every frame is examined and the bounding box is calculated. If the bounding box is too close to the edge of the image, the frame is not used. Otherwise, the height in pixels and location of each bounding box is stored for regression as described in 3.3.2. Each training video is then classified to calculate the distribution of heights that best models the heights measured for that person. The training videos are analyzed one final time to generate the confusion matrix for height classification which can be seen in Figure 4-15.

[2]The choice of polynomial kernel here is determined by the SVM implementation that is used for real time processing.

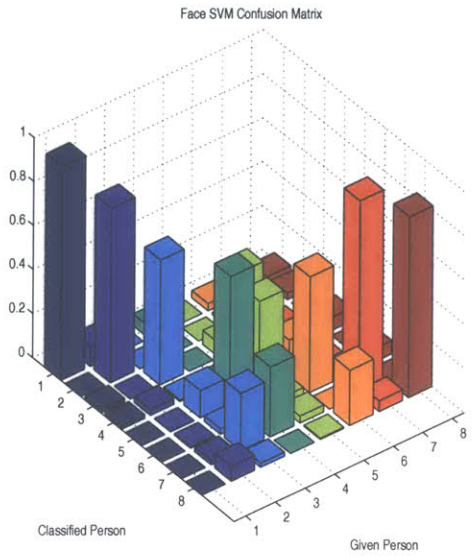

Figure 4-14: In the face classifier confusion matrix, the decently strong set of diagonal bars tells us that the face classifier is good at classifying half the people in the dataset. However, the four people in the middle often get confused for other people.

The training procedure for the color models is very similar to the training procedure for the height classifier and is detailed more in Section 3.3.3. The confusion matrix for this classifier can be seen in Figure 4-16. The only additional step required for color models is the storage of recording times. Given that color modeling is fairly time sensitive, it is necessary for the Supervisor module to be aware of the time at which the color model is based.

Classification then occurs on the remaining four days of videos that compose the test data. Each classifier is called on every frame and the outputs are given to the Supervisor which then uses confusion matrices and heuristics to determine how the outputs should be combined as outlined in Chapter 3.

4.3.2 Performance

The performance of the individual classifiers can be seen in Figures 4-17 - 4-19, where the ROC curves for baseline performance and temporal integration strategies are shown. From Figure 4-17, we see that baseline performance of the face recognition classifier is not particularly strong. However, using the average and voting temporal

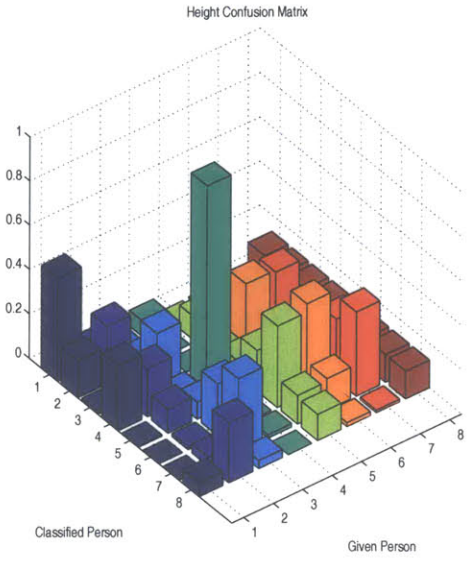

Figure 4-15: In the height classifier confusion matrix, we see that the height classifier has very poor performance in that it can only accurately classify person 4.

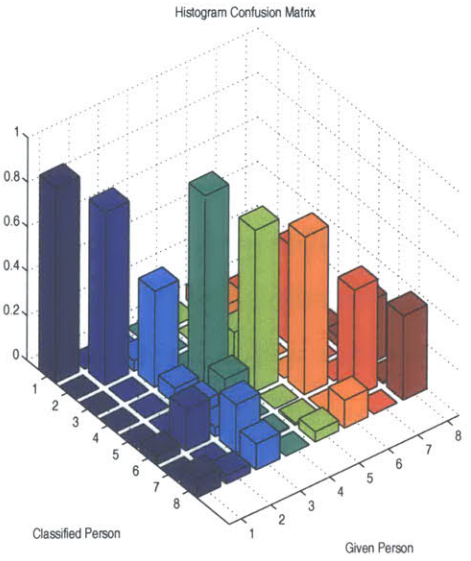

Figure 4-16: The color classifier confusion matrix shows excellent performance for same day classification.

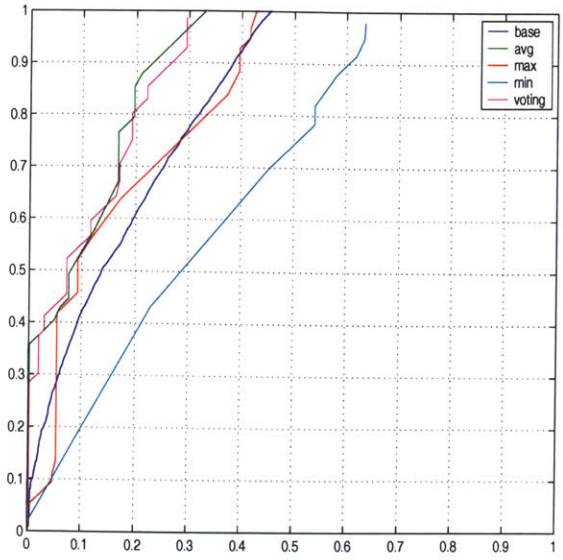

Figure 4-17: ROC curves for face classifier using temporal integration. For this classifier, temporal integration improves classification significantly.

integration strategies improves performance significantly. At 1% error rate, baseline acceptance rate is less than 10%. Using voting, the acceptance rate jumps to 28% and with averaging it goes even higher to 36%. Looking at the total acceptance rates, voting and averaging give 29% and 33% error rates, respectively, whereas baseline has a 46% error rate.

The height classifier performs poorly which can by seen by its ROC curves (Figure 4-18) and confusion matrix (Figure 4-15). Despite this, temporal integration yields noticeable improvement. At 1% error rate, the baseline height classifier has only 2% acceptance rate. However, averaging and voting have 17% and 9% acceptance rates respectively.

The color classifier has a baseline error rate of 37% at full acceptance. Using temporal integration, this error rate can be reduced to 32%. At the 1% error rate, baseline performance is close to 0 whereas the average strategy has 6% acceptance and the minimum strategy has 10% acceptance. It is important to note that these ROC curves represent the best possible performance of the color classifier because they assume that each video seen before has been labeled correctly. In the case that this assumption is not true, the performance of this classifier would decrease.

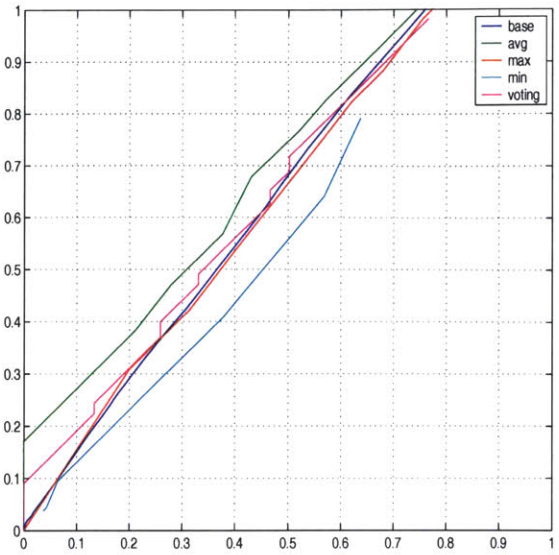

Figure 4-18: ROC curves for height classifier using temporal integration. Even for a very poor classifier, temporal integration improves classification.

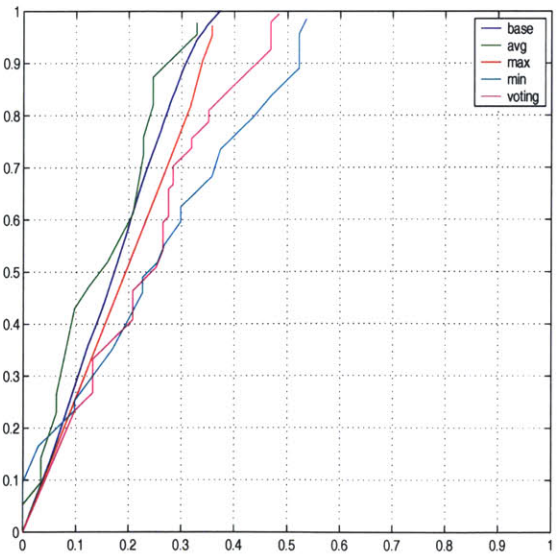

Figure 4-19: ROC curves for color classifier using temporal integration. For this classifier, the only the maximum strategy outperforms baseline.

Performance of the complete system where classifier combination and temporal integraion are used can be seen in Figure 4-20. Baseline performance, in this case, is classification based solely on classifier combination. At full acceptance, baseline performance has a 70% error rate. Using the maximum temporal integration strategy reduces the error rate to 45%. Meanwhile, all other strategies have error rates between 45% and 70% at full acceptance. At 1% error rate, baseline has 1% acceptance whereas maximum strategy has 22% acceptance rate and averaging has a 9% acceptance rate.

Temporal integration clearly improves our method of combining classifiers based on empirical prior and classifier weighting. In Figure 4-21, we see that our temporally integrated combination of classifiers performs better than each classifier individually. To simply replot all the ROC curves in a single graph does not allow for accurate comparison since the curves from the different classifiers will be measured on different sets of frames. To "normalize" the plots, ROC curves were generated for the individual classifiers based on the set of frames seen by the classifier combination method which only requires one classifier to be present per frame. At 1% error rate, temporal integration of combined classifiers has an acceptance rate of 23% whereas the best single classifier has only an acceptance rate of 9%. Evaluating at full acceptance is not possible since no individual classifier is able to classify all frames. However, it is worth noting that each all individual classifier ROC curves lie underneath the ROC curve for temporal integration of combined classifiers.

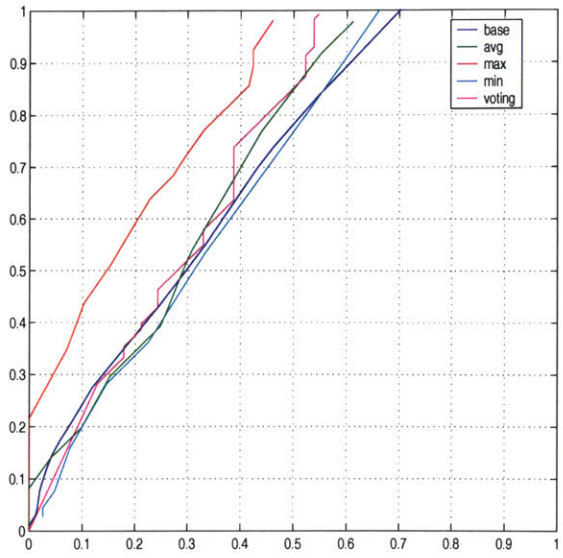

Figure 4-20: ROC Curves for Classifier Combination and Temporal Integration. Using the maximum temporal integration strategy improves performance significantly over baseline which represent classifier combination.

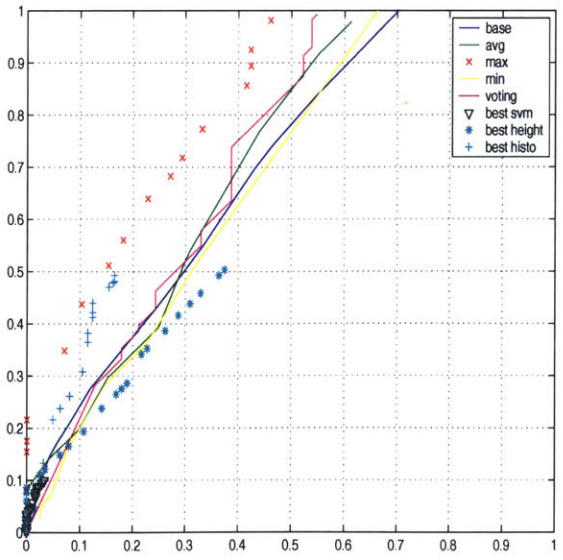

Figure 4-21: ROC Curves for Classifier Combination and Individual Classifiers. Here we see that the performance of using temporal integration and classifier combination (denoted by the red x's) outperforms all individual classifiers (denoted by +'s, stars, and triangles).

Chapter 5

Discussion and Conclusion

In this thesis, classifier combination and temporal integration were studied in three stages. In the first stage, experiments in temporal integration were conducted on clean, controlled data. In these tests, we saw that accuracy improved up to 40% depending on the class separability and operating regime of the classifier. In the second stage, real data in the form of faces and voice recordings yielded similar results. In the final stage where actual surveillance type videos were used, there was still significant performance increases in spite of poor classifiers.

5.1 Synthetic Data Experiments

From the synthetic data experiments, a few observations can be made. Under the assumption of gaussian noise, the same temporal integration methods give the same performance across the datasets with varying separability. Further, for a verification task where low error rate is the main concern, the minimum strategy was found to always perform best. For maximum classification, using the maximum temporal integration strategy always yielded the lowest error rate. Improvements found were significant and are outlined in the Table 5.1.

Given the significant difference in performance gains between sequence based and non-sequence based temporal integration strategies, it is clear that knowing the sequence boundaries is necessary for temporal integration. Otherwise, the sequence of

Improvements from Temporal Integration in Synthetic Data		
Category	*Baseline Performance*	*Temporal Integration*
Error rate at 100% acceptance	22%	10%
Acceptance rate at 1% error	15%	70%

Table 5.1: Summary of Performance Improvement in Synthetic Data

Improvements from Temporal Integration in Face and Audio Classification		
Face Data		
Category	*Baseline Performance*	*Temporal Integration*
Error rate at 100% acceptance rate	3%	1%
Acceptance rate at 1% error rate	88%	99%
Audio Data		
Category	*Baseline Performance*	*Temporal Integration*
Error rate at 100% acceptance rate	5%	1.5%
Acceptance rate at 1% error rate	20%	86%

Table 5.2: Summary of Performance Improvement in Face and Audio Data

sequences that are classified may as well be random points.

5.2 Face and Audio Data Experiments

From studying face and audio data, we saw that the general trends observed from the synthetic data experiments are preserved. Actual performance improvements from temporal integration are summarized in Table 5.2.

5.3 System Data Experiments

In the system data experiments, temporal integration and classifier combination were able to improve accuracy despite having poor classifiers. Using temporal integration, the performance benefits for each classifer can be seen in Table 5.3.

One area that deserves further discussion is the poor performance of the height classifier. Given the realistic data presented to the classifier which contained greatly varied lighting conditions, accurate background subtraction containing the entire person was difficult to perform. As a result, bounding box measurements that are used in calculating the height are off by several pixels causing the estimated height to be

Improvements from Temporal Integration in System Classifier		
Face Classifier		
Category	*Baseline Performance*	*Temporal Integration*
Error rate at 100% acceptance rate	46%	29%
Acceptance rate at 1% error rate	10%	36%
Height Classifier		
Category	*Baseline Performance*	*Temporal Integration*
Error rate at 100% acceptance rate	78%	74%
Acceptance rate at 1% error rate	2%	17%
Color Classifier		
Category	*Baseline Performance*	*Temporal Integration*
Error rate at 100% acceptance rate	37%	32%
Acceptance rate at 1% error rate	1%	10%

Table 5.3: Summary of Performance Improvement in System Classifiers

inaccurate. Improvement of this classifier is discussed in the Future Work section.

Meanwhile, looking at overall results. Temporal integration was able to reduce the error rate of our classifier cominbation scheme by 35% when classifying all frames in the test set. Temporal integration of the combined classifiers also outperformed the best temporal integration method of any individual classifier.

5.4 Further Work and Improvements

This system can be improved in several areas. First, better classifiers should be used. The face classifier can easily be improved by adding more images per person with varied lighting conditions to the training data. In addition, the height classifier should be improved by using more constant lighting conditions. Additional accuracy can also be gained by using true camera calibration for measuring heights rather than the method described in this thesis.

One important area of future work is to expand on how the classifiers should be weighted (ie. setting $P(\lambda|x)$ even more dynamically). This can be done through the use of additional heuristics for determining classifier confidence or through modelling the distributions of the weights.

Bibliography

[1] B. Achermann and H. Bunke. Multimodal people id for a multimedia meeting browser. In *ACM Multimeida '99*, pages 159–168, 1999.

[2] P. Ho B. Heisele and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *International Conference on Computer Vision (ICCV'01)*, volume 2, pages 688–694, 2001.

[3] T. Poggio B. Heisele and M Pontil. Face detection in still gray images. Technical Report A.I. Memo No. 1687, MIT, May, 2000.

[4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. Technical Report http://www.ics.uci.edu/~mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[5] S.M. Omohundro C. Bregler and Y. Konig. A hybrid approach to bimodal speech recognition. In *28 Asilomar Conference on Signals, Systems and Computers*, volume I, pages 556–560, 1994.

[6] B. Heisele C. Nakajima, M. Pontil and T. Poggio. Full-body person recognition system.

[7] B. Heisele C. Nakajima, M. Pontil and T. Poggio. People recognition in image sequences by supervised learning. Technical Report A.I. Memo No. 1688, MIT, June, 2000.

[8] C. Neti *et al.* Large vocabulary audio-visual speech recognition: A summary of the johns hopkins summer 2000 workshop. In *Workshop on Multimedia Signal Processing*, pages 619–624, 2001.

[9] S. Bengio H. Bourlard and K. Weber. Towards robust and adaptive speech recognition models. Technical Report IDIAP Research Reprort No. IDIAP-PR 02-01, 2002.

[10] M. Hatef J. Kittler and R. Duin. Combining classifiers. In *Proceedings of ICPR '96*, pages 897–901, 1996.

[11] R. Gross J. Kominek Y. Pan J. Yang, X. Zhu and A. Waibel. In *Combination of Face Classifiers for Person Identification*, pages 416–420, 1996.

[12] J. Kitter and F. Alkoot. Sum versus vote fusion in multiple classifier systems. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25, pages 110–115, Jan 2003.

[13] H. Attias M. Beal and N. Jojic. Audio-visual sensor fusion with probabilistic graphical models. In *ECCV'02*, May 2002.

[14] C. Papageorgiou and T. Poggio. Trainable pedestrian detection. In *Proceedings of International Conference on Image Processing*, 1999.

[15] R. Rifkan. Svmfu: Support vector machine package. Technical Report http://fpn.mit.edu/SvmFu/index.html, MIT, 2000.

[16] S. Marcel S. Bengio, C. Marcel and J. Mariethoz. Confidence measures for multimodal identity verification. Technical Report IDIAP Research Reprort No. IDIAP-PR 01-38.

[17] V. Vapnik. *Statistical Learning Theory.* Springer-Verlag, 1998.