Mobile Biometrics (MoBio): Joint Face and Voice Verification for a Mobile Platform


Abstract—The Mobile Biometrics (MoBio) project combines real-time face and voice verification for better security of personal data stored on, or accessible from, a mobile platform.

I. INTRODUCTION

Modern smartphones not only have the memory capacity to store large amounts of sensitive data (e.g. contact details, personal photographs) but also provide access, via mobile internet, to even larger quantities of personal data stored elsewhere (e.g. in social networking, internet banking and e-mail). Though passwords provide some protection against unauthorized access to this data, the sheer number of such sites make it impractical to remember a different password for each, yet using the same password for all is risky (all sites are compromised if the password is discovered) and storing the password on the device is not advisable for mobile devices that are easily lost or stolen.

An alternative approach is to use biometrics: physical characteristics (e.g. fingerprints) that are unique to you and easy to remember, yet hard for you to lose or for someone else to steal. Though biometric systems require data capture facilities, modern smartphones conveniently come equipped with a video camera and microphone. The Mobile Biometrics project exploits these features to combine face and voice biometrics for secure yet rapid user verification to ensure that that someone who wants to access data is entitled to do so.

A major challenge is to capture the signal in a way that is not confused by day-to-day variations. A face, for example, looks different depending on expression and lighting, and change over time (e.g. when growing a beard); the voice sounds different depending on your condition (e.g. having a sore throat) and is difficult to separate from background noise in noisy environments. We must also make the system robust to ‘spoofing’ by impostors; checking that the lips move, for example, ensures that photographs do not fool the system.

The MoBio project combines a software verification layer that uses your face and voice, captured by a mobile device, to ensure that you are who you claim to be (Figure 1). This article summarizes the state of the art methods we have developed that authenticate the face and voice, combine these biometrics to make the system more robust, and update system models to allow for changes in appearance over time – all within the hardware limitations of a consumer-grade mobile platform. Though other studies have investigated face and voice authentication [5], [19], MoBio is the first to assess bimodal authentication under the challenging conditions imposed by a mobile architecture (e.g. limited processing power, shaking of the handheld camera).

The key technical contributions of the project include:

• Rapid face detection, with improved performance under adverse lighting [18] and an exponential reduction in false positives [2].
• A threefold reduction in computation time (from ~45ms to ~15ms per image) during facial feature localization on the Nokia N900 mobile device, thus achieving framerate performance [20].
• Novel image descriptors for improved face verification [1], [4], [8].
• Efficient speaker verification via a 25-30 times speedup to the i-vector modelling approach [7], a novel pairwise feature that is 100-1000 times more efficient than a Gaussian Mixture Model approach [16], and methods to decouple core speaker recognition from session variability modelling when training data is limited [9].
• A reduction of up to 30% in Equal Error Rate (EER) by adapting the classifier score to the estimated capture conditions, and a reduction of up to 50% in EER by adapting the model to accommodate a wider range of capture conditions [15].
• In-depth studies of Data Fusion algorithms, taking into account cost and signal quality [13], [14].
• A new, publicly available bimodal database containing videos of 150 subjects, captured over 18 months on a

1In this article, we briefly summarize key results – full technical details and experimental results are available from the references.
handheld mobile device in relatively uncontrolled conditions.

II. FACE ANALYSIS

A. Face Detection

To capture the user’s appearance, we begin with an image that contains (somewhere) the user’s face. Our first job is then to localize the face in the image to get a rough estimate of its position and size (Figure 2). This is hard because appearance varies in the image and our system must detect faces regardless of shape and size, identity, skin colour, expression and lighting. Ideally, it should also handle different orientations and occlusion, but in mobile verification we assume the person is looking almost directly into the camera most of the time.

We approached this problem by classifying every plausible region of the image as either face or not face based on its image properties using modern Pattern Recognition methods to learn the characteristics that differentiate faces from non-faces. Two key considerations were how to summarize the image region in a compact form (i.e. compute its feature vector) and how to classify the image region based on its features.

When searching an image for the first time, there are hundreds of thousands of possible locations for the face and it is important that each image region be summarized quickly. We used a variant of the Local Binary Pattern [12] to summarize local image statistics around each pixel with a binary code indicating the image gradient direction with respect to its eight neighbours (Figure 4). We then fed the histogram over transformed values for each patch into a classifier to decide whether the patch should be labelled as ‘face’ or ‘not face’, using a cascade of increasingly complex classifiers [21] to discard non-face regions efficiently (Figure 2). In practice, a cascade rejects most image regions (that look nothing like a face) using simple but efficient classifiers its early stages; the more accurate but computationally demanding classifiers are reserved for the more challenging image regions that look most face-like.

Our experiments on standard datasets (e.g. BANCA [3] and XM2VTS [11]) suggest that we detect over 97% of the true faces. In our application, we also prompt the user to keep their face in the centre of the image so that we can restrict the search to a smaller region, thus permitting a more discriminative image representation that increases detection rates further.

Developing this baseline system further, we showed that applying Haar-like features to the Local Binary Pattern descriptor can improve performance, especially under adverse lighting conditions [18]. We also developed a principled system that exponentially reduced false positives (background regions labelled as ‘face’) and clusters of several detections around the same true face, with little or no reduction in the true acceptance rate [2].

B. Face Normalization

Though we could try to recognise the user from the rectangular image region that approximately surrounds the face in the image, performance will be impaired by factors such as background clutter, lighting, and facial expression. Our next task, therefore, is to remove as many of these effects as possible by normalizing the face so that it has similar properties (with respect to shape and texture) to the stored model of the user (Figure 3). First, we localize individual facial features such as the eyes, nose, mouth and jawline, and use these to remove any background that is not relevant for verification. Next, we stretch the face region to fit a pre-defined shape, thus compensating for differences due to the orientation of the head in three dimensions, the person’s expression, and the shape of their face (a weak cue for verification). Finally, we normalize the lighting by adjusting brightness and contrast to some fixed values. The resulting normalized image can then be directly compared with a similarly normalized model image for accurate verification.

To locate facial features, we fitted a deformable model to the image using a novel version of the Active Appearance Model (AAM [6]) that we developed specifically for a mobile architecture using modern Machine Learning techniques [20]. The AAM uses statistical models of shape and texture variation – learned from a set of training images where facial features have been labelled by hand – to describe the face using only a few parameters; it also learns to detect when the model is in the wrong place, and to apply an appropriate correction to the parameters to align the model with the image. To predict these corrections, we trained a linear regressor to learn the relationship between sampled image data and true parameter values using image samples with known misalignments.

When fitting the model to a new image, we initially aligned the model with the coarse face detection result, then sampled and normalized the corresponding part of the image (Figure 3).
The AAM then predicted and applied a correction to the shape and pose parameters to align the model more closely to the image. By repeating this sample-predict-correct cycle several times we converged on the true feature locations, giving a normalized texture sample for verification.

In a comparison with existing methods, our approach achieved comparable or better accuracy (typically within 6% of the distance between the eyes) while achieving a threefold speedup on a Nokia N900, reducing processing time from 44.6ms to 13.8ms and therefore reaching frame-rate performance [20]. This performance was achieved using a generic model, trained from publicly available datasets. Although we could adapt this model to any specific user by treating the estimated feature positions as ‘ground truth’ labels to retrain the predictor (offline, if necessary), our results suggest that performance would improve little in return for the added computational cost.

C. Face Verification

Given a normalized image of the face, the final step is to assign a score describing how well it matches the stored model for the claimed identity, and to use that score to decide whether to accept or reject that person’s claim (Figure 4). Again, we treat this as a classification problem but here we want to label that person as a client or an impostor based on how they look, as summarized by their image features. Clients are given access to the resource they need; impostors are not.

Since illumination conditions can drastically alter someone’s appearance, we apply gamma correction, Difference of Gaussian filtering and variance equalization to remove as many lighting effects as possible. For added robustness we subdivide the processed images into non-overlapping subwindows to make the descriptor more robust to occlusion, and compute the Local Binary Pattern (LBP) value for every pixel over three scales. We then summarize every window by its LBP histogram and concatenate all histograms to give a feature vector for the image (Figure 4).

When classifying an observed feature vector as client or impostor, we compute its distance to the stored model of the claimed identity. Although we could make a decision based on this similarity measure alone, we instead use a robust likelihood ratio whereby the distance to a background model provides a reference that expresses how much more than average the observation matches the claimed identity, thus indicating our confidence in the classification. Applying this system to the BANCA dataset [3], we were able to achieve half total error rates (where false acceptances are as likely as false rejections) of around 5%.

In developing these methods, we proposed a number of novel image descriptors that were shown to improve recognition performance. One based on Local Phase Quantization was designed to work with out-of-focus images and achieved a recognition rate of 93.5%, compared with 70.1% for Local Binary Patterns on a face image blurred with a Gaussian of standard deviation $\sigma = 2$ pixels [1]. We then developed this descriptor further to include information at multiple scales, improving recognition rates in some cases from 66% to 80% on a more challenging dataset with widely varying illumination [4]. We also showed that, when available and used correctly, colour information can be exploited to reduce error rates by several percent [8].

III. Voice Analysis

Though face verification technology is maturing, we also exploit the fact that we have a microphone at our disposal by including voice-based speaker verification in our system.

A. Voice Activity Detection

Given a sound sample that was captured using the mobile’s microphone, our first step is to separate speech (which is useful for speaker recognition) from background noise (which is not). As in face detection, however, speech detection is complicated by variation from speaker to speaker (e.g. due to characteristics of the vocal tract, learned habits and language) and from session to session for the same speaker (e.g. as a result of having a cold).

To summarize someone’s voice, we represent the slow variation in the shape of the vocal tract by a feature vector summarizing frequency characteristics over a small window (on the order of a few tens of milliseconds) around any given time. Cepstral analysis computes this spectrum via a Fourier Transform and decomposes its logarithm by a second Fourier Transform or a Discrete Cosine Transform. Mapping the spectrum into the mel scale (where distances more closely match perceived differences in pitch) before the second decomposition gives mel-frequency cepstral coefficients (MFCCs).

A popular approach to classifying these feature vectors as speech or non-speech uses a Gaussian Mixture Model (GMM) for each of the two classes, discarding the temporal ordering of feature vectors and low-pass smoothing the output classifier. Though this has proved to be an effective technique for examples with a high signal-to-noise ratio, in environments with a lot of background noise more complex methods are required that use more than the signal energies.

We instead use an Artificial Neural Network to classify MFCC vectors, derived from a longer temporal context of around 300ms, as either one of 29 phonemes or as non-speech to give an output vector of posterior probabilities corresponding to the 30 classes. These vectors are smoothed.
over time using a Hidden Markov Model to account for the (language-dependent) known frequency of phoneme orderings learnt from training data, and the 29 phoneme classes are then merged to form the ‘speech’ samples.

Because this approach is computationally demanding (and therefore not well suited to an embedded implementation), we also proposed a simpler feature set, denoted Boosted Binary Features [17], that were based on the relationship between pairs of filter responses and achieved performance at least as good as existing methods (~65% correct classification over 40 possible phonemes) while requiring only modest computation.

B. Speaker Verification

Having discarded background noise, we then use the useful segments of speech to compute how well the person’s voice matches that of the claimed identity, and decide whether to accept or reject their claim. To describe the voice, we use 19 MFCCs (computed over a 20ms window) plus an energy coefficient, augmented with their first and second derivatives. After removing silence frames via voice activity detection, we apply a short-time cepstral mean and variance normalization over 300 frames.

As a baseline, to classify the claimant’s feature vector we use Joint Factor Analysis based on a parametric Gaussian Mixture Model, where the weights and covariances of the mixture components are learnt at the outset but the centres are specified as a function of the data (ideally, of the speaker and the session). The weights, covariances and means of the mixture components are learnt using a large cohort of known speakers, pooling over sessions to reduce inter-individual variability. The session-subspace is then learnt from training data, and the 29 phoneme classes are then estimated (since we are not interested in whether the sessions match, only the speaker) and compute the likelihood of the test session. The session-subspace is then learnt from most of what is left.

When testing, we use every training example to estimate the speaker and session, and generate a client-specific model by modifying the world model. We then discard the session estimate (since we are not interested in whether the sessions match, only the speaker) and compute the likelihood of the test example given the speaker-specific model. Score normalization then gives a measure for us to use for classification.

On the BANCA dataset [3], this baseline system achieved equal error rates of around 3-4% for speaker verification. One of ways in which we developed this method was based on the related i-vector estimation approach (the current state of the art in speaker recognition), where we showed that it was possible to make speaker modelling 25-50 times faster using only 10-15% of the memory, yet with only a small penalty in performance (typically increasing Equal Error Rate from only 10-15% of the memory, yet with only a small penalty in performance [9]). This is a useful result since it opens up the possibility for optimizing the models independently, possibly using different optimization criteria, since the two are largely uncorrelated. Decoupling parameters also results in more stable systems where training data are limited.

In a third study we showed that the pairwise features [17] could produce performance comparable with GMM-based approaches in speaker verification [16]. Specifically, this boosted binary feature-based system achieved a Half Total Error Rate (HTER) of 17.2% compared with a mean HTER of 15.4% across 17 alternative systems, yet was 100-1000 times more efficient.

IV. Model Adaptation

One challenge with biometric verification is accommodating factors that change someone’s appearance over time – either intentionally (e.g. makeup) or unintentionally (e.g. wrinkles) – as well as external influences in the environment (e.g. lighting, background noise) that affect system performance. Therefore, the model of the user that was created when they enrolled cannot remain fixed – it must adapt to current conditions and adjust its criteria for accepting or rejecting a claim accordingly.

In experiments with face verification, we began by building a generic model of appearance from a large corpus of training data including many individuals; this enabled us to model effects such as lighting and head pose that were not present in every individual’s enrolment data. We then adapted this generic model to each specific user by adjusting model parameters based on user-specific training data. In our case, we used a Gaussian Mixture Model to represent facial appearance of the known conditions most closely matched the current environment, and adapted the classifier score accordingly.

To account for changes in the capture environment (e.g. the BANCA dataset contains examples captured under controlled, adverse and degraded conditions), we computed parameters of error distributions for each condition, \( q \), independently during training and used score normalization such as the Z-norm,

\[
z_q(y) = \frac{y - \mu_q}{\sigma_q},
\]

or Bayesian-based normalization (implemented via logistic regression),

\[
P(q|y) = \frac{1}{1 + \exp(-\alpha_q y - \beta_q)},
\]
to reduce the effect of capture conditions (where \( \mu_q, \sigma_q, \alpha_q \) and \( \beta_q \) are parameters estimated by learning). When testing, we computed measures of signal quality that identified which of the known conditions most closely matched the current environment, and adapted the classifier score accordingly.

In our experiments [15], normalizing the score reduced the equal error rate in some tests by 20-30% (from 19.59% to 15.31% for the face; from 4.80% to 3.38% for speech) whereas adapting the model to capture conditions had an even greater effect on performance, reducing equal error rates by over 50% in some trials (from 19.37% to 9.69% for the face; from 4.80% to 2.29% for speech).

V. Data Fusion

At this point, for every sample in the video sequence we have a score that tells us how much the person looks like their
claimed identity and another score for how much they sound like their claimed identity. To give a system that performs better than either biometric on its own, we fuse these two modalities either by classifying each modality independently and feeding the resulting pair of scores into a third classifier (score-level fusion), or by fusing the features and passing the result to a single classifier (feature-level fusion). Since we are concerned with video sequences it is also beneficial to fuse scores (or features) over time.

A naïve approach to score-level fusion pools data over time by averaging scores over the sequence whereas more principled methods model the distribution of scores over the observed sequence and compare this to distributions, learnt from training data, that correspond to true and false matches. As a baseline, we computed non-parametric statistics (such as mean, variance and inter-quartile range) of the score distributions and separated true and false matches using a classifier trained via logistic regression. Again, score normalization can be used to ensure that the outputs from different sensing modalities are comparable, while also taking into account the quality of the signal [13], [14].

Though score-level fusion is popular when using proprietary software (where the internal classifier workings are hidden), feature-level fusion can capture relationships between the two modalities. Feature-level fusion may, however, result in a large joint feature space where the ‘curse of dimensionality’ becomes problematic, and we must also take care when fusing sources with different sampling rates (e.g. video and audio).

We therefore developed a novel feature-level fusion technique (dubbed the ‘boosted slice classifier’) that searches over the space of feature pairs (one face and one speech) to find the pair for which quadratic discriminant analysis (QDA) minimizes misclassification rate, iteratively reweighting training samples in the process. Although in some experiments this performed no better under controlled conditions, it outperformed the baseline score-level fusion system when one modality was corrupted, confirming that fusion does indeed make the system more robust.

In a different experiment, the benefit of fusing modalities was more pronounced (Figure 5a), as indicated by its Detection Error Tradeoff (DET) curve. This illustrates the tradeoff between false rejections and false acceptances for varying thresholds of the classifier score – accepting more claimants reduces false rejections but increases false acceptances (and vice versa).

VI. MOBILE PLATFORM IMPLEMENTATION

Since we want to run the system on a mobile device, we need to consider the limitations of the available hardware such as low-power processing, a fixed-point architecture and limited memory. We therefore carried out experiments that looked at the effect on accuracy when making approximations that would make the system more efficient.

One very effective modification was to implement as many methods as possible using fixed-point (rather than floating-point) arithmetic; although some modern devices are equipped with floating-point units, they are far from common and are less efficient. Other ways to reduce computation included applying an early stopping criterion for the face detection and reducing the number of iterations used in facial feature localization. Because reducing memory consumption also has performance benefits, we made further gains by reducing parameters such as the number of LBP scales, the dimensionality of feature vectors and the number of Gaussian mixture components for speech verification.

As a quantitative evaluation of these approximations, we rated 1296 scaled systems (48 face × 27 speech) by two criteria: an abstract cost reflecting both memory consumption and speed; and resultant generalization performance measured by equal error rate. As expected, increasing efficiency came at a cost in accuracy whereas increasing complexity resulted in much smaller gains (Figure 5b).

To test the system under real conditions, we developed a prototype application for the Nokia N900 that has a front-facing VGA camera for video capture, a Texas Instruments OMAP3 microprocessor with a 600MHz ARM Cortex-A8 core, and 256Mb RAM. Using GTK for the user interface and gstreamer to handle video capture (Figure 6), we achieved near frame-rate operation for the whole identity verification system and frame-rate operation for some modules (e.g. facial feature localization).
related projects is that the MoBio system is a bimodal system that uses the face and the voice, and therefore needs a bimodal dataset on which to evaluate performance. Many publicly available datasets, however, contain either face data or voice data but not both. Even those few that do include both video and audio [5], [19] capture the data using high-quality cameras and microphones under controlled conditions, and are therefore not realistic for our application; we are limited to a low quality, hand-held camera. The few that come close (e.g. the BANCA dataset [3]) use a static camera and so do not have the image ‘jitter’, caused by small hand movements, that we have to deal with.

Since we anticipate other mobile recognition and verification applications in the future, we used a handheld mobile device (the Nokia N93i) to collect a new database that is realistic and is publicly available3 for research purposes (Figure 7). This database was collected over a period of 18 months from six sites across Europe, contains 150 subjects and was collected in two phases for each subject: the first phase includes 21 videos per session for six sessions; the second contains 11 videos per session for six sessions. A testing protocol is also supplied with the data, defining how the database is split into training, development and test sets, and how evaluation scores are computed. This protocol was subsequently applied in a competition entered by fourteen sites: nine for face verification and five for speaker verification [10].

VIII. SUMMARY

We have outlined a new system for identity verification on a mobile platform that uses both face and voice characteristics. Specifically, state-of-the-art video modules detect, normalize and the face, while audio modules segment speech and verify the speaker. To ensure the system is robust, we adapt models to the estimated capture conditions and fuse signal modalities, all within the constraints of a consumer-grade mobile device.

The aim of the MoBio project was to develop a robust and secure verification system for mobile applications. Mobile internet is an obvious example where biometric verification may complement (or replace) traditional access methods such as passwords. Other potential applications include using biometrics to lock and unlock the phone, and mobile money transactions.

VIII. S. DATABASE AND PROTOCOL

One major difference between the MoBio project and other related projects is that the MoBio system is a bimodal system that uses the face and the voice, and therefore needs a bimodal dataset on which to evaluate performance. Many publicly available datasets, however, contain either face data or voice data but not both. Even those few that do include both video and audio [5], [19] capture the data using high-quality cameras and microphones under controlled conditions, and are therefore not realistic for our application; we are limited to a low quality, hand-held camera. The few that come close (e.g. the BANCA dataset [3]) use a static camera and so do not have the image ‘jitter’, caused by small hand movements, that we have to deal with.

Since we anticipate other mobile recognition and verification applications in the future, we used a handheld mobile device (the Nokia N93i) to collect a new database that is realistic and is publicly available3 for research purposes (Figure 7). This database was collected over a period of 18 months from six sites across Europe, contains 150 subjects and was collected in two phases for each subject: the first phase includes 21 videos per session for six sessions; the second contains 11 videos per session for six sessions. A testing protocol is also supplied with the data, defining how the database is split into training, development and test sets, and how evaluation scores are computed. This protocol was subsequently applied in a competition entered by fourteen sites: nine for face verification and five for speaker verification [10].

Acknowledgements

This work has been performed by the MOBIO project 7th Framework Research Programme of the European Union (EU), grant agreement number 214324. The authors would like to thank the EU for the financial support and the consortium partners for a fruitful collaboration, especially Visidon Ltd. for their invaluable work in developing the mobile phone user interface. For more information about the MOBIO consortium please visit http://www.mobipointproject.org.

References


3http://www.idiap.ch/dataset/mobio

