Mobile Biometrics:

Combined Face and Voice Verification for a Mobile Platform

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.

Phil Tresadern
and Timothy F. Cootes
University of Manchester

Norman Poh
University of Surrey

Pavel Matejka
Brno University of Technology

Abdenour Hadid
University of Oulu

Christophe Lévy
University of Avignon

Christopher McCool
and Sébastien Marcel
Idiap Research Institute

Modern smartphones not only have the memory capacity to store large amounts of sensitive data (such as contact details and personal photographs), they also provide access, via the mobile Internet, to personal data stored elsewhere (for example, on social networking sites, through Internet banking, and in email). Although passwords provide protection against unauthorized access to this data, the sheer number of such sites makes it impractical to remember a different password for each one, yet using the same password for all is risky (all sites would be compromised if the password were discovered). Furthermore, storing the password on the device isn't advisable, because mobile devices are easily lost or stolen.

An alternative is to authenticate yourself using biometrics—physical characteristics (such as fingerprints) unique to you and easy to remember yet not easily lost or stolen. Although biometric systems require data-capture facilities, modern smartphones come equipped with a video camera and microphone. The Mobile Biometrics (MoBio) project exploits these features to combine face and voice biometrics for secure yet rapid user verification.

A major challenge, however, is to capture the signal in a way that isn't confused by day-to-day variations. A face, for example, looks different depending on the expression, and lighting and can change over time (such as when growing a beard). Similarly, a voice can sound different depending on the user's health (for example, if the user has a sore throat) and is difficult to separate from background noise in loud environments. We must also make the system robust to spoofing by impostors; checking that the lips move, for example, ensures that photographs don't fool the system.

The MoBio project provides 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 (see Figure 1). Although other studies have investigated face and voice authentication,[1,2] MoBio is the first to assess bimodal authentication under the challenging conditions of a mobile architecture.

Face Analysis

Using the integrated camera on the mobile device, we can verify that users are who they claim to be from their facial biometrics. First, however, we isolate the part of the image that contains their face so that we can ignore the background that's irrelevant.

Face Detection

To capture the user's appearance, we begin with an image that contains the user's face, which

Published by the IEEE CS ■ 1536-1268/13/$31.00 © 2013 IEEE

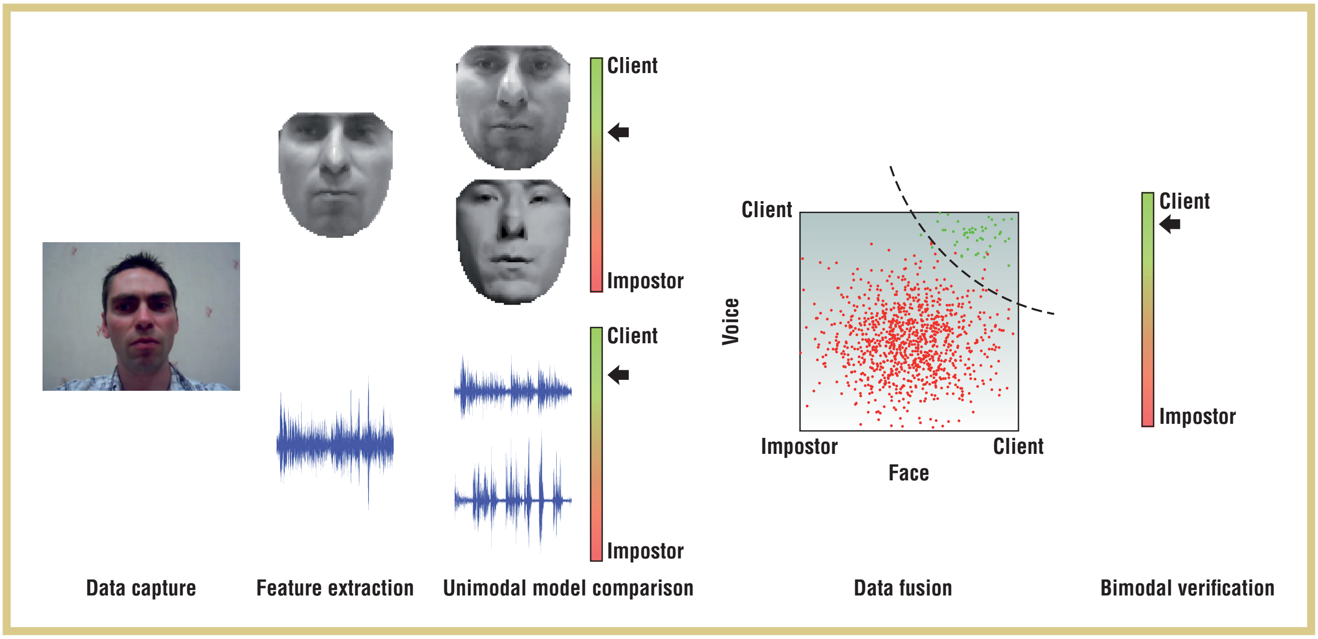


Figure 1. The Mobile Biometrics (MoBio) identity verification system computes feature vectors from the captured and normalized face and voice, compares the features to stored models, fuses the scores for improved robustness, and performs a bimodal verification.

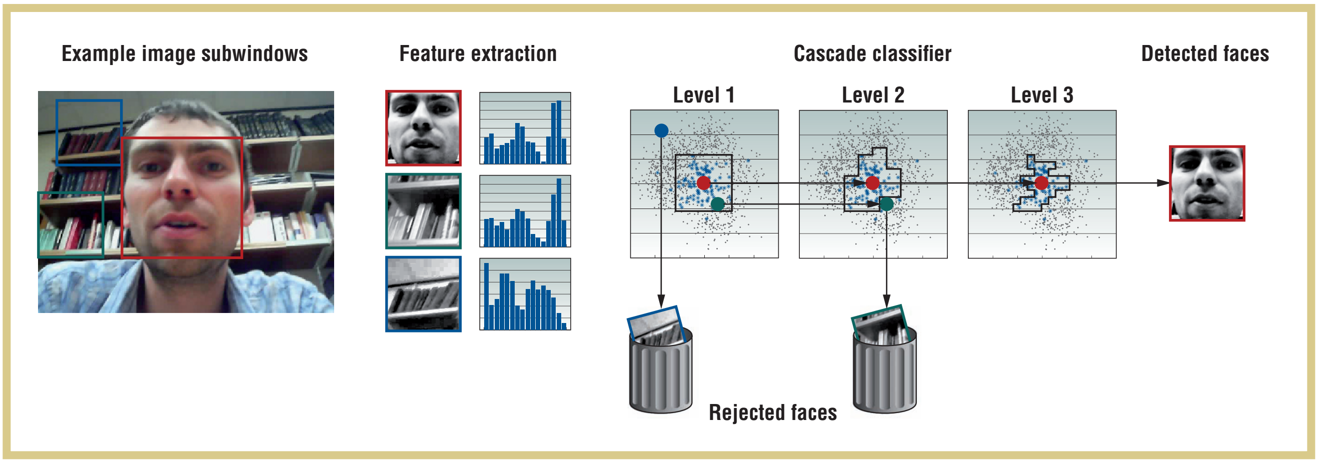


Figure 2. A "window" slides across the image, and the underlying region is sampled and reduced to a feature vector. This feature vector feeds into a simple classifier that rejects obvious nonfaces. Subwindows that are accepted then feed into a succession of more complex classifiers until all nonfaces have been rejected, leaving only the true faces.

we localize to get a rough estimate of its position and size (Figure 2). This is difficult because faces vary in appearance depending on their shape and size, skin color, facial expression, and lighting conditions. Our system must be able to detect all faces regardless of these factors. Ideally, the system should 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 region of the image as either "face" or "not face," using modern pattern-recognition methods to learn the image characteristics that differentiate faces from nonfaces. We considered how to summarize the image region in a compact form (that is, compute its feature vector) and classify the image region based on its features.

When searching an image, there are thousands of possible locations for the face, and it's important to quickly summarize every image region. Using a variant of the Local Binary Pattern,[3] we represented local image statistics around each pixel with a binary code indicating the image gradient direction with respect to its eight neighbors. We then computed the histogram over transformed values for each patch and fed this into a classifier to decide whether the patch was "face" or "not face." In practice, we used a cascade of increasingly complex classifiers[4] to reject most image regions (that look nothing like a face) using simple, but efficient, classifiers in the early stages. We reserved the more accurate, but computationally demanding, classifiers for the more challenging image regions that look more like a face.

Our experiments on standard datasets—for example, the *b*iometric *a*ccess control for *n*etworked and ecommerce *a*pplications (Banca) and e*x*tended *m*ulti*m*odal *v*erification for *t*eleservices and *s*ecurity applications (XM2VTS) datasets—suggested that these methods detect more than 97 percent of the true faces. In our application, however, we also prompted the user to keep his or her face in the center of the screen so that we could restrict the search to a smaller region, thus reducing false positives and permitting more discriminative image representations to further increase detection rates.

To extend this baseline system, we developed a principled system that exponentially reduced false positives (background regions that were wrongly given the "face" label) and clusters of several detections around the same true face, with little or no reduction in the true acceptance rate.[5]

Face Normalization

Although we could try to recognize the user from the rectangular image region approximately surrounding the face, factors such as background clutter, lighting, and facial expression could affect performance. We therefore remove as many of these effects as possible by normalizing the face so that it has properties (with respect to shape and texture) similar to the stored user model (see Figure 3).

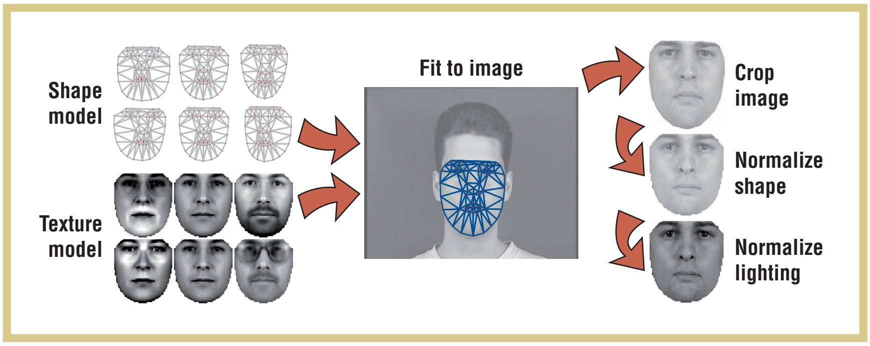


Figure 3. Statistical models of shape and texture are estimated from training data and fitted to a new image using the Active Appearance Model. The underlying image can then be sampled to remove background information, warped to remove irrelevant shape information (for example, due to expression), and normalized to standardize brightness and contrast levels.

First, we localize individual facial features, such as the eyes, nose, mouth, and jawline, using them to remove any irrelevant background. Next, we stretch the face to fit a predefined shape, thus compensating for differences due to the direction the person is facing, his or her expression, and the shape of the face (a weak cue for verification). Finally, we normalize lighting by adjusting brightness and contrast to some fixed values. The resulting image can then be directly compared with a similarly normalized model image for accurate verification.

To locate facial features, we fitted a parametric model of the face to the image using a novel version of the Active Appearance Model (AAM) that we developed specifically for a mobile architecture using modern machine-learning techniques.[6] The AAM uses statistical models of shape and texture variation—learned from a set of training images with hand-labeled feature locations—to describe the face using only a few parameters. It also learns to detect when the model is in the wrong place and adjust 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 we fitted the model to a new image, we initially aligned the model with the coarse face-detection result. Then, we sampled and normalized the corresponding part of the image (see Figure 3). Our method 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.

Compared with the AAM, our approach achieved similar or better accuracy (typically within 6 percent of the distance between the eyes) and a threefold speedup on a Nokia N900, reducing processing time from 44.6 ms to 13.8 ms, and thus achieving frame-rate performance.[6] Although we achieved this performance using a model that was trained from publicly available datasets and that could be adapted to a specific user by retraining the predictor (online or offline), our results suggest that performance would improve little in return for the added computational cost.

Face Verification

Given a normalized facial image, the final step is to assign a score describing how well the image matches the stored model for the claimed identity. We then use that score to decide whether to accept or reject the person's claim

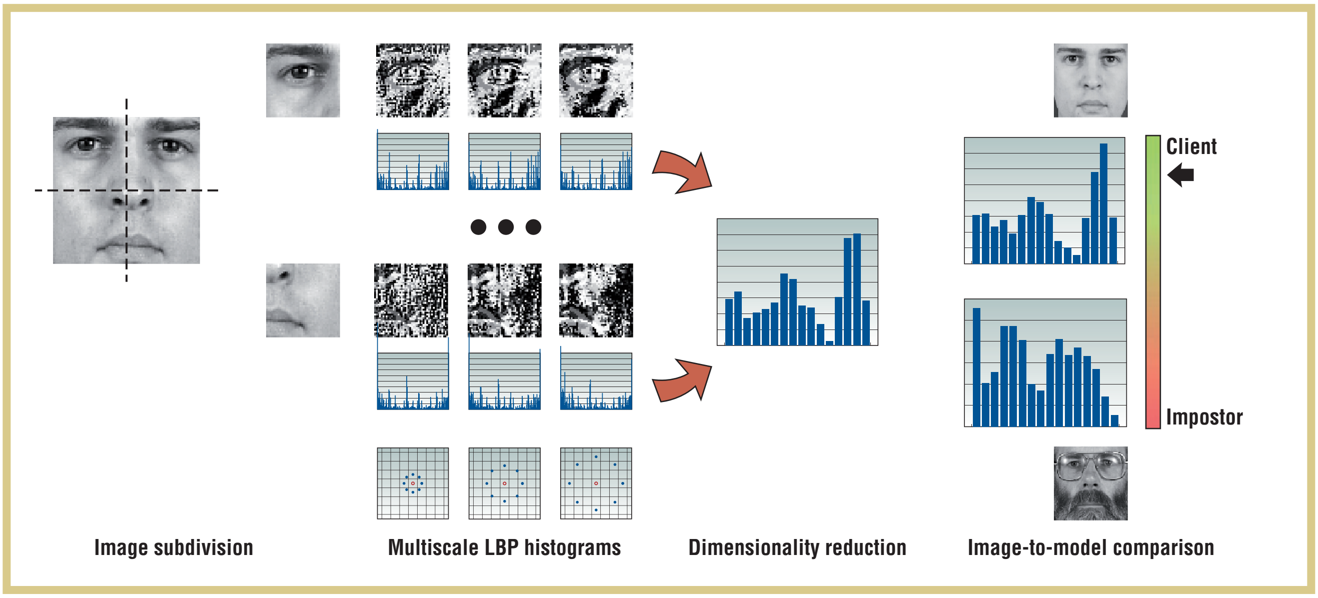


Figure 4. A cropped face window is subdivided into blocks, each of which is processed with a Local Binary Pattern operator at several scales. We then capture the distributions of LBP values in histograms that we concatenate and reduce in dimensionality (for example, via principal component analysis) before comparing with the stored model.

(see Figure 4). Again, we treat this as a classification problem, but here we want to label the person as a client or an impostor based on his or her appearance, as summarized by the image features. Clients are given access to the resource they need; impostors aren't.

Because illumination conditions affect appearance, we applied gamma correction, difference of Gaussian filtering, and variance equalization to remove as many lighting effects as possible. For added robustness, we subdivided the processed images into nonoverlapping subwindows to make the descriptor more robust to occlusion, and we computed the Local Binary Pattern (LBP) value for every pixel over three scales. We then summarized every window by its LBP histogram and used the concatenated histograms as a feature vector for the whole image (Figure 4).

To classify an observed feature vector, we computed its distance to the stored model of the claimed identity. Although we could make a decision based on this similarity measure alone, we instead used a robust likelihood ratio, whereby the distance to a background model provided a reference that expressed how far above average the observation matched the claimed identity, thus indicating our confidence in the classification. Using this method, we achieved half-total error rates (where false acceptances are as likely as false rejections) of approximately 5 percent using the Banca dataset.

We also developed several novel image descriptors that improved recognition performance. One based on Local Phase Quantization was designed for out-of-focus images and achieved a recognition rate of 93.5 percent (compared with 70.1 percent for LBP) on a face image that had been blurred.[7] Further developing this descriptor to include information at multiple scales, we improved recognition rates—in some cases, from 66 to 80 percent—on a more challenging dataset with widely varying illumination.[8]

Voice Analysis

Although 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.

Voice Activity Detection

Given a sound sample captured using the mobile device's microphone, our first step is to separate speech from background noise. As in face detection, however, speech detection is complicated by variation from speaker to speaker (for example, due to characteristics of the vocal tract, learned habits, and language) and from session to session for the same speaker (for example, as a result of having a cold).

To summarize someone's voice, we represented the 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. More specifically, we used cepstral analysis to compute this spectrum via a Fourier transform and decompose its logarithm by a second Fourier transform, mapping the spectrum into the mel scale (where distances more closely match perceived differences in pitch) before the second decomposition to give mel-frequency cepstral coefficients (MFCCs).

We then used a Gaussian Mixture Model (GMM) to classify a feature vector as speech or nonspeech, discarding the temporal ordering of feature vectors and low-pass smoothing the output. Although this proved to be an effective technique for examples with a high signal-to-noise ratio, environments with a lot of background noise demanded more complex methods that use more than the signal energies.

We therefore used an artificial neural network to classify MFCC vectors, derived from a longer temporal context of around 300 ms, as either one of 29 phonemes or as nonspeech to give an output vector of posterior probabilities corresponding to the 30 classes. These vectors were smoothed over time using a Hidden Markov Model to account for the (language-dependent) known frequency of phoneme orderings learned from training data, and the 29 phoneme classes were merged to form the "speech" samples.

Because this approach was computationally demanding (and therefore not well suited to an embedded implementation), we also proposed a simpler feature set, denoted Boosted Binary Features,[9] based on the relationship between pairs of filter responses and performing at least as well as existing methods (65 percent correct classification over 40 possible phonemes) while requiring only modest computation.

Speaker Verification

After discarding background noise, we can then apply 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 the claim.

To describe the voice, we used 19 MFCCs (computed over a 20 ms window), plus an energy coefficient, each augmented with its first and second derivatives. After removing silence frames via voice activity detection, we applied a short-time cepstral mean and variance normalization over 300 frames.

As a baseline, to classify the claimant's feature vector, we used Joint Factor Analysis based on a parametric GMM, where the weights and covariances of the mixture components were optimized at the outset but the centers were specified as a function of the data. These weights, covariances, and means were learned using a large cohort of individuals, and the subject-subspace was learned using a database of known speakers, pooling over sessions to reduce intersession variability. The session-subspace was then learned from what was left.

When testing, we used every training example to estimate the speaker and session, and we adapted a generic model to be user specific. We then discarded the session estimate (because we weren't interested in whether the sessions matched—only in the speaker) and computed the likelihood of the test example given the speaker-specific model. Score normalization then gave a measure to use for classification.

On the Banca dataset, this baseline system achieved equal error rates of approximately 3 percent for speaker verification. We showed, however, that we could improve the related i-vector estimation approach (the current state of the art in speaker recognition) to make speaker modeling 25 to 50 times faster using only 10 to 15 percent of the memory, with only a small penalty in performance (typically increasing the equal error rate from 3 to 4 percent[10]).

The user model can't remain fixed; it must adapt to current conditions and adjust its criteria for accepting or rejecting a claim accordingly.

We also demonstrated that decoupling the core speaker recognition model from the session variability model—letting us optimize the two models independently and giving a more stable system with limited training data—resulted in little or no penalty in performance.[11] Finally, we showed that using pairwise features achieved a half total error rate (HTER) of 17.2 percent compared to a mean HTER of 15.4 percent across 17 alternative systems, despite being 100 to 1,000 times more efficient.[12]

Model Adaptation

One challenge in biometric verification is accommodating factors that change someone's appearance over time—either intentionally (makeup, for example) or unintentionally (such as wrinkles)—as well as external influences in the environment (such as lighting or background noise) that affect performance. Therefore, the user model created when the person first enrolled can't 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 training data that included many individuals, letting us model effects such as lighting and head pose that weren't present in every individual's enrollment 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 GMM to represent facial appearance because of its tolerance to localization errors. We also adapted the model a second time to account for any expected variation in capture conditions.

To account for changes in the capture environment (the Banca dataset, for example, contains examples captured under controlled, adverse, and degraded conditions), we computed parameters of error distributions for each condition independently during training, and used score normalizations such as the Z-norm,

$$z_q(y) = \frac{y - \upsilon_q}{\sigma_q},$$

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

$$P\left(q|y\right) - \frac{1}{1 + \exp\left(-\alpha_q y - \beta_q\right)},$$

to reduce the effect of capture conditions (where υ_q, σ_q, α_q, and β_q are parameters estimated, by learning, for condition q). During 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.

To run the system on a mobile device, we must consider the limitations of the available hardware, such as low-power processing.

In our experiments,[13] normalizing the score reduced the equal error rate in some tests by 20 to 30 percent (from 19.59 to 15.31 percent for the face; from 4.80 to 3.38 percent for speech), whereas adapting the model to capture conditions had an even greater effect on performance, reducing equal error rates by more than 50 percent in some trials (from 19.37 to 9.69 percent for the face, and from 4.80 to 2.29 percent for speech).

Data Fusion

At this point, every sample in a video sequence has a score that tells us how much the person looks like his or her claimed identity and another score for how much he or she sounds like the claimed identity. To create 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). Because we're concerned with video sequences, it's 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; more principled methods model the distribution of scores over the observed sequence and compare this to distributions, learned from training data, that correspond to true and false matches. As a baseline, we computed nonparametric statistics (such as the mean, variance, and interquartile 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 considering the signal's quality.[14]

Although 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 can, 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 (such as video and audio).

We therefore developed a novel feature-level fusion technique, dubbed the "boosted slice classifier," that searched over the space of feature pairs (one face and one speech) to find the pair for which quadratic discriminant analysis minimized the misclassification rate, iteratively reweighting training samples in the process. Although this approach had only a small effect 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, as indicated by the detection error tradeoff curves shown in Figure 5a. 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).

Mobile Platform Implementation

To run the system on a mobile device, we must 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 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're 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 1,296 scaled systems (all possible combinations of 48 face and 27 speech) by two criteria. First, we used an abstract cost reflecting both memory consumption and speed. Second, we used the resultant generalization performance as

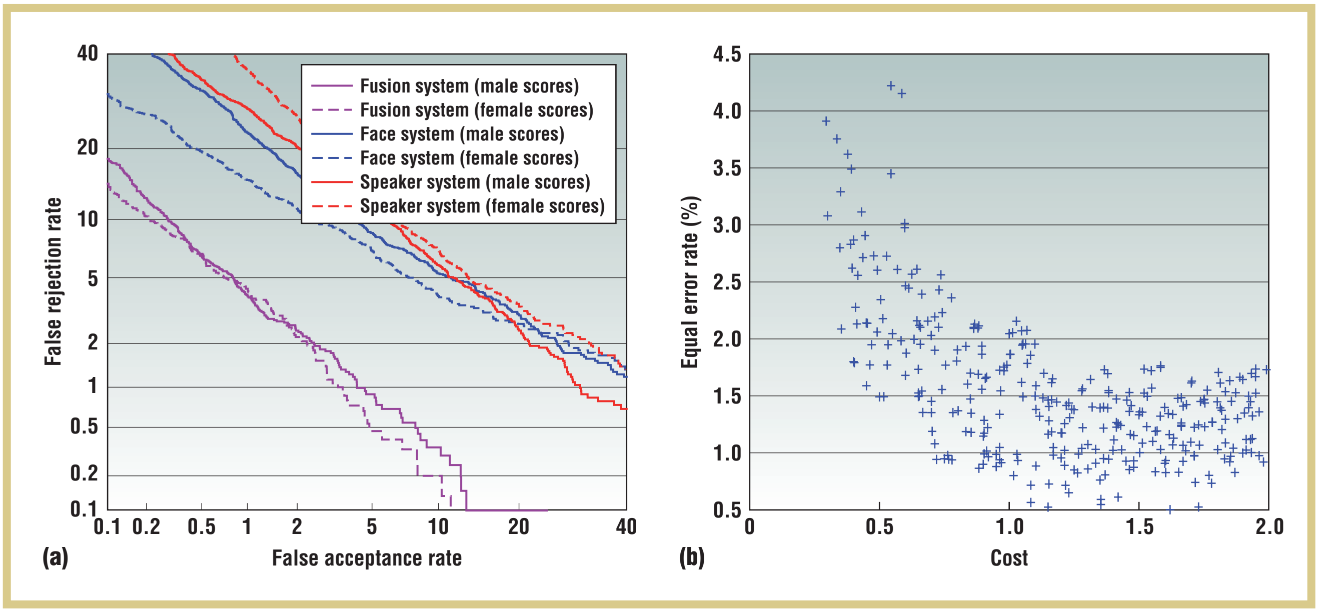


Figure 5. A detection error tradeoff (DET) curve plots false acceptance rate against false rejection rate for a range of decision thresholds on a logarithmic scale, rather than a linear scale as in the receiver operating characteristic (ROC) curve. This makes the DET almost linear and gives a more uniform distribution of points, making interpretation easier. The equal error rate (EER) for a given curve occurs at the point where the false acceptance rate and false rejection rate coincide; the lower-left point is optimal. (a) DET curves for unimodal and fused bimodal systems tested on the MoBio database. (b) EER versus efficiency for various scaled systems, confirming that better accuracy comes at a cost, defined as the lower of two proportions (memory consumption and time taken) with respect to a baseline system.

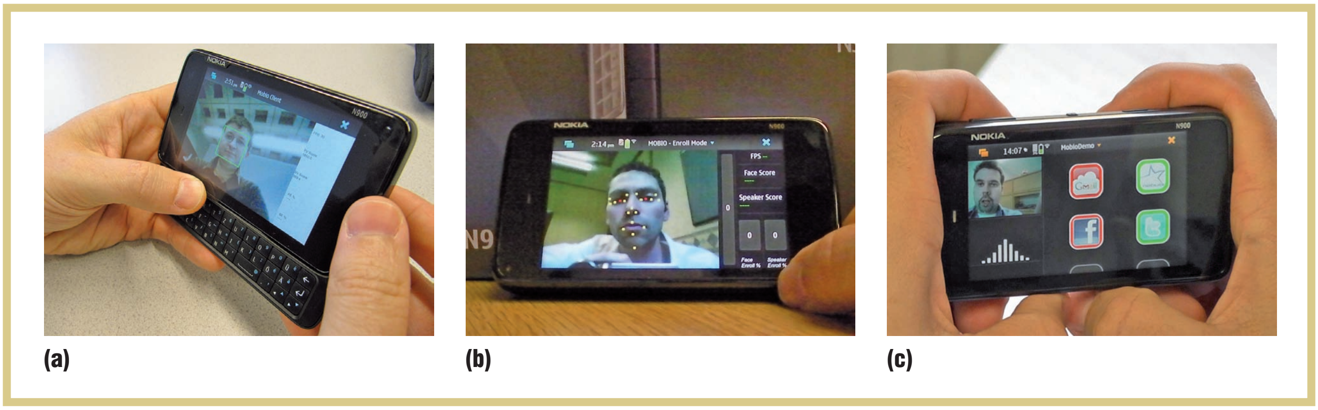


Figure 6. Mobile Biometrics interface demonstrating (a) face detection, (b) facial feature localization (for shape normalization), and (c) the user interface with automated login and logout for popular websites such as email and social networking.

measured by the equal error rate (EER). As expected, increasing efficiency came at a cost in accuracy, whereas increasing complexity resulted in much smaller gains (see Figure 5b).

To test the system under real conditions, we developed a prototype application (see Figure 6) for the Nokia N900 that has a front-facing VGA camera for video capture, a Texas Instruments third-generation Open Multimedia Applications Platform (OMAP3) microprocessor with a 600 MHz ARM Cortex-A8 core, and 256 Mbytes RAM. Using GTK for the user interface and gstreamer to handle video capture, we achieved near frame-rate operation for the identity verification system.

MoBio 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 or voice data

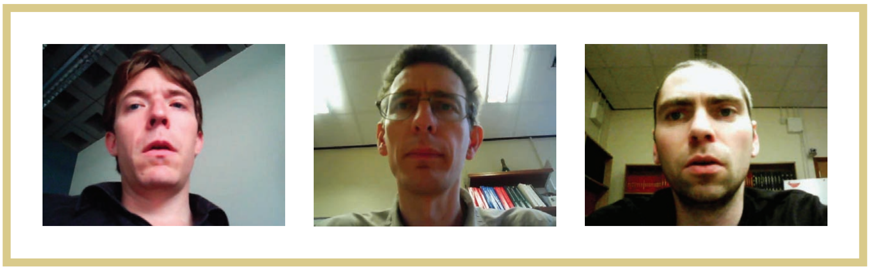


Figure 7. Screenshots from the new database, showing the unconstrained nature of the indoor environments and uncontrolled lighting conditions.

but not both. Even those few that do include both video and audio[1,2] captured the data using high-quality cameras and microphones under controlled conditions, and thus aren't realistic for our application—we're limited to a low-quality handheld camera. The few that come close (such as the Banca dataset) use a static camera, and thus don't have the image jitter—caused by small hand movements—that we must deal with.

Because 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's realistic and publicly available (www.idiap.ch/dataset/mobio) for research purposes (see Figure 7). We collected this database over 18 months, from six sites across Europe. It contains 150 subjects and was collected in two phases for each subject: the first phase includes 21 videos per session for six sessions, and the second contains 11 videos per session for six sessions. A testing protocol is also supplied with the data, defining how the database should be split into training, development, and test sets, and how evaluation scores should be computed. This protocol was subsequently applied in a competition entered by 14 sites: nine for face verification and five for speaker verification.[15]

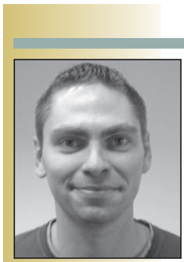
The MoBio project aimed to develop a robust and secure verification system for mobile applications (for full technical details and experimental results, see www.mobioproject.org). The mobile Internet is an obvious example where biometric verification can complement (or replace) traditional access methods, such as passwords. Other potential applications include using biometrics to lock and unlock the phone, and mobile money transactions.

ACKNOWLEDGMENTS

This work was performed by the MoBio project 7th Framework Research Programme of the European Union (EU), grant agreement number 214324. The authors thank the EU for the financial support and the consortium partners for a fruitful collaboration, especially Visidon for their invaluable work in developing the mobile phone user interface.

REFERENCES

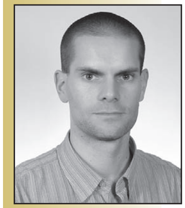
1. G. Chetty and M. Wagner, "Multi-Level Liveness Verification for Face-Voice Biometric Authentication," *Proc. Biometric Symp.*, IEEE, 2006; doi:10.1109/BCC.2006.4341615.
2. A.B.J. Teoh, S.A. Samad, and A. Hussain, "A Face and Speech Biometric Verification System Using a Simple Bayesian Structure," *J. Information and Science Eng.*, vol. 21, 2005, pp. 1121–1137.
3. M. Pietikainen et al., *Computer Vision Using Local Binary Patterns*, Springer, 2011.
4. P. Viola and M.J. Jones, "Robust Real-Time Face Detection," *Int'l J. Computer Vision*, vol. 57, no. 2, 2004, pp. 137–154.
5. C. Atanasoaei, C. McCool, and S. Marcel, "A Principled Approach to Remove False Alarms by Modelling the Context of a Face Detector," *Proc. British Machine Vision Conf.*, BMVA Press, 2010; doi:10.5244/C.24.17.
6. P.A. Tresadern, M.C. Ionita, and T.F. Cootes, "Real-Time Facial Feature Tracking on a Mobile Device," *Int'l J. Computer Vision*, vol. 96, no. 3, 2011, pp. 280–289.
7. T. Ahonen et al., "Recognition of Blurred Faces Using Local Phase Quantization," *Proc. IEEE Int'l Conf. Pattern Recognition*, IEEE, 2008; doi:10.1109/ICPR.2008.4761847.
8. C.H. Chan and J. Kittler, "Sparse Representation of (Multiscale) Histograms for Face Recognition Robust to Registration and Illumination Problems," *Proc. Int'l Conf. Image Processing*, IEEE, 2010, pp. 2441–2444.
9. A. Roy, M. Magimai-Doss, and S. Marcel, "Phoneme Recognition Using Boosted Binary Features," *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, IEEE, 2011, pp. 4868–4871.
10. O. Glembek et al., "Simplification and Optimization of i-Vector Extraction," *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, IEEE, 2011, pp. 4516–4519.
11. A. Larcher et al., "Decoupling Session Variability Modelling and Speaker Characterisation," *Proc. 11th Ann. Conf. Int'l Speech Communication Assoc.* (Interspeech 10), ISCA, 2010, pp. 2314–2317.
12. A. Roy, M. Magimai-Doss, and S. Marcel, "A Fast Parts-Based Approach to Speaker Verification Using Boosted Slice Classifiers," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 1, 2011; doi:10.1109/TIFS.2011.2166387.
13. N. Poh et al., "Model and Score Adaptation for Biometric Systems: Coping with Device Interoperability and Changing Acquisition Conditions," *Proc. 2010 20th Int'l Conf. Pattern Recognition*, IEEE, 2010, pp. 1229–1232.
14. N. Poh, J. Kittler, and T. Bourlai, "Quality-Based Score Normalization with Device Qualitative Information for Multimodal Biometric Fusion," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, 2010, pp. 539–554.
15. S. Marcel et al., "On the Results of the First Mobile Biometry (MoBio) Face and Speaker Verification Evaluation," *Proc. 20th Int'l Conf. Recognizing Patterns in Signals, Speech, Images, and Videos* (ICPR 10), Springer, 2011, pp. 210–225.



Phil Tresadern is a research associate at the University of Manchester. His research interests include computer vision and machine learning. Tresadern received his DPhil in information engineering from the University of Oxford. Contact him at philip.tresadern@manchester.ac.uk.

Timothy F. Cootes is a professor of computer vision at the University of Manchester. His research interests include the construction and use of models of appearance and their application to the medical domain. Cootes received his PhD in engineering from Sheffield City Polytechnic. Contact him at t.cootes@manchester.ac.uk.

Norman Poh is a research fellow with the Centre for Vision, Speech, and Signal Processing (CVSSP) at the University of Surrey, UK. He's also a work-package leader in the Mobile Biometry (MoBio) project, responsible for designing adaptive multimodal biometric systems. His research interests include pattern recognition, video processing, biometric authentication, and information fusion. Poh received his PhD in computer science from the Swiss Federal Institute of Technology, Lausanne. He won the Researcher of the Year 2011 Award from the University of Surrey and is an IEEE Certified Biometrics Professional (IEEE CBP). Contact him at normanpoh@ieee.org.



Pavel Matejka is a senior researcher in the Speech@FIT research group in the Department of Computer Graphics and Multimedia at Brno University of Technology. His research interests include robust speaker verification, language identification, and speech recognition—namely, phone recognition based on novel feature extractions (temporal patterns) and neural networks. He is active also in keyword-spotting and on-line implementation of speech processing algorithms. Matejka received his PhD from Brno University of Technology. He's a member of IEEE and ISCA. Contact him at matejkap@fit.vutbr.cz.

Abdenour Hadid is an adjunct professor and senior researcher in the Center for Machine Vision Research at the University of Oulu. His research interests include biometrics and facial image analysis, local binary patterns, manifold learning, human-machine interaction, and mobile applications. Hadid received his Doctor of Science in Technology in electrical and information engineering from the University of Oulu. He's a member of the Pattern Recognition Society of Finland and the International Association for Pattern Recognition. Contact him at hadid@ee.oulu.fi; www.ee.oulu.fi/~hadid.

Christophe Lévy is a researcher in the Computer Laboratory (LIA) at the University of Avignon. His research activities focus on speech recognition, especially for embedded devices such as cell phones, GPS, and PDAs. Lévy received his PhD in computer science from the University of Avignon. Contact him at christophe.levy@univ-avignon.fr.

Christopher McCool is a postdoctoral researcher in biometrics and machine learning at the Idiap Research Institute. His research interests include 2D and 3D face authentication, face detection, and computer vision. McCool received his received his PhD from the Speech, Audio, Image and Video Technologies (SAIVT) group at Queensland University of Technology. Contact him at christopher.mccool@idiap.ch.

Sébastien Marcel is a senior research scientist at the Idiap Research Institute, where he leads a research team and conducts research on face recognition, speaker recognition, and spoofing attacks detection. He's currently interested in pattern recognition and machine learning, with a focus on multimodal biometric person recognition. Marcel received his PhD in signal processing from the Université de Rennes I. Contact him at sebastien.marcel@idiap.ch.