MOBIO

Mobile Biometry

http://www.mobioproject.org/

Funded under the 7th FP (Seventh Framework Programme)
Theme ICT-2007.1.4
[Secure, dependable and trusted Infrastructure]

D5.2: Description and evaluation of scalable systems for uni-modal authentication

Due date: 30/09/2010 **Submission date:** 30/09/2010
Project start date: 01/01/2008 **Duration:** 36 months
WP Manager: Jean-François Bonastre **Revision:** 1

Author(s): Christophe Lévy & Anthony Larcher (LIA)

Project funded by the European Commission in the 7th Framework Programme (2008-2010)		
Dissemination Level		
PU	Public	Yes
RE	Restricted to a group specified by the consortium (includes Commission Services)	No
CO	Confidential, only for members of the consortium (includes Commission Services)	No

D5.2: Description and evaluation of scalable systems for uni-modal authentication

Abstract:

This deliverable describes scalable systems for uni-modal authentication.The deliverable provides a description of each scalable system and the studies of each scalable parameter. It includes evaluation and results compared to baseline face authentication and speaker authentication systems. Performance obtained when varying the scalable parameters are evaluated on the publicly available BANCA bimodal database. The evaluation of these algorithms/systems will provide an estimation of performances of the audio and video biometric experts according to the CPU and memory consumption required for the different configurations

Contents

KEY

1. AMS - Adaptive Mean Shift
2. ASM - Active Shape Model
3. CFD - Context-based Face Detector
4. CLM - Constrained Local Model
5. c-MCT-C - cascaded-Modified Census Transform-Classifier
6. DCT - Discrete Cosine Transform
7. DET - Detection Error Trade-off
8. EFLDM- Enhanced Fisher Linear Discriminant Model
9. EM - Expectation Maximization
10. EPC - Expected Performance Curve
11. FA - Factor Analysis
12. FAR - False Acceptance Rate
13. FLD - Fisher Linear Discriminant
14. FRR - False Rejection Rate
15. GMM - Gaussian Mixture Model
16. HLDA - Hierarchical Linear Discriminant Analysis
17. HMM - Hidden Markov Models
18. HTER - Half Total Error Rate
19. JFA - Joint Factor Analysis
20. LBP - Local Binary Pattern
21. LFB-GMM - Local Frequency Band Gaussian mixture model
22. LDM - Linear Discriminant Model
23. LFA - Latent Factor Analysis
24. LFCC - Linear Frequency Cepstral Coefficients

25. LLR - Log Likelihood Ratio
26. LPQL - Local Phase Quantization Label face detector
27. MCT - Modified Census Transform
28. MFCC - Mel Frequency Cepstral Coefficients
29. MRF - Markov Random Field
30. PCA - Principle Component Analysis
31. PS_MLBPHLDA_tnorm - Multi-scale Local Binary Pattern Histogram Discriminant Analysis with Score Normalization
32. SVM - Support Vector Machines
33. UBM - Universal Background Model
34. VAD - Voice Activity Detection
35. VJFD - Viola Jones Face Detector

1 Introduction

MoBio project is targeting an application where the Bi-Modal Biometric Authentication system (BMBA) is embedded in the device to authenticate its user in order to allow the use of the device and/or to access data. This project focus on both face and speech modalities as well as bi-modal authentication combining these two modalities. The components of the face biometric engine considered in this project are face detection, face localisation and face verification. The speech authentication system requires a voice activity detection component and a speaker verification module.

Portable devices such as mobile phones suffer from limitations in terms of memory allocation and computational power while BMBA systems require large amount of memory to store face and voice templates and a powerful CPU to execute floating point operations.

The first part of this report presents a study of the scalability of the uni-modal systems resulting from the WP3 and WP4 and evaluating the degradations due to limited resources available into the mobile phone (memory and CPU). Then, the last section of this report presents a scalability study at the multi-modal level. It proposes a way to compare the complexity of each system relatively to the baseline system. Scalability of the uni-modal and bi-modal authentication systems is evaluated by measuring the ratio between the authentication performance and the complexity in terms of memory requirement and computational time.

2 Face detection with Modified Census Transform

The baseline system used for this work is an MCT face detection system [2] based on the work of Rodriguez [17]. The system was developed at Idiap using Torch3Vision [12]. The original face detection algorithm has been altered to a face localisation algorithm by taking the best matching region (*scan window*) and then merging this with at most 10 other detections which have a surface overlap of more than 50%; the best matching region is considered to be the region which has the highest confidence score.

2.1 Baseline system

The implementation of the MCT face localiser consists of four stages. At each stage M weak classifiers are used to help accept or reject a *scan window* as being a face or non-face region. This is the same architecture that Viola and Jones used to derive their real-time object tracker in [20] and is sometimes referred to as a cascade of classifiers. For the derived MCT face localiser there are four stages of the cascade with each stage consisiting of $m = [2, 10, 50, 200]$ weak classifiers per stage. We provide the performance of this baseline system, for several databases, in the Appendix A (Table 21).

2.2 Experimental protocol

For time and computational consumption estimation we ran all tests on a standard PC. The PC characteristics are presented in Table 1. All experiments for memory and time consumption were performed on the BANCA English database.

model name	Intel® Core2 Duo
cpu MHz	2,200
cache size	4 MB
cpu cores	2 **(used only 1)**
memory	2 GB

Table 1: Parameters of the computer the tests were run on.

2.3 Fixed Point Implementation

A major limitation with mobile devices is that the processors they use often have no support for floating point arithmetic. For instance the ARM4/4I instruction set is currently the most widespread among existing devices and it only provides fixed-point arithmetic. Some chipset enhancements targeting multimedia applications are being introduced, such as those in the iPhone chipset. However, there is currently no standard method to add these multimedia enhancements and no clear standard is expected for at least the next few years. Therefore for any computer vision algorithm to migrate to a mobile device a conversion from its floating point maths to integer based maths needs to be made.

2.3.1 Description and Experiments

We conducted several experiments to determine how many bits would be needed to accurately represent the parameters of the MCT face detector. The number of bits for accuracy refers mainly to the number of B used to represent the fraction (decimal places) of the real number (as we can then use as many bits as necessary to represent the non-decimal places). We ran a set of localisation experiments using 32 bit numbers which means that $32 - B$ bits were used to represent the non-decimal part of the real number. From these experiments it was found that $B = 16$ bits was sufficient to represent the accuracy of the parameters for the weak classifiers (the thresholds and LUTs) as this resulted in no loss of accuracy, see Table 2.

Results

	Value						
Number of bits	8	12	14	16	18	20	22
Difference in Accuracy (%)	100	8	0	0	0	0	0
Memory (%)	100	100	100	100	100	100	100
# parameters	100	100	100	100	100	100	100
Computational Time (%)	100	100	100	100	100	100	100

Table 2: Evolution of performance and computational and memory consumption for varying the number of bits used to represent a real number in fixed point (the number of bits used to represent the decimal portion of the number). Resource consumption are given in terms of percentage relative to the baseline system.

Conclusion The effect of converting the arithmetic to fixed point is significant because mobile devices often have very limited or no floating point units available for use. Therefore this parameter had to be explored in detail (and also required significant time to implement). The raw experimental results show that provided the implementing fixed point arithmetic uses at least 16 bits to represent the decimal part then this parameter will have no impact on the overall performance of the face localisation system.

2.4 Efficient Face Localisation

The idea behind efficient face localisation is that the scanning algorithm (to find a face) can be stopped once a face has been found. This requires a fundamental change to the underlying algorithm as scanning algorithms exhaustively scan the image. An example of an exhaustive scanning algorithm is presented in Algorithm 1 where all S scales are processed. However, in our task we consider that there is only one face of interest as we are performing verification: this means that one identity is claimed and furthermore we can likely assume that the most prominent face in the image is of the person of interest.

Algorithm 1 Exhaustive Localisation

```
Require: Input Image $I$
  $s = 1$, $Detections = []$
  $S = [S_{largest}, ..., S_{smallest}]$
  while $s < num\_scales$ do
    $Detections_s =$ face_matches_at_scale($Scales[s]$,$I$)
    $Detections = [Detections; Detections_s]$
    $s = s + 1$
  end while
  return best_match($Detections$)
```

2.4.1 Description and Experiments

The proposed algorithm is to stop the scanning process once a face has been found and is presented as pseudo-code in Algorithm 2. This change to the scanning process means that the first face that is found is assumed to be the face of interest. Considering the domain for this technology, which is a mobile device, it is fair to assume that the first face of interest would also be the largest face in the image, or the most prominent face. Choosing the largest face in the image means that the search should begin by scanning the largest search windows and then scanning progessively smaller search windows. This helps to define how the scanning algorithm should be structured, however, the essential part of this efficient face localisation algorithm is to define an efficient criteria for stopping the scanning process.

The initial stopping criteria trialled in this work was to assume that if two detected regions had a surface overlap greater than 60% then a face region had been found. Tests were then conducted to analyse the localisation accuracy and efficiency of such an algorithm. It was found that this simple change to the scanning process reduced the computational time (when a face was found in the image) by an order of magnitude. However, it came at the cost of localisation accuracy. By examining the errors in localisation it was found that a gross number of localisation errors were caused by inaccurately localising the face rather finding the incorrect face. In fact, a consistent trend was that the size of the final scanning window was too large. This suggested that the initial stopping criteria supplies a good estimate of where the face of interest is, however, it is does not provide a sufficiently accurate estimate. To improve the accuracy of the localisation S more scales were scanned to refine or improve the localisation result.

Algorithm 2 Efficient Localisation

```
Require: Input Image $I$
  $s = 1$, $Detections = []$, stop_criteria=FALSE
  $S = [S_{largest}, ..., S_{smallest}]$
  while ($s < num\_scales$) AND (stop_criteria==FALSE) do
    $Detections_s$ = face_matches_at_scale($Scales[s]$,$I$)
    $Detections = [Detections; Detections_s]$
    if found_face($P$) then
      stop_criteria=TRUE
    end if
    $s = s + 1$
  end while
  return best_match($Detections$)
```

This change led to a significant improvement in performance and still yielded a much more efficient scanning algorithm. The tradeoff between localisation accuracy and computational efficiency is presented below (Table 3) where the CPU performance improvement (compared to the baseline system) and the relative decrease in localisation accuracy (compared to the baseline system) is presented.

From the results in Table 3 it can be seen that altering the number of extra scales searched can have a significant impact upon the speed and accuracy of the system. For instance if the number of extra scales searched is set to 0 then the accuracy decreases by 20.67% but it takes approximately one third of the time to process a video. Given the conflicting need of accuracy and reduced computation time using $S > 0$ could be reasonable, however, for our case since we still consider accuracy to be important $S = 2$ appears to be the most useful tradeoff.

Results

	Value					
Number of extra scales S	0	1	2	3	4	5
Decrease in Accuracy (%)	20.67	2.43	0.84	0.64	0.26	0.03
Memory (%)	100	100	100	100	100	100
# parameters	100	100	100	100	100	100
Computational Time (%)	33.82	40.82	50.00	61.50	75.30	89.21

Table 3: Evolution of performance and computational and memory consumption for using a different number of extra scales S searched to refine the localisation. Resource consumption are given in terms of percentage relative to the baseline system.

2.5 Best compromise

Two sets of experiments have been conducted: one for the using fixed point arithmetic (with a varying number of bits B) and one for the number of extra scales S to search (to refine the localisation). For the best compromise it was decided that the number of bits for fixed point arithmetic should be set $B = 18$ to ensure that extra accuracy was retained, this had no impact on the resources or computational time. The number of extra scales searched was set to be $S = 2$ as this provides a system that is twice as fast with a reduction in accuracy of 0.84%. The results for this best system are presented in Table 4.

Results

	Value	
	Baseline	Best Compromise
Accuracy (%)	99.89	99.05
Memory (%)	100	100
# parameters	100	100
Computational Time (%)	100	50.00

Table 4: Performance and computational and memory consumption for the optimal configuration of the system. Resource consumption are given in terms of percentage relative to the baseline system.

2.6 Conclusion

The optimised face detection provides a tradeoff between accuracy and performance. The raw performance increase is achieved mainly through altering the number of extra scales searched (S). By setting $S = 2$ the system is twice as fast but only loses 0.84% in detection accuracy. Further work was also conducted to remove floating point operations, these changes are covered in implementing the fixed point arithmetic. This is considered significant because many mobile devices have no (or limited) floating point units and so such operations will cause a significant increase in computational time. The final choice of the in the number of bits ($B = 18$) to represent the number in fixed point was made so as to allow for retaining the same accuracy of detection (and of the floating point numbers) without impacting on memory or computational time.

3 Viola-Jones face detection in fixed point arithmetic

The purpose of this work was to produce a fast fixed point implementation of the Viola-Jones face detector [21] with no dependencies on external libraries. The baseline of this work was the face detector in the OpenCV library and the haarcascade_frontalface_alt from the OpenCV library.

3.1 Baseline system

The baseline method for face detection is based on the Viola-Jones face detector [21]. This detector is well known for its high detection accuracy under limited computational overload. An OpenCV library [14] based implementation of the face detector is also available.

Viola and Jones use features that resemble Haar wavelet responses as an input for their detectors. These features, albeit very simple, seem to provide enough information for reliable face detection. The most prominent advantage of these features is their speed: using so called *integral images*, these features can be computed in constant time from any subwindow of an image.

AdaBoost [10] is used to select the most prominent features among a large number of extracted features and construct a strong classifier from boosting a set of weak classifiers. The use of a cascade of classifiers made their system one of the first real-time frontal-view face detector. The system has resulted in a large amount of research and publications concerning face detectors of similar nature. In summary, the three factors behind the success of this type of a detector are:

- Haar-like features which are very fast to compute – can be computed in constant time.
- A fast and reliable classifier resulting from boosting
- Cascade of classifiers where most windows can be discarded in a very early stage of the cascade resulting in fast processing.

3.2 Experimental protocol

The scalable system is compared to baseline implementation of the Viola-Jones detector in the OpenCV library.

The face detection accuracy of the scaled system is measured using the same protocol as in MOBIO Project deliverables D3.2 and D3.4. The performance measure is based on predicted locations of eye centers, p_l and p_r, and the corresponding ground truth locations $q_l$and q_r. The normalised maximum distance is then used as the performance measure:

$$d_{max} = \frac{\max(|p_l - q_l|, |p_r - q_r|)}{|q_l - q_r|}. \tag{1}$$

The median and 90th percentile statistics are reported for each test image dataset as a performance measure in addition to the number of missed detections.

The face detector outputs a *face box* described by its center point coordinates (c_x, c_y), width w and height h. These are converted into eye coordinates as

$$\begin{aligned} p_l &= (c_x - 0.18w, c_y - 0.12h) && (2) \\ p_r &= (c_x + 0.18w, c_y - 0.12h)\,. && (3) \end{aligned}$$

To measure the scalability of the face detector, we use two numbers: face detector running time on frames of BANCA videos and the memory consumption of the face detector. These numbers are compared to those of the baseline system (OpenCV Viola-Jones face detector), so that for example number 40 % in the detection time means that the scaled system needs 40 % of the running time of the baseline system (i.e. it is 60 % faster). The experiments are conducted using a standard PC whose characteristics are described in Table 5.

model name	Intel® Core2 Duo
cpu MHz	2,000
cache size	6 MB
cpu cores	2 **(used only 1)**
memory	2 GB

Table 5: Parameters of the computer the tests were run on.

3.3 Fixed point implementation

The scalable Viola-Jones face detector was implemented in 32 bit fixed point arithmetic. The fixed point system uses the Haar classifier cascade from OpenCV library, i.e. the classifier was not re-trained. The detection results should thus be very comparable to the baseline system.

Experiments

The purpose of this experiment was to confirm the correct fixed point implementation of the face detector. It was expected that there is no significant difference in the detection performance or in the running time on PC platform, and that memory consumption should be smaller.

The detection results for the baseline and fixed point implementation of the detector can be found in Table 6. There is no significant difference in the detection accuracy between the floating point baseline detector and the fixed point implementation. The face detection time per frame is 25 % smaller and memory consumption is 53 % smaller in the fixed point implementation .

	time	memory	Missed detections	d_{max}	
	ms / frame	MB		Med.	90%
Baseline	102 (100 %)	8.14 (100 %)	177 (2.8%)	0.088	0.15
Fixed point	76 (75 %)	3.85 (47 %)	176 (2.8%)	0.083	0.14

Table 6: Detection time, memory usage, missed detections (i.e. images where no face was detected at all) and statistics (specifically the median value and 90th percentile over the entire dataset) of d_{max} for the eye points

3.4 Sliding window step size

This parameter controls the step size of the detection window. The step size does not need to be equal in x and y directions, and it is assumed that a larger step size will result in smaller detection time at the cost of less accurate detections and missed faces. The actual step between two window locations is

$$step_y = s_y \frac{window\_height}{20}, step_x = s_x \frac{window\_width}{20}.$$

In practice, the actual step size increases when the search window is larger, and the minimum possible search window size is 20×20pixels.

Experiments

Three different step sizes were tested: 1) $s_y = 1, s_x = 1$; 2) $s_y = 1, s_x = 2$; 3) $s_y = 2, s_x = 2$. The results can be found in Appendix (Table 22).

3.5 Window scaling step size

Another step size parameter is the window scaling between two successive scales. In this implementation, search is started at the largerst possible scale and the window is then down-scaled successively until the selected minimum window size is achieved.

Experiments

In the experiments, the window was scaled by 1.05, 1.10, 1.15, 1.20, 1.30, and 1.40 between the successive scales. It can be observed that, similarly to x and y step sizes, a larger step size results in shorter detection time but less accurate detections. The detection results can be found in Appendix (Table 23).

3.6 Image downscaling

For the purposes of face verification in the MOBIO project, we are not interested in finding the very small faces of sizes less than about 60×60 pixels. For this reason, we can downscale the input image prior to performing face detection, resulting in smaller memory footprint and also shorter detection time, due to more efficient memory cache usage.

Experiments

In the experiments, the input image was downscaled by 1, 2, 3, 4 and 5. The minimum detection window size was changed accordingly (64 for $d = 1$, 32 for $d = 2$, 20 for $d = 3, 4, 5$). Downscaling the image results in more missed detections, and also slight loss of accuracy, but both the detection time and memory consumption are significantly reduced. The detection results can be found in Appendix (Table 24).

3.7 Stopping at largest face found

As the fourth option to speed up face detection, the search can be stopped when the first cluster with sufficient number of overlapping detections is found. This is expected to make the search much faster for those frames where a large face can be found.

The system stops at largest detections, so it outputs face boxes slightly larger than the systems performing exhaustive search. For this reason, we use different parameters when transforming the face box into eye coordinates:

$$\begin{aligned} p_l &= (c_x - 0.15w, c_y - 0.1h) & (4)\\ p_r &= (c_x + 0.15w, c_y - 0.1h)\,. & (5) \end{aligned}$$

Experiments

Face detector stopping at the largest face found was tested with different image downscaling and window step parameters. In all the experiments, the window downscaling step was set to $s_{\text{scale}} = 1.10$. It was observed that the face detection is less accurate than in exhaustive search, but the number of missed detections is not affected. Overall, very good results were obtained, for instance with the parameters $d = 2, s_y = 1, s_x = 1$, we get less missed detections than with the baseline system. The accuracy of detections is not as high, but the detection time is 7.4 % and the memory consumption is 19 % of that of the baseline system. All the results can be found found in Appendix (Table 25).

3.8 Best compromise

Based on the results presented above, we propose the fixed point Viola-Jones face detector, stopping at largest face found, as the best compromise between accuracy and speed. More specifically, the detector with the parameters

- Stop at largest face found: Yes
- Image downscaling factor: 2
- Window scaling step size: 1.1
- Search step size: $s_y = 1, s_x = 1$

seems to provide a good compromise scaled face detector (See Table 7). The accuracy of that detector (as measured by the d_{max} values) is not as good as that of the best other options, but the number of missed detections is very low. That is explained by the fact that we can perform dense search ($s_y = 1, s_x = 1$) and still do detection very quickly because the search can be stopped early in most cases. This detector runs in less than a tenth of the time taken by the baseline system, and it uses about a fifth of the memory, so it is very efficient both in terms of computational cost and memory usage.

	time ms / frame	memory MB	Missed detections	d_{max}	
				Med.	90%
Baseline	102 (100 %)	8.14 (100 %)	177 (2.8%)	0.088	0.15
Compromise scaled system	7.5 (7.4 %)	1.57 (19 %)	125 (2%)	0.13	0.21

Table 7: The detection results for the baseline and the best compromise fixed point implementation

4 Face alignment

In this section, we evaluate the effect of varying parameters within the model that have an impact on the accuracy, speed and memory requirements of the face alignment module. The quality of the facial feature localization has repercussions for later modules in the system. For example, if the eyes are not localized accurately then the normalization required by the verification system will also be in error. Without an accurate normalization, verification performance is likely to decline.

4.1 Baseline system

The baseline system uses the Constrained Local Model algorithm to locate deformable objects (such as faces) in an image. Initialized using the corners of the detected face region, individual facial feature locations are predicted in the image. These points are then optimized using a non-linear minimization approach under the constraints that the points and their surrounding texture lie within a lower dimensional linear subspace; this prevents the model from overfitting to the data.

Therefore, the parameters we choose to vary are as follows:

- The number of modes of the appearance (*i.e.* shape and texture) subspace;
- The number of iterations performed during the non-linear minimization;
- The number of facial features that we localize;
- The size of the image template associated with each feature;
- How the texture is represented;
- How we predict the points from the provided bounding box of the face;
- How we optimize the points to reach a local minimum.

4.2 Experimental protocol

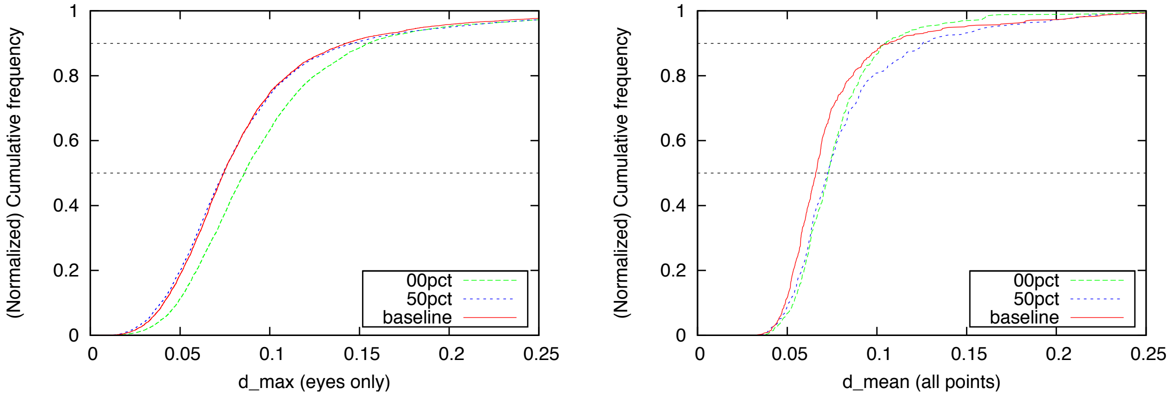
These experiments repeat the evaluations presented in deliverable D3.1 whilst varying system parameters to influence accuracy and efficiency. Specifically, we report results using (a) Session 1 of the XM2VTS dataset (a total of 590 images labelled with all 22 facial features) and (b) the BANCA images dataset (over 6500 images with eye points labelled). The feature localizer was initialized using the faces detected by the implementation of the Viola-Jones detector provided by UOULU (in the case of XM2VTS) and the Modified Census Transform detector provided by IDIAP (in the case of BANCA images).

model name	Intel® Core2 Quad
cpu MHz	2,660
cache size	6 MB
cpu cores	4 **(used only 1)**
memory	3.25 GB

Table 8: Parameters of the computer the tests were run on.

4.3 System Parameters

4.3.1 Number of Modes in PCA Model



(a) (b)

Figure 1: No. of PCA model modes: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

The face model is characterized by several eigenmodes that permit variation in the shape and texture of the face. Using more modes increases the accuracy of the fit (since the model is given more flexibility to fit to the data) but requires more memory and reduces efficiency. Furthermore, using too many modes allows the model to overfit to the data, reducing the usefulness of a prior shape and texture model.

The baseline model computes a subspace that retains 95% of the shape information and 60% of the texture. The coefficients in these two spaces are then concatenated (with appropriate scaling) and a joint subspace computed that captures 95% of the total variance. In this experiment, we vary both the shape and joint variance parameter and compare performance for alternative values of 0% (*00pct*) and 50% (*50pct*) since these would offer efficiency savings in memory requirements and speed.

From these results, we see that the lower variance captured does indeed speed up localization, though memory requirements are largely unaffected. For the BANCA dataset, capturing only 50% of the variance leaves accuracy largely unchanged. For the XM2VTS dataset, however, there is a penalty in accuracy when using fewer modes. This suggests that reducing the size of the shape model could have benefits for lower quality image data (such as in the MoBio database).

4.3.2 Number of Iterations of Optimization

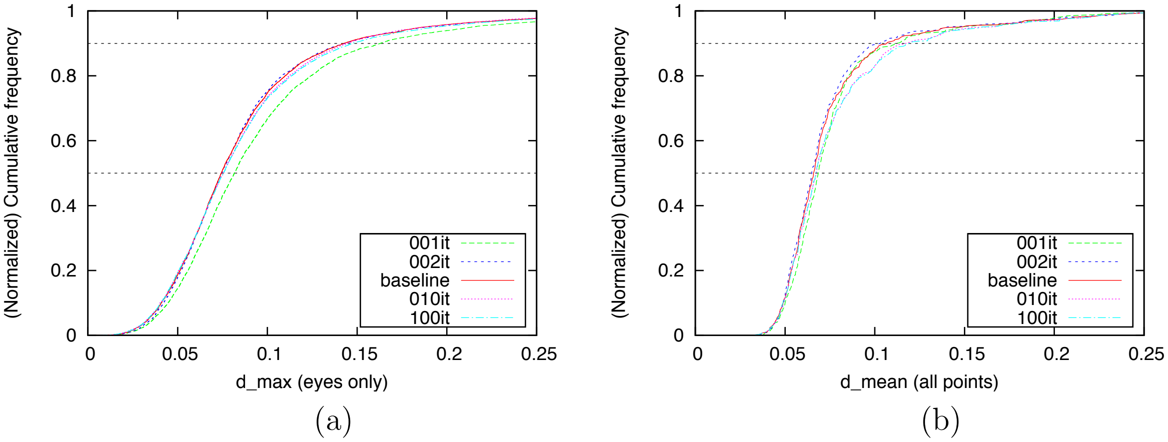


Figure 2: No. of iterations of optimization: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

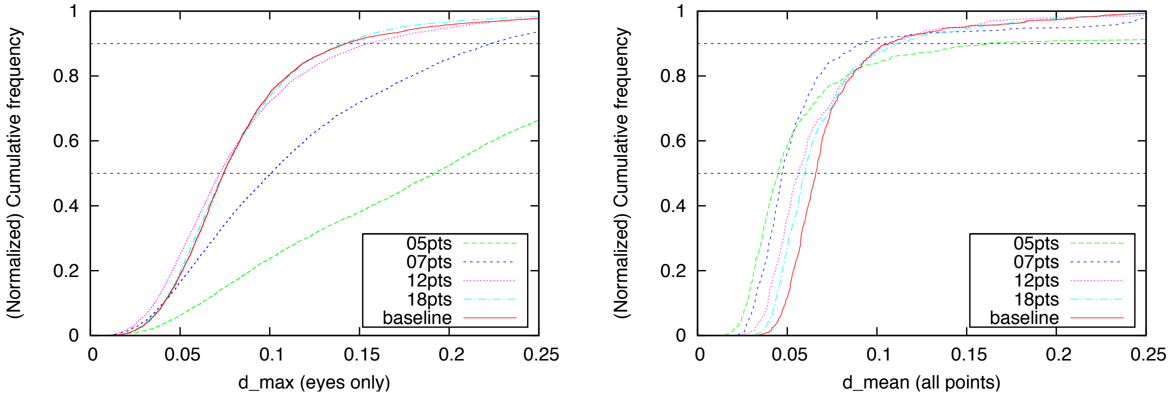
Since the baseline system employs an iterative optimization scheme, it is useful to know how many iterations are needed for effective performance. In this experiment, we vary the maximum number of iterations and examine the effect on accuracy and efficiency. Note that, due to the presence of a termination criterion (upon which convergence is declared and no more iterations are performed), this limit may not be reached in every case.

The baseline system uses a maximum of 3 iterations of optimization. In this experiment we compare performance when applying a limit of one (*001it*), two (*002it*), 10 (*010it*) and 100 (*100its*) iterations. Our results suggest that the error does not decrease considerably beyond two iterations, where an efficiency saving of around 20% is available. The number of iterations has no influence on the memory requirements of the system.

4.3.3 Number of Features

An important pre-processing step in face verification is normalization with respect to position, orientation and scale. Though a coarse estimate of position is provided by the detector, a fine localization of individual features provides a more accurate estimate of pose in the image plane. Furthermore, knowing all feature locations gives us the option of applying a non-linear warp of the image data to simulate a frontoparallel image of the face, thus making verification more robust.

In this experiment, we investigate the effect of varying the number of features localized by the algorithm. As the number of features increases, so does the complexity of the system and the efficiency suffers as a result. However, it may also be the case that the additional



(a)

(b)

Figure 3: No. of facial features: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

points provide greater stability in the estimation of key points such as the centres of the eyes. Alternatively, the less salient points may contaminate the result and increase the error on the key points.

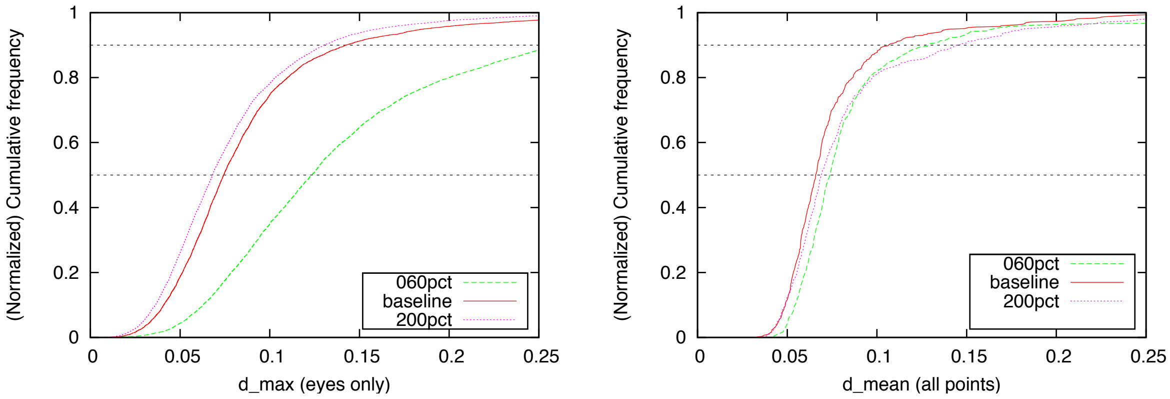
By default, the baseline system predicts 17 points based on the image data before estimating the position of five more points; all 22 points are then optimized. In this experiment, we compare performance to that when predicting and optimizing five (*05pts*), seven (*07pts*), 12 (*12pts*) and 18 (*18pts*) points.

If it determined that all points are necessary to normalize the face for accurate verification, the 22 points model will be necessary. Using only five points is clearly insufficient and susceptible to gross error. However, a seven-point model may be sufficient if image capture conditions are favourable (*e.g.* as in the XM2VTS dataset). For more challenging datasets (*e.g.* BANCA) it is likely that more features are necessary to provide additional constraints on the solution.

4.3.4 Size of Template

Each feature point is associated with an image template that is used to localize the feature in the image. Increasing the size of this image template provides more data with which to compare the image and should therefore improve localization accuracy. However, this also increases the size of the model (and its associated memory requirements) and the complexity of the system, thus reducing efficiency.

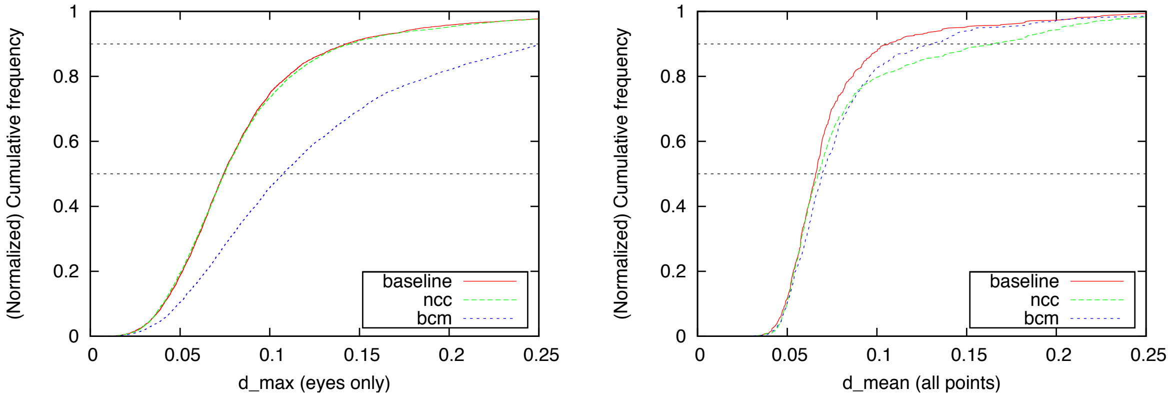
In this experiment, we express the template size with respect to the baseline, investigating templates that are 60% (*060pct*) and 200% (*200pct*) of the baseline size. Again, despite potential savings in efficiency the degradation in accuracy does not justify using a smaller template than used by the baseline. This is especially evident in the lower quality datasets such as BANCA and thus is likely to apply also to data typical of the MoBio project.



(a) (b)

Figure 4: Template size: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

4.3.5 Texture Model



(a) (b)

Figure 5: Texture model: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

In order to compare the model image template for each feature with the observed data in the image, we require a representation of the image data. In this experiment, we compare approaches of differing complexity.

The baseline system computes image gradients in X and Y before computing the normalized cross correlation in both planes to compare an image patch and the stored template. A more cost-effective approach is to use the normalized correlation over raw greyscale values (*ncc*). We also compare the methods with a census transform-based approach (*bcm*) which may also provide efficiency savings due to its use in other modules.

From the results, we see that although normalized correlation is slightly more efficient it also degrades performance with respect to accuracy on high quality datasets (*e.g.* XM2VTS). However, on lower quality datasets there is a much smaller difference such that using normalized cross correlation (rather than the more complex gradient correlation) may be worth considering as a compromise.

The census transform model is both more complex and less accurate than the baseline method of normalized correlation with gradient images. We again note that the use of the census transform in other modules (*e.g.* face detection and verification) would eliminate a pre-processing overhead and therefore increase efficiency. However, quantifying this saving is difficult at this stage of development.

4.3.6 Point Prediction Method

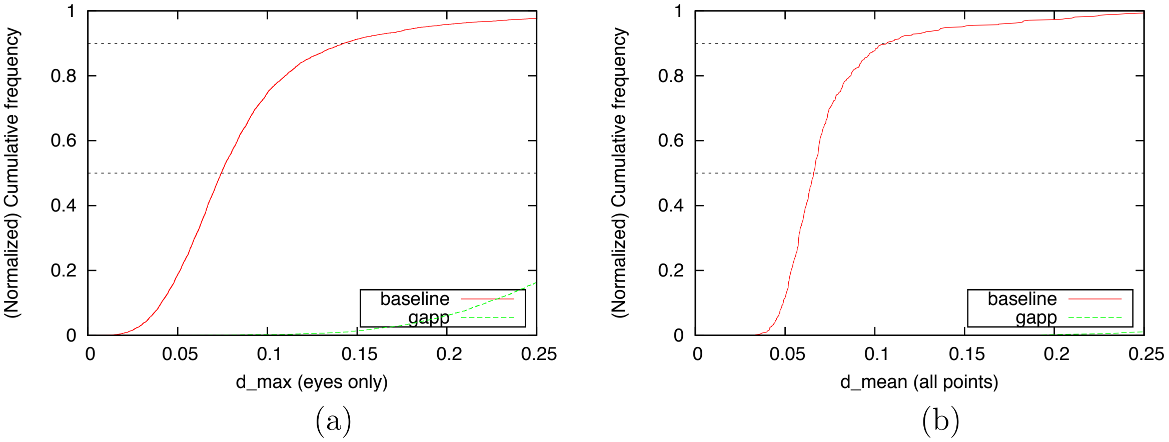


Figure 6: Point predictions method: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

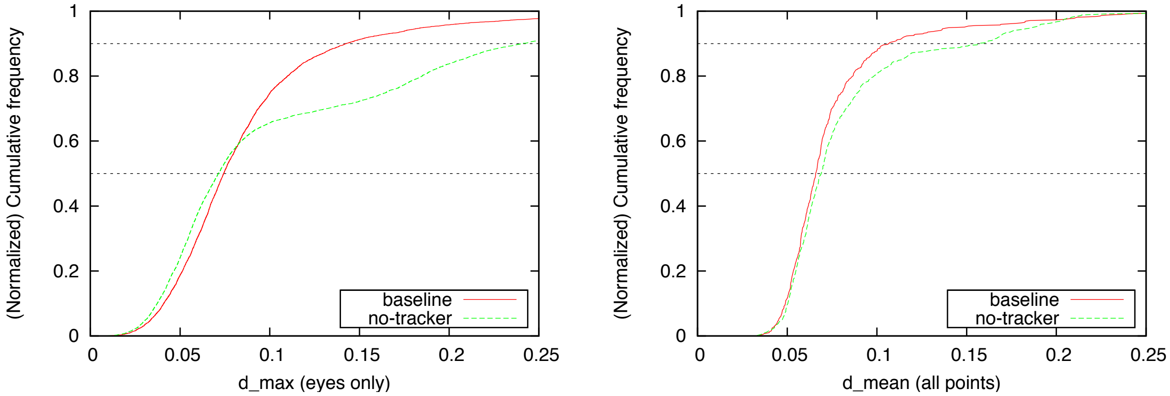
The predictor used in the baseline system is based on the Pictorial Structures Model [9] that uses image information via a Markov Random Field to estimate feature locations. We compare this to a more naive approach (*gapp*) that simply specifies predicted feature locations as fixed points within the co-ordinate frame of the face. This approach is, of course, faster but less accurate.

It is clear that using image data to predict an accurate initialization for the optimizer is crucial to the success of the feature localizer and must be retained.

4.3.7 Optimization Method

A non-linear minimization is used to optimize the predicted feature locations. This experiment investigates the effect of using a different optimizer on efficiency and accuracy.

The default option used in the baseline system is based on the Nelder-Mead Simplex algorithm. In this experiment, we compare this to a system that applies no optimization



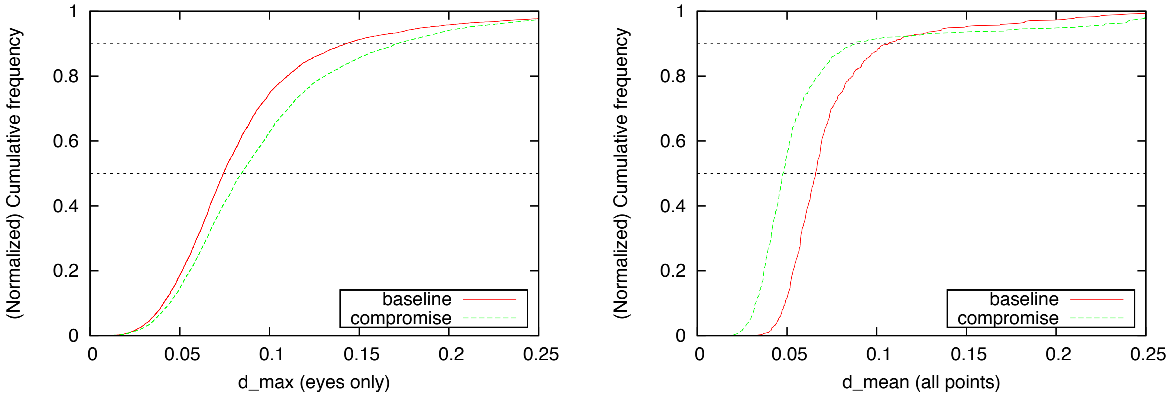
(a) (b)

Figure 7: Optimization method: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

whatsoever (*no-tracker*). Note that this is not the same as reducing the number of iterations to zero since there are additional memory savings to be gained by not loading a tracker at all.

As in the predictor evaluation, we see that although not optimizing points leads to a large efficiency saving the loss in accuracy is too high a penalty. In particular, the high 90th percentile errors indicate a large number of gross failures in the absence of an optimization strategy.

4.4 Best Compromise



(a) (b)

Figure 8: Compromise system: (a) Maximum error over eye points using BANCA; (b) Mean error over all points using XM2VTS.

We conclude that the best compromise between accuracy and efficiency can be achieved by modifying the baseline system to track only a few features (depending on image quality) and iterate only twice. Our results suggest that the eyes can be localized with comparable accuracy (Figure 8, Table 9 and Table 11) in approximately one-third of the time and using just over half the memory (Table 10 and Table 12) using these modifications.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
compromise	0.084	0.171	-	-	-	-	0.066	0.135

Table 9: Accuracy of compromise system for BANCA dataset.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	97	100	13.4531
compromise	36	40	8.39038

Table 10: Efficiency of compromise system for BANCA dataset.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
compromise	0.047	0.123	0.089	0.204	-	-	0.048	0.088

Table 11: Accuracy of compromise system for XM2VTS dataset.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
compromise	33	30	7.62894

Table 12: Efficiency of compromise system for XM2VTS dataset.

The increase in accuracy over all points for XM2VTS is attributable to the fact that the more difficult points to localize (*i.e.* those that exhibit greatest error) have been excluded. We note that although only a few salient features are tracked, they should still provide some information with respect to the 3D orientation of the head which will be of importance if multiple models are implemented for verification.

In contrast, the compromise system performs less well than the baseline on the BANCA dataset. This can be attributed to the fact that points that are more difficult to localize are required for this dataset and therefore increase the overall error.

In conclusion, any compromise will be dependent on the dataset to which the method is applied though there are clear gains in efficiency that can be achieved with little penalty in accuracy.

BANCA videos Finally, we compare the efficiency of the baseline and 'best compromise' systems using the 20 videos selected from the BANCA dataset. As with the other datasets, we see that the compromise system is approximately three times as efficient as the baseline system. In the absence of labelled feature points for the BANCA videos, we are unable to assess the accuracy of the two methods for this dataset.

	Time (ms)	
	Med.	Mean
baseline	108	110
compromise	34	40

Table 13: Efficiency of compromise system (processing time per frame).

5 Scalable mobile phone-based system for Face verification

There is increasing demand to apply biometric recognition to enhance usability and security of handheld devices such as mobile phones, and as these devices are typically equipped with intrgrated video cameras, the face is a natural biometric to be applied. However, these devices are used in unconstrained environmental conditions, and the quality of integrated cameras as well as the computation power and memory size of these devices are often limited Therefore, implementing relablie face verification system in handheld-based device is very challenging. In this report, Multi-scale Local Binary Pattern Histogram Discriminant Analysis with Score Normalization for robust face recognition (PS MLBPHLDA tnorm) reported in D3.4 is evaluated for a scalable system.

5.1 Multi-scale Local Binary Pattern Histogram Discriminant Analysis with Score Normalization for robust face recognition (PS_MLBPHLDA_tnorm)

The method first normalizes the face image to a canonical form in which the illumination variations are suppressed. Then the image is represented by the multi-scale local binary pattern histogram discriminative descriptor. Accordingly, the similarity score of each query image is normalized by the test norm. For a sequence of frames in a video, the final similarity score is the average of frame-based similarities. A brief description of this system is given in following.

5.1.1 Preprocessing sequence approach

In this report, a preprocessing method [19] based on a series of steps presented in Figure 9, designed to reduce the effects of illumination variation, local shadowing and highlights, while still keeping the essential visual appearance information for use in recognition is used.

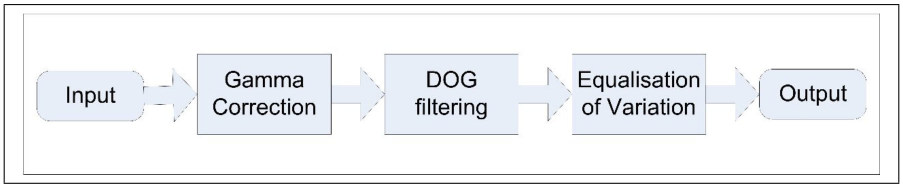


Figure 9: The block diagram of the Preprocessing sequence approach.

This process first applies a gamma correction, which is a nonlinear gray level transformation replacing the pixel value in $\mathbf{I}$ with $\mathbf{I}^{\gamma}$ where $\gamma > 0$. The objective of this process is to enhance the local dynamic range of the image in dark and shadow regions, while suppressing the bright region. In our work, γ is set to 0.2. Then the image is processed by a band-pass filter that is the difference of Gaussian filtering, shown in Equ 6, to remove the influence of intensity gradients such as shading effects, while homomorphic filtering uses the high-pass filter.

$$DoG = (2\pi)^{-\frac{1}{2}}[\sigma_1^{-1}e^{-\frac{x^2+y^2}{(2\sigma_1)^2}} - \sigma_2^{-1}e^{-\frac{x^2+y^2}{(2\sigma_2)^2}}] \tag{6}$$

The reason of choosing the band-pass filter is that it not only suppresses low frequency information caused by illumination gradient, but also reduces the high frequency noise due to aliasing artifacts. In our work, σ_1 is set to 1 and σ_2 is set 2. Then, the two stage contrast equalisation presented in Equ 7 and Equ 8 is employed to further re-normalise the image intensities and standardise the overall contrast.

$$\mathbf{J}(x,y) = \frac{\mathbf{I}(x,y)}{(mean(|\mathbf{I}(x,y)|^a))^{\frac{1}{a}}} \tag{7}$$

$$\widehat{\mathbf{J}}(x,y) = \frac{\mathbf{J}(x,y)}{(mean(min(\tau, |\mathbf{J}(x,y)|)^a))^{\frac{1}{a}}} \tag{8}$$

a, set to 0.1, is used to reduces the influence of large values and τ,set to 10, is a threshold used to truncate large values after the first stage of normalisation. Lastly, a hyperbolic tangent function in Equ 9 is applied to suppress the extreme values and limit the pixel values in normalised image,$\widehat{\mathbf{I}}$, to a range between $-\tau$ and τ

$$\widehat{\mathbf{I}}(x,y) = \tau tanh(\frac{\widehat{\mathbf{J}}(x,y)}{\tau}) \tag{9}$$

5.1.2 Multi-scale Local Binary Pattern Histogram Discriminant Descriptor

The multi-scale local binary pattern histogram (MLBPH) representation with Linear Discriminant Analysis, LDA [6] is used in this report. Local binary pattern operators at R scales are first applied to a face image. This generates a grey level code for each pixel at every resolution. The resulting LBP images are cropped to the same size and divided into non-overlapping sub-regions, $\mathbf{M}_0$, $\mathbf{M}_1$,..$\mathbf{M}_{J-1}$. The regional pattern histogram for each scale is computed based on Equ (10)

$$\mathbf{h}_{P,r,j}(i) = \sum_{x',y' \in \mathbf{M}_j} B(LBP_{P,r}(x',y') = i) \quad | \quad i \in [0, L-1], r \in [1, R], j \in [0, J-1],$$
$$B(v) \begin{cases} 1 & \text{when } v \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$B(v)$ is a Boolean indicator. The set of histograms computed at different scales for each region, $\mathbf{M}_j$, provides regional information. L is the number of histogram bins. By concatenating these histograms into a single vector, we obtain the final multiresolution regional face descriptor presented in Equ(11)

$$\mathbf{f}_j = [\mathbf{h}_{P,1,j}, \mathbf{h}_{P,2,j}, \cdots, \mathbf{h}_{P,R,j}] \tag{11}$$

This regional facial descriptor can be used to measure the face similarity by fusing the scores of local similarity of the corresponding regional histograms of the pair of images being compared. However, by directly applying the similarity measurement to the multiscale LBP (MLBP) histogram [13], the performance will be compromised. The reason is that this histogram is of high dimensionality and contains redundant information. By adopting the idea from [3], the dimension of the descriptor can be reduced by employing the principal component analysis (PCA) before LDA. PCA is used to extract the statistically independent information as a prerequisite for LDA to derive discriminative facial features. Thus a regional discriminative facial descriptor, $\mathbf{d}_j$, is defined by projecting the histogram information, $\mathbf{f}_j$, into LDA space $\mathbf{W}_j^{lda}$, i.e.

$$\mathbf{d}_j = (\mathbf{W}_j^{lda})^T \mathbf{f}_j \tag{12}$$

This discriminative descriptor, $\mathbf{d}_j$, gives 4 different levels of locality: 1) the local binary patterns contributing to the histogram contain information at the pixel level, 2) the patterns at each scale are summed over a small region to provide information at a regional level, 3) the regional histograms at different scales are concatenated to produce multiresolution information, 4) the global description of face is established by concatenating the regional discriminative facial descriptors. Our results show that combining Multi-scale Local Binary Pattern Histogram with LDA is more robust in the presence of face mis-alignment and a uncontrolled environment.

5.1.3 Similarity measurement

After projecting the regional histogram into LDA space, the similarity measurement between query image $\mathbf{I_n}$ and the average of m template images, $Sim(\mathbf{I}, \mathbf{I_n})$ is obtained by taking the sum of the normalised correlation between the average of the regional discriminative descriptor $\mathbf{d}_j$ of the template images, and the regional discriminative descriptor $\mathbf{d}'_j$ of probe image respectively which is presented below.

$$Sim(\mathbf{I}, \mathbf{I_n}) = \sum_{j=0}^{J-1} \frac{\mathbf{d}_j \mathbf{d}'_j}{\|\mathbf{d}_j\| \|\mathbf{d}'_j\|} \tag{13}$$

5.1.4 Score Normalisation in each frame

In verification, the similarity score is degraded by many factors, such as a change of pose, illumination, occlusion and the characteristic of different persons enrolled in the system, and it will degrade the system performance as a predefined threshold for making a decision to accept or reject the claimed identity is chosen in an off-line training stage. Interestingly, although client specific thresholds can achieve a better adaptation to class specific distributions, as exemplified by Yang *et al.*'s Z-norm [22]. these methods are not effective when imaging conditions such pose, environment and sensor change. To cope with these problems, we propose to postprocess the similarity scores by test-normalisation(T-norm)[1] because it removes the score variation caused by condition changes. The T-norm is defined as:

$$Norm(\mathbf{I},\mathbf{I_n}) = \frac{Sim(\mathbf{I},\mathbf{I_n}) - \mu^C}{\sigma^C} \tag{14}$$

where the parameters, μ^C and σ^C, are the mean and standard deviation of the distribution of the similarity between a cohort impostor templates and an incoming image. Thus, T-norm is a test dependent approach. In this work, the cohort impostor templates are all the subject templates in the enrolment set except the template(s) for the claimed subject during testing and this is called Gallery norm. Lui *et al.* [11] has recently proposed nonlinear T-norm which is mapping the normalised score to the sigmoid function to improve the accuracy of the face verification and our simple T-norm results also show that the performance of our proposed methods mentioned in [7] can be boosted up by over 70%.

5.2 Scaling the Multi-scale Local Binary Pattern Histogram Discriminant (MLBPHLDA) Descriptor

In this report, two parameters of MLBPH descriptor are available to evaluate the algorithm performance in the scalable system. The first parameter is the total number of multi-scale operators. A small number of operators not only reduces the dimensionality of the combined histogram, but also degrades the system accuracy because of the associated loss of information. In this experiment, three different MLBPHLDA deescriptors are developed. The first system, called *MLBP1*, employs nine LBP operators by adjusting the radius of operator from 1 to 9. The second system, called *MLBP2*, employs five LBP operators by adjusting the radius of operator from 1 to 9 with step of two. The last system, called *MLBP4*, employs three LBP operators by adjusting the radius of operator from 1 to 9 with step of four. The second parameter determines the $k \times k$ non-overlapping rectangle size regions. A large number of regions increases the computation time and memory size as well as degrading the system accuracy in the presence of face localization errors. Therefore, the accuracy, robustness, memory consumption and time of the system are also evaluated in the report.

5.3 Experimental protocol

Two databases including their protocols and evaluation framework, are used for evaluating the algorithm.

5.3.1 Feret Image Database for evaluating the system accuracy and robustness

The Feret database[15] was collected at George Mason University and the US Army Research Laboratory facilities. The Colorado State University(CSU) face identification evaluation framework[18] used this database extensively, and an extensive set of performance figures achieved on this database is available for a range of research algorithms and commercial face recognition systems. The images are captured in grey scale at resolution 256 by 384. The database contains 14,126 images of which 3,816 are frontal images. This database is divided into a gallery set and four probe sets as summarised in Table 14.

Partition	Count	Description
Gallery (FA set)	1,196	Images taken with one of two facial expressions: neutral versus smile.
FB probe set	1,195	Images taken with other facial expressions.
FC probe set	194	Images taken under different illumination.
Dup I probe set	722	Subjects taken between a minute and 1031 days after their gallery entries.
Dup II probe set	234	Subjects taken at least 18 months after their gallery entries

Table 14: Description of the subsets of the FERET Database.

The open-source publicly available evaluation framework was utilised to test and benchmark the performance of our methods with others. In our work, two statistical measures are used to compare the performance of the methods. These statistical measures, namely the mean recognition rate at rank 1 and the probability of the algorithm outperforming another, are evaluated using a set of probe images and a set of gallery images. In this statistical test, a probe-gallery image pair for each subject is drawn from the corresponding 12 image pairs in each experiment involving 160 subjects and each subject has 4 images. In order to properly infer the quality of generalisation to a larger population of subjects, a permutation approach, generating a sampling distribution of the recognition rate for different rank order by repeatedly computing the recognition rate from different drawn datasets in 10,000 trials, is used. The mean of the recognition rate at rank 1 defined in [4] is the average of the recognition rate at rank 1 in total 1000 trials.

Let us denote the probability of the algorithm outperforming another in rank 1 by P(Alg1 > Alg 2). In order to estimate this quantity, the signed difference between the recognition rate of Alg1 and Alg 2 is computed in each trial, from a total of 1000 trials available[4]. Then, the quantity P(Alg1 > Alg 2) is determined by summing the probabilities of the differences greater than 0. The difference between Alg1 and Alg2 is considered *statistically significant* if the probability of P(Alg1 > Alg 2) is greater than or equal to 0.95. Otherwise, the performance of both algorithms is considered similar.

5.3.2 BANCA Video Database for evaluating the memory consumption and time taken

The BANCA Video database contains 52 subjects. The database is divided into two groups: Group 1 (G1) and Group 2 (G2). Each group contains 13 males and 13 females, i.e., they are

model name	IIntel® Core2 Duo
cpu MHz	3,000
cache size	6 MB
cpu cores	2 **(used only 1)**
memory	4.00 GB

Table 15: Parameters of the computer the tests were run on.

gender balanced. Each subject participated in 12 recording sessions in different conditions and with different cameras. Sessions 1-4 contain data under Controlled conditions while sessions 5-8 and 9-12 contain Degraded and Adverse scenarios respectively. Each session contains two recordings per subject, a true client access and an informed impostor attack.

5.4 Experiments in Face Identification

The objective of this experiment is to evaluate the system robustness and accuracy as the proposed parameters are adjusted. This experiments applied the CSU standard training set to estimate the parameters of the LDA. In this test, the mean recognition rate with 95% confidence interval and the probablility of the algorithm outperforming another are used to compare the system performance. The results of three MLBPHLDA systems with different $k \times k$ regions are plotted in Figure 10.

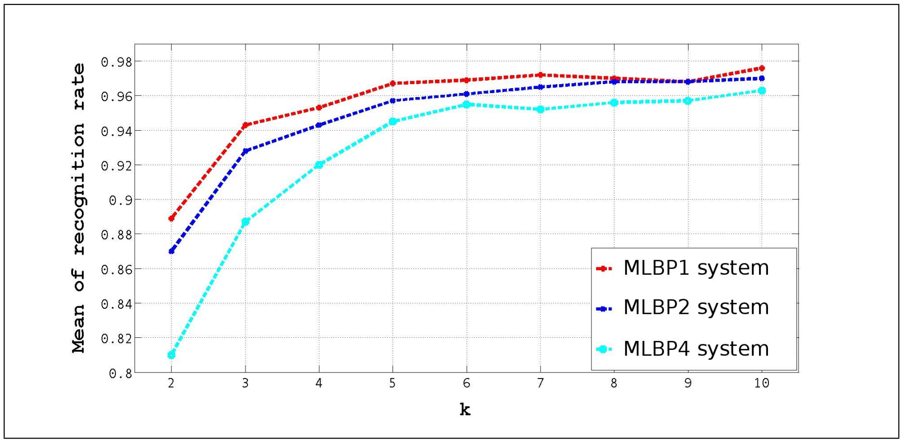


Figure 10: The mean recognition rate with 95% confidence interval for three MLBPHLDA systems against the number of ($k \times k$) of regions.

Results It is clearly shown that the more LBP operators are involved, the higher recognition is achieved, but the system complexity increases. As expected for the MLBPHLDA methods, the mean recognition rate is robust for a wide range of values of k. For examples, in the MLBP1 system, using 9 LBP operators, the accuracy of k =10 and k=5 is not significantly different as P((k=10) > (k=5))=0.6777. In other words, changing the number of regions, k, only affects the length of the feature vector and the computation time. In the presence of the face localization error, the performance of the face recognition method involving spatial information as an input degrades; however, the MLBPHLDA system using smaller k can be expected to maintain the recognition accuracy. These finding are discussed further in the next section.

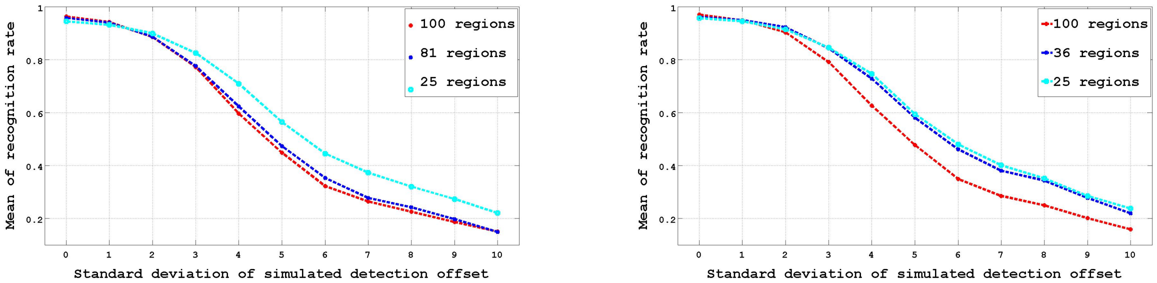
Robustness to face localization error

A generic face recognition system first localizes and segments a face image from the background before recognizing it. However, a perfect face localization method is very difficult to achieve, and therefore a face recognition method capable of working well in the presence of localization errors is highly desired. The training images and the gallery images in the FA set, are registered using the groundtruth eye coordinates but the probe sets (FB, FC, Dup I and II) are registered using simulated eye coordinates. To simulate the error caused by the translation, rotation, occlusion and scale effects to the normalized face image, left and right eye coordinates in the second test are computed by adding different random vectors (δX_{eyeL}, δY_{eyeL}, δX_{eyeR}, δY_{eyeR}) of disturbances to the groundtruth eye locations. These vectors are uncorrelated and normally distributed with a zero mean and standard deviation, σ, from 0 to 10. In this experiment, MLBP1 at $k = 5$ and 10, MLBP2 at $k = 5, 6$ and 10 and MLBP4 at $k = 5, 9$ and 10 are evaluated.

Results The mean recognition rates of MLBP2 and MLBP4 systems using respective values of parameter k against the standard deviation of the simulated localization are plotted in Figure 11a and 11b. As expected, the larger region size and the small number of regions, the better the recognition rate as the localization error increases. However, if the k is fixed, the trend of the recognition rates of all 3 MLBPHLDA systems against the standard deviation of the simulated localization plotted in Figure 12a and 12b is very similar. However, the more LBP operators employed in the system, the better the recognition rate that can be achieved, but the computation time and the memory consumption will increase.

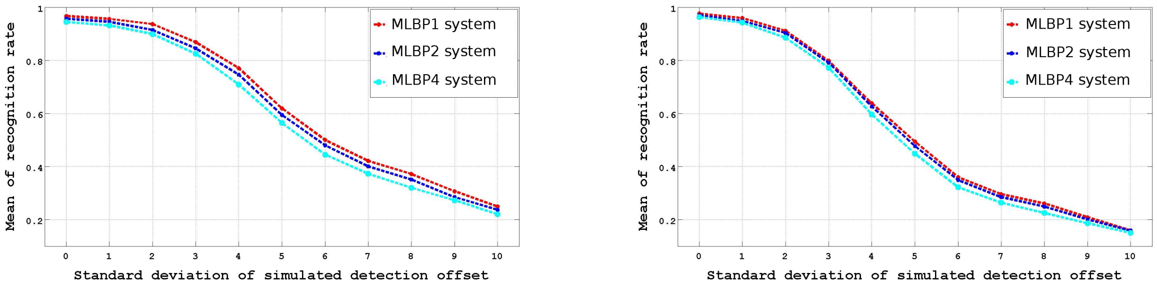
5.5 Experiment in Face Verification

The objective of this experiment involving BANCA Videos is to evaluate the memory consumption and the computation time as the proposed parameters are adjusted. In this experiment, G1 enrollment set containing the average of frame templates for each subject and 20 videos in G1 test set are used. The hardware platform of this experiment is Intel Core2 Duo CPU E8400@3GHz. The memory consumption is measured by Valgrind. The memory consumption and the computation time of MLBP1 with $k = 5$ and 10, MLBP2 with $k = 5$ and 10 and MLBP4 operators with $k = 5$ and 10 and their corresponding T-norm version (TN) is reported in Table 54 and 55 (in the Appendix).



(a) MLBP2 (b) MLBP4

Figure 11: The mean recognition rate with 95% confidence interval for 2 MLBPHLDA systems with different number of non-overlapping regions against the standard deviation of the simulated localization error.



(a) k = 5 (b) k = 10

Figure 12: The mean recognition rate with 95% confidence interval for 3 MLBPHLDA systems with $k = 5, 10$ against the standard deviation of the simulated localization error.

Results As expected, using larger k consumes more memory. For example, the memory consumption of those systems at $k = 10$ is more than double that at $k = 5$, but the accuracy is slightly better in manually annotated face images. On the other hand, the accuracy for $k = 10$ is rapidly degraded as the localization error increases. In conclusion, smaller k is prefered in terms of the memory consumption, computation time and the robustness of the system.

It is clearly shown that with more LBP operators employed by the system, a better recognition rate can be achieved but the processing takes more memory and computation time. In balance, the MLBP2 system at $k = 5$ is preferred because the characteristic is similar to the MLBP1 system at $k = 5$ but takes less memory and computation time. Comparing the system with T-norm, the memory requirement differ by only around 1 or 4 MB different but the system with T-norm provides better recognition rate reported in D3.4.

5.6 Conclusions and Future work

In this report, the total number of multi-scale operators and the $k \times k$ non-overlapping rectangle size regions in MLBPHLDA system have been evaluated in term of accuracy, system robustness, memory consumption and computation time. The best compromise configuration is the MLBP2 system which uses 5 LBP operators at $k = 5$. The bottleneck of the MLBPHLDA system is in the LDA projection process where it takes around 90% memory usage and computation time in the whole process. Therefore, LDA projection process should be improved in the future work.

6 Speaker verification

This section present the different ways explored for the scalability of a speaker recognition system. Both systems presented by BUT and by LIA are studied.

6.1 BUT

6.1.1 Baseline system

The baseline system for speaker verification is a GMM system that is based on standard GMM-UBM paradigm [16]. It employs number of techniques that has previously proven to improve GMM modeling capability and help fight against the eternal problem in speaker verification - diversity in channel and acoustic condition. The system contain UBM with 2048 Gaussians which model 13 Mel frequency cepstral coefficients (MFCC) coefficients (including zero'th cepstral coefficients, 20ms window, 10ms shift, 23 bands in Mel filter bank) augmented with their delta, double and triple deltas followed by HLDA with dimensionality reduction from 52 to 39. We used eigen-channel compensation for coping with the session variability. For detailed description of the system see deliverable D3.2 [5].

6.1.2 Experimental protocol

For time and computational consumption estimation we ran all tests on a standard PC. The PC characteristics are presented in Table 16.

model name	Intel® Core2 Duo
cpu MHz	3,000
cache size	6 MB
cpu cores	4 **(used only 1)**
memory	2 GB

Table 16: Parameters of the computer the tests was run on.

All experiments are performed on the BANCA database and on the data from NIST 2006 Speaker Recognition Evaluation, because of the comparison with previous already published results and because the test set is much bigger then in the BANCA database. The tests were run on the offline system. The memory consumption for the online system will be smaller, because we will not load whole recording and features to the memory, but it will be computed online.

6.1.3 Voice activity detection

The Voice activity detection (VAD) is the first step in the speaker verification process. The role of VAD is to extract only the parts where the speaker is speaking.

We used three types of VAD in our experiments:

- baseline NN based - downscaled version of our phoneme recognizer - 1 state per phone, 62 phones in phone set, 200 neurons in hidden layer.

- NN fast - downscaled phoneme recognizer - 1 state per phone, reduced phone set (32), 50 neurons in hidden layer.

- GMM based - analyzing energy contour of the audio (see VAD subsection in [5]) - the system is trained on the audio itself.

The comparison of the complexity of the VAD can be seen in the Table 56. The Energy based VAD is in terms of complexity the simplest one, because it consists of 3 GMM components trained on 1 dimensional features, giving 9 parameters in total (a mean, variance and weight for each component) and since they are estimated for each utterance separately we do not need to store them. On the other hand the NN based recognizer has 200 neurons in hidden layer and 3 Neural networks in the structure, thus needing 91254 parameters to represent the VAD. This kind of VAD can, however, better handle non-speech events like ringing tones and fax transmissions.

The results of speaker verification module with this three VAD and the evaluation of different VAD in the speaker verification system in the Table 57.

The complexity of the VAD does not affect much the complexity of whole system, because speaker verification (SV) module has several times more parameters. The energy VAD in concatenation with SV module works more then 2 times faster then the baseline system. But its performance significantly drops, probably because the SV module was not trained with such VAD. However, tests with more a advanced SV module show promising results with energy based VAD where we do not see any degradation of performance even though it is much faster.

6.1.4 Number of Gaussians

The number of Gaussian components is the one of the most promising parameters with which we can easily downscale the system in terms of memory and speed requirements. For this experiment we used an eigen-channel matrix with 50 eigen-vectors - the one which comes from the baseline system and will be further examine in the Section 6.1.6. The detailed results are in the Table 58. We varied the number of Gaussian components from 2048 (baseline) to 256. The memory consumption dropped to 30% from the baseline and speed to 75% while the performance dropped for all test sets by about 1% absolute. The conclusion from this analysis is that 512 Gaussian components is very good compromise.

6.1.5 Feature dimensionality

The feature dimensionality is also very good start mainly for memory. We have used downscale system from the previous experiment for this one. We used 512 Gaussian component with 50 eigen-channels. There are 3 experiments we have carried out. We used MFCC coefficients (13 coefficients) and appended them with first derivatives (MFCC_0D, 26), first and second derivatives (MFCC_0DA,39) and first, second and third derivatives followed by HLDA (52) with dimensionality reduction (MFCC_0DAT_HLDA, final 39 coefficients). The detailed analysis is in the Table 59. The outcome of the experiment is that it is not worth decreasing the feature dimensionality for this system, because of the performance drop with no significant improvement in memory and speed usage.

6.1.6 Number of vectors in the eigen-channel matrix

The main expectation of decreasing the number of vectors in eigen-channel matrix is the memory consumption. We varied the number of vectors from 50 (baseline) to 10 with a step of 10 vectors. Each step reduction of the matrix by 10 vectors is about 20% reduction in memory from the baseline. Which means that 10 vectors is about 20% of parameters of the baseline model. The performance decrease is negligible till 30 vectors, after that the performance of the system starts decreasing significantly. Good compromise is 30 vectors with respect to performance and memory usage. The detailed analysis is in the Table 60.

6.1.7 Best compromise

Considering the previous results, and tuning each parameter with respect to the others, the optimal downscaled configuration of the system is:

- using faster VAD
- model size reduction from 2048 to 512 Gaussians
- keep the same features as in the baseline
- smaller eigen-channel matrix - reduction from 50 to 30 vectors

The computation resources of downscaled system is reduced by 50% from the baseline system. Memory consumption decrease to 29% and performance decrese in average about 10% relative or 1% absolute.

	Baseline	Best Compromise
EER BANCA G1 (%)	7.16	8.48
EER BANCA G2 (%)	5.27	5.19
EER NIST 2006 (%)	5.31	6.52
Memory (%)	100	29.0
Memory (MB)	48.48	14.04
Computational Time = RT	0.0522	0.0277
Computational Time (%)	100	53.1

Table 17: Performance and computational and memory consumption for the optimal downscaled configuration of the system. Resource consumption are given in terms of percentage relative to the baseline system.

6.2 LIA

6.2.1 Baseline system

The system for the LIA speaker verification system is a standard GMM-UBM approach based on the open-source biometric platform MISTRAL/ALIZE[1] [8]. The front-end processing consists of extracting parameters from the speech signal by using a filter-bank analysis (linear filter). Acoustic features are composed of 50 Linear Frequency Cepstral Coefficients (LFCC): 19 static coefficients (c), 19 delta (Δc) and 11 delta-delta ($\Delta\Delta c$) and the delta energy (Δe). Coefficients are obtained as follows: 24 filter bank coefficients are first computed over 20ms Hamming windowed frames at a 10ms frame rate. Bandwidth is limited to the 300-3400Hz range. A classical energy-based frame pruning system is applied to normalise the recordings, file-by-file (cepstral mean subtraction and variance normalisation). The UBM model size is set to 512 components (with diagonal covariance matrix). In the specific context of BANCA evaluation, no post-processing is performed (no score normalisation). For a full description of this system refer to deliverable D3.2 [5].

6.2.2 Experimental protocol

model name	Intel® Core2 Quad
cpu MHz	2000
cache size	3 MB
cpu cores	4 **(used only 1)**
memory	1 GB

Table 18: Parameters of the computer the tests was run on.

[1] `http://mistral.univ-avignon.fr`

For time and computational consumption estimation, all tests are performed on a standard PC. The PC characteristics are presented in Table 18.

All experiments presented in 6.2.3, 6.2.4 and 6.2.5 are performed on the BANCA database.

6.2.3 Number of Gaussian components in the UBM

The number of components per GMM explicitly determines the memory occupation for model storage as well as computational time. Experiments were performed in order to evaluate the influence of the number of components of the GMM on the overall performance.

Table 61 gives the performance of the GMM/UBM system in terms of equal error rate, according to the number of Gaussian distributions per GMM (from 512 to 32). This table also links the number of components to the memory occupation and CPU consumption.

Downscaling the number of components per GMM from 512 to 32 increases the EER from 3.48% up to 5.15% for the group 1 of Banca. Nevertheless, this reduction allows strong successive improvements in reducing memory occupation and computational time. Reducing the number of components to 128 allows to keep performance equivalent to the baseline system while dividing by 2.9 the memory and by 3 the computational time. According to these results, 128 Gaussian components per GMM is a good compromise.

6.2.4 Size of the acoustic vector

The size of the acoustic vectors is another crucial parameter which determines the time and memory consumption. For this reason, several configurations mixing c, Δc, $\Delta\Delta c$, Δe coefficients, have been evaluated. As the number of combination is very important, only four configurations are proposed (in addition to the reference one). To only study the influence of the feature vector size/composition, the UBM is composed of 512 Gaussian components.

Table 62 gives the detail of each configuration and provides the resulting score (% EER) and the related resources for each of them. Removing part of the acoustic coefficients allows to reduce drastically the memory and CPU time consumptions. For example, the resource saving goes from 27% for memory consumption to 13% for computational time by processing only 30-dimensional acoustic vectors (instead of 50 for the baseline). In the same time the EER still remains under 5%.

6.2.5 Number of selected frames

Acoustic features are generated every 10ms as described in Section 6.2.1. Each of these features are used by the baseline system in order to estimate the accumulated log-likelihood on the test segment. We propose to skip features during the scoring process. Instead of using each parameters frame to compute likelihood, we use only 1 frame of 2 or 1 frame of 4. Nevertheless, The parameterisation step remains unchanged as the Δc and $\Delta\Delta c$ computation requires the extraction of every LFCC vectors. Results obtained using only 1 frame each 2 or 4 are given in Table 63.

Processing only 1 frame each 2 or 4 allows a significant saving in terms of CPU time consumption (77%) but not in terms of memory. Moreover, EER still remains under 5% when processing 1 frame out of 4, which is still an acceptable performance deterioration. Detailed results are presented in Table 63.

The frame-skipping only consists in downsampling the scoring process. We can assume that a better frame selection could lead to strongly improve the performance. However, the simple frame skipping was chosen as a better frame selection would also increase computational time.

6.2.6 Conclusion

Regarding these results, two systems which seems to be good compromises are proposed here.

Minimal system A minimal configuration has been designed for computational power and memory saving. Performance of this configuration are given in Table 19. In this configuration, GMM are composed of 32 Gaussian components, the dimension of feature vectors is 20 and 1 frame over 4 is processed for likelihood computation. Results obtained with this configuration show that EER could still remains under 8% while saving up to 83% of memory and 88.3% of the computational time.

Best compromise system A second downscaled configuration which seems to be optimal according to the MOBIO constraints is the following:

- model size reduction from 512 to 128
- 30 dimensional features ($10c+ 10\Delta c + 10\Delta\Delta c$)
- 50% of frames processed

Results obtained with this configuration are presented in Table 19 and show that memory consumption could be reduced by 72% and computational time by 82% from the baseline system while keeping EER under 5%.

	Baseline	Best Compromise	Minimal
EER BANCA G1 (%)	3.48	4.77	7.72
EER BANCA G2 (%)	2.94	2.94	7.34
Memory (%)	100	28	17
Memory (MB)	7.84	2.19	1.37
Computational Time = RT	0.0052	0.0006	0.0001
Computational Time (%)	100	11.7	1.7

Table 19: Performance and computational and memory consumption for the optimal downscaled configuration of the system. Resource consumption are given in terms of percentage relative to the baseline system.

7 Multimodal Scalability

7.1 Methodology

Having presented the unimodal scaled systems, this section presents a scalability study at the multimodal level.

In order to achieve the above, there should be a way to compare the complexity of each system. By complexity, we understand that this is a *relative* notion, i.e., the complexity of a system is defined as one that is relative to the baseline system. Thus, the complexity of the baseline system always has a unit cost of one. A lighter system, either taking less memory allocation (the "space" criterion) or faster in speed (the "time" criterion), has a complexity cost of less than one.

Since both space and time are equally important constraints, assigning a single cost is a debatable subject. To this end, we opt for the following strategy:

Between the space and time complexity, choose the one that is the most significant in changes with respect to the baseline system.

For example, referring to Table 58, which shows the system complexity as a function of the number of Gaussian components, the space complexity is clearly more *important* than the time complexity. For instance, by reducing the number of Gaussian components from 2048 to 256, the memory consumption is reduced to 30.1% whereas the time complexity is reduced only to 75.0%. In this case, since the memory reduction is more significant, the system with 256 Gaussian component has a cost of 0.3 (recalling that the baseline system has a cost of 1).

Another cost assignment strategy is to take the weighted sum between the two complexities. However, in this case, fixing the coefficient associated with each cost is again subject to debate and highly application- and policy-dependent. Our objective here is not to explore all possible cost assignment strategies, but to show that by adopting a reasonable cost assignment strategy, multimodal scalability study can be performed somewhat more objectively.

7.2 Exhaustive Fusion Performance Analysis

We list the cost assignments of the face systems without facial alignment, those with facial alignments, the scaled speaker verification systems developed by BUT and those developed by LIA (UPV) in Tables 64–67, respectively (in the Appendix). We summarize these tables by plotting a cost vs. performance curve, as shown in Figure 13. Each scaled system (whether face or speech) is plotted as a point in this figure.

Since there are 48 face systems and 27 speech systems, an exhaustive bimodal fusion will result in $48 \times 27 = 1296$ combinations. We have conducted all these fusion experiments. The idea is then to find among these systems, which pair of combination (the fusion system) will be optimal in terms of performance.

From the 1296 possible fusion systems, we chose the top 20 systems in each cost band

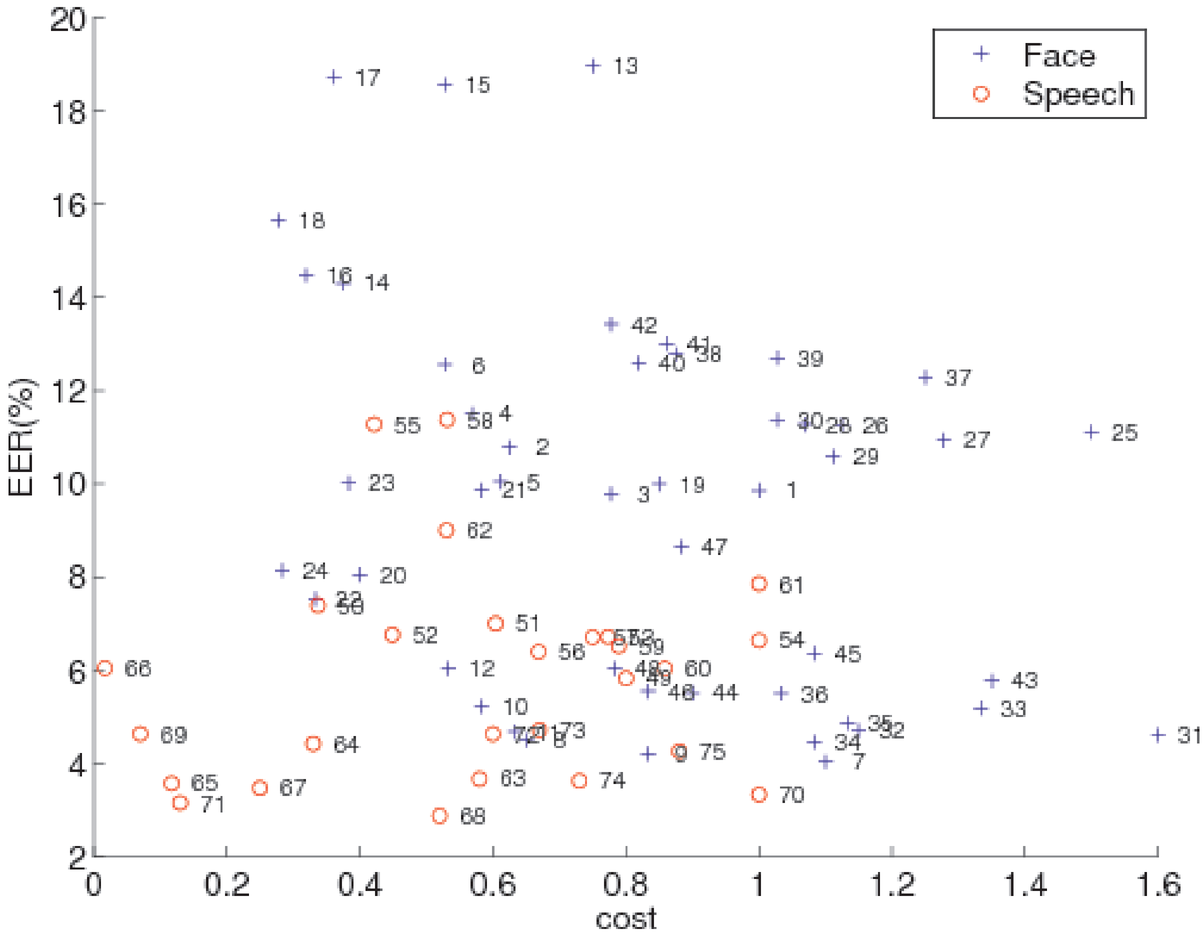


Figure 13: A plot of cost versus performance. The number associated with each point is the system index as listed in Tables 64–67.

in the following ranges: $0 < cost \leq 0.1$, $0.1 < cost \leq 0.2$ and so on until $1.9 < cost \leq 2$. The results are shown in Figure 15. As can be observed, higher system costs generally entail better performance (lower EER). This shows that the cost definition we adopted is reasonable and corresponds well to the actual scenario. Figure 15 can be used as a guideline for engineers and system designers to decide which system to deploy under a given constraint and performance expectation.

7.3 Subset of Selected Fusion Systems

From the 1296 possible fusion systems, we chose a subset exhibiting a reasonable trade-off between complexity and performance. We used the following criteria:

- unit cost less than 0.6

- performance less than 10 EER%

These criteria lead to the selection of 6 face systems and 11 speech systems, as shown in Figure 14. Note that these systems are but a subset of those shown in Figure 13.

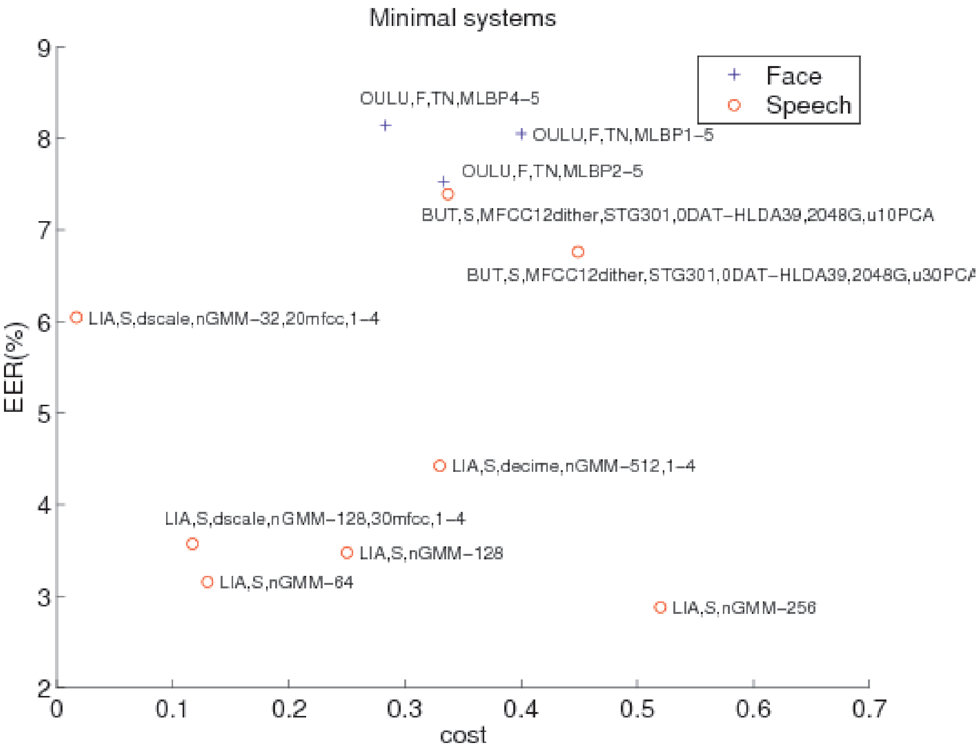


Figure 14: A of cost versus performance for cost < 0.6 and EER < 10%

The fusion experiments among 6 face systems and 11 speech systems are summarized in Table **??**. As can be observed in this table, the overall best performing fusion system for this subset is face system 21 (OULU,F,TN,MLBP2-10) and speech system 68 (LIA,S,nGMM-256). The total cost of these two systems is $0.58 + 0.52 = 1.10$.

7.4 Summary

In this section, we presented a methodology for multimodal biometric scalability. The idea consists of first defining an *abstract* cost, allowing different systems to be compared on common ground, on the basis of the relative memory and speed-up with respect to the baseline systems. By adopting this cost definition, it becomes relatively straightforward to assess the cost of a given multimodal system by linearly adding the costs of the constituent systems. We demonstate the effectiveness of this strategy on 1296 fusion systems carried out the bimodal BANCA (speaking face) and speech database.

Speech	Face systems					
systems	10	12	20	21	22	24
50	2.72	2.80	**2.55**	3.03	3.23	3.40
52	**2.52**	2.72	2.53	2.82	3.15	2.94
62	**3.10**	3.24	4.44	4.11	4.63	4.85
63	1.86	1.86	**1.70**	1.74	1.71	2.14
64	2.09	**2.01**	2.52	2.06	2.42	2.51
65	**1.30**	1.51	1.56	1.49	1.64	1.91
66	**1.60**	2.00	2.66	2.59	3.34	3.20
67	**1.15**	1.30	1.81	1.98	2.00	2.02
68	1.33	1.43	1.05	$\ast$ **0.93**	1.25	1.43
69	**1.45**	1.82	2.05	2.19	1.86	2.14
71	**1.04**	1.48	1.73	2.04	1.75	2.19

Table 20: Pairwise fusion performance, in terms of averaged EER (%) between g1 and g2, consisting of 6 face systems and 11 speech systems.

Note: the smallest number in a row is printed in bold. $\ast$ denotes the best system

8 Summary

This report contains all algorithms dedicated to scalable system developed during the MOBIO project.

One aim of this project is the integration of a biometric authentication system into a mobile phone. However, classical systems are not designed to fit with embedded constraints like memory size or computational time. For this reason, each step of the authentication system has been studied to allow the integration into a mobile (the Nokia N900©). Steps related to face authentication are detailed in Section 2, 3, 4 and 5. Speaker recognition is presented in Section 6. Last, fusion system improvements are described in Section 7.

All system modifications allowing to fit with mobile phone constraints have been evaluated on the well known BANCA database.

As a conclusion, the main parameters of the authentication system have been downscaled to be integrated on the mobile phone. This has been shown to allow a relative decrease in memory consumption of up to 70%/90% (Sections 3,4 and 6). Considering computational time, downscaling the speaker verification engine saves between 50% and 90% of the time required for speaker verification. Experiments performed on the BANCA database have shown that downscaling the biometrics engine does not increase the error rate by much, which is still acceptable considering the specificity of embedded context.

Acknowledgements

The authors would like to thank Honza Cernocky (BUT), Pavel Matejka (BUT), Sebastien Marcel (IDIAP), Chris McCool (IDIAP), Jean-Francois Bonastre (LIA), Driss Matrouf (LIA), Matti Pietikinen (UOULU), Abdenour Hadid (UOULU), Tim Cootes (UMAN), Phil Tresadern (UMAN), Josef Kittler (UNIS), Norman Poh (UNIS), Chi-Ho Chan (UNIS), Markus Turtinen and Mika Helistekangas for their contributions to and review of this report.

References

[1] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42 – 54, 2000.

[2] A. Ernst B. Froba. Face detection with the modified census transform. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 91–96, 2004.

[3] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, Jul 1997.

[4] J.R. Beveridge, K. She, B.A. Draper, and G.H. Givens. A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–535–I–542 vol.1, 2001.

[5] Harish Bhaskar and Philip A. Tresadern. D3.2: Report on the description and evaluation of baseline algorithms for unimodal authentication. Technical report, The MOBIO project, 2008.

[6] Chi-Ho Chan, Josef Kittler, and Kieron Messer. Multi-scale local binary pattern histograms for face recognition. In Seong-Whan Lee and Stan Z. Li, editors, *ICB*, volume 4642 of *Lecture Notes in Computer Science*, pages 809–818. Springer, 2007.

[7] Chi Ho CHAN, Josef Kittler, Norman Poh, Timo Ahonen, and Matti Pietikainen. (multiscale) local phase quantisation histogram discriminant analysis with score normalisation for robust face recognition. In *ICCVWS*, 2009.

[8] Eric Charton, Anthony Larcher, Christophe Lévy, and Jean-Francois Bonastre. Mistral: open source biometric platform. In *Symposium on Applied Computing (ACM)*, Sierre (Switzerland), march 2010.

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vis.*, 61(1):55–79, January 2005.

[10] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of the Japanese Society for Artificial Intelligence*, 14(5):771–780, September 1999.

[11] Yui Man Lui, J. Ross Beveridge, Bruce A. Draper, and Michael Kirby. Image-set matching using a geodesic distance and cohort normalization. In *FG*, pages 1–6, 2008.

[12] S. Marcel and Y. Rodriguez. `http://torch3vision.idiap.ch`.

[13] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[14] OpenCV library. http://opencvlibrary.sourceforge.net/.

[15] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, 2000.

[16] D. A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Eurospeech*, pages 963–966, Rhodes, Greece, September 1997.

[17] Yann Rodriguez. *Face detection and verification using local binary patterns.* PhD thesis, Idiap/EPFL, 2006.

[18] Marcio Teixeira Ross Beveridge, David Bolme and Bruce Draper. *The CSU Face Identification Evaluation System User's Guide: Version 5.0.*, 2003.

[19] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In Shaohua Kevin Zhou, Wenyi Zhao, Xiaoou Tang, and Shaogang Gong, editors, *AMFG*, volume 4778 of *Lecture Notes in Computer Science*, pages 168–182. Springer, 2007.

[20] Paul Viola and Michael Jones. Robust real-time object detection. In *Second International Workshop on Statistcal and Computational Theories of Vision - Modelling, Learning, Computing, and Sampling*, 2001.

[21] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vis.*, 57(2):137–154, 2004.

[22] Fei Yang, Shiguang Shan, Bingpeng Ma, Xilin Chen, and Wen Gao. Using score normalization to solve the score variation problem in face authentication. In *IWBRS*, pages 31–38, 2005.

A Face detection with Modified Census Transform

In the following table (Table 21) we provide the accuracy of the Baseline face detector on several databases.

	Accuracy (%)
BANCA English	99.88%
BANCA French	99.83%
BANCA Spanish	99.50%
Purdue	99.98%
XM2VTS Frontal	99.75%
XM2VTS Darkened	99.79%
BioSign	99.82%

Table 21: The accuracy of the baseline face localisation system.

B Viola-Jones face detection in fixed point arithmetic

	time	memory	Missed detections	d_{max}	
	ms / frame	MB		Med.	90%
$s_y = 1, s_x = 1$	143 (140 %)	3.85 (47 %)	59 (0.95%)	0.079	0.13
$s_y = 1, s_x = 2$	76 (75 %)	3.85 (47 %)	176 (2.8%)	0.083	0.14
$s_y = 2, s_x = 2$	42 (41 %)	3.85 (47 %)	541 (8.7%)	0.096	0.16

Table 22: Results for different window step sizes

	time	memory	Missed detections	d_{max}	
	ms / frame	MB		Med.	90%
$s_{\text{scale}} = 1.05$	140 (137 %)	3.85 (47 %)	71 (1.1%)	0.08	0.13
$s_{\text{scale}} = 1.10$	76 (75 %)	3.85 (47 %)	176 (2.8%)	0.083	0.14
$s_{\text{scale}} = 1.15$	67 (66 %)	3.85 (47 %)	269 (4.3%)	0.087	0.14
$s_{\text{scale}} = 1.20$	43 (42 %)	3.85 (47 %)	563 (9%)	0.15	0.26
$s_{\text{scale}} = 1.30$	50 (49 %)	3.85 (47 %)	753 (12%)	0.13	0.26
$s_{\text{scale}} = 1.40$	26 (25 %)	3.85 (47 %)	934 (15%)	0.13	0.23

Table 23: Results for changing the window scale parameter

	time	memory	Missed detections	d_{max}	
	ms / frame	MB		Med.	90%
$d = 1$	76 (75 %)	3.85 (47 %)	176 (2.8%)	0.083	0.14
$d = 2$	76 (75 %)	1.57 (19 %)	216 (3.5%)	0.083	0.14
$d = 3$	43 (41 %)	1.07 (13 %)	243 (3.9%)	0.084	0.14
$d = 4$	22 (22 %)	0.90 (11 %)	323 (5.2%)	0.087	0.14
$d = 5$	17 (17 %)	0.82 (10 %)	462 (7.4%)	0.097	0.16

Table 24: Face detection results when downscaling the input image

	time	memory	Missed detections	d_{max}	
	ms / frame	MB		Med.	90%
$d = 1, s_y = 1, s_x = 2$	9.2 (9.0 %)	3.85 (47 %)	176 (2.8%)	0.14	0.23
$d = 2, s_y = 1, s_x = 1$	7.5 (7.4 %)	1.57 (19 %)	125 (2%)	0.13	0.21
$d = 2, s_y = 1, s_x = 2$	6.1 (6.0 %)	1.57 (19 %)	216 (3.5%)	0.14	0.23
$d = 3, s_y = 1, s_x = 1$	5.3 (5.2 %)	1.07 (13 %)	100 (1.6%)	0.14	0.24

Table 25: Results for the detector stopping at largest face

C Face alignment

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
00pct	0.086	0.155	-	-	-	-	0.069	0.121
50pct	0.074	0.147	-	-	-	-	0.058	0.111

Table 26: Accuracy with respect to number of appearance model modes for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
00pct	82	80	13.3926
50pct	84	90	13.3858

Table 27: Efficiency with respect to number of appearance model modes for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
00pct	0.064	0.119	0.179	0.293	0.126	0.204	0.073	0.105
50pct	0.063	0.133	0.186	0.342	0.129	0.249	0.072	0.126

Table 28: Accuracy with respect to number of appearance model modes for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
00pct	78	80	13.4409
50pct	82	80	13.4341

Table 29: Efficiency with respect to number of appearance model modes for XM2VTS.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
001it	0.082	0.163	-	-	-	-	0.066	0.131
002it	0.075	0.144	-	-	-	-	0.060	0.118
010it	0.076	0.144	-	-	-	-	0.059	0.116
100it	0.076	0.147	-	-	-	-	0.059	0.119

Table 30: Accuracy with respect to varying number of iterations for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
001it	66	70	13.451
002it	89	90	13.4419
010it	157	140	13.4419
100it	212	140	13.4419

Table 31: Efficiency with respect to number of iterations for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
001it	0.064	0.136	0.178	0.314	0.124	0.208	0.068	0.114
002it	0.052	0.110	0.181	0.317	0.122	0.205	0.065	0.103
010it	0.049	0.114	0.198	0.382	0.127	0.240	0.067	0.118
100it	0.049	0.120	0.196	0.389	0.127	0.249	0.067	0.120

Table 32: Accuracy with respect to number of iterations for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
001it	65	60	13.4902
002it	85	90	13.4902
010it	139	130	13.4902
100it	162	130	13.4902

Table 33: Efficiency with respect to number of iterations for XM2VTS.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
05pts	0.192	0.384	-	-	-	-	0.147	0.340
07pts	0.101	0.223	-	-	-	-	0.083	0.186
12pts	0.072	0.155	-	-	-	-	0.058	0.126
18pts	0.074	0.141	-	-	-	-	0.059	0.112

Table 34: Accuracy with respect to number of facial features localized for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
05pts	29	30	6.33414
07pts	39	40	7.62141
12pts	66	70	10.2672
18pts	93	90	13.6517

Table 35: Efficiency with respect to number of facial features localized for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
05pts	0.056	0.262	0.078	0.328	-	-	0.045	0.161
07pts	0.043	0.117	0.094	0.214	-	-	0.047	0.092
12pts	0.048	0.118	0.123	0.288	0.095	0.214	0.056	0.107
18pts	0.067	0.140	0.147	0.314	0.116	0.254	0.060	0.112

Table 36: Accuracy with respect to number of facial features localized for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
05pts	27	30	6.34351
07pts	38	40	7.62893
12pts	63	60	10.2766
18pts	86	90	13.661

Table 37: Efficiency with respect to number of facial features localized for XM2VTS.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
060pct	0.123	0.263	-	-	-	-	0.103	0.229
200pct	0.068	0.130	-	-	-	-	0.053	0.103

Table 38: Accuracy with respect to size of feature point template for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
060pct	70	70	10.2063
200pct	267	270	27.6327

Table 39: Efficiency with respect to size of feature point template for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
060pct	0.063	0.123	0.199	0.360	0.137	0.261	0.074	0.129
200pct	0.048	0.126	0.215	0.449	0.134	0.310	0.069	0.147

Table 40: Accuracy with respect to size of feature point template for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
060pct	69	70	10.206
200pct	263	260	27.5935

Table 41: Efficiency with respect to size of feature point template for XM2VTS.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
bcm	0.107	0.252	-	-	-	-	0.086	0.220
ncc	0.075	0.144	-	-	-	-	0.058	0.119

Table 42: Accuracy with respect to texture model for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
bcm	119	120	17.7042
ncc	96	100	10.7791

Table 43: Efficiency with respect to texture model for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
bcm	0.053	0.151	0.206	0.381	0.134	0.269	0.069	0.132
ncc	0.051	0.219	0.195	0.402	0.129	0.314	0.067	0.166

Table 44: Accuracy with respect to texture model for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.4901
bcm	118	120	17.7524
ncc	94	100	10.7952

Table 45: Efficiency with respect to texture model for XM2VTS.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
gapp	0.356	0.478	-	-	-	-	0.305	0.400

Table 46: Accuracy with respect to point prediction method for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
gapp	75	80	10.4256

Table 47: Efficiency with respect to point prediction method for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
gapp	0.264	0.321	0.946	1.273	0.694	0.896	0.389	0.522

Table 48: Accuracy with respect to point prediction method for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
gapp	75	80	10.435

Table 49: Efficiency with respect to point prediction method for XM2VTS.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.074	0.143	-	-	-	-	0.059	0.116
no-tracker	0.071	0.240	-	-	-	-	0.059	0.168

Table 50: Accuracy with respect to optimization method for BANCA.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	106	110	13.459
no-tracker	32	30	0.191595

Table 51: Efficiency with respect to optimization method for BANCA.

	Eye points		All points					
	d_{max}		d_{max}		d_{90}		d_{mean}	
	Med.	90%	Med.	90%	Med.	90%	Med.	90%
baseline	0.049	0.110	0.189	0.331	0.123	0.231	0.066	0.107
no-tracker	0.053	0.272	0.224	0.374	0.161	0.319	0.069	0.158

Table 52: Accuracy with respect to optimization method for XM2VTS.

	Time (ms)		Mem. (Mb)
	Med.	Mean	Peak
baseline	107	110	13.5013
no-tracker	33	30	11.7549

Table 53: Efficiency with respect to optimization method for XM2VTS.

D Scalable mobile phone-based system for Face verification

	$k = 5$	$k = 10$
MLBP1	133.356	412.805
MLBP2	88.169	229.691
MLBP4	64.020	135.929
TN,MLBP1	134.394	416.947
TN,MLBP2	89.140	233.474
TN,MLBP4	64.839	139.135

Table 54: Memory consumption in term of mega byte (MB) for the proposed parameters in 3 MLBHLDA systems.

	$k = 5$	$k = 10$
MLBP1	93.17	98.26
MLBP2	53.24	55.84
MLBP4	33.35	34.58
TN,MLBP1	93.08	98.28
TN,MLBP2	53.46	56.02
TN,MLBP4	33.35	34.85

Table 55: Computation in milliseconds for the proposed parameters in 3 MLBHLDA systems in a frame matching excluding LDA Matrix loading process and image loading process.

E Speaker verification

E.1 BUT system

VAD type	parameters
baseline NN based	91254
fast NN based	25788
GMM based	9

Table 56: Evolution of the number of parameters in the VAD (Voice activity detection) system only.

VAD	baseline	NN-fast	Energy
EER BANCA G1 (%)	7.16	6.97	13.21
EER BANCA G2 (%)	5.27	5.92	9.00
EER NIST 2006 (%)	5.31	6.12	10.26
Memory (%)	100	98.64	96.35
Memory (MB)	48.48	47.82	46.71
Computational Time = RT	0.0522	0.0349	0.0220
Computational Time (%)	100	66.86	42.15

Table 57: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the VAD (Voice activity detection).

#Gaussians	2048	1024	512	256
EER BANCA G1 (%)	6.97	7.36	8.27	8.21
EER BANCA G2 (%)	5.92	4.34	4.40	4.83
EER NIST 2006 (%)	6.20	6.28	6.81	7.85
Memory (%)	100	60.3	40.2	30.1
Memory (MB)	45.19	27.25	18.16	13.62
Computational Time = RT	0.0332	0.0285	0.0262	0.0249
Computational Time (%)	100	85.84	78.9	75.0

Table 58: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the number of Gaussions in the model.

Feature kind	MFCC_0DAT_HLDA	MFCC_0DA	MFCC_0D
Dimensionality	52 reduced to 39	39	26
EER BANCA G1 (%)	7.62	8.27	9.76
EER BANCA G2 (%)	3.65	4.40	7.87
EER NIST 2006 (%)	6.52	6.81	8.88
Memory (%)	100	96	89
Memory (MB)	18.83	18.16	16.75
Computational Time = RT	0.0262	0.0262	0.0255
Computational Time (%)	100	100	97

Table 59: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the feature dimensionality.

#vectors	50	40	30	20	10
EER BANCA G1 (%)	6.97	6.97	6.97	7.46	7.68
EER BANCA G2 (%)	5.92	6.08	6.16	6.16	6.73
EER NIST 2006 (%)	6.12	6.44	6.58	7.22	7.91
Memory (%)	100	77.3	60.4	44.9	33.7
Memory (MB)	47.82	36.96	28.90	21.48	16.12
Computational Time = RT	0.0349	0.0344	0.0340	0.0334	0.0330
Computational Time (%)	100	98.6	97.3	95.6	94.4

Table 60: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the number of vectors in eigenchannel compensation matrix.

E.2 LIA system

#Gaussians	512	256	128	64	32
EER BANCA G1 (%)	3.48	2.19	3.86	4.23	5.15
EER BANCA G2 (%)	2.94	3.32	3.32	2.19	3.85
Memory (%)	100	57	36	25	20
Memory (MB)	7.84	4.29	2.70	1.90	1.50
Computational Time = RT	0.0052	0.0027	0.0013	0.0007	0.0004
Computational Time (%)	100	52	25	13	7

Table 61: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the number of Gaussians in the model.

AV Size		50	41	30	25	20
AV composition	c	19	15	10	15	10
	Δc	19	15	10	10	10
	Δe	1				
	$\Delta\Delta c$	11	11	10		
EER BANCA G1 (%)		3.48	4.77	3.48	5.52	5.15
EER BANCA G2 (%)		2.94	3.85	4.23	3.85	4.23
Memory (%)		100	88	73	67	60
Memory (MB)		7.84	6.60	5.48	4.99	4.46
Computational Time = RT		0.0052	0.0048	0.0045	0.0045	0.0043
Computational Time (%)		100	95	87	86	83

Table 62: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the acoustic vector size.

% of frame	100	50	10
EER BANCA G1 (%)	3.48	3.86	4.23
EER BANCA G2 (%)	2.94	3.48	4.60
Memory (%)	100	100	100
Memory (MB)	7.84	7.84	7.84
Computational Time = RT	0.0052	0.0030	0.0017
Computational Time (%)	100	58	33

Table 63: Performance (relative to baseline) in terms of computational and memory consumption for different scales of the number of selected frame for likelihood estimate.

F Multimodal Scalability

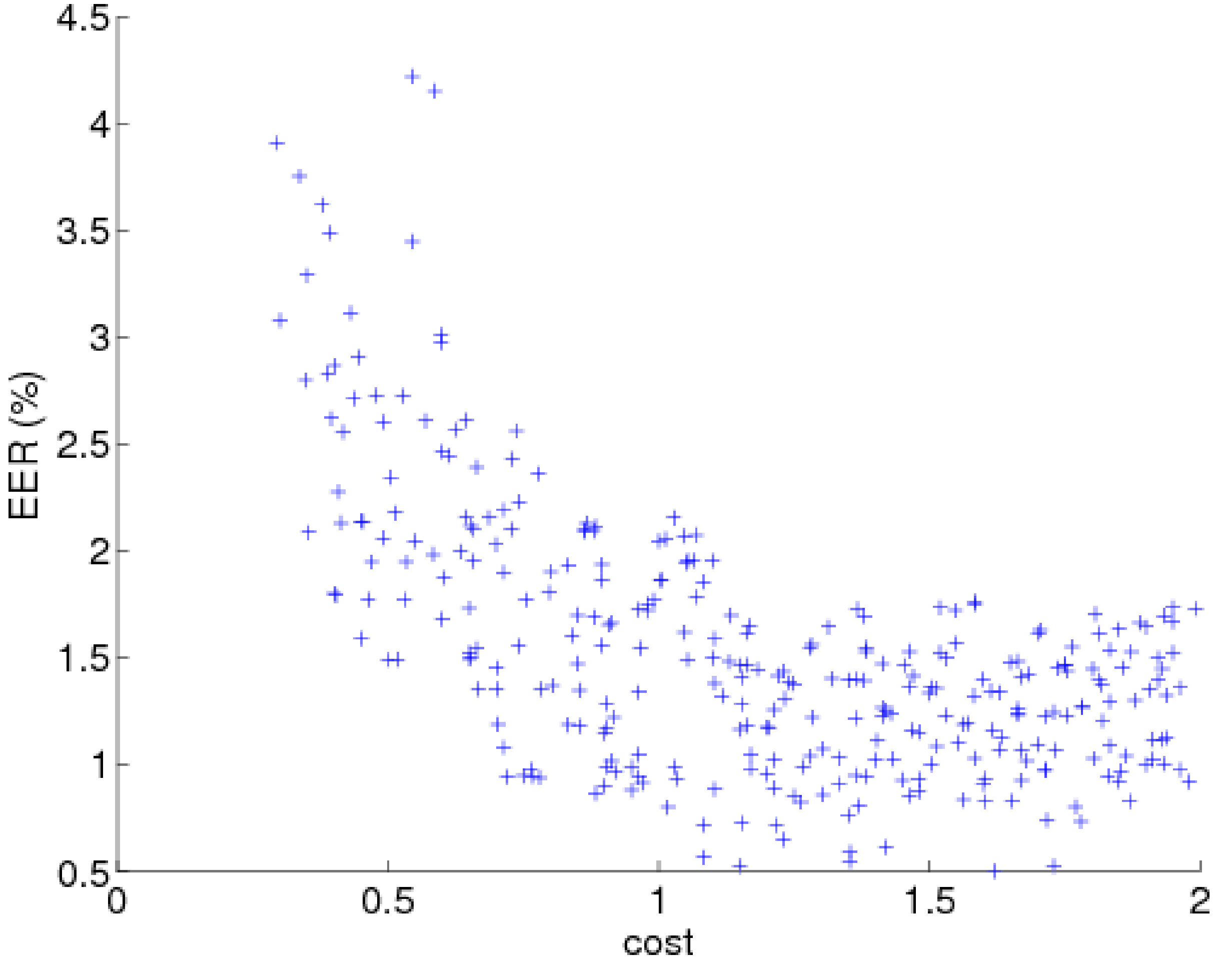


Figure 15: EER of bimodal fusion systems versus their combined cost. For a given cost band of 0.1 unit length, only the top 20 performing systems are shown.

No.	System	cost (unit)
1	IDIAP,F,NN,MLBP1-10	1.00
2	IDIAP,F,NN,MLBP1-5	0.62
3	IDIAP,F,NN,MLBP2-10	0.78
4	IDIAP,F,NN,MLBP2-5	0.57
5	IDIAP,F,NN,MLBP4-10	0.61
6	IDIAP,F,NN,MLBP4-5	0.53
7	IDIAP,F,TN,MLBP1-10	1.10
8	IDIAP,F,TN,MLBP1-5	0.65
9	IDIAP,F,TN,MLBP2-10	0.83
10	IDIAP,F,TN,MLBP2-5	0.58
11	IDIAP,F,TN,MLBP4-10	0.63
12	IDIAP,F,TN,MLBP4-5	0.53
13	OULU,F,NN,MLBP1-10	0.75
14	OULU,F,NN,MLBP1-5	0.38
15	OULU,F,NN,MLBP2-10	0.53
16	OULU,F,NN,MLBP2-5	0.32
17	OULU,F,NN,MLBP4-10	0.36
18	OULU,F,NN,MLBP4-5	0.28
19	OULU,F,TN,MLBP1-10	0.85
20	OULU,F,TN,MLBP1-5	0.40
21	OULU,F,TN,MLBP2-10	0.58
22	OULU,F,TN,MLBP2-5	0.33
23	OULU,F,TN,MLBP4-10	0.38
24	OULU,F,TN,MLBP4-5	0.28

Table 64: Cost assignments for the face verification systems *without facial alignment*

No.	System	cost (unit)
25	UMAN,F,base,NN,MLBP1-10	1.50
26	UMAN,F,base,NN,MLBP1-5	1.12
27	UMAN,F,base,NN,MLBP2-10	1.28
28	UMAN,F,base,NN,MLBP2-5	1.07
29	UMAN,F,base,NN,MLBP4-10	1.11
30	UMAN,F,base,NN,MLBP4-5	1.03
31	UMAN,F,base,TN,MLBP1-10	1.60
32	UMAN,F,base,TN,MLBP1-5	1.15
33	UMAN,F,base,TN,MLBP2-10	1.33
34	UMAN,F,base,TN,MLBP2-5	1.08
35	UMAN,F,base,TN,MLBP4-10	1.13
36	UMAN,F,base,TN,MLBP4-5	1.03
37	UMAN,F,comp,NN,MLBP1-10	1.25
38	UMAN,F,comp,NN,MLBP1-5	0.88
39	UMAN,F,comp,NN,MLBP2-10	1.03
40	UMAN,F,comp,NN,MLBP2-5	0.82
41	UMAN,F,comp,NN,MLBP4-10	0.86
42	UMAN,F,comp,NN,MLBP4-5	0.78
43	UMAN,F,comp,TN,MLBP1-10	1.35
44	UMAN,F,comp,TN,MLBP1-5	0.90
45	UMAN,F,comp,TN,MLBP2-10	1.08
46	UMAN,F,comp,TN,MLBP2-5	0.83
47	UMAN,F,comp,TN,MLBP4-10	0.88
48	UMAN,F,comp,TN,MLBP4-5	0.78

Table 65: Cost assignments for the face verification systems *with facial alignment*

No.	System	cost (unit)
49	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,0512G,u50PCA	0.80
50	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u10PCA	0.34
51	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u20PCA	0.60
52	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u30PCA	0.45
53	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u40PCA	0.77
54	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u50PCA	1.00
55	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u50PCA,VADen	0.42
56	BUT,S,MFCC12dither,STG301,0DAT-HLDA39,2048G,u50PCA,VADnn200	0.67
57	BUT,S,MFCC12dither,STG301,0DA,0256G,u50PCA	0.75
58	BUT,S,MFCC12dither,STG301,0DA,0512G	0.53
59	BUT,S,MFCC12dither,STG301,0DA,0512G,u50PCA	0.79
60	BUT,S,MFCC12dither,STG301,0DA,1024G,u50PCA	0.86
61	BUT,S,MFCC12dither,STG301,0DA,2048G,u50PCA	1.00
62	BUT,S,MFCC12dither,STG301,0D,0512G,u50PCA	0.53

Table 66: Cost assignments for the speaker verification developed by BUT

No.	System	cost (unit)
63	LIA,S,decime,nGMM-512,1-2	0.58
64	LIA,S,decime,nGMM-512,1-4	0.33
65	LIA,S,dscale,nGMM-128,30mfcc,1-4	0.12
66	LIA,S,dscale,nGMM-32,20mfcc,1-4	0.02
67	LIA,S,nGMM-128	0.25
68	LIA,S,nGMM-256	0.52
69	LIA,S,nGMM-32	0.07
70	LIA,S,nGMM-512	1.00
71	LIA,S,nGMM-64	0.13
72	LIA,S,vsize,nGMM-512,20mfcc	0.60
73	LIA,S,vsize,nGMM-512,25mfcc	0.67
74	LIA,S,vsize,nGMM-512,30mfcc	0.73
75	LIA,S,vsize,nGMM-512,41mfcc	0.88

Table 67: Cost assignments for the speaker verification developed by LIA (UPV)