

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Real-Time Video Face Recognition for Embedded Devices

Gabriel Costache, Sathish Mangapuram, Alexandru
Drimbarean, Petronel Bigioi and Peter Corcoran
*Tessera, Galway,
Ireland*

1. Introduction

This chapter will address the challenges of real-time video face recognition systems implemented in embedded devices. Topics to be covered include: the importance and challenges of video face recognition in real life scenarios, describing a general architecture of a generic video face recognition system and a working solution suitable for recognizing faces in real-time using low complexity devices. Each component of the system will be described together with the system's performance on a database of video samples that resembles real life conditions.

2. Video face recognition

Face recognition remains a very active topic in computer vision and receives attention from a large community of researchers in that discipline. Many reasons feed this interest; the main being the wide range of commercial, law enforcement and security applications that require authentication. The progress made in recent years on the methods and algorithms for data processing as well as the availability of new technologies makes it easier to study these algorithms and turn them into commercially viable product. Biometric based security systems are becoming more popular due to their non-invasive nature and their increasing reliability. Surveillance applications based on face recognition are gaining increasing attention after the United States' 9/11 events and with the ongoing security threats. The Face Recognition Vendor Test (FRVT) (Phillips et al., 2003) includes video face recognition testing starting with the 2002 series of tests.

Recently, face recognition technology was deployed in consumer applications such as organizing a collection of images using the faces present in the images (Picassa; Corcoran & Costache, 2005), prioritizing family members for best capturing conditions when taking pictures, or directly annotating the images as they are captured (Costache et al., 2006).

Video face recognition, compared with more traditional still face recognition, has the main advantage of using multiple instances of the same individual in sequential frames for recognition to occur. In still recognition case, the system has only one input image to make the decision if the person is or is not in the database. If the image is not suitable for recognition (due to face orientation, expression, quality or facial occlusions) the recognition result will most likely be incorrect. In the video image there are multiple frames which can

be analyzed in order to have greater recognition accuracy. Even if some frames are not suitable for recognition there is a high probability that some of them will work and the decision made will have a high degree of confidence. Once a face is recognized, it remains recognized in the scene by tracking techniques.

The disadvantage in the video imaging technique is in most cases the quality and size of the input frames are inferior compared to the still images.

2.1 General architecture of a VFR system

Most face recognition systems for still and video image technology follow the same classical workflow:

1. The faces have to be detected in the images.
2. The faces are normalized to the same size and usually same in-plane orientation.
 - a. Before or after (2), a pre-processing step tries to minimize the effect of illumination over the face.
3. Features are extracted from the facial region.
4. Test faces are compared with a database of people.

The first difference between the video and still image technology is that video scenarios can use a tracking algorithm together with a detection algorithm in order to keep track of all the faces in the video sequence. Using face tracking combined with face detection has three main advantages:

1. It allows the system to follow the faces across a wide range of variations in pose and lighting where tracking can be done easier than detection.
2. The time and memory requirements of a face tracking algorithm are lower than those of a face detection algorithm. Freed resources can be accessed once a face is detected in a frame. Tracking from that moment forward is a very important aspect when achieving real-time functionality.
3. Once a face in a particular frame is recognized with a high degree of confidence, that particular face does not need to be processed for the next frames. Only track the face and keep the association between the recognized person and the tracked face.

In the classification stage of the video imagery, a history of the recognized face offers greater accuracy than that of a still image.

Figure 1 shows a typical architecture of a video face recognition system.

Below are brief descriptions of each component together with the requirements that need to be satisfied in order to have a robust real-time face recognition system which can be integrated into an embedded device.

2.1.1 Face detection & tracking

The face detection and tracking component is very important in designing the recognition system. The properties of the detection algorithm (detection rate, robustness to variations, speed and memory requirements, etc.) will directly affect the properties of the overall recognition system. It is clear that undetected faces will not be recognized. Also considering the goal of real-time functionality on embedded devices where limited resources are available, spending most of that early on will reduce the application of the other blocks in the diagram in real time.

The main challenges associated with the detection and tracking algorithm are determined by the following factors:

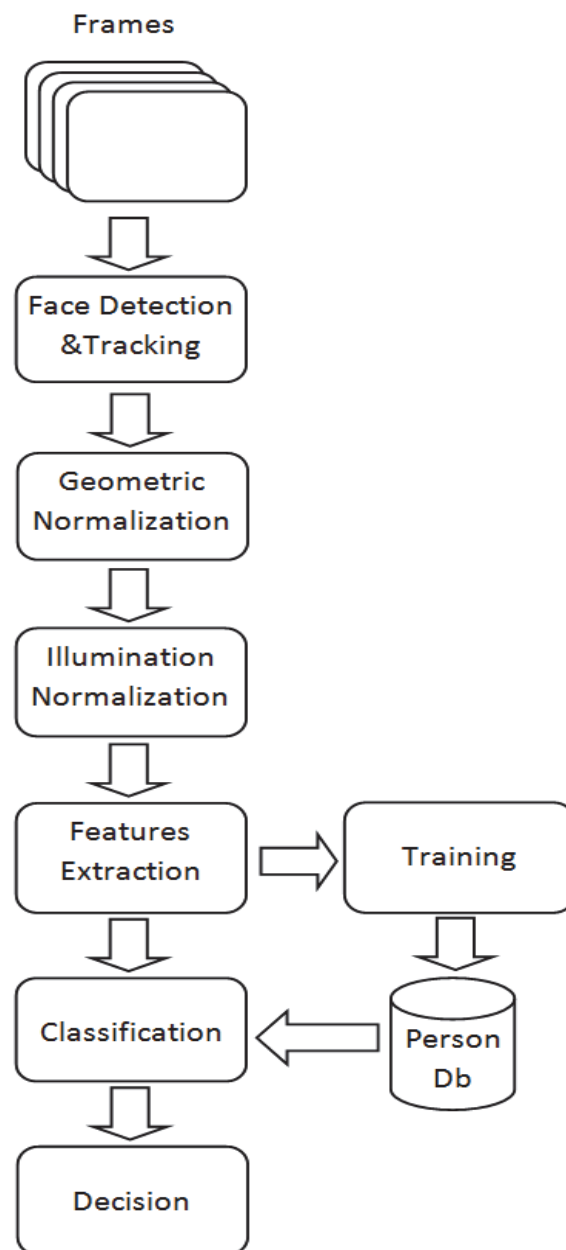


Fig. 1. Architecture of a VFR system

- Face orientation (pose). The appearance of the face may differ in many ways when the orientation of the face changes from frontal to profile or extreme view angles, where face components like the eyes, the nose or the ear may be occluded. It is difficult to detect a face at these extreme angles although face tracking is achievable.
- Changes in facial appearance. Examples include beards, moustaches or glasses. Women may use make-up which can significantly alter the face color and texture. These factors together with the potential for variability in shape, size or color of the face makes face detection challenging.
- Facial expression. The appearance of the face is directly affected by the person's facial expression. Tracking has to be robust to these variations as it is likely to be encountered in normal consumer videos.

- Occlusions. Different components from the face may be occluded in the image by other objects or faces. These have to be addressed by the tracking algorithm.
- Capture conditions. Factors that are involved in capturing the image such as lighting conditions, camera characteristics or quality of the captured image may have a big influence in the detection process.
- Face size or distance to subject. For video face detection and tracking consider the capture resolution and the distance from the capture equipment to the subject. For normal working resolution (qVGA, VGA) the faces can be very small even for relatively short distances.

The detection algorithm should have a high detection rate and robustness to variations such as changes in appearance, capture conditions and face size. The tracking algorithm should improve the robustness to face orientation, expressions and occlusions.

All the above requirements are difficult to fulfil especially for real-time scenarios in embedded devices. In the last few years there has been much progress in this area and now face detection and tracking is a common feature in most consumer cameras and mobile phones.

Tessera's OptiML™ Face Tools Face Tracking and Face Recognition technologies represent a perfect example of state-of-the-art technology in this area.

Some of the relevant parameters of Tessera's Face Tools technology that affect the performance of the overall recognition system include:

- Face tracking for up to 10 faces per frame, with less than 0.1 seconds lock time
- Minimum face size: 14x14 pixels
- Real-time face tracking up to 30 frames per second
- Faces detected in a wide range of orientations including rotation-in-plane and out-of-plane

Together with the detection rate, another metric used to describe the performance of a face detection algorithm is the false positive rate which represents the number of regions falsely reported as faces by the algorithm.

The false positive rate is not as important as the detection rate because the recognition algorithm should be able to differentiate between faces and non-faces when trying to classify the false positive candidates.

2.1.2 Geometric normalization

It is very important to detect and track the faces in all conditions and variations. When comparing local regions between faces, an image registration step must be performed so corresponding facial features are synchronised.

Simple geometric normalization usually involves bringing the faces to a standard size and rotating them in-plane in order to bring the eyes on the same horizontal line. Figure 2 shows some face samples before and after applying the geometric normalization.

More complex normalization scenarios (Corcoran et al., 2006b) can use 3D face models to rotate the face in the out-of-plane space to have identical orientation (i.e only frontal faces). This will have a higher computational requirement and could only be used when there is enough processing power. Figure 3 shows an example of the output of this complex normalization which can help recognition for large pose variations.

All other processing steps applied after geometric normalization should have the same affect on each face.



Fig. 2. Geometric normalization (Before (top), After (bottom))

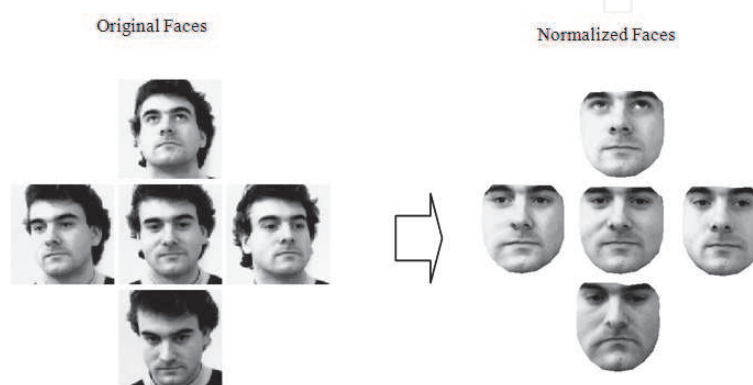


Fig. 3. Complex geometric normalization

2.1.3 Illumination normalization

If we can control the image capturing environment and impose strict requirements regarding lighting conditions (i.e. control access), recognition accuracy can be improved. In most scenarios where video face recognition is employed, the variations in lighting conditions when the faces are captured can range between dark and bright extremes. The profile face samples for each person to be recognized are captured in very different conditions with still images than those used in video imagery. A pre-processing algorithm should be used to minimize the effect of the lighting conditions when capturing the video images.

Depending on the resources available and capturing conditions, the illumination normalization algorithm can vary from simple algorithms such as: histogram equalization (HE), contrast limited adaptive histogram equalization (CLAHE) (Pizer et al., 1987; Corcoran et al., 2006a), logarithm transformed combined with suppressing DCT coefficients (LogDCT) or retinex (Land, 1986) based approaches, to more complex algorithms that can model the effects of lighting over facial regions (Lee et al., 2001; Smith & Hancock, 2005).

For embedded devices the simple normalization is a good compromise between execution speed and robustness to lighting variations.

Figure 4 displays the output of simple normalization techniques for two images affected by extreme side illumination.

An important issue to be considered when designing an illumination normalization algorithm is the balance between minimizing the effect of illumination and the inherent loss of information; information which is useful for classification. For instance, a face may appear dark because of dark lighting conditions or because the person has dark skin. Usually after normalization this information may not be recovered.



Fig. 4. Simple illumination normalization examples

The validation sets for most of the algorithms that try to minimize the effect of lighting (i.e. Yale database (Georghiades et al., 2001)) consists of faces captured in very different lighting conditions where the normalization algorithms have better results compared with using the original faces without illumination normalization. In real life conditions, the faces can be compared to similar lighting conditions where applying the illumination normalization should not have a negative impact over the recognition results.

It is very important to have a validation set that has a variation distribution close to those most likely to be encountered in the scenarios that the recognition system is designed for.

2.1.4 Feature extraction

Together with the useful information that can be used to differentiate between individuals, the face images described by the pixel values contain redundant information and information that can be ignored in the classification stage. By extracting only the useful information in this step we improve the accuracy of the recognition and also lower the storage requirement for each face.

Below are the main requirements for the feature selection algorithm:

- Good discriminative property. The features need to be able to differentiate between people. This translates into large variations between the value distributions for each person.
- Consistency. Features should not be modified between different images of the same person. This allows for recognition accuracy across large variations. Quantitatively this translates into small variation in the feature distribution for multiple faces of the same person.
- Small size. These features need to be stored and compared. Small size will allow fast comparison and low storage requirement.
- Fast computation. In order to achieve real-time recognition in video images, the faces need to be processed quickly.

The first two requirements will improve the accuracy of the recognition system and the last two requirements will ensure real-time recognition in embedded devices.

Classical approaches for still image recognition were also applied to the video image scenario with good results. These include: Principal Component Analysis (PCA) (Turk & Pentland, 1991), Linear Discriminate Analysis (LDA) (Belhumeur et al., 1996) and Discrete Cosine Transform (DCT) (Podilchuk & Zhang, 1996).

The DCT approach is of particular interest because of the speed of DCT transformation. Most of the capturing devices have DCT already implemented as part of JPEG compression module for storing captured images. More recent approaches like Local Binary Patterns (LBP) (Ojala et al., 2001) and Histogram of Oriented Gradients (HOG) (Lowe, 2004) have been used for face recognition.

2.1.5 Classification

In the case of still image recognition, the system makes a decision if the test face belongs to one of the people in the database and if so, which one (based on comparing the features computed in the previous step for test faces and a database of people).

In the case of video image recognition, the system compares the series of test faces with those in the sample database. Most commonly this is implemented as a series of still images derived comparisons and at each frame the confidence of our decision is modified based on the history of previous comparisons.

Simple classification algorithms like distance between feature vectors are preferred because of their simplicity and speed. More complex learning algorithms can be used if there are enough computation resources.

The classification algorithm is divided into two stages:

1. Training. Prototypes are constructed for each person in the database. The prototypes can be built from single or multiple face samples. Using multiple samples improves the quality of the prototype. The prototype can be represented by a series of feature vectors (such as distance-based classification) or can be represented by statistical models trained with multiple samples (such as learning-based classification algorithms).
2. Testing. Test samples are compared with each person prototype and similarity scores are computed. A decision is made using these similarity scores and the history of previous scores.

If, for a specific scenario, there is a fixed database that does not modify or update on the same platform where recognition is executed, a more complex algorithm for training can be used (i.e. training a learning algorithm) and performed offline. The result of this training algorithm is used during the recognition phase. When the database needs to be updated online at any time, the training algorithm needs to be less complex to be run on the embedded device. The result of the training algorithm, either the feature vectors or the person model, needs to be stored in the training database. This will influence the storage requirement of the recognition system.

An interesting algorithm that can be used for video face recognition is to model not only the appearance of the person at each frame but also the transition from frame to frame. A multi-dimensional Hidden Markov Model (HMM) (Nefian & Hayes, 1998) is used in order to model this type of transition. At the moment the complexity of HMMs makes it less favourable for embedded implementation.

2.2 Performance testing

Comparing two face recognition systems is a difficult task because there are many parameters that can describe the performance of a particular recognition system. Usually one system or the other is superior using different sets of parameters.

Depending on the specific application where the recognition system is deployed, some specific performance metrics are more important than others. For example, a security system based on face recognition will have, as a main priority, a very low false acceptance rate, whereas a photo sorting application implemented on an embedded platform based on face recognition, will have its priorities of high recognition rates and low complexity.

The performance of a recognition system can be described by two types of parameters: accuracy parameters that describe how accurate the system is in recognizing faces and technical parameters that represent characteristics such as how fast the system will process a face, etc.

Some of the accuracy parameters that can be used to describe a video face recognition system include:

- Recognition rate. This is the main measurement to describe the accuracy of a recognition system. It represents how many faces are correctly recognized from the total number of faces. For video recognition this is a little more complex as it can be computed as the total number of frames where the faces are recognized.
- False positive rate. For specific applications this parameter can be more important than the recognition rate. This is usually computed as the number of mistakes made by the system. It can be further classified as a false acceptance rate in verification applications where an unknown individual is classified as one person from the database and as a false rejection rate where a person from the database is classified as unknown.
- Receiver Operating Characteristic (ROC) curve. In most cases there is a trade-off between the recognition rate and a false positive rate. For a high recognition rate, tune the recognition system to increase the recognition rate. This will inevitably increase the false positive rate as well and the other way around. The ROC curve represents the recognition rate for each possible false positive rate and only by displaying the ROC curves can a comparison of the two recognition algorithms be made.
- Minimum face size to be detected and recognized. When working with normal video resolution the faces can be very small, even at short distances from the capture equipment. Imposing a high minimum size for the face in order to be recognized can lead to a high rate of faces that are ignored or not recognized in the video images.
- Range of pose variations to recognize a face. Depending on the application to apply the recognition, a higher or lower range of pose variations is needed to recognize the faces.

There are many technical parameters that can be used to describe a video face recognition system including:

- Processing time. This represents the time required to detect, process and classify all faces in a frame. This parameter depends on the platform where the recognition is implemented and will dictate if real-time functionality is available or not. For video frames, the time available for real-time recognition is the time between consecutive frames.
- Memory requirements. This represents the storage requirement for the system and includes the size of the feature vectors, person prototypes and other constants used in the algorithm.
- Number of faces recognized in each frame. The time required for detecting all faces in a frame is constant. This parameter will be influenced by the time required to process and recognize one face after it is detected.

The accuracy parameters depend on the database used for testing and the technical parameters depend on the specific platform where the recognition system is implemented.

Without using the same database and same platform, two recognition systems cannot be compared only by the performance parameters.

3. Proposed recognition system

The goal is to build a video face recognition system running in real-time on low computational power embedded platforms capable of recognizing multiple faces in video sequences. The main use case scenario intended for this system is tagging faces in consumer images as they are captured by digital cameras and mobile phones. The main requirements for the recognition system are high recognition rate, high robustness to variation types and low computational complexity for the algorithms used in the system. For this scenario, the input video stream can vary in size from small (qVGA) to high (full HD). Faces can also vary in size from tens of pixels in width to hundreds of pixels. Large variations in face pose, expression and illumination are also likely to be present.

Tessera's Face Tracking and Detecting technology is used in this experiment (Tessera, 2010). For geometric normalization, a computationally attractive approach is used, which involves: gray-scale transformation of the image, rotation of the face image to align the eyes on horizontal direction and resizing the face image to a small fixed size (i.e. 32x32 pixels). This size will allow for recognition of faces at a range of distances from the camera.

To minimize the effect of lighting variations, use a variant of the retinex (Land, 1986) illumination normalization algorithm. This is done by using a fixed variance matrix computed offline from a large database of images. This approach is very fast to apply and insures that the features computed in the next stage are more robust to large variations in illumination.

The features used for classification, in this chapter, are a variant of the Local Binary Pattern (LBPs) (Ojala et al., 2001) features which have been recently employed, with good results, for face recognition (Ahonen et al., 2006).

The classic approach of using LBP features in face recognition involves computing these features for each pixel in the face image, dividing the face image into small regions (separated or overlapped), and for each region computing the distribution of the LBP values. Often, only a small subset of all the features is used (uniform LBPs) in order to compute the region distribution. The classification involves comparing these distributions between corresponding regions from the test faces and the face samples used to build the prototypes in the training stage.

One approach is completely different. It is based on selecting from all possible features, those features that maximize the two properties defined in Section 2.1.4, namely: consistency and discriminancy. The training stage is split into two stages: 1) Off-line training, using a very large database of faces, in order to determine and select the most consistent features. 2) On-line training, using the face samples in the database that need to be recognized.

The weights for each selected feature are computed in the off-line stage, each weight representing how discriminating the respective features are for the people in the database. Look for the best features that are globally consistent for a very large database (off-line) of people and weight them according to how well they can discriminate for a given database (on-line). Both training stages are presented in the next sections.

For classification a similarity measure between two faces is computed by looking for identical corresponding features in the off-line training stage. For each identical feature value add the similarity between the energy of the features multiplied by the weights computed in the on-line training stage.

3.1 Feature extraction – LBP features

The Local Binary Patterns (LBP) (Ojala et al., 2001) features have been used in the system. These features are computed based on comparing the central pixel with its neighbours, concatenating the binary comparison results and computing the decimal number from the binary string.

Figure 5 shows how the feature is computed from an image.

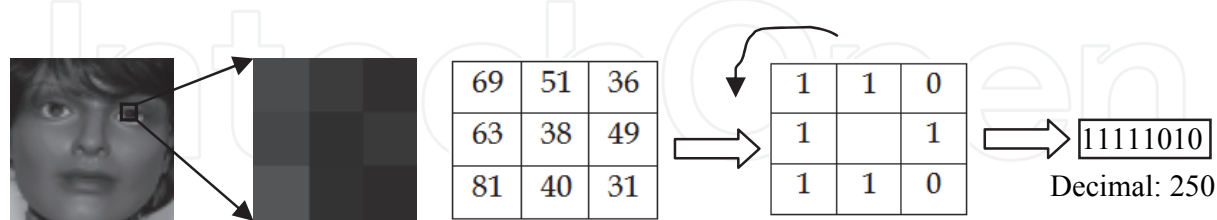


Fig. 5. LBP feature.

The LBP features extract local information from the face region and due to the binary comparisons are robust to changes in illumination.

3.1.1 Multi-scale LBPs and extended LBPs

In order to capture more information from the face region, features are computed at different face resolutions beginning with the standard size used for geometrical normalization, down to smaller scales by downsampling the face image with different factors (i.e. 2, 4 etc). The LBP feature are extracted at each resolution using their 8 closest neighbours together with their extended variants using 8 more distant neighbours. Figure 6 illustrates an example of the normal LBP feature together with its first order extended LBP feature.

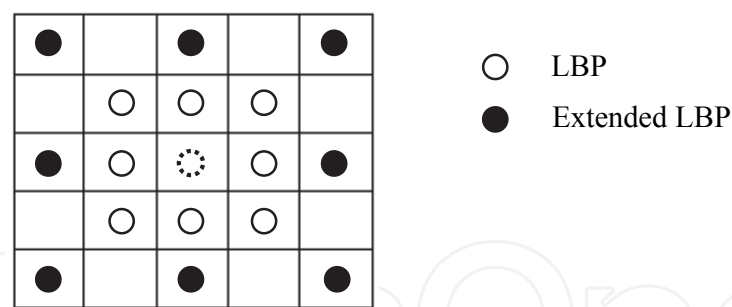


Fig. 6. LBP and Extended LBP

For each pixel in the normalized face image, compute multiple feature values. Do the same for the other scales.

3.1.2 LBP energy

The binary comparisons used for computing the features make them very robust to illumination variations. They also cause loss of information about the similarity of the local regions. For example, a very strong feature will have same value as a very faded feature. For this, calculate the normalized energy of the feature that will be used when comparing identical features for similarity between faces. The energy is computed using the formula:

$$e = \sum I_i^2 / \max(I_i^2)$$
(1)

where I_i represents the value of the neighbour pixel i used when computing the feature.

The energy is computed for both normal LBP and extended LBP.

Because small size face images are used, the features are not grouped in the face image by dividing the face into regions, but the corresponding features are compared for classification. The features are corresponding if they are computed at the same location, same scale and if they are normal or extended.

The feature vector after this analysis consists of the normal and extended LBPs computed at each location and at each scale together with their energies.

3.2 Off-line training – consistency analysis

Good features are those that do not change between images of the same person. This increases the accuracy of the recognition for different variations of the facial image.

This algorithm ranks a set of features given a large database of facial images. The order of the features is given considering their intra-class variation from low to high. The first features will be the most consistent between faces of same individuals. For recognition these features are more robust to variations.

Assume there is a collection of m people (P_1, P_2, \dots, P_m), each with multiple facial images. For each face compute all N possible features described in the previous section (normal and extended LBPs at all resolutions). Note the feature vector for person P_i image j as:

$$F_{ij} = (F_{ij1}, F_{ij2}, \dots, F_{ijN}) \quad (2)$$

The features in the system are the LBP features. This algorithm can be extended to any other type of feature or combinations between features. Below is the general form of the algorithm.

For each feature, define a measure of intra-class consistency $S = (S_1, S_2, \dots, S_N)$

The steps of the algorithm are:

- Reset all scores.

$$S = (0) \quad (3)$$

- Update scores. For every feature k , for every person i , for every m image of person i :
 - Compare the feature F_{imk} with the same feature k of the remaining j images of person i (F_{ijk} with $j=1:N_i$) where N_i is the number of images for person i .
 - If $|F_{imk} - F_{ijk}| < thr$ then increment S_k

$$S_k = S_k + 1 \quad (4)$$

At the end of this process order all features according to their score. Depending on the constraints, either keep a fixed number of features for classification in the latter stage or impose a threshold over the consistency measure.

The $|F_{imk} - F_{ijk}|$ term represents the distance between the feature values. In this case, search for identical features so $thr = 0$. For other types of features, a distance measure needs to be defined and a suitable thr needs to be chosen.

For best results, meaning best globally consistent features, the input database should be very large with all types of variation. Because it is executed off-line it does not affect the speed performance of the overall system.

3.3 On-line training – discriminative analysis

Together with consistency, the features also need to be able to discriminate between the people in the database. The same value for one feature across a database means both perfect consistency and no discriminative power. This algorithm assigns weights to each previously selected feature. The weights determine how well the feature can discriminate between the person in the database and which can be used in the classification stage.

Apply this algorithm to any type of feature using a suitable distance measure. The algorithm description is generic.

Assume m people in the database (P_1, P_2, \dots, P_m) , each with at least one representative facial image. For each face, N representative features selected by the off-line training procedure and their corresponding discriminative scores. For example, for person P_i we have:

- feature vector $F_i = (F_{i1}, F_{i2}, \dots, F_{iN})$
- score vector $S_i = (S_{i1}, S_{i2}, \dots, S_{iN})$

The steps of the algorithm are:

- Reset all scores.

$$S_i = (0), i = 0 \dots m \quad (5)$$

- Update scores. For every feature k , for every person i ,
 - Compare the feature F_{ik} with the same feature k of the remaining people (F_{jk} with $j=1:N$)
 - If $|F_{ik} - F_{jk}| > \text{thr}$ then increment S_{ki}

$$S_{ki} = S_{ki} + 1 \text{ or } S_{ki} = S_{ki} + |F_{ik} - F_{jk}| \quad (6)$$

In order to make these scores independent of the number of people and faces in the database, normalize them using the maximum sum of scores for a person using the next equation where N represents the number of features and m the number of people in the database:

$$S_{ki} = S_{ki} / \max(\text{sum}(S_{ln})_{n=1:N})_{l=1:M} \quad (7)$$

The same observations from the previous section for the terms: $|F_{ik} - F_{jk}|$ and thr are valid. In the case of searching for identical LBPs, the parameter $\text{thr} = 0$.

3.4 Classification

Having computed the features described in Section 3.1 and the discriminative scores of all features selected in Section 3.2, using the algorithm described in the previous section, compute a similarity (S_{ij}) between two faces (trained face f_i and test face f_j) by counting how many identical corresponding features there are between the two faces using this formula:

$$S_{ij} = \sum w_k * |e_{ik} - e_{jk}| * g_{ij}, k=1:N \quad (8)$$

where:

- g_{ijk} is equal to 1 if feature k is identical between faces i and j , and 0 otherwise.
- e_{ik}, e_{jk} represent the energy of the feature k from image i and j respectively computed using eq. (1).
- w_k represents the discriminative score for features k of trained face i .

Comparing the test face with all face samples from the trained database, return the most similar person with the test face. By imposing a decision threshold, control the recognition

rate versus the false positive rate, depending on the application mentioned in Section 2.2. Once the similarity measure between the test face and the most similar person from the database is higher than the decision threshold, decide that the face is recognized and continue or not the recognition process over the next frames.

4. Results

In order to assess the performance of the recognition system a large database of videos with systematic variations was used, including: pose, illumination, face size/distance to subject, and facial expressions.

In the training stage, a single image was used to train each person. The training face is frontal, good size, normal illumination and good quality. Tests were run for different numbers of people in the database from low (3) to high (100).

For each test, the recognition rate (RR) was measured, as the number of correct classifications, false positive rate (FP2) as wrong classifications and undecided rate (MD) as number of test faces which were not classified.

Figure 7 shows the recognition and error rates as a function of head yaw angle. As specified above, training was conducted at head yaw angles of zero degrees and testing was done with 0°, 10°, 20° and 30° yaw angles.

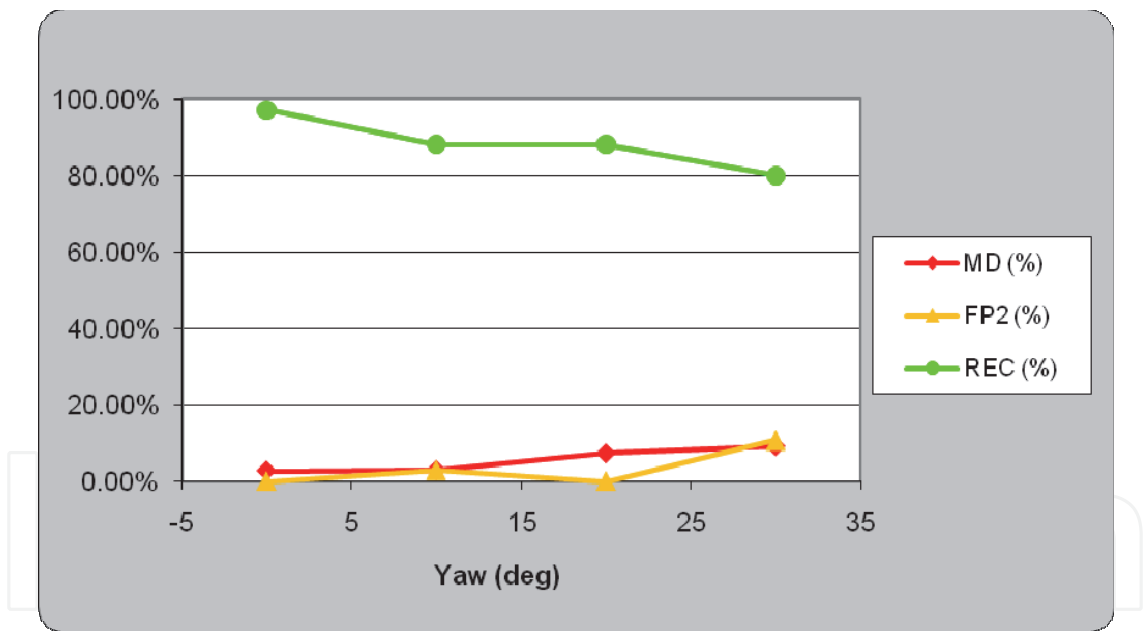


Fig. 7. Recognition performance for different yaw angles

Figure 8 shows the recognition and error rates as a function of head pitch angle. Training was conducted at head pitch angles of zero degrees.

Figure 9 shows the recognition and error rates as a function of different facial expressions. The training faces had no facial expression.

Figure 10 shows the recognition and error rates as a function of different illumination conditions. The approximate EV values for the given conditions are: LowLight (2.4EV) and StrongLight (9EV), which can be considered extreme lighting conditions. Training was conducted using normal indoors ambient lighting.

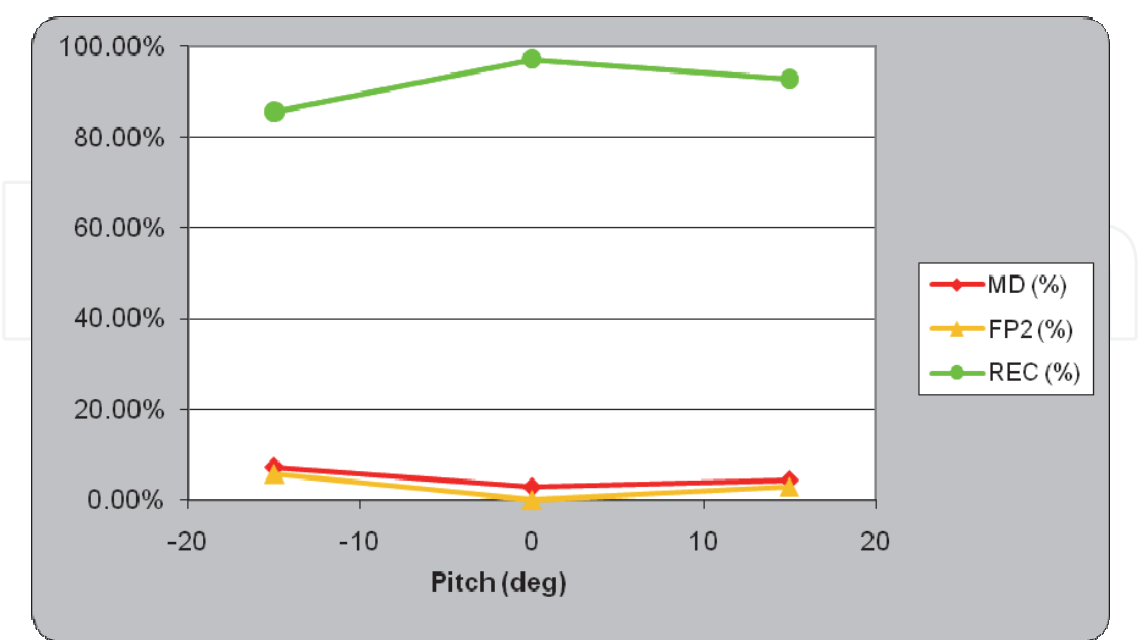


Fig. 8. Recognition performance for different pitch angles

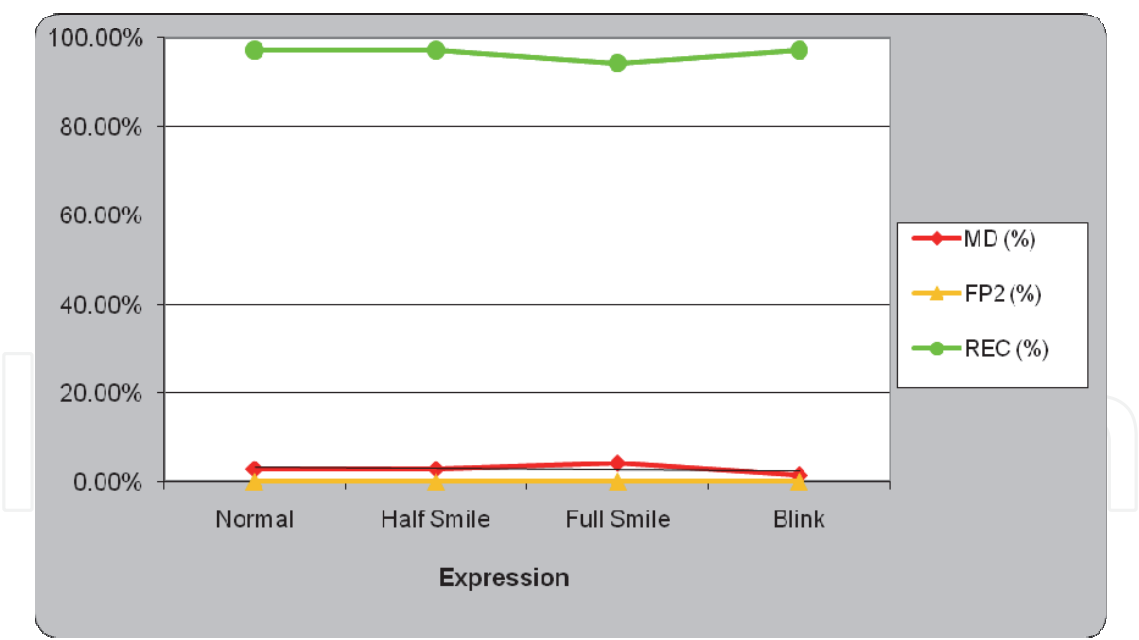


Fig. 9. Recognition performance for different face expressions

The main technical parameters for the system were implemented on an ARM9 platform (266 MHz CPU), the processing time for a face depending on the size of the input frame varied between 8 and 15 milliseconds for qVGA and VGA input frame size which is well within real-time requirements. The size of the features vector for each analyzed face is about 2Kb which is very small.

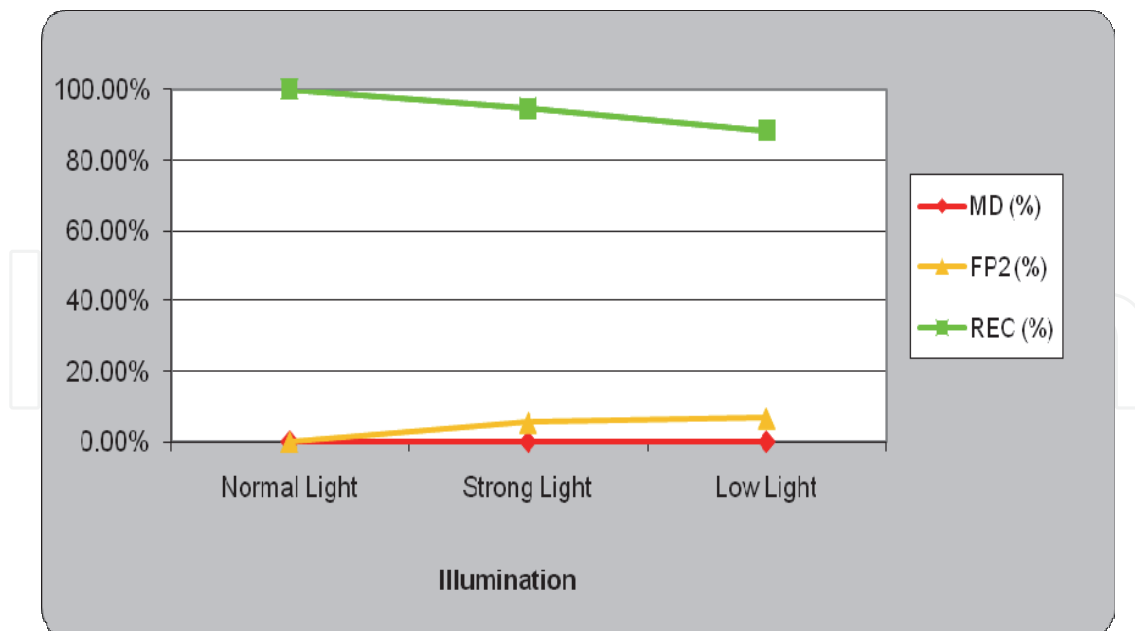


Fig. 10. Recognition performance for illumination conditions

5. Conclusion

This chapter presented the challenges of implementing a real-time video face recognition system on an embedded platform. The first section presented the main issues that need to be addressed when designing such a system and possible solutions. The second part described a working solution based on using LBP features which are fast to compute, robust to variations and able to extract useful information from the face region. In order to obtain a robust recognition system, only the features which have the same value across multiple variations of the same person were extracted. In order to increase the accuracy of the system, weights were associated to the selected features based on their discriminative power between the people from the database.

Results for this system were tested and implemented on an embedded platform, which shows good accuracy across large variations of the input data and technical parameters which satisfy the condition for real-time processing.

6. References

- Ahonen, T.; Hadid, A. & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 28, pp. 2037-2041, December 2006.
- Belhumeur, P.N.; Hespanha, J.P. & Kriegman, D.J. (1996). Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection, *Proc. of the 4th European Conference on Computer Vision, ECCV'96*, 15-18 April 1996, Cambridge, UK, pp. 45-58
- Corcoran, P. & Costache, G. (2005). Automated sorting of consumer image collections using face and peripheral region image classifiers, *Consumer Electronics, IEEE Transactions on* Vol. 51, Issue 3, Aug. 2005, pp. 747 - 754.

- Corcoran, P.; Iancu, C. & Costache, G. (2006a). Improved hmm based face recognition system. *International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, May 2006.
- Corcoran, P.; Ionita, M. & Costache, G. (2006b) Pose-invariant face recognition using AAMs, *International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, May 2006.
- Costache, G.; Mulryan, R.; Steinberg, E. & Corcoran, P. (2006). In-camera person-indexing of digital images, *Consumer Electronics ICCE '06 Digest of Technical Papers. International Conference on*, 7-11 Jan. 2006.
- Georghiades, A.S.; Belhumeur, P.N. & Kriegman, D.J., (2001). From few to many: illumination cone models for face recognition under variable lighting and pose, *IEEE Trans. Pattern Anal. Mach. Intelligence*, Vol. 23, No. 6, pp 643-660.
- Google Picassa (n.d.), <http://picasa.google.com/>
- Land, E(1986). An alternative technique for the computation of the designator in the retinex theory of color vision. *Proc. Nat. Acad. Sci.*, vol. 83, 3078-3080
- Lee, K.; Ho, J.; & Kriegman, D. (2001). Nine points of light: Acquiring subspaces for face recognition under variable lighting. *In Proc. of CVPR*, pp 519-526.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. 226 *International Journal of Computer Vision* 60 (2), 91.
- Nefian, A.V. & Hayes III, M.H. (1998). Hidden Markov Models for Face Recognition, *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, Vol. 5, 12-15 May 1998, Seattle, Washington, USA, pp. 2721-2724
- Ojala, T.; Pietikäinen, M. & Mäenpää, T. (2001). A generalized Local Binary Pattern operator for multiresolution gray scale and rotation invariant texture classification, *Advances in Pattern Recognition, ICAPR 2001 Proceedings*, Springer, 397 – 406
- Phillips P. J.; Grother, P.; Micheals, R.; Blackburn, D.M.; Tabassi, E. & Bone, M. (2003). Face Recognition Vendor Test 2002, *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, p.44, October 17, 2003
- Pizer, S.; Amburn, E.; Austin, J.; Cromartie, R.; Geselowitz, A.; Greer, T.; Romeny, B.; Zimmerman, J. & Zuiderveld. K. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, Vol. 39 pp. 355-368.
- Podilchuk, C. & Zhang, X. (1996). Face recognition using DCT-based feature vectors, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, vol. 4, pp. 2144-2147, May 1996.
- Smith, W.A.P.; & Hancock, E.R. (2005). Single image estimation of facial albedo maps. *BVAI*, pp. 517-526.
- Tessera OptiML FaceTools (2010).
http://tessera.com/technologies/imagingandoptics/Documents/OptiML_faceTools.pdf
- Turk, M.A. & Pentland, A.P. (1991). Face Recognition using Eigenfaces, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586-591, June 1991.
- OptiML is a trademark of Tessera Inc. In the United States and other countries.



New Approaches to Characterization and Recognition of Faces

Edited by Dr. Peter Corcoran

ISBN 978-953-307-515-0

Hard cover, 252 pages

Publisher InTech

Published online 01, August, 2011

Published in print edition August, 2011

As a baby, one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. We accept face recognition later as an everyday ability. We realize the complexity of the underlying problem only when we attempt to duplicate this skill in a computer vision system. This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section presents an architecture for face recognition based on Hidden Markov Models; it is followed by an article on coding methods. The next section is devoted to 3D methods of face recognition and is followed by a section covering various aspects and techniques in video. Next short section is devoted to the characterization and detection of features in faces. Finally, you can find an article on the human perception of faces and how different neurological or psychological disorders can affect this.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Gabriel Costache, Sathish Mangapuram, Alexandru Drimborean, Petronel Bigioi and Peter Corcoran (2011). Real-Time Video Face Recognition for Embedded Devices, New Approaches to Characterization and Recognition of Faces, Dr. Peter Corcoran (Ed.), ISBN: 978-953-307-515-0, InTech, Available from: <http://www.intechopen.com/books/new-approaches-to-characterization-and-recognition-of-faces/real-time-video-face-recognition-for-embedded-devices>

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen