Weighted multi-method user identification in gaming applications

Jurgen Baert, Ludovic Espeel, Steven Puttemans, Jan Staelens
Katholieke Hogeschool Oostende-Brugge; corresponding author: jurgen.baert@khbo.be

Abstract – **In this paper, considerations and methodology for user identification in gaming applications is discussed. Typical user identification processes operate through input devices that yield qualitative audio and video data in a controlled environment. In gaming applications, low cost hardware and uncontrolled environmental conditions pose a serious obstacle towards efficient user identification. Through a combination of several methods, including speaker recognition, facial feature extraction and the eigenface recognition approach for face recognition, a robust algorithm can be developed. Weighing the relative importance of each method leads to more robust recognition, despite the limitations associated with the application.**

Keywords – **speaker recognition, eigenfaces, feature extraction, eigenfeatures, user identification, gaming**

I. INTRODUCTION

Computer-aided recognition of people can be seen as a collection of several, often well-researched domains of digital signal processing. Based on input from camera, scanner, microphone and other sources, the computer system attempts to identify the user through their fingerprints [1], facial characteristics [2] and other biometrics [3], and voice [4]. Many of these methods require high quality input data and/or a controlled environment [5] in order to guarantee good results.

Modern game consoles are often equipped with low-cost audio and video input devices. Yet these devices can benefit from automatic sign-in procedures through recognition of the user. With security being relatively unimportant in gaming applications, the consequences of false positives are quite low, while the low-cost input devices make recognition harder.

A. Video input devices and techniques

The camera modules available on game consoles provide a color image with a relatively low resolution, especially considering that faces of people will not take up a large part of the image surface. Indeed, as the camera module is typically set up near the television set, the user will typically be between 2 and 5 meters away from the camera. This means that typical faces will only take up a few hundred pixels in total.

Super-resolution techniques attempt to create a high resolution image to work with, starting with the low resolution input. Most often, this is achieved by only looking at a sequence of low-resolution images [6], but it is also possible to use prior knowledge of what the result should look like (in this case: a face) by using a training set [7]. This is, for the application of face processing, called face hallucination. The details in the high-resolution version of these hallucinated faces are not always accurate, as the high-frequency part is partially fabricated!

The environmental circumstances under which the video input is collected is another important factor in determining the overall quality of the recognition system. High definition systems often have relatively controlled environments with lighting conditions that remain similar over time; they also often have a static background. In gaming applications, lighting conditions as well as background are hard to control. This has a significant influence on the performance of the recognition algorithms, as illumination has been proven [8] to have a higher significance on recognition methods than actual facial features.

Techniques to combat illumination variations have been proposed and tested, often requiring information about the light source and/or a set of training images. These techniques are applied as a form of preprocessing on the image, such as in [9].

Several popular techniques such as fingerprint and retina scans are highly impractical for gaming applications, given the presence of a single low quality video source at a large distance from the target. This limits the range of possibilities to robust face recognition, body structure parameterization and motion-based recognition [10].

B. Audio input devices and techniques

Speaker recognition requires an audio input source. It is much easier to condition audio signals than video signals. In a first approximation, the only important environmental factor affecting the input signal quality is noise.

Microphone systems connected to game consoles are typically of sufficient quality, but it is hard to control background noise in a home environment.

In a system combining video-based and audio-based recognition, locating people through audio can help identifying the speaker visually as well. This requires multiple microphones to be set up in an array structure. The time delay and amplitude variations between the input from the different sources can help determine with relative accuracy [11] where the speaker is located relative to the microphone array. Using multiple microphones also improves signal quality [12]. The audio signal can finally be analyzed, extracting parameters characterizing the speaker's voice.

II. APPROACH

In order to create a robust recognition system under uncontrolled conditions and with low quality input data, a combination of algorithms is suggested. By using multiple methods and combining the results using weighing algorithms, it is expected that the inaccuracy of the separate methods can be partially offset.

As a first phase in the research towards this goal, individual algorithms are implemented and their performance is optimized for the application. This is applied for three algorithms in Section III.

In a second phase, the results from these algorithms are combined using several methods. Section IV details the possible methods.

In the third and final stage, discussed in Section V, research towards optimizing the algorithms for combined use should further increase efficiency and accuracy of the resulting system.

Section VI outlines future work on this topic, while Section VII presents the conclusion of the presented work.

III. EXISTING ALGORITHMS

A. Eigenface-based face recognition

Principal component analysis (PCA) [13] is used to construct a set of eigenvectors from a library of faces; these are called eigenfaces. Faces are characterized by projecting an image on the basis of eigenfaces and storing the coordinates (or "weight pattern") in a database. Recognition is then the process of mapping a newly taken image (containing the face that is to be recognized) onto the set of eigenfaces and comparing the coordinates of this image with the coordinates of people in the database.

A brief discussion of the algorithm follows. A training set of images consists of M faces, each of a given resolution N_x x N_y. These images, named F_1, F_2, …, F_M are offset with the average image A, so that $G_n = F_n - A$ represent the difference between each face and the average. The set of eigenfaces contains M orthonormal vectors $\boldsymbol{u_n}$, calculated as the eigenvectors of the covariance matrix C:

$$C = \frac{1}{M}\sum_{n=1}^{M} G_n G_n^T \qquad (1)$$

Calculating these eigenvectors can get very unwieldy due to the sheer size of the C matrix. Luckily, the calculations can be reduced significantly by calculating the eigenvectors and eigenvalues of an M x M matrix. Details are left out and the reader is referred to [14].

Once the eigenfaces have been calculated, a new face image F can be projected onto a subset of size M' of the face space, again after correcting the unknown face with the average A.

$$f_k = \iota_k^T (F - 4) \qquad (2)$$

The resulting M' values form a weight vector or a set of coordinates, which can be compared to those coordinates that have been stored for known individuals. Note that $M' \leq M$, where the value of M' depends on the eigenvalues of the different eigenfaces.

B. Haar-like feature extraction for facial recognition

Haar-like features [14, 15] represent areas in an image that show differences in intensity between two or more rectangular areas. Edge features, line features and center-surround features are used to classify edges, lines and centroids respectively.

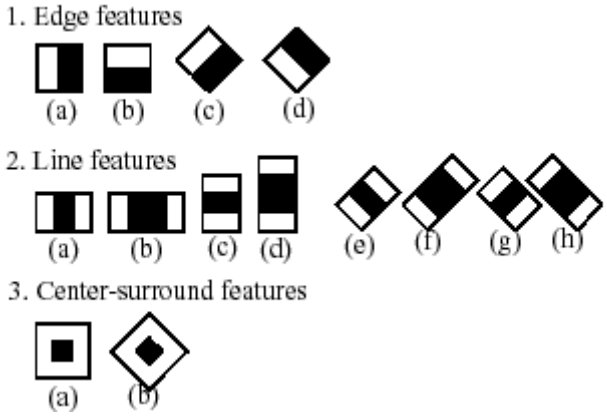

FIGURE I: HAAR-LIKE AND CENTER-SURROUND FEATURE PROTOTYPES
Source: [17]

By cascading and combining several Haar-like features, Haar-classifiers can be created. These classifiers are used to detect faces, but also eyes, nose, ears, mouth and other facial features and are constructed as a decision tree where each stage of the cascade accepts and rejects a certain percentage of input images. Constructing such classifiers is hard and extremely computationally intensive, but Haar-classifiers can reach very good results in detecting faces and facial features.

After finding facial features, distance measurements can be performed and the results can be stored in a database for known individuals, for later comparison. Like eigenfaces, Haar-like feature systems have been extensively researched.

As recognition paradigms, feature detection and distance measurements based thereon are quite limited in accuracy, especially when used on low resolution input images. Section V will show how Haar-cascades can be combined with eigenface methodology to create better results however.

C. Speaker recognition using mel frequency cepstral coefficients (MFCC)

Speech is usually analyzed over short time windows, typically about 20-30ms (with overlap). During such a window, the speech signal is indeed quasi-stationary. The difference between speakers can be studied most easily in the frequency domain.

Peaks in the frequency domain denote dominant frequency components in the speech signal; these peaks are called formants and are the prime identifiers for a sound signal. In order to extract these formants, a smooth curve known as the spectral envelope is created from the frequency spectrum. Speaker identification is then performed by comparing the spectral envelope and formants from the unidentified speech signal with those stored in the database of known speakers.

Calculating the spectral envelope can be done by calculating the cepstrum of the signal. This is the FFT (or IFFT) of the log magnitude spectrum of that signal and, when a low-pass filter is applied to it, yields the spectral envelope after reverse transformation [16].

However, human hearing is not linearly sensitive to frequencies. Instead, it acts as a filter bank concentrating

on certain frequency components, where these components are not uniformly spaced on the frequency axis. More filters (with a smaller bandwidth each) are present in the low frequency regions, and less filters (with broader bandwidth) in the high frequency regions. To compensate for this, mel-frequency transformation is introduced, leaving a mel-spectrum with linear perceptual importance. The above-mentioned ceptral analysis is then applied to this transformed spectrum rather than on the regular, linear spectrum. The resulting cepstral coefficients are referred to as MFCC, or mel-frequency cepstral coefficients.

As a final step in the parameterization process, the spectral envelope and formants (now represented by MFCCs) need to be compressed and stored. Vector quantization (VQ) is used to accomplish this [17]. Vector quantization techniques attempt to form encoding regions, characterized by centroids for the vectors resulting from the cepstral analysis. This way, the MFCCs of known individuals can be transformed to codebook-storable data.

IV. Multi-Method Identification

In Section III, several existing algorithms have been discussed. It is obvious that other techniques and variants exist, which have not been discussed. A challenge lies in combining results from the different algorithms in order to accomplish a more robust and accurate identification system.

In general, most techniques will be codebook-based. This means that individuals are characterized by a set of parameters, which are stored in a database. Each algorithm will have a number of associated parameters, and each known individual will be linked to a set of parameters for every identification method. For eigenface recognition, the coordinates in the face space are stored; for speaker identification, the MFCCs are stored through VQ methodology, etc.

When combining N recognition methods, where every method has M_k ($k = 1...N$) associated codebook parameters, several metrics can be applied to determine the most likely result. This section discusses two families: rank-based and norm-based methods. A verification method for the algorithm's fidelity is also given.

A. Ranking-based methods

Each algorithm will yield (beside other data) an output matrix containing either the percentages of similarity, or the distances between the unknown input data and the known entities in the codebook. As such, the results of each algorithm can be represented as a sorted list, from the most likely to the least likely match.

When using only this ranking, final scores for each result need to be assigned using a point-based method, where, for example, a first place is awarded 3 points, a second place 2 points and a third place 1 point. Alternative scoring schemes are possible.

By also assigning weights to each of the employed identification methods, relative to the method's reliability, a final classification can be made. The assignment of weight values to the different algorithms can either be done through training during the design phase, through training during the deployment phase, or adaptive based on continuous user feedback.

B. Norm-based methods

As an alternative to ranking-based methods, which do not take the distance between the input data and the codebook data into account, norm-based methods will calculate one of several possible norms (Euclidean norm, maximum norm, p-norm, …) based on those distances. Data from the different recognition methods will need to be normalized before the norm can be calculated.

By moving away from classic norm calculations towards other norm-like functions, it is also possible to give a different weight to the different methods.

Each method uses several codebook parameters, but those parameters do not necessarily have an equal weight in determining the end result for a given method (compare to DCT where less vital information is stored in the higher frequencies). Instead of calculating distances for each algorithm separately and combining those results, one generalized distance function can be distilled, which takes the importance of each codebook parameter, for each algorithm into account.

If a weighted norm-like function is used, the weights can again be determined either during design, training or adaptively in the field.

C. Algorithm fidelity

Since the employed methods are ideally independent (or largely so, at least), there is no guarantee that each method will individually yield the same result. Because the results of each method are available however, it is possible to correlate them to determine the reliability of the final solution. If many methods present the same or similar results, it is likely that the final result is correct; but if the correlation between the results is low, the final result is more than likely merely the result of a mathematical manipulation rather than a reliable indication.

Additionally, correlation between the different methods can be used as a means to (heuristically) determine the weights mentioned earlier. This can again be done during either the design phase, the training phase or after deployment of the system.

V. Inter-Algorithmic Optimization

When manipulating input data, each algorithm uses only data generated by that algorithm. In a closed system with several of these algorithms cooperating however, this need not be the case. Inter-algorithmic optimization is possible and will lead to increased efficiency and/or accuracy.

By using the Haar-cascades, not only faces can be found, but features such as the nose, mouth, chin or eyes can be identified within each face. By extracting parts of the image and applying eigenface techniques to these subimages, a better mapping can be achieved. Indeed, since there is a better separation of facial features, they will be compared to other features of the same type, rather than areas in the same spot on different images.

Another example of inter-algorithmic optimization can be found by combining audio and video data for audio source localization. Two methods immediately present themselves: the video stream can be analyzed to determine which of the persons in the picture are speaking (through gestures or mouth movement); *or* in case a microphone array is present, the different audio signals can be correlated to determine the location of the speaker within the picture.

VI. Future Work

The work presented in this paper is part of a lab project in the KHBO Master of Science in Engineering program for students of electronics and ICT in their final year of studies and has in part been carried out by three of these students.

The algorithms presented in Section III have been implemented and tested, but further optimization for the low-resolution constraint needs to be done. Super-resolution and hallucination techniques can be added and parameter optimization needs to be researched.

Much of what Sections IV and V describe has not been implemented and is presented as a theoretical study for now.

VII. Conclusion

A framework for combining multiple user identification algorithms has been presented. These algorithms function under the constraint of a typical gaming application, with low-cost input devices. Further research is required to determine the viability of the different suggested processes.

J. Baert is with the ECOREA research lab, Department of Industrial Sciences and Technology, Catholic University College Brugge–Oostende (KHBO) – Zeedijk 101, 8400 Oostende, Belgium, e-mail: jurgen.baert@khbo.be

L. Espeel, S Puttemans and J. Staelens have received the M.Sc. in Engineering degree at the Catholic University College Brugge–Oostende (KHBO) in 2011.

References

[1] N.Yager, A.Amin. *Fingerprint classification: a review*, Pattern Analysis & Applications, vol. 7, no. 1, pp. 77-93, 2004.

[2] W.Zhao, R.Chellappa, P.J.Phillips, A.Rosenfeld. *Face recognition: a literature survey*, ACM Computing Surveys, vol. 35, no. 4, pp. 399-458, 2003.

[3] A.Jain, L.Hong, S.Pankanti. *Biometric identification*, Communications of the ACM, vol. 43, no. 2, pp. 90-104, 2000.

[4] D.Reynolds. *An overview of automatic speaker recognition technology*, IEEE Acoustics, Speech and Signal Processing (ICASSP), Proc. Book 3, pp. 4072-4075, 2002.

[5] P.J.Phillips P.Flynn, T.Scruggs et al. *Overview of the face recognition grand challenge*, IEEE Computer Vision and Pattern Recognition (CVPR), Proc., vol. 1, pp. 947-954, 2005.

[6] S.Park, M.Park, M.Kang. *Super-resolution image reconstruction: a technical overview*, IEEE Signal Processing, vol. 20, no. 3, pp 21-36, 2003.

[7] B.Gunturk, A.Batur, Y.Altunbasak, M.Hayes, R.Mersereau. *Eigenface-Domain Super-Resolution for Face Recognition*, IEEE Transactions on Image Processing, vol. 12, no. 5, pp. 597-606, 2003.

[8] W.Zhao. *Robust face recognition using symmetric shape-from-shading*, 1999.

[9] T.Chen, W.Yin, X.S.Zhou, D.Comaniciu, T.S.Huang. *Total variation models for variable lighting face recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, col.

[10] L.Wang, H.Ning, T.Tan, W.Hu. *Fusion of static and dynamic body biometrics for gait recognition*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 2, pp. 149-158, 2004.

[11] M.Omologo, P.Svaizer. *Acoustic event localization using a crosspower-spectrum phase based technique*, IEEE Acoustics (ICASSP), Proc, Book 2, pp. 273-276, 1994.

[12] N.Cheng, W.Liu, P.Li, B.Xu. *An effective microphone array post-filter in arbitrary environments*, Interspeech, Proc, pp. 439-442, 2008.

[13] M.Turk, A.Pentland. *Eigenfaces for recognition*, J. Cognitive Neuroscience, pp. 71-86, 1991.

[14] R.Lienhart, J.Maydt. *An extended set of Haar-like features for rapid object detection*, IEEE Image Processing (ICIP), Proc, pp. 900-903, 2002.

[15] T.Mita, T.Kaneko, O.Hori. *Joint Haar-like features for face detection*, IEEE Computer Vision (ICCV), vol. 2, pp. 1619-1626, 2005.

[16] L.Muda, M.Begam, I.Elamvazuthi. *Voice recognition algorithms using Mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques*, J. Computing, vol. 2, no. 3, pp 138-143, 2010.

[17] H.B.Kekre, V.Kulkarni. *Speaker identification by using Vector Quantization*, J. Engineering Science and Technology, vol. 2, no. 5, pp. 1325-1331, 2005.