Study from implementing algorithms for face detection and recognition for embedded systems

I. E. Ceuca[1], A. Tulbure[1], O. Domsa[1], C. Covaciu[2]
[1]University "1 DECEMBRIE 1918" Alba Iulia, eceuca@uab.ro, aditulbure@uab.ro,ovidiu@uab.ro
[2]IPEC S.A. Alba Iulia, cosmin@ipec.ro

***Abstract*-These Face detection and recognition research has attracted great attention in recent years. Automatic face detection has great potential in a large array of application areas, including banking and security system access control, video surveillance, and multimedia information retrieval. In this paper, we discuss the opportunity and methodology for implementation of an embedded face detection system like web-camera or digital camera. Face detection is a widely studied topic in computer vision, and advances in algorithms, low cost processing, and CMOS imagers make it practical for embedded consumer applications. As with graphics, the best cost-performance ratio is achieved with dedicated hardware. The challenges of face detection in embedded environments include bandwidth constraints set by low cost memory, or processor speed and a need to find optimal solution because applications need reliability, and low cost for final solution.**

I. INTRODUCTION

Recent years have witnessed the steady progress of both theoretical study and practical applications in computer vision.

Many workable software and hardware systems have been proposed for human-computer interaction and surveillance, or intelligent traffic measurement. Besides conventional vision applications, popularity of handheld devices with cameras creates potential for fantastic new applications in PDA's, cell phones, or any small battery driven device. So, it is a good opportunity from both a technical and commercial perspective to tailor algorithm development to the needs of low cost embedded vision systems.

For handheld cameras, human faces are a very common topic of interest, a lot of camera adjustments will be included based on this information. In addition, frontal face detection is usually the first step to initialize many computer vision tasks like tracking, recognition and image analysis. In this paper we investigate the parallelism in the state-of-the-art Adaboost based face detection algorithm and neural networks algorithms.

The challenges include efficient system design, since most published vision algorithms don't take hard real-time and parallel processing into consideration. The latter involves pipeline design, data flow arrangement and parallel acceleration, while for a hard real-time design, we propose first to obtain a desktop solution, because is more simple to gain. After the solution is obtained we will implement on a embedded work platform and optimize the algorithm for best results.

Simulation results show this real-time detection system achieves 75%-80% detection rate for group portraits. The Adaboost based face detection algorithm and some related hardware face detection systems was described briefly in related works.

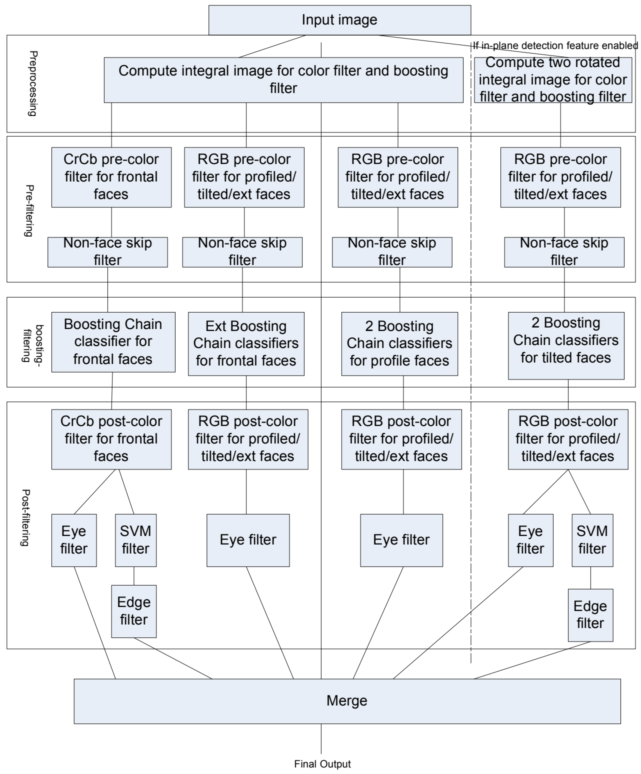

Figure 1. A schematic owerview of face detector algorithm.

A. Adaboost based face detection

The goal of face detection is to locate any faces present in still images. This has long been a focus of computer vision research and has seen great success. Please refer to [3] until [12] for detailed surveys. Among face detection algorithms, the Adaboost [2] based method proposed by Viola and Jones [1] gains great popularity due to a high detection rate, low complexity and solid theoretical basis. The fast speed of Adaboost method is due to the use of simple Haar-like features and a cascaded classifier structure, which excludes most of the image window hypotheses quickly.

In a pre-processing stage, an auxiliary image, Ii, called the integral image or summed-area table [5] is calculated from the original image, Io, where the value Ii(i, j) is the sum of Io pixels above or to the left of position (i, j) in Io. Using Ii, the sum of Io pixel intensities in any rectangle can be calculated in constant time. Afterwards, each image window w, at all positions and all scales, is fed into a cascaded classifier. At each stage, the classifier response h(w) is the sum of a series of features responses hj(w).

$$h(w) = \sum_{j=1}^{n_i} h_j(w), h_j(w) = \begin{cases} \alpha_{j1} for\_f_j(w) < t_j \\ \alpha_{j2}\_else \end{cases} \quad (1)$$

where fj(w) is the feature response of the jth Haar feature and αj1, αj2 are the feature weight coefficients. If h(w) is less than a threshold t, the window w will be regarded as non-face and thrown away, otherwise proceed to the next classifier. Fig. 1 shows the block diagram. Please refer to [1] and [5] for the details.

Beyond performance, Adaboost face detector popularity comes from low average execution time. As will be shown, it can be modified to allow for steady data flow and a neat hardware implementation. However, there are challenges for an embedded system. The algorithm assumes random access of the large integral image and considerable processing power for multiplication and floating-point.

B. Hardware limitations

Essentially, the face detection problem is a pattern classification problem. In addition to the discrimination power of classifiers, the number of image windows evaluated plays a significant role in performance, so computational efficiency is critical to the success of detection. The number of features examined at run-time is an input image-dependent variable. So, a complexity control scheme is indispensable to meet hard real-time deadlines.

In addition, to reduce hardware complexity by real-time restrictions, some approximations to the Adaboost based algorithm will be useful to use.

In the literature, there are hardware implementations of face detection based on tone color detection [3, 7] and Neural Networks [10]. For skin-color methods, in the training stage a statistical skin-color model in a certain color space is learned with labeled skin pixels. During detection, skin pixels are extracted with this model, and then heuristics based on face edge template [3] or connected component analysis [7] are applied to determine face regions. Generally, skin-color methods are efficient and robust to geometric transformation. But, no matter the color space used, skin-color models are not reliable in unconstrained environments since illumination and variation among individuals cause face color changes.

C. Literature prospects

The solutions of the above problems are given by many partial approaches. These are changing because a lot of constraints are required. And also, does not exist a general applicable method. Some of the used or proposed approaches are presented in the following.

- Eigenfaces (Karrhunen-Loeve decompositions)

The eigenface approach is a very used and early implemented method. It is based on the decomposition of the face in a base of faces, computed from the database images. The unknown face is written as a weighted sum of the base images. The weights will characterize the face from the image.

Also was proposal to improve face recognition under a new condition, not captured by the prototypes coherently. They suggest using a second level linear combination of the prototypes belonging to the same faces class, to treat the prototypes coherently.

- Feature vector based methods

The Gabor-filter based feature vector extraction approaches are one way of investigations. After the face features were localized Gabor transformations are applied to the regions of the features.

- Template based, multiple view

Represents faces with templates from multiple model views that cover different poses from the viewing sphere. The face pose is detected using a 5 level pyramidal approach. Every level gives a better approximation of the features' position. After the pose is estimated, the face matches are determined by comparing the affine transformation – conform the pose – of the relevant regions of the image.

D. Phases of Face Detect and Recognition

The complexity of the face recognition problem implies that it has to be divided in more, simple phases. These phases are studied split, and the results of them are joined because are the meeting points between them. Every phase contains one or more data processing steps, which transforms the input of the phase into the desired output, if the transformation can be realized.

Image acquisition

This phase does the capturing of the pictures using peripheral capturing devices. The image can be obtained from many sources.

There are obvious ways to obtain digitalized images: scanning a photo-camera made picture, saving web-cam generated static or dynamic images, capturing with high quality video camera, or using other sources.

Preprocessing

After an image is available, some processing steps are welcomed. These steps ensure easier processing in the next phases. These preprocessing steps are usually reducing the image errors, they are applying color and histogram processing. The resulting images easier interpreted by humans or are easier processed in the following automatic processing steps.

Segmentation

In face recognition, but in other image processing problems too, it is important to segment the image into object regions and into the background, or to separate the image into known or searched objects. The segmentation is a very important phase, because it can decide if the useful information is extracted or it is ignored: the following phases can be easily compromised if this step doesn't do accurate job. But the use of this step can reduce extremely the next phases processing efforts.

Feature extraction

The features of the image are a set of statistically manner determinable values. The features can be distances between two points, region areas, perimeter lengths, center of mass, moments, bounding regions, etc, or can be the results of other, global or local applied transformations parameters. The selected feature is very important because they are used in de classifying process: a bad results or identification is obtained in a wrong way. A good choice leads to easy, simple and exact identification method. After the features are extracted, they have to be organized into a coherent form, what can describe precisely a face, can compare the parameter sets to determine if they are representing the same person or if they are not, or they can determine if the face is in the existing database.

II. Related Steeps of Work

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies.

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.

Early face-detection algorithms focused on the detection of frontal human faces, whereas newer algorithms attempt to solve the more general and difficult problem of multiview face detection. That is, the detection of faces that are either rotated along the axis from the face to the observer (in-plane rotation), or rotated along the vertical or left-right axis (out-of-plane rotation), or both.

Face detection is used in biometrics, often as a part of (or together with) a facial recognition system. It is also used in video surveillance, human computer interface and image database management. Some recent digital cameras use face detection for auto focus.

Starting from literature, The algorithm proposed by Paul Viola and Michael Jones is a solution based on three key parts:

- new image representation called the Integral Image which allows the features used by the detector to be computed very quickly;
- a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. AdaBoost, short for Adaptive Boosting, is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire. It is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance;
- a method for combining increasingly more complex classifiers in a cascade which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest.

Statistical model (classifier) - based training takes multiple instances of the object class of interest, or positive samples, and multiple negative samples, images that do not contain objects of interest. Positive and negative samples together make a training set. During training, different features are extracted from the training samples and distinctive features that can be used to classify the object are selected. This information is compressed into the statistical model parameters. If the trained classifier does not detect an object (misses the object) or mistakenly detects the absent object (gives a false alarm), it is easy to make an adjustment by adding the corresponding positive or negative samples to the training set.

We start the research in Matlab Platform for understanding, modifications and simplifications of the applied algorithms.

Specifically, we *rescale* the integral image on the fly; *fixed-point* is used instead of floating-point; the Haar feature pool is reduced; and we approximate the normalization factor.

The algorithm is not novel and has been reported in many scientific papers. The downscaled image is smoothed with Gaussian filter to reduce noise, then the search is performed over 19x19 rectangles which are histogram equalized before the classification process.

In the first step the system uses an adaptive pre-filter to eliminate candidate rectangles in the input that it can confidently determine do not contain faces.

In the second step the system feeds the remaining rectangles to an improved implementation of an algorithm based on boost. Finally, in the third step the system attempts to eliminate false positives. It applies a color filter and an edge filter to improve the precision of the detection. It uses a support filter, plus lighting correction and histogram equalization, to further eliminate false positives. The system then outputs each rectangle, which contains a detected face, along with its confidence in each detection.

In order to simplify the face detection problem, the system divides the target faces to be detected into four categories: upright frontal face, extend frontal face set with difficulty illumination/lighting and expression, profiled face (left profile and right profile), and in-plane rotated face (clockwise and anti-clockwise). Three sets of filters are applied to each kind of face. After the filtering stage, regions detected are merged to generate final result.

The project was started for few possible applications:

a. In a video-conference scenario, a passive camera with a wide angle lens provides a global view of the persons seated around a conference table. An active camera provides a close-up of whichever person is speaking. Face detection, operating on the wide-angle images, can be used to estimate the location of conference participants around the table. When a participant speaks, the high resolution camera can zoom onto the face of the speaker and hold the face in the center of the image.

b. In the real-world usage of video surveillance, multiple cameras are utilized in a wide spectrum of applications for the purpose of monitoring. As a result, multiple monitors are used to display the various output video streams. Manual monitoring of multiple screens is tedious and operators are prone to fatigue. Face detection can help to reduce the data from multiple screens into one main screen for monitoring.

c. Better surveillance system where Face Detection is the first step for face recognition.

The project proposed aims at the following general objectives: to follow a new direction, applications on embedded mobile systems. Hardware limitations and low memory, involve a detailed research of mobile platform and smart solutions for software. Applications of face detections, tracking, and recognitions will have impact about future development of those systems. A lot of applications will grove like: camera surveillance, automatic zoom of digital cameras, person's identifications in banking systems. The cost of these applications will be lower if these solutions are included in camera and run in real time.

Specific objectives of project:

- promote fundamental research resulting in modern, interdisciplinary applications.

- recover and become in line with the most recent tendencies of the field, by choosing a flexible instrument for study and communication.

- preserve the identified material, propose patents for standardization.

- actively collaborate with doctoral students and direct them towards archive research.

Expanding the horizon of interpretation on the subject at the present time implies firstly a recovery at the level of main instruments to investigate that problem, by making a rigorous inventory of the materials and documents available in libraries and archives in the country and abroad, which reflect the best the directions and is an important step aimed to supply consistency and to stimulate new approaches of embedded systems.

III. Experimental Evaluation

During training, different features are extracted from the training samples and distinctive features that can be used to classify the object are selected. This information is "compressed" into the statistical model parameters. If the trained classifier does not detect an object (misses the object) or mistakenly detects the absent object (i.e., gives a false alarm), it is easy to make an adjustment by adding the corresponding positive or negative samples to the training set.

Early face-detection algorithms focused on the detection of frontal human faces, whereas newer algorithms attempt to solve the more general and difficult problem of multi-view face detection.

This project has been started using Matlab 7.0, the interface created (in Figure 3.) and modules has been presented in the Figure 4.

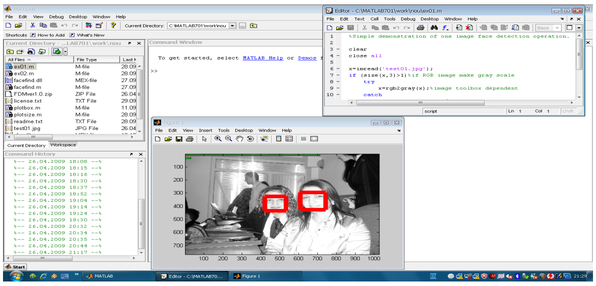

Figure 2. Interface created in Malab

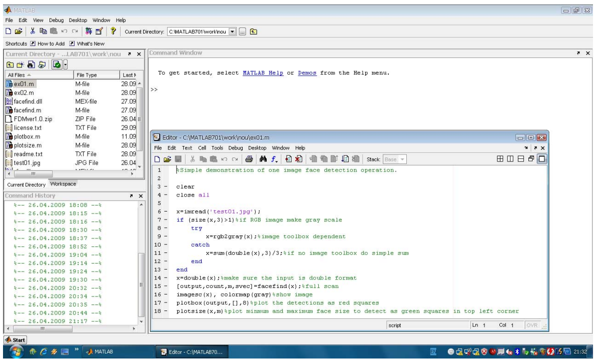

Figure 3. Code created in Malab

Face detection can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). In face detection, one does not have this additional information. In the following pictures we presents the real image and the face detection image.

Figure 4. Original image.(one of sample)

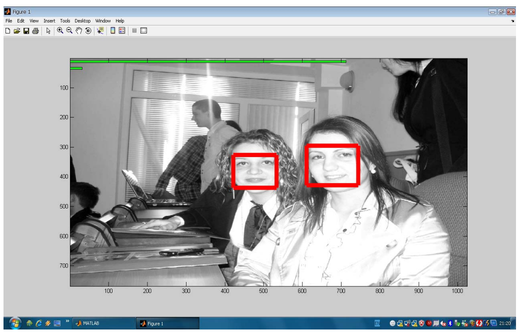

Figure 5. The output result of algorithm.

Another aspect was to have a point of view regarding used algorithms. One team was focused to compare applications of detection face available to users, using the same lot of images, but this time using more complex picture in terms of focus, the occurrence of flash, images processed, etc., and finally to suggest the best solution for implementation.

First were studied Pittsburgh Pattern Recognition, a robust algorithm, and the average of running this application is concluded in the next table:

TABLE I

AVERAGE FOR PITTSBURGH PATTERN RECOGNITION

Face Type	Appearance		
	Location in figure	Size in Pixels	Confidence
Frontal	(159,99)	64x64	3.1
Left profile	(242, 78)	29x36	0,5
Right profile	(145, 65)	45x54	0,6

Another algorithm tested was WaldBoost, in this case is necessary to mention the eyes detection is included.

TABLE II

AVERAGE FOR WALD BOOST

Face Type	Appearance		
	Location in figure	Size in Pixels	Confidence
Frontal	(29,29)	16x16	0,0
Frontal	(201,113)	15x15	0,1
Frontal	(145, 65)	45x54	0,6
Frontal	(228,66)	18x18	1,3
Frontal	(386, 84)	19x19	1,8
Left profile	(201,113)	15x15	0,1
Right profile	(145, 65)	16x16	1,2

Best performance where obtained with Pittsburgh Pattern Recognition. These graphs demonstrate that the eye position is better localized using feature search than raw prediction from the mean, but that for eyes the additional shape constraints do not make a significant difference. However, with mouth corners the individual feature search is poor, but is greatly improved by the use of shape constraints.

The facial feature models are constructed using the same method as each individual level of the Viola-Jones Detector cascade. The method of building an AdaBoost template from simple Haar wavelet like features is described by Viola and Jones [ ]

Timings were carried out on a set of 320*262 pixel images, using modest hardware, a 2 GHz Pentium Dual Core processor. The speed of the system is dependent on the pixel values in the image, but can broadly be broken down as shown in table III.

The time to find the best point candidate stage varies greatly, because it implements the search strategy outlined in algorithm.

TABLE III

SEARCH TIME 2 GHZ PROCCESOR

Global search	~ 25 ms
Local feature search	~ 15 ms
Find best candidate set	Up to 10 ms

IV. CONCLUSIONS

The shape constrained search significantly improves on the feature detection accuracy possible when using only the best response from each feature detector. For features that exhibit large variation in location and appearance on the face, such as the corners of the mouth, local image structure is not enough to allow reliable detection. The shape of a set of feature points must be used as a constraint to remove false matches.

A method was proposed to search the space of possible feature point combinations in an efficient manner. After the shape constrained search has taken place on the data set, the average feature distance is within 10% of the eye separation for 60% of faces, 15% for 80% of faces and 15% for 90% of faces and the method improves the initial feature points predicted by the global search detector in 83 % of cases. Similar results are observed when applying the search to more difficult images, which contain more head pose variation. The method is also quick requiring less than 0.5 secs for a 320*262 image on a 1GHz PIII processor.

In the above we have described results for only four feature points. We have performed further experiments with seven points, and obtained similar results in terms of the improvement obtained by using the shape constraints.

REFERENCES

[1] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *In Computer Vision and Pattern Recognition Conference 2001, volume 1,* pages 511–518, Kauai, Hawaii, 2001;

[2] P. Viola and M. Jones. Robust real-time object detection. In *In ICCV Workshop on Statistical and Computation Theories of Vision*, Vancouver, Canada, 2001;

[3] M. H. Yang, N. Ahuja. Detecting human Faces in Color Images. Beckman Institute and Department of Electrical and Computer Engineering University of Illinois at Urbana-Champaign., Ubrana, IL 61801.

[4] R. C. Verma, C. Schmid, and K. Mikolajcayk, Face detection and tracking in a video by propagating detection probabilities, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25 (2003), no. 10, 1216–1228;

[5] D. Cristinacce, Tim Cootes - "Facial feature detection using AdaBoost with shape constraints" - Dept. Imaging Science and Biomedical Engineering University of Manchester, Manchester, M13 9PT, U.K.;

[6] S.M. SMITH, J.M. Brady, "SUSAN – A New Approach to Low Level Image Processing", *Int. Journal of Computer Vision*, 23(1):45–78, May 1997;

[7] I. S.Z. Li, J. Lu, "Face Recognition Using the Nearest Feature Line Method", *IEEE Trans. on Speech and Audio Processing,* Vol.10, pp.439–443, September 2000;

[8] .Z. Li, "Face Recognition Based on Nearest Linear Combinations", In-*Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 839–844,. June 23–25, 1998;

[9] Pavlovic V., Garg A. Efficient Detection of Objects and Attributesusing Boosting.*IEEE Conf. Computer Vision and Pattern Recognition*.2001.

[10] H. Rowley, S. Baluja and T. Kanade. Neural network-based face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 22-38, 1998;

[11] M.H. Yang, D. Kriegman, N. Ahuja, Detecting Faces in Images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24 n± 1, January 2002 pag 91;

[12] T. Pham, M. Worring. Face Detection Methods, A Critical Evaluation.Intelligent Sensory Information Systems, Department of computer science.Amsterdam 2002;

[13] J. Cai, A. Goshasby, Detecting human faces in color images. *Image and-Vision Computing*, nr18:63-74, 1999