DSC 2006 (Workshop on Distributed Smart Cameras), Boulder, CO, USA, October 31 2006

REAL-TIME FACE DETECTION ON A "DUAL-SENSOR" SMART CAMERA USING SMOOTH-EDGES TECHNIQUE

[1] Vincent Jeanne, [2] Francois-Xavier Jegaden, [3] Richard Kleihorst, [4] Alexander Danilin, [5] Ben Schueler

[1] vincent.jeanne@isen.fr, [2] francois-xavier.jegaden@isen.fr
ISEN (Institut Superieur d'Electronique et du Numerique)
29 rue Cuirasse Bretagne, 29200 Brest, France
[3,4,5] Philips Research Laboratories,
High Tech Campus 5, 5656 AE, Eindhoven, The Netherlands

ABSTRACT

In this paper, we propose a new feature-based approach to detect and track faces in real-time applications using a massively parallel processor architecture. Due to the fact that skin color is not a sufficient feature for detecting faces, our technique is based on a smooth-edges detection algorithm. The main advantage, when applied on luminance values, is that it's robust against homogenous variations of the illuminating conditions. Our detection method is implemented on a very compact wireless smart camera. The algorithm uses the output image of two sensors of this camera. With data-fusion the false detection rate is significantly reduced.

Key words: face detection, smooth edges, real-time, varying illumination, pattern recognition, smart camera

1. INTRODUCTION

Face detection has been a very active research domain for more than 25 years because it can be applied in several fields such as surveillance, commercial applications and health care. Although it is easy for a human to differentiate face and non-face, it requires complex algorithms and a lot of computations to perform it on hardware platforms. Indeed, the face image can vary considerably in terms of facial expression and skin color and the quality also depends on the sensor used.

For face detection, two groups of methods are used.
The first group of methods is based on the knowledge of the face: the shape [1], color information [2][3], particulars points' position (for example: existence of a symmetry between the eyes, ratio between mouth and eyes [4]). The main disadvantage of these methods is that all relevant features depend on the environmental conditions. For example, experiments have shown that it can be detrimental to just rely on skin color. In fact, skin color as seen by the sensor is a result of the reflecting light around it. Even if a color space appears to be designed for skin detection [5], it is not reliable enough to be one of the main features to perform face detection.

The second group of methods is based on learning algorithms using database and statistics tools [6] [7]. These techniques work well but require the use of a large amount of memory.

The method presented in this article is about smooth edges detection in feature-based approach.

2. ARCHITECTURE

We implemented our face detection algorithm on a wireless low-power 802.14 camera with embedded vision processor, the IC3D chip a new version of the Xetal chip [8]. This section deals with the architecture of the WICA camera description, shown in Figure 1.

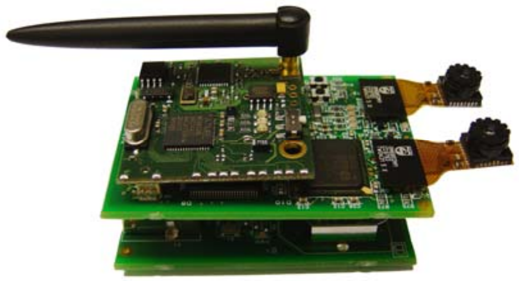


Figure 1: WICA camera (6x4x3cm) shown with 2 VGA sensors

In a vision system we have to distinguish between two levels of processing: low level and high level. Whereas low-level deals with pixel crunching issue, high–level is about making decisions. The low-level operations necessary for natural scene analysis demand a high-performance processor. We are able to achieve this high-performance at low power by using an SIMD processor (Philips IC3D) for the low-level operations. The IC3D takes advantage of the inherent data-parallelism present in the pixel processing. This stand-alone high-performance processor is capable of supplying a pixel performance of up to 50GOPS at 300mW.

In the high-level part of the analysis measurements are performed on the faces found and decisions are taken. These tasks can be performed by a general purpose DSP/CPU.
Currently we are using an upper class 8051 microprocessor to perform this task. This processor has on-board non-volatile memory to store the camera programs and settings, and several I/O pins to control external devices.
The camera can broadcast the events in the scene using an on-board 802.14 transceiver. For this, we used the AquisGain module from Philips. All processors are connected to a dual port RAM, large enough to store video frames and to perform tasks like background subtraction, image pyramids and image registration. A block diagram of the system is shown in Figure 2.

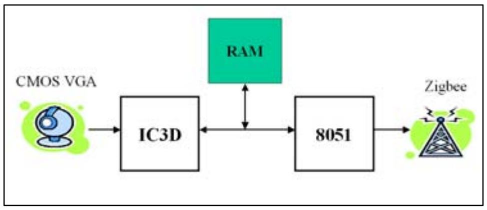


Figure 2: Architecture diagram of the WICA

3. OUR APPROACH

In this paper, we describe a new method for face detection based on smooth edges for pattern recognition. The main advantage of this algorithm is that it is robust to varying illumination. The only requirement we have to is that the light distribution from above is homogenous. A dual sensor approach improves the reliability of the detection. Figure 3 shows an overview of the proposed face detection method.

Our algorithm can be divided in four parts:

- Smooth edges detection
- Boundary detection
- Vertical & horizontal pattern recognition
- Noise removal

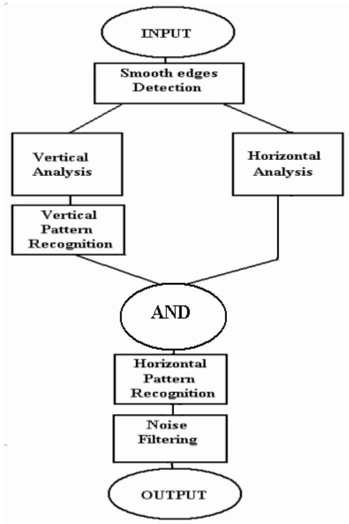


Figure 3: Overview of the Algorithm

3.1 Smooth edges detection

As with several methods for face detection, ours is also based on edge processing [9]. Most of the time, edge information is used in face detection to extract the main features of the face shape as shown in Figure4.

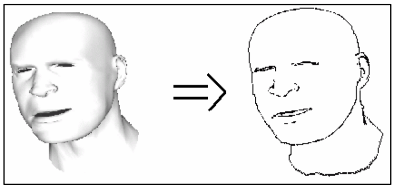


Figure 4: Edge detection

Then by looking at some specific face contours, it is possible to find faces.

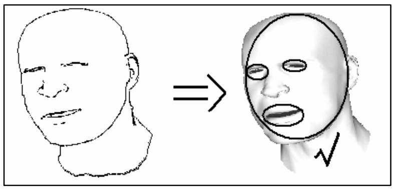


Figure 5: Finding specific Contour on a face

In our case, we are not looking at the high gradient value that represents the main face's edges. In fact, we look at the small gradient values in order to highlight the smooth edges [10]. The main reason for that process is that due to their shapes, human faces are internally mostly composed of smooth edges. These are typically described with little increasing and decreasing luminance along the face. This smooth variation of luminance can mathematically be described in this way:

Let a Pixel in the picture be described by:
I(i,j) with: i ∑ [0;640]; j ∑ [0;480]

A pixel will be seen as a smooth pixel if:

$$\mathbf{| I(i , j) - I(i+1 , j) | < Horizontal\ Threshold}$$
$$\mathbf{| I(i , j) - I(i , j +1) | < Vertical\ Threshold}$$

This smooth edges operator is applied on the luminance signal. The chrominance signals (U & V) contain colour information which it has been proven unreliable to use as a detection means. By using the luminance signal, we reduce the influence of the illumination variations. And since all human beings have almost the same face silhouette, it is possible to find a feature from the smooth edge map that applies for everybody.

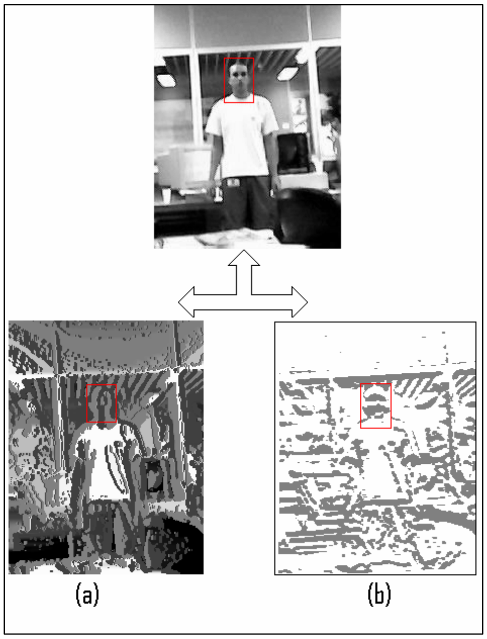


Figure 6: Result of the Smooth Edges Operator

As it can be seen in Figure 6, the result of the smooth operator applied horizontally and vertically is very interesting. Indeed, we can find a particular pattern at the head of the people.
This pattern is actually a smooth mapping of the head:
_ Horizontally **[a]** the head can be divided in six parts, alternatively increasing and decreasing part of luminance.
_ Vertically **[b]** we can distinguish between four or five part. This direction shows more variation, for example having a beard can create a new part of decreasing luminance.

3.2 Boundary Detection

Once the smooth operator is applied, it is not necessary to keep all the data. To reduce it we use a simple operator in order to match the boundaries between the different parts of the smooth mapping. This operator can mathematically be described by:

<u>A pixel will be seen as a boundary if:</u>

$$\left| \frac{\partial I(i,j)}{\partial i} - \frac{\partial I(i+1,j)}{\partial i} \right| \mathrel{!=} 0 \quad \text{// Horizontally}$$

$$\left| \frac{\partial I(i,j)}{\partial j} - \frac{\partial I(i,j+1)}{\partial j} \right| \mathrel{!=} 0 \quad \text{// Vertically}$$

As we can see in Figure 7 and 8, the main advantage of the boundary operator is that it removes data in the picture without removing the required feature. This operator keeps the boundary between two different smooth variations of luminance in order to easily process it.

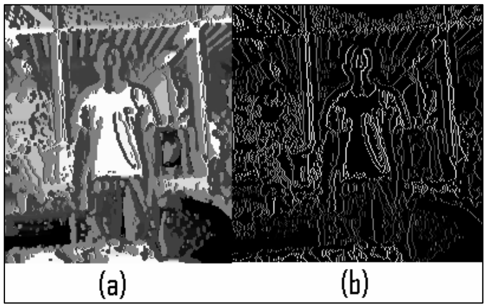


Figure 7: Result of the Boundary Operator applied horizontally

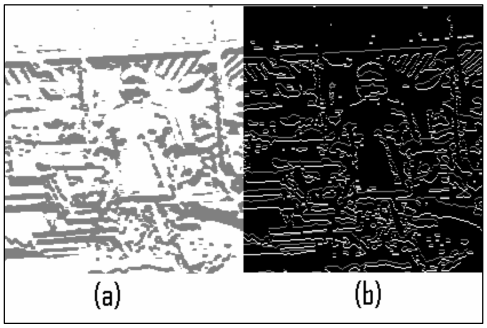


Figure 8: Result of the Boundary Operator applied vertically

3.3 Pattern Recognition

At this step of the face detection algorithm, we now have two streams of data: the binary outputs of the boundary operator applied horizontally and vertically.
For the pattern recognition, we put as a requirement that the system will search a pattern in a 19x19 pixels window. To look for different head sizes we will use a pyramid approach.

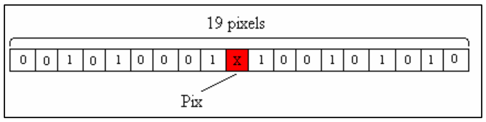
3.3.1 Vertical Pattern Recognition

As it can be seen in the Figure 7, the data contained in the head are very specific compare to the rest of the picture. After applying the boundary operator horizontally, we can notice that six vertical lines can be easily found in the head even when the head rotate between –30° and 30°, because the shadows follow the light.

So the next step is using a 19 by 19 pixels window to look at those seven lines. Due to the single instruction multiple data architecture of our DSP, this pattern recognition has to be implemented in a specific way described in the following part.

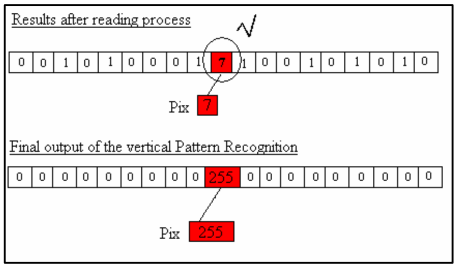
Pattern Recognition processed on SIMD processor:

1.The first step is for each pixel to select an area of 19 pixels wide in a line.



2. Once the area is selected, for every pixel in a line we read the value of its nine left and right neighbours. Each time we will read a pixel's value different from 0, we will add 1 in the center processor.
Note : this process is applied on all processors in parallel.

3. The final process of the pattern recognition is about checking output lines of the previous step. Then, if a pixel's value is 7 we found the correct number of lines, we put it at 255, otherwise 0.



Finally at the end of this process, our output is a binary picture:

Pix_Value= | 255 if the Vertical pattern is found
| 0 otherwise

As it can be seen in the Figure 9, even if the pattern is found at different places on the picture, the biggest amount of detected patterns is on the head.

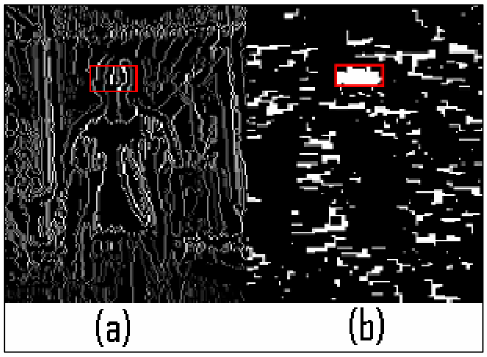


Figure 9: (A) Input of the Pattern Recognition applied horizontally,(B) Result of the Pattern Recognition applied horizontally

3.3.2 Horizontal Pattern Recognition

As shown in the previous paragraph, the vertical pattern recognition contains the most significant response of our detection, but it still not robust enough to be used as optimal face detection (see the noise in the Figure 9). The use of the horizontal pattern recognition is needed in the way that we can combine both techniques to remove the noise and increase the robustness of the detection. After the horizontal process, we are looking for a boundary between the eyes and the forehead in the "previous detected region", using the vertical process.
Considering that the common part between the two processes is significant in the entire image, an additional constraint is needed. We can notice that the line between the forehead and the eyes is surrounded by several black pixels, and is present over a width of 19 pixels. So we use this information to remove the noise, as we know that this pattern is only present in the region of the eyes.

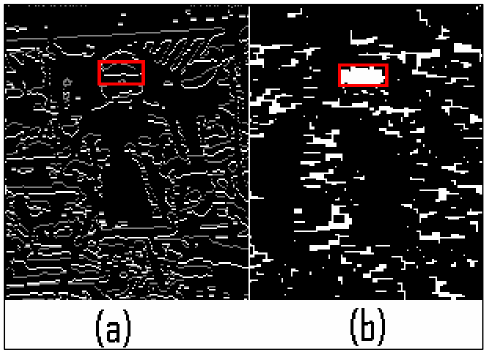


Figure 10: Combination of the horizontal process and vertical process.

3.4 Noise removing part: Large colour window & distributed face detection

The combination of the horizontal process, the vertical process, and a large skin colour domain (Hue value in the HSV domain between 0 and 35, i.e all the reddish colours) succeeds to remove most part of the noise. Notice that the use of the skin tone is only here as a filter to remove the noise and not as a main feature to find a face. As we are processing in real time, and as we are not using a high-resolution sensor, we still have a bit of noise, but it's randomised over time in the entire picture, which helps us to get the real coordinate of the face. To remove this noise we takes benefit of the WICAs architecture. Indeed, the WICA smart Camera has two VGA sensors, and the IC3D is able to process both of the inputs in real-time.
In order to use this distributed smart camera network, we apply the algorithm alternatively on both inputs as shown in Figure 11.

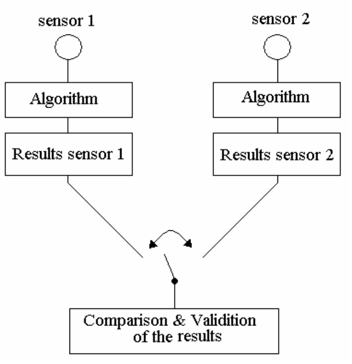


Figure 11: Distributed Face Detection

The video frame rate of the WICA is 24 frames per second, thus for the distributed face detection the algorithm is applied on the first sensor during the first twelve frames, and on the second sensor during the last twelve frames. Finally, by combining the results of both detections, we can notice that the small distance between the two sensors (about 2cm) is enough to remove the noise. Then, by processing on two slightly different angles of view is sufficient to considerably remove the false detections.

The results of this approach are shown in Figure 12: the red curve corresponds to the first sensor and the blue curve corresponds to the second sensor. The number of detections for a row is represented on the Y-axis, and X-axis represents the row index for a 640x480 picture's frame.

The right position of the face can be found by comparing the amount of detection for both sensors. If this amount is bigger than a defined threshold for both curves at the same index, we can be sure that it is a face, as presented here for the row index 200, then the rest can be considered as noise.

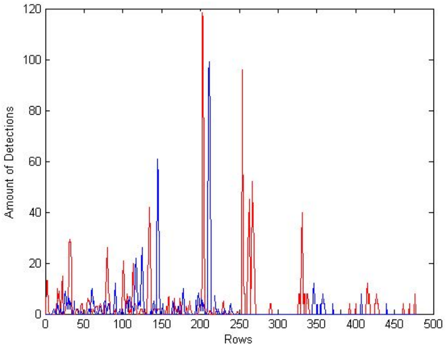


Figure 12: Results of the Distributed Face Detection

4. CONCLUSION

In this work we implemented on a dual-sensor smart camera -a face detection algorithm using soft edges techniques. As we use this soft-edges technique on grey-level images, it means that it is robust to illumination changes. Nevertheless, the light condition has to be quite homogeneous from above, as we have to get the soft edges from the both sides of the face. The system also requires that the people are facing straight ahead. In all our development, we have taken care just to use one threshold, applied on the luminance difference between two consecutive pixels, needed for the smooth mapping. One of the directions that future research might take care is to develop better ways to integrate the shape of the face combined with the shape of the shoulders together, or with the human figure.

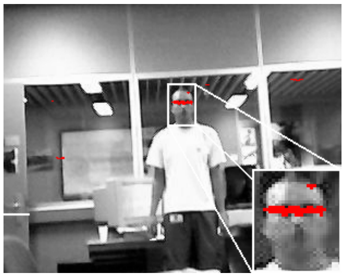


Figure 13: Final output, on one frame, of the Face Detection's Algorithm

5 REFERENCE

[1] M.H Yang, D.J Kriergman, N.Ahuja. Detecting faces in images : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[2] Q. Zhu, K-T. Cheng, C-T Wu, Y-L Wu. Adaptive Learning of an Accurate Skin-Color Model. *IEEE Conference on Automatic Face and Gesture Recognition*, 2004.

[3] M. Soriano, B Martinkauppi, S. Huovinen, M. Laaksonen. Skin detection in video under changing lighting conditions, 2000.

[4] E. Saber, A. Tekalp. Face detection and facial feature extraction using color, shape and symmetry based cost functions. *Proc. International Conference on Pattern Recognition, Vienna*, 1996.

[5] S.K. Singh1, D. S. Chauhan, M. Vatsa, R. Singh. A Robust Skin Color Based Face Detection Algorithm. *Tamkang Journal of Science and Engineering, 2003*

[6] T. Heseltine, N. Pears, J. Austin. Evaluation of pre-processing techniques for eigenfaces based face recognition. *Second International Conference on Image and Graphics*, 2002.

[7] Paul Viola, Michael Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. *IEEE Conference on Computer Vision and Pattern Recognition,* 2001.

[8] A.A. Abbo, R.P. Kleihorst, L. Sevat. A Low-¨Power Parallel Processor IC for Digital Video Cameras. *ISCAS,* 2001.

[9] Y. Suzuki, T. Shibata. An Edge-based Face Detection Algorithm robust against Illumination, focus and scale variations. *EUSIPCO,* 2004.

[10] A. Tankus, Y. Yeshurn, N. Intrator. Face detection by Direct Convexity Estimation, 2003. [http://www.math.tau.ac.il/~hezy/papers/j23.pdf]