Automatic features detection for overlapping face images on their 3D range models

Raffaella Lanzarotti, Paola Campadelli*
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41 20135 Milano, Italy
{lanzarotti, campadelli}@dsi.unimi.it

N. Alberto Borghese
INB, Lab. Human Motion Analysis
and Virtual Reality, CNR
c/o LITA Via Cervi, 93 20090 Segrate (Mi), Italy
borghese@inb.mi.cnr.it

Abstract

We describe an algorithm for the automatic features detection in 2D color images of human faces. The algorithm proceeds with subsequent refinements. First, it identifies the sub-images containing each feature (eyes, nose and lips). Afterwards, it processes the single features separately by a blend of techniques which use both color and shape information. The method does not require any manual setting or operator intervention.

1 Introduction

Human faces are characterized by both 3D geometrical shape and color appearance. Therefore, to reproduce with high realism a 3D digital model of a face, the 3D shape and the color field have to be acquired. 3D scanners can be used to the purpose of acquiring and registering color and 3D shape information [3]. Apart from their high cost, 3D scanners require that the real human face is available at scanning time. There are many situations where this cannot be the case. For instance, when reconstructing 3D clones of celebrities [2] the 3D shape can be acquired from wax models, but the color field has to be acquired from footage or old color images which are not registered with the 3D shape. Another important case is the reproduction of human face on a daily basis for web applications. In this case, it would be desirable to have a 3D shape model to which a color image could be sampled and applied on frequently and automatically. In general, a minimum of scanning acquisitions is preferable since they are time consuming and the setup is expensive. On the contrary, as image acquisition can be carried out quickly and with low-cost digital photo cameras, the acquisition of color images and their application to the 3D model [9][8] can be done often. In this case the 3D model and the color field have to be registered. To accomplish this task, the projective transformation realized by acquiring the 2D image has to be derived [5][13]. A minimum of 5 points is required to compute the 9 parameters of the transformation[14].

Analyzing the surface local curvature, points on the 3D surfaces can be identified. As eyes, lips and nose all present marked curvature, they can be identified by using proper spatial filters [7]. More difficult is the extraction of the same features from 2D face images such as "tricks" have been widely used e.g. carefully controlling the internal parameters of the camera or using lip-stick for the lips and markers for the other features [7].

In this framework we propose a more robust method, which works on natural face images. It is based on a novel integration of standard image processing techniques which use both color and shape information.

2 Methodology

The method we propose works on images of faces' foregrounds. We thus ignore the problem of localizing the faces in more complex scenes [10][6]. We acquire the images with homogeneous and light-colored background and only little rotations of the head are accepted (quasi-vertical and frontal position). This conditions are common to most of the face analysis systems (e.g. [15], [17]). The method works through two hierarchical processing modules: the first identifies four sub-images, each tightly containing one of the features of interest (the two eyes, the nose and the lips); the other three modules are specialized in localizing with high accuracy repere points on the lips, the nose and the eyes.

*Work supported by project "Disegno e analisi di algoritmi" (ex MURST 60% 2000)

2.1 Identification of sub-images

Starting from a face foreground, we want to localize the eyes, the nose and the lips. At this stage the use of templates [4][18] is computationally too expensive.

An alternative solution is to search the parts of interest by their color. The difficulty here is in the definition of proper color ranges. Inter-individual differences and variations in the illumination produce large variation in the color range of the face components, and does not allow the color characterization of them [16]. For these reasons we moved to consider the grey-level images [Fig.1(a)].

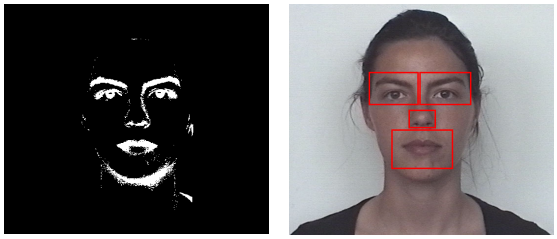
The first consideration to make is that the skin grey-levels are lighter than the ones of the features of interest. Thus it is possible to find a threshold which binarize the image extracting all the features[4]. The question now is how to determine the threshold. As we did not make any assumption regarding the illumination, it would be desirable to obtain an adaptive threshold based on the distribution of grey-levels in a particular image. This led us to adopt a competitive learning algorithm [1] which divides the image in a predetermined number of clusters. In this case we search for three clusters: one represents the background (the lightest), one the skin (the intermediate grey-level), and the third one the features and the remaining dark pixels of the images (for example the hair) [Fig.1(b)].

(a) (b)

Figure 1. (a) Grey-level image; (b) Clustered image

To localize the features of interest, first, the largest connected region of the intermediate grey-level cluster, S, (corresponding to the skin) is found. Then all the pixels surrounded by pixels of S and belonging to the darkest grey-level cluster are identified and set to 1. All the others are set to 0 [Fig.2(a)]. Finally boxes containing the features of interest are defined iterating horizontal and vertical projections: first we localize the vertical position of the eyes which corresponds to the highest peak in the horizontal projection of the image in figure 2(a). Then, on the band surrounding the peak, we make the vertical projection in order to localize the two eyes separately. The process continues applying the projections on the remaining sub-image. The peak we now obtain in the horizontal projection corresponds to the lips, and the last peak corresponds to the nose. The result of this module is shown in figure 2(b).

At this stage, more refined algorithms are applied to the pixels inside the four sub-images to precisely identify the features of interest and locate repere points.



(a) (b)

Figure 2. (a) Features extracted; (b) Image Sub-division

2.2 Lips

Our goal for the lips is the determination of the corners and of their middle point. To get a more robust estimate, the entire outline is determined. Classical edge detectors (Sobel, Roberts, Canny, Perona) experimentally fail to produce consistent results both in the determination of the outline and of the repere points. Therefore a novel more refined technique has been developed. This is based on the following steps: localization of the lips cut and of the lips lowest point, localization of the lips corners and of the lips outline.

2.2.1 Recognition of the lips cut and of the lips lowest point

To determine the lips cut we apply the Sobel vertical derivative operator to the mouth sub-image. This gives better results than determining the Sobel gradient considering that lips cut line is approximately horizontal. We then cluster the vertical derivative values into three clusters: one associated to light-to-dark vertical transition, one to dark-to-light and one to no meaningful transition [Fig.3(a)]. We then determine the largest dark connected region and we extract, for every x, the upper pixel belonging to it [Fig.3(b)]. The line L connecting these pixels well represents the lips cut apart from its extremes: in some cases, the line ends before the corners, in other, it exceeds them. At this point we are able to recognize the lips lowest point as the lower extreme of the largest connected white region under the line L

[Fig.3(b)]. This point is significant both because it gives the vertical position of the middle point and for the lips outline recognition.

2.2.2 Recognition of the lips corners and of the lips outline

The key idea here is to arrive to the corners by subsequent refinements. We go back to the original color image considering the mouth box only and clustering it into two clusters. We apply the algorithm in the CIE-Luv color space, being the one which has given experimentally the best results consistently, with respect to RGB, CIE-Lab and HSV color spaces. This lead to obtain one cluster, M, which is roughly associated to the lips and the other to the surrounding skin. The outline of the region M is quite precise in describing the upper lip, but it is not suitable to describe the corners and the outline of the lower lip.

We then apply a second time the same clustering algorithm to the sub-image which tightly contains the region M. In this case we require four clusters: the darkest one identifies the shadow corresponding to the lips cut and gives a precise information regarding the horizontal position of the corners. Combining this with the information given by the line L, we obtain the correct corners position [Fig.3(c)].

Using all the information we have gathered up to now, we are able to define with sufficient precision the lips' outline. The lower lip is identified as the parabola through the corners and the lips lowest point. For the upper lip, we take the upper part of the outline of M, straining it to finish in the found corners [Fig.3(d)].

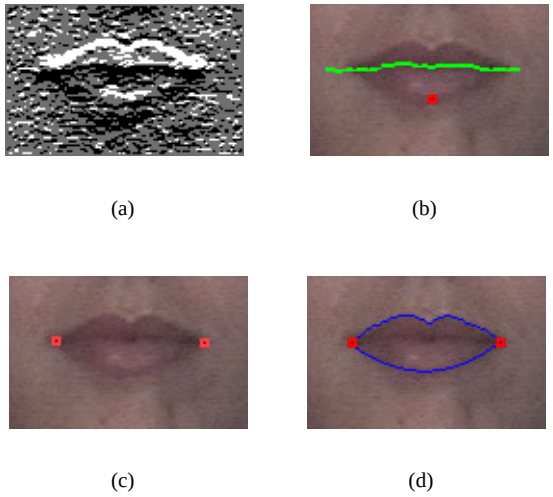


Figure 3. Processing of the lips' image: (a) Clustering of the vertical derivative with 3 clusters; (b) Detection of the lips cut and of the lips lowest point; (c) Corners recognition; (d) Outline.

Figure 4. (a) Horizontal derivative; (b) Eye sub-image

2.3 Eyes

Also in this module we proceed with subsequent refinements: first the eyes are located more precisely, eliminating the eyebrows from the eyes sub-images, second pupils and lower and upper extremes of the eyes are recognized, finally the eyes corners and the upper arcs are defined.

2.3.1 Eyes localization

To obtain a more precise localization of the eye, we apply the Sobel horizontal derivative operator to the grey-level image [Fig.4(a)]. We can notice that, corresponding to one side of the iris, the values of the derivative are always the highest. We can therefore localize such pixels using a threshold which binarize the image putting at 1 only the 5% of the brightest pixels and locate the interesting sub-image in the corresponding band [Fig.4(b)].

2.3.2 Pupil and eye extreme points recognition

Once we have better localized the eye, we proceed looking for the pupil. Some authors (e.g. [11], [15]) suggest finding the darkest pixels. However, this approach is suitable only to particular illumination conditions. Instead, in general, we cannot guarantee this: as can be seen in figure 4(b), a reflex spot makes several pixels internal to the pupil very bright. As we did not want to make any assumption on the illumination, we have resorted to template matching. As we look for the iris, the template has the shape of a circle [1]. The correlation is calculated between the template and the binarized image of the eye [Fig.5(a)], obtained clustering it

[1] we observe here that we adopt a circle of ray 8, but this can easily be generalized by taking a ray which length is proportional to the face dimension.

into two clusters. The center of the pupil, P, is localized in correspondence with the maximum value of the correlation.

Then, the lower extreme of the iris is looked for on the vertical axis A passing throw P: it is localized in the position on the axis A corresponding to the lowest pixel belonging to the white pixels of figure 5(a). It is not possible to look for the upper extreme of the eye on figure 5(a), because it would exceed the correct position. What we do is to calculate the absolute value of the vertical derivative, $\triangle_y$, of the eye image and to threshold it keeping the 10% of the pixels with the highest values. The upper extreme of the eye is localized in correspondence to the intersection of the axis A and the region obtained by the thresholding.

2.3.3 Eye corners and upper arc determination

To define the shape of the eye, we apply an edge following algorithm based on the idea of the hysteresis thresholding [12]. The two thresholds are automatically determined in order to keep the 10% and the 40% of the pixels with the highest values of $\triangle_y$. The contour C obtained in this way [Fig.5(b)] well-defines the shape of the eye but it does not identify precisely the corners. This is the reason why we have the necessity to adopt another method to localize the corners. We start from the consideration that the white internal part of the eye is quite evident. In order to emphasize the contrast between it and the surrounding regions, we equalize the grey-level image and then cluster it requiring 3 clusters [Fig.5(c)]. The upper border, U, of such white part [Fig.5(d)] gives a good indication to detect both the internal corner of the eye and its outline: combining U and the contour C, we determine the parabola which better approximates them. Then we localize the eye internal corner on the parabola at the height of the end of U, and the external corner on the parabola at the height of the end of C. This allows us to find a good approximation of the upper part of the eye outline[Fig.5(e)].

2.4 Nose

Regarding this feature, we are interested in finding the nose tip. Observing numerous faces' images, we concluded that the nose is characterized by two dark regions, corresponding to the nostrils, and a light region, corresponding to the reflex of the light on the nose tip. In order to identify these regions, we applied the clustering algorithm to the nose grey-level sub-image requiring four clusters. The lightest grey-level pixels correspond to the reflex, and the darkest grey-level pixels correspond to the nostrils [Fig.6]. Moreover we observed that the middle point between the nostrils gives a good vertical localization of the nose tip and that the lower extreme of the region corresponding to the reflex gives a good horizontal localization. The intersection of these two axis gives the nose tip [Fig.6].

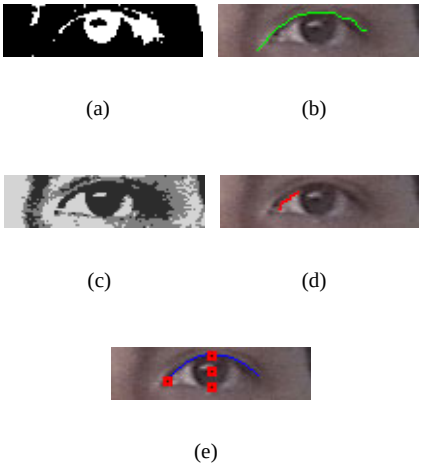


Figure 5. Processing of the eye's image: (a) Binarization; (b) Edge-following; (c) Clustering with 3 clusters; (d) Detection of a piece of border; (e) Detection of the eye upper arc.

Figure 6. Clustering and Tip of the nose

3 Results and conclusion

We have applied the method described above to a set of 10 images acquired in different illumination conditions. They represent Caucasian women and men frontal faces. We asked to the people to have a neutral expression, keeping the mouth closed and the eyes opened. Moreover we have not dealt with the case of men with beard or mustaches. All the images have been taken at about the same scale. Working on images at different scales would not create any problem since the algorithm assumes a given resolution just for the template matching step, which can be easily extended to be adapted to the resolution of the image. Figure 7 shows some example results.The method allows identifying three repere points on the mouth, one on the nose and four on each eye for a total of twelve repere points. This is far more than the five required to align the 3D mesh with the 2D image. In general all the repere points were identified successfully.

In order to evaluate quantitatively the error carried out by the algorithm, we asked to three subjects to locate on the face images the position of the corners of the eyes and of the lips, and the tip of the nose. In case the answers were different, we considered the average value.

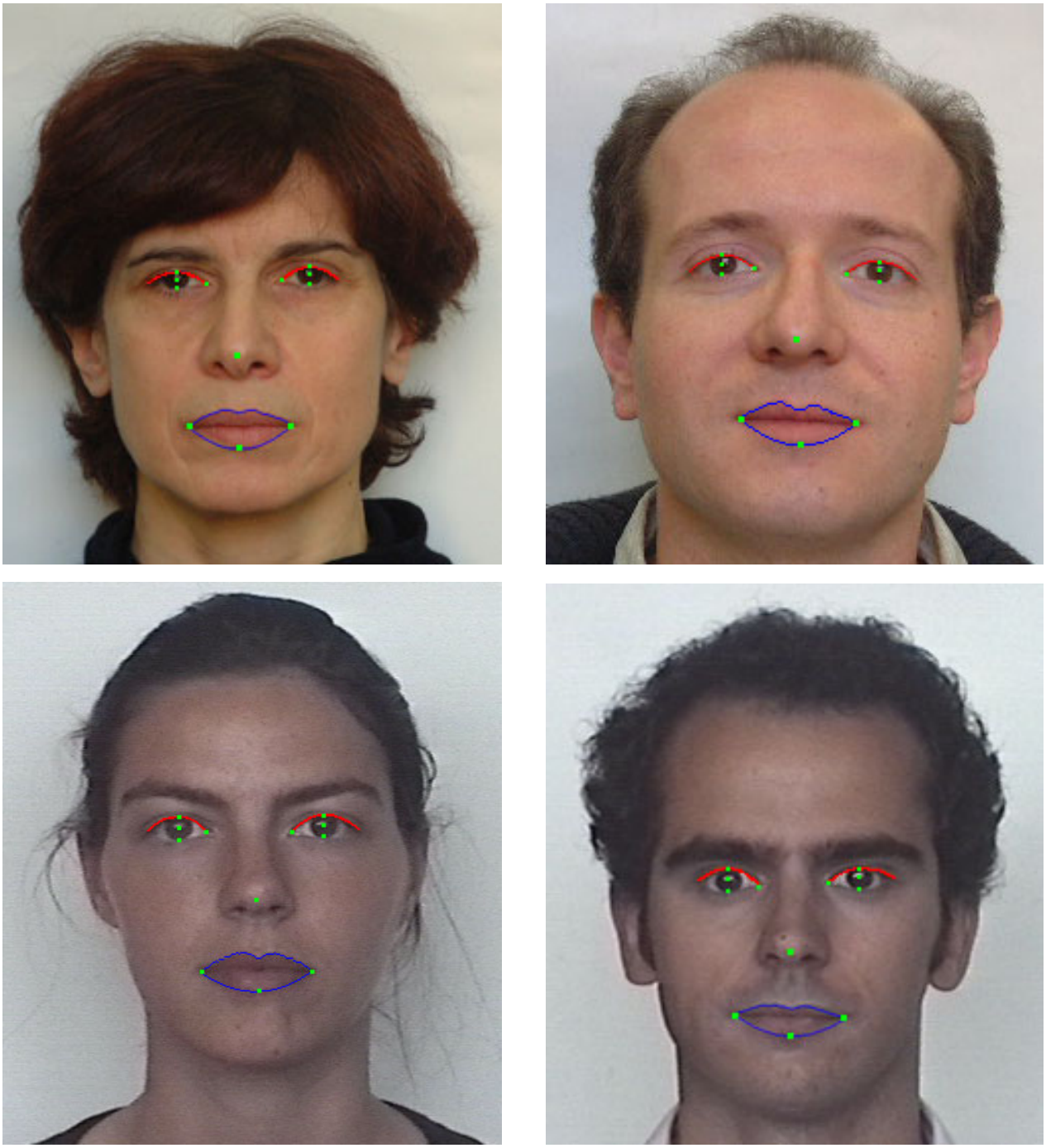


Figure 7. Some results

For each repere point the error has been valued as the distance between the coordinates (x_a, y_a) determined by the algorithm, and the coordinates (x_s, y_s) located by the subjects. We adopted the d_4 metrics[2]. Regarding the mouth's corners, we obtained no error in 9 images, an error equal to 1 in 2 images and an error equal to 6 in 1 case (right mouth corner in the right-low image of figure 7).

The tip of the nose has been determined incorrectly only in 1 case with an error equal to 3.

The estimate of the eye internal corners is reasonable: we obtained no error in 8 images, an error lower than 3 in 2 images and an error equal to 5 in 2 images.

The most critical point is the determination of the eye external corners which results correct only in 2 cases. In the other images the error is lower than 6. Normalizing the error with respect to the eye dimension (width, height), we obtain, in the worst case, an error of 10%.

We believe we can conclude that the precision is high for lips and nose, while for the eyes the method is less accurate.

As regard the problem of registering the 3D model and the image, this inaccuracy can be faced considering that we have much more points than we need. Therefore each point contribution can be weighed taking into account its reliability.

A final consideration has to be done about the program running time. The time necessary to process one image of 640×480 pixels on a Pentium II, 200MHz, 64Mb of RAM is about 1 minute. This result is acceptable considering this process works off-line. Moreover the program has been implemented with IDL, an interpreted language. The running time can be certainly reduced developing the algorithm in a compiled language and optimizing the code.

References

[1] M. Arbib and T. Uchiyama. Color image segmentation using competitive learning. *IEEE Transactions on pattern analysis and machine intelligence*, 16:1197–1206, 1994.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. *Proceedings ACM Siggraph, Los Angeles*, 1999.

[3] A. Borghese, G. Ferrigno, G. Baroni, R. Savarè, S. Ferrari, and A. Pedotti. Autoscan: A flexible and portable scanner of 3d surfaces. *IEEE Computer Graphics and Applications*, pages 38–41, 1998.

[4] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions PAMI*, 15(10):1042–1062, 1993.

[5] M. Levoy, S. Rusinkiewicz, M. Gintzon, J. Ginsberg, K. Pulli, D. Koller, S. Anderson, J. Shage, B. Curless, L. Pereira, J. Davis, and D. Fulk. The digital michelangelo project: 3d scanning large statues. *Proceedings ACM Siggraph', 2000, St. Luis.*, 2000.

[6] T. Poggio and K. Sung. Example-based learning for view-based human face detection. *Proceedings of the ARPA Image Understanding Workshop*, II:843–850, 1994.

[7] M. Proesmans and V. G. Luc. One-shot 3d shape and texture acquisition of facial data. *Proceedings ICCV*, 1998.

[8] P. Rigiroli, P. Campadelli, A. Pedotti, and N. Borghese. Mesh refinement with colour attributes. *Computers and Graphics*, 25(3), 2001.

[9] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Multiple textures stiching and blending on 3d objects. *Proceedings 10th Eurographics Rendering Workshop, Springer-Verlag*, pages 119–130, 1999.

[10] H. Rowley, S. Baluja, and T. Kanade. Neural networkbased face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[11] R. Stiefelhagen, J. Yang, and A. Weibel. Tracking eyes and monitoring eye gaze. *Proceedings of the Workshop on Perceptual User Interfaces (PUI)*, pages 98–100, 1997.

[12] E. Trucco and A. Verri. *Introductory techniques for 3d computer vision*. Prentice-Hall, Inc., New Jersey, 1998.

[13] F. Weinhaus and V. Devarajan. Texture mapping 3d models of real-world scenes. *ACM Computing Surveys*, 29(4):325–365, 1997.

[14] P. Wolf. *Elements of photogrammetry*. New York, McGraw-Hill, 1983.

[15] Y. Yan and K. Challapali. A system for the automatic extraction of 3-d facial feature points for face model calibration. *Proceedings ICIP*, 2000.

[16] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. *Technical Report CMUCS*, 97(146), 1997.

[17] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Real-time face and facial feature tracking and applications. *Proceedings AVSP*, pages 79–84, 1998.

[18] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International journal of computer vision*, 8(2):99–111, 1992.

[2] $\text{error} = |x_a - x_s| + |y_a - y_s|$