Estimation of Pose and Illuminant Direction for Face Processing

R. Brunelli
Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Povo, Trento, ITALY

Abstract— **In this paper[1] three problems related to the analysis of facial images are addressed: the estimation of the illuminant direction, the compensation of illumination effects and, finally, the recovery of the pose of the face, restricted to in-depth rotations. The solutions proposed for these problems rely on the use of computer graphics techniques to provide images of faces under different illumination and pose, starting from a database of frontal views under frontal illumination.**

Appeared as MIT AI. Memo No.1499

1. Introduction

Automated face perception (localization, recognition and coding) is now a very active research topic in the computer vision community. Among the reasons, the possibility of building applications on top of existing research is probably one of the most important. While recent results on localization and recognition open the way to automated security systems based on face identification, breakthroughs in the field of facial image coding are of practical interest for teleconferencing and database applications.

In this paper three tasks will be addressed: learning the direction of illuminant for frontal views of faces, compensating for non frontal illumination and, finally, estimating the pose of a face, limited to in-depth rotations. The solutions we propose to these tasks have in common two important aspects: the use of learning techniques and the synthesis of the examples used in the learning stage. Learning an input/output mapping from examples is a powerful general mechanism of problem solving once a suitably large number of meaningful examples is available. Unfortunately, gathering the needed examples is often a time consuming, expensive process. Yet, the use of a-priori knowledge can help in creating new, valid, examples from a (possibly limited) available set. In this paper we use a rough model of the 3D head structure to generate from a single, frontal, view of a face under uniform illumination, a set of views under different poses and illumination using ray-tracing and texture mapping techniques. The resulting extended sets of examples will be used for solving the addressed problems using learning techniques.

2. Learning the illuminant direction

In this section the computation of the direction of the illuminant is considered as a learning task (see [1], [2], [3], [4] for other approaches). The images for which the direction must be computed are very constrained: they are frontal views of faces with a fixed interocular distance [5]. Once the illuminant direction is known it can be compensated for, obtaining an image under *standard illumination* which can be more easily compared to a database of faces using standard techniques such as cross-correlation. Let us introduce a very simple lighting model [6]:

$$I = (A + L\delta \cos\omega)\Lambda \tag{1}$$

where I represents the emitted intensity, A is the ambient energy, δ is 1 if the point is visible from the light source and 0 otherwise, ω is the angle between the incident light and the surface normal, Λ is the surface albedo and L is the intensity of the directional light. Let us assume that a frontal image I_A of a face under diffuse ambient lighting ($L = \delta = 0$) is available:

$$I_A = A\Lambda \tag{2}$$

The detected intensity is then proportional to the surface albedo. Let us now assume that a 3D model of the same face is available. The corresponding surface can be easily *rendered* using ray-tracing techniques if the light sources and the surface albedo are given. In particular, we can consider a constant surface albedo Λ_0 and use a single, directional, light source of intensity L in addition to an appropriate level of ambient light A'. By changing the direction $\boldsymbol{\Omega} = (\theta, \phi)$[2] of the emitted light, the corresponding synthetic image $S(\theta, \phi, A')$ can be computed:

$$S(\theta, \phi, A') = (A' + L\delta\cos\omega)\Lambda_0 \tag{3}$$

Using the albedo information I_A from the real image, a set of images $I(\theta, \phi, A')$ can be computed:

$$I(\theta, \phi, A') = \frac{1}{\Lambda_0 A} S(\theta, \phi, A') I_A \propto S(\theta, \phi, A') I_A \tag{4}$$

In the following paragraphs it will be shown that even a very crude 3D model of a head can be used to generate images for training a network that learns the direction of the illuminant. The effectiveness of the training will be demonstrated by testing the trained network on a set of real images of a different face. The resulting estimates are in good quantitative agreement with the data.

[1] This report describes research done by the author at CBCL-MIT and at IRST.

[2] The angles θ and ϕ correspond to left-right and top-down displacement respectively.

From a rather general point of view the problem of learning can be considered as a problem of function reconstruction from sparse data [7]. The points at which the function value is known represent the examples while the function to be reconstructed is the input/output dependence to be learned. If no additional constraints are imposed, the problem is ill posed. The single, most important constraint is that of *smoothness*: similar inputs should be mapped into similar outputs. Regularization theory formalizes the concept and provides techniques to select appropriate family of mappings among which an approximation to the unknown function can be chosen. Let us consider the reconstruction of a scalar function $y = f(\boldsymbol{x})$: the vector case can be solved by considering each component in turn. Given a parametric family of mappings $G(\boldsymbol{x}; \boldsymbol{\alpha})$ and a set of examples $\{(\boldsymbol{x}_i, y_i)\}$ the function which minimizes the following functional is chosen:

$$E(\boldsymbol{\alpha}) = \sum_i (y_i - G(\boldsymbol{x}_i; \boldsymbol{\alpha}) - \boldsymbol{p}(\boldsymbol{x}_i))^2 \tag{5}$$

where $y_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{p}(\boldsymbol{x}_i)$ represent a polynomial term related to the regularization constraints. A common choice for the family G is that of linear superposition of translates of a single function such as the Gaussian:

$$G(\boldsymbol{x}; \{c_j\}, \{\boldsymbol{t}_j\}, W) = \sum_j c_j e^{-(\boldsymbol{x}-\boldsymbol{t}_j)^T W^T W (\boldsymbol{x}-\boldsymbol{t}_j)} \tag{6}$$

where $W^T W$ is a positive definite matrix representing a *metric* (the polynomial term is not required in this case). The resulting approximation structure can also be considered as an HyperBF network (see [7] for further details). In the task of learning the illuminant direction we would like to associate the direction of the light source to a vector of measurements derived from a frontal image of a face.

In order to describe the intensity distribution over the face, the central region (see Figure 1) was divided into four patches, each one represented by an average intensity value computed using Gaussian weights (see Figure 2). The domain of the examples is then $\mathcal{R}^4$. Each input vector $\boldsymbol{x}$ is normalized to length 1 making the input vectors independent from scaling of the image intensities so that the set of images $\{I(\theta, \phi, A')\}_{\theta\phi}$ can be replaced by $\{S(\theta, \phi, A')I_A\}_{\theta\phi}$.

The normalization is necessary as it is not the global light intensity which carries information on the direction of the light source, but rather its spatial distribution. Using a single image under (approximately) diffuse lighting, a set of synthetic images was computed using eqn. (3). A rough 3D model of a polystyrene mannequin head was used to generate the constant albedo images[3]. The direction of the illuminant spanned the range $\theta, \phi \in [-60, 60]$ with the examples uniformly spaced every 5 degrees. The illumination source used for ray tracing was modeled to match the environment in which the test images were acquired (low ambient light and a powerful studio light with diffuser).

[3] A public domain rendering package, *Rayshade 4.0* by Craig Kolb, was used.

From each image $S(\theta, \phi, A')I_A$ an example $(\boldsymbol{x}_{\theta\phi}, \theta)$ is computed. The resulting set of examples is divided into two subsets to be used for training and testing respectively. The use of two independent subsets is important as it allows to check for the phenomenon of overfitting usually related to the use of a network having too many free parameters for the available set of examples. Experimentation with several network structures showed that a HyperBF network with 4 centers and a diagonal metric is appropriate for this task. A second network is built using the examples $\{(\boldsymbol{x}_{\theta\phi}, \phi)\}_{\theta\phi}$. The networks are trained separately using a stochastic algorithm with adaptive memory [8] for the minimization of the global square error of the corresponding outputs:

$$\begin{aligned} E_\theta(\boldsymbol{\alpha}) &= \sum_{\theta\phi} (\theta - G_\theta(\boldsymbol{x}_{\theta\phi}; \boldsymbol{\alpha}))^2 \\ E_\phi(\boldsymbol{\alpha}) &= \sum_{\theta\phi} (\phi - G_\phi(\boldsymbol{x}_{\theta\phi}; \boldsymbol{\alpha}))^2 \end{aligned}$$

The error E_θ for the different values of the illuminant direction is reported in Figure 4. The network trained on the θ angle is then tested on a set of four real images for which the direction of the illuminant is known (see Figure 5). The response of the network is reported in Figure 6 and is in good agreement with the true values.

Once the direction of the illuminant is known, the synthetic images can be used to correct for it, providing an image under standard (e.g. frontal) illumination. The next section details a possible strategy.

3. Illumination compensation

Once the direction of the illuminant is computed, the image can be corrected for it and transformed into an image under standard illumination, e.g. frontal. The compensation can proceeds along the following steps:

1. compute the direction (θ, ϕ) of the illuminant;
2. establish a pixel to pixel correspondence between the image to be corrected X and the reference image used to create the examples I_A

$$(x_X, y_X) \xrightarrow{\mathcal{M}} (x_{I_A}, y_{I_A}) \tag{7}$$

3. generate a view $I(\theta, \phi, A)$ of the reference image under the computed illumination;
4. compute the transformation due to the change in illumination between I_A and $I(\theta, \phi, A)$:

$$\Delta(x, y) = I_A(x, y) - I(\theta, \phi, A; x, y) \tag{8}$$

5. apply the transformation Δ to image X by using the correspondence map $\mathcal{M}$ in the following way [9]:

$$X(x, y) \rightarrow X(x, y) + \Delta(\mathcal{M}_x(x, y), \mathcal{M}_y(x, y)) \tag{9}$$

The pixel to pixel correspondence $\mathcal{M}$ can be computed using optical flow algorithms [10], [11], [12], [13]. However, in order to use such algorithms effectively, it is often necessary to pre-adjust the geometry of image X to that of I_A [14]. This can be done by locating relevant features of the face, such as the nose and mouth, and *warping* image X so that the location of these features is the same as in the reference image. The algorithm for locating the *warping* features should not be sensitive to the illumination under which the images are taken and should be able to locate the features without knowing the identity of the represented person. The usual way to locate a pattern within an image is to search for the maximum of the normalized cross-correlation coefficient ρ_{xy} [15]. The sensitivity of this coefficient to changes in the illumination can be reduced by a suitable processing of the images prior to the comparison (see Appendix A). Furthermore, the identity of the person in the image is usually unknown so that the features should be located using generic templates (a possible strategy is reported in Appendix B). After locating the nose and mouth the whole face is divided into four rectangles with sides parallel to the image boundary: from the eyes upwards, from the eyes to the nose base, from the nose base to the mouth and from the mouth downwards. The two inner rectangles are stretched (or shrunk) vertically so that the nose and mouth are aligned to the corresponding features of the reference image I_A. The lowest rectangle is then modified accordingly. The image contents are then mapped using the rectangles affine transformations and a hierarchical optical flow algorithm is used to build a correspondence map at the pixel level. The transformations are finally composed to compute the map $\mathcal{M}$ and image X can be corrected according to eqns. (8-9). One of the examples previously used is reported in Figure 8 under the original illumination and under the *standard* one obtained with the described procedure.

4. Pose Estimation

In this section we present an algorithm for estimating the pose of a face, limited to in depth rotations. The knowledge of the pose can be of interest both for recognition systems, where an appropriate template can then be chosen to speed up the recognition process [14], and for model based coding systems, such as those which could be used for teleconferencing applications [16]. The idea underlying the proposed algorithm for pose estimation is that of quantifying the asymmetry between the aspect of the two eyes due to in-depth rotation and mapping the resulting value to the amount of rotation. It is possible to visually estimate the in-depth rotation even when the eyes are represented schematically such as in some cartoons characters where eyes are represented by small bars. This suggests that the relative amount of gradient intensity, along the mouth-forehead direction, in the regions corresponding to the left and right eye respectively provides enough information for estimating the in-depth rotation parameter.

The algorithm requires that the location of one of the eyes be approximately known as well as the direction of the interocular axis. Template matching techniques such as those outlined in Appendix B can be used to locate one of the eyes even under large left-right rotations and the direction of the interocular axis can be computed using the method reported in [17]. Let us assume for simplicity of notation that the interocular axis is horizontal. Using the projection techniques reported in [18], [5] we can approximately localize the region were both eyes are confined. For each pixel in the region the following map is computed:

$$V(x,y) = \begin{cases} |\partial_y C(x,y)| & \text{if } |\partial_y C(x,y)| \geq |\partial_x C(x,y)| \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $C(x,y)$ represent the *local contrast* map of the image computed according to eqn. (13) (see Appendix A). The resulting map assigns a positive value to pixels where the projection of gradient along the mouth-forehead direction dominates over the projection along the interocular axis. In order to estimate the asymmetry of the two eyes it is necessary to determine the regions corresponding to the left and right eye respectively. This can be done by computing the projection $P(x)$ of $V(x,y)$ on the horizontal axis given by the sum of the values in each of the columns. The analysis of the projections is simplified if they are smoothed: in our experiments a Gaussian smoother was used. The resulting projections, at different rotations are reported in Figure 9. These data are obtained by rotating the same 3D model used for the generation of the illumination examples: texture mapping techniques are then used to project a frontal view of a face onto the rotated head (see [13], [19] for alternative approaches to the estimation of pose and synthesis of non frontal views).

The figure clearly shows that the asymmetry of the two peaks increases with the amount of rotation. The asymmetry $\mathcal{U}$ can be quantified by the following quantity:

$$\mathcal{U} = \frac{\sum_{x<x_m} P(x) - \sum_{x>x_m} P(x)}{\sum_{x<x_m} P(x) + \sum_{x>x_m} P(x)} \tag{11}$$

where x_m is the coordinate of the minimum between the two peaks. The value of $\mathcal{U}$ as a function of the angle of rotation is reported in Figure 10. Using the approximate linear relation it is possible to quantify the rotation of a new image. The pose recovered by the described algorithm from several images is reported in Figure 11 where for each of the testing images a synthetic image with the corresponding pose is shown.

5. Conclusions

In this paper three problems related to the analysis of facial images have been addressed: the estimation of the illuminant direction, the compensation of illumination effects and, finally, the recovery of the pose of the face, restricted to left-right rotations. The solutions proposed for these problems rely on the use of computer graphics techniques to provide images of faces under different illumination and pose starting from a database of frontal views under frontal illumination. The algorithms trained using synthetic images have been successfully applied to real images.

Acknowledgements The author would like to thank A. Shashua and M. Buck for providing the optical flow code and help in using it. Special thanks to Prof. T. Poggio for many ideas, suggestions and discussions.

Appendix

A. Illumination sensitivity

A common measure of the similarity of visual patterns, represented as vectors or arrays of numbers, is the normalized cross correlation coefficient:

$$\rho_{xy} = \frac{\mu_{xy}}{\mu_{xx}\mu_{yy}} \tag{12}$$

where μ_{xy} represent the second order, centered moments.

The value of $|\rho_{xy}|$ is equal to 1 if the components of two vectors are the same modulo a linear transformation. While the invariance to linear transformation of the patterns is clearly a desirable property (automatic gain and black level adjustment of many cameras involve such a linear transformation) it is not sufficient to cope with the more general transformations implied by changes of the illumination sources. A common approach to the solution of this problem is to process the visual patterns before the estimation of similarity is done, in order to preserve the necessary information and eliminate the unwanted details. A common preprocessing operation is that of computing the intensity of the brightness gradient and use the resulting map for the comparison of the patterns. Another preprocessing operation is that of computing the *local contrast* of the image. A possible definition is the following:

$$C = \begin{cases} C' & \text{if } C' \leq 1 \\ 2 - \frac{1}{C'} & \text{if } C' > 1 \end{cases} \tag{13}$$

where

$$C' = \frac{I}{I * K_{G(\sigma)}} \tag{14}$$

and $K_{G(\sigma)}$ is a Gaussian kernel whose σ is related to the expected interocular distance. It is important to note that C saturates in region of high and low *local contrast* and is consequently less sensitive to noise.

Recently some claims have been made that the gradient direction field has good properties of invariance to changes in the illumination [20]. In the case of the direction field, where a vector is associated to each single pixel of the image, the similarity can be computed by measuring the alignment of the gradient vectors at each pixel. Let $\boldsymbol{g}_1(x,y)$ and $\boldsymbol{g}_2(x,y)$ be the gradient fields of the two images and $\|\cdot\|$ represent the usual vector norm. The global alignment can be defined by

$$\mathcal{A} = \frac{1}{\sum_{(x,y)} w(x,y)} \times \tag{15}$$
$$\times \sum_{\|\boldsymbol{g}_1(x,y)\|,\|\boldsymbol{g}_2(x,y)\|>0} w(x,y)\frac{\boldsymbol{g}_1(x,y)\cdot\boldsymbol{g}_2(x,y)}{\|\boldsymbol{g}_1(x,y)\|\|\boldsymbol{g}_2(x,y)\|}$$

where

$$w(x,y) = \frac{1}{2}(\|\boldsymbol{g}_1(x,y)\| + \|\boldsymbol{g}_2(x,y)\|) \tag{16}$$

The formula is very similar to the one used in [20] (a normalization factor has been added). The following preprocessing operators were compared using either the normalized cross-correlation-coefficient ρ or the alignment $\mathcal{A}$:

plain: the original brightness image convolved with a Gaussian kernel of width σ;

contrast: each pixel is represented by the *local image contrast* as given by eqn.(13);

gradient: each pixel is represented by the brightness gradient intensity computed after convolving the image with a Gaussian kernel of standard deviation σ:

$$\|\nabla(N_\sigma \star I(x,y))\| \tag{17}$$

gradient direction: each pixel is represented by the brightness gradient of $N_\sigma \star I(x,y)$. The similarity is estimated through the coefficient $\mathcal{A}$ of eqn.(16)

laplacian: each pixel is represented by the value of the laplacian operator applied to the intensity image convolved with a Gaussian kernel.

For each of the preprocessing operators, the similarity of the original image under (nearly) diffuse illumination to the synthetic images obtained through eqn. (4) was computed. The corresponding average values are reported in Figure 12 for different values of the parameter σ of the preprocessing operators. The *local contrast* operator turns out to be the less sensitive to variations in the illuminant direction. It is also worth mentioning that the minimal sensitivity is achieved for an intermediate value of σ: this should be compared to the monotonic behavior of the other operators. Further experiments with the template-based face recogntion system described in [5] have practically demonstrated the advantage of using the *local contrast* images for the face recognition task.

B. Alternative Template Matching

The correlation coefficient is quite sensitive to noise and alternative estimators of pattern similarity may be preferred [21], [22]. Such measures can be derived from distances other than the Euclidean, such as the L_1 norm defined by:

$$d_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{18}$$

where n is the dimension of the considered vectors. A similarity measure based on the L_1 norm can be introduced:

$$l(\mathbf{x}',\mathbf{y}') = \frac{1}{n}\sum_i \left(1 - \frac{|x'_i - y'_i|}{|x'_i| + |y'_i|}\right) \tag{19}$$

that satisfies the following relations:

$$\begin{aligned} l(\mathbf{x}',\mathbf{y}') &\in [0,1] \\ l(\mathbf{x}',\mathbf{y}') = 1 &\Leftrightarrow \mathbf{x}' = \mathbf{y}' \\ l(\mathbf{x}',\mathbf{y}') = 0 &\Leftrightarrow \mathbf{x}' = -\mathbf{y}' \end{aligned}$$

where $\mathbf{x}'$ and $\mathbf{y}'$ are normalized to have zero average and unit variance. The characteristics of this similarity measure are extensively discussed in [23] where it is shown that it is less sensitive to noise than ρ_{xy} and technically *robust* [24]. Hierarchical approaches to the computation of correlation, such as those proposed in [25] are readily extended to the use of this alternative coefficient.

The influence of template shape can be further reduced by slightly modifying $l(x, y)$. Let us assume that the template T and the corresponding image patch are normalized to zero average and unit variance. We denote by $\Omega_I(\mathbf{x})$ a the 4-connected neighborhood of point $\mathbf{x}$ in image I and $F_{\Omega_I(\mathbf{x})}(w)$ the intensity value in $\Omega_I(\mathbf{x})$ whose absolute difference from w is minimum: if two values qualify, their average (w) is returned. A modified $l(x, y)$ can then be introduced:

$$l'(\mathbf{y}) = \frac{1}{n} \sum_{\mathbf{x}} \left(1 - \frac{|F_{\Omega_I(\mathbf{x}+\mathbf{y})} - (T(\mathbf{x}))\, T(\mathbf{x})|}{|F_{\Omega_I(\mathbf{x}+\mathbf{y})}| + |(T(\mathbf{x}))\, T(\mathbf{x})|} \right) \quad (20)$$

The new coefficient introduces the possibility of *local deformation* in the computation of similarity (see also [26] for an alternative approach).

References

[1] A. P. Pentland. Local Shading Analysis. In *From Pixels to Predicates*, chapter 3. Ablex Publishing Corporation, 1986.

[2] A. Shashua. Illumination and View Position in 3D Visual Recognition. In *Advances in Neural Information Processing Systems 4*, pages 572–577. Morgan Kaufmann, 1992.

[3] P. W. Hallinan. A Low-Dimensional Representation of Human Faces For Arbitrary Lighting Conditions. Technical Report 93-6, Harvard Robotics Lab, December 1993.

[4] A. Shashua. On Photometric Issues in 3D Visual Recognition From A Single 2D Image. *International Journal of Computer Vision*, 1994. to appear.

[5] R. Brunelli and T. Poggio. Face Recognition: Features versus Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[6] J-P. Thirion. Realistic 3d simulation of shapes and shadows for image processing. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 54(1):82–90, 1992.

[7] T. Poggio and F. Girosi. Networks for Approximation and Learning. In *Proc. of the IEEE, Vol. 78*, pages 1481–1497, 1990.

[8] R. Brunelli and G. Tecchiolli. Stochastic minimization with adaptive memory. Technical Report 9211-14, I.R.S.T, 1992. To appear on *Journal of Computational and Applied Mathematics*.

[9] T. Poggio and R. Brunelli. A Novel Approach to Graphics. A.I. Memo No. 1354, Massachusetts Institute of Technology, 1992.

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Morgan-Kauffman, editor, *Proc. IJCAI*, 1981.

[11] J. R. Bergen and R. Hingorani. Hierarchical, computationally efficient motion estimation algorithm. *Journal of The Optical Society of America*, 4:35, 1987.

[12] J. R. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center, 1990.

[13] D. J. Beymer, A. Shashua, and T. Poggio. Example Based Image Analysis and Synthesis. A.I. Memo No. 1431, Massachusetts Institute of Technology, 1993.

[14] David J. Beymer. Face Recognition under Varying Pose. A.I. Memo No. 1461, Massachusetts Institute of Technology, 1993.

[15] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, NJ, 1982.

[16] K. Aizawa, H. Harashima, and T. Saito. Model-based analysis synthesis image coding (mbasic) system for a person's face. *Signal Processing Image Communication*, 1:139–152, 1989.

[17] W. T. Freeman and Edward H. Adelson. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[18] R. Brunelli. Edge projections for facial feature extraction. Technical Report 9009-12, I.R.S.T, 1990.

[19] A. Shashua and S. Toelg. The Quadric Reference Surface: Applications in Registering Views of Complex 3D Objects. Technical Report CAR-TR-702, Center for Automation Research, University of Maryland, 1994.

[20] Martin Bichsel. *Strategies of Robust Object Recognition for the Identification of Human Faces*. PhD thesis, Eidgenossischen Technischen Hochschule, Zurich, 1991.

[21] M. Boninsegna and M. Rossi. Similarity measures in Computer Vision. Technical Report 9312-06, IRST, 1993. to appear on Pattern Recognition Letters.

[22] R. Brunelli and S. Messelodi. Robust Estimation of Correlation: an Application to Computer Vision. Technical Report #9310-05, I.R.S.T, 1993. to appear on Pattern Recognition.

[23] R. Brunelli and S. Messelodi. Robust Estimation of Correlation: an Application to Computer Vision. Technical Report 9310-05, I.R.S.T, 1993. to appear on *Pattern Recognition*.

[24] P. J. Huber. *Robust Statistics*. Wiley, 1981.

[25] P. J. Burt. Smart sensing within a pyramid vision machine. *Proceedings of the IEEE*, 76(8):1006–1015, 1988.

[26] Alan L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70, 1991.

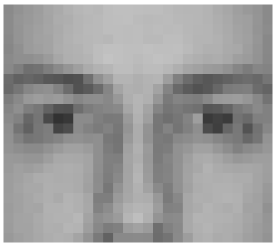

Fig. 1. The facial region used to estimate the direction of illuminant. Four intensity values are derived by computing a weighted average, with Gaussian weights, of the intensity over the left (right) cheek and left (right) forehead-eye regions.

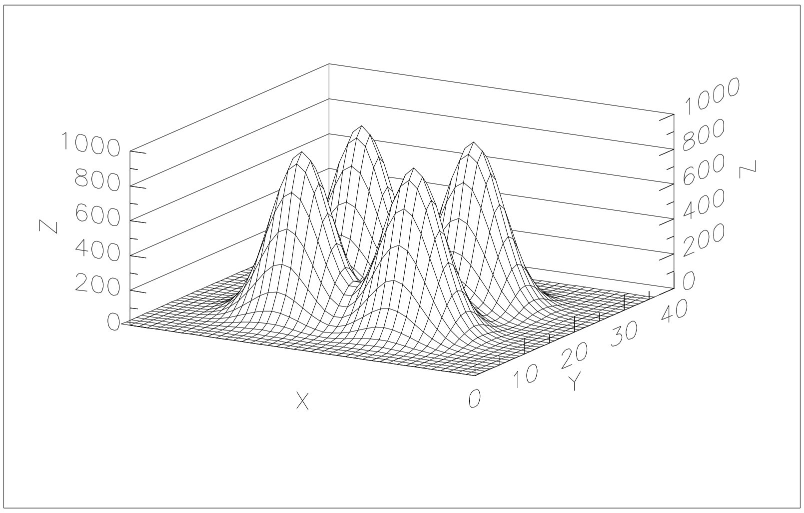

Fig. 2. Superimposed Gaussian receptive fields giving the four dimensional input of the HyperBF network. Each field computes a weighted average of the intensity. The coordinates of the plot represent the image plane coordinates of Figure 1.

Fig. 3. Computer generated images (left) are used to modulate the intensity of a single view under approximately diffuse illumination (center) to produce images illuminated from different angles (right). The central images are obtained by replication of a single view. The right images are obtained by multiplication of the central and left images (see text for a more detailed description).

Fig. 4. Error made by a 4 units HyperBF network on estimating the illuminant direction on the 169 images of the training set. The horizontal axis represents the left-right position of the illuminant while the vertical axis represents its height. The intensity of the squares is proportional to the squared error: the lighter the square, the greater the error is.

Fig. 5. Some real images on which the algorithm trained on the synthetic examples has been applied.

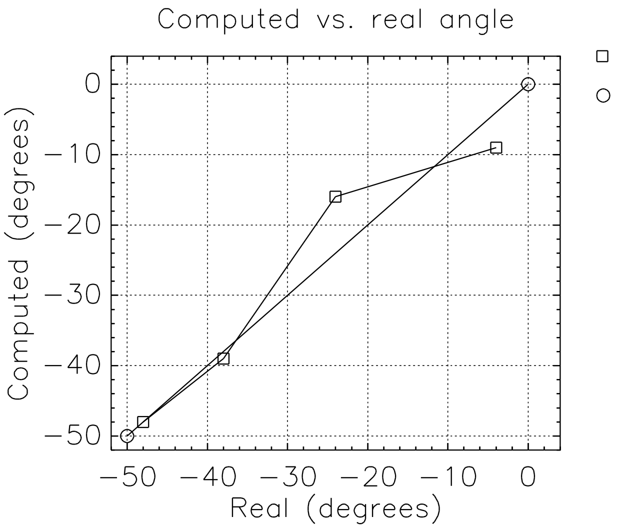

Fig. 6. Illuminant direction as estimated by the HyperBF network compared to the real data for the four test images.

Fig. 7. The first three images represent respectively the original image, the image obtained by fixing the nose and mouth position to that of the reference image (the last in the row) and the refined warped image obtained using a hierarchical optical flow algorithm.

Fig. 8. The original image (left) and the image corrected using the procedure described in the text.

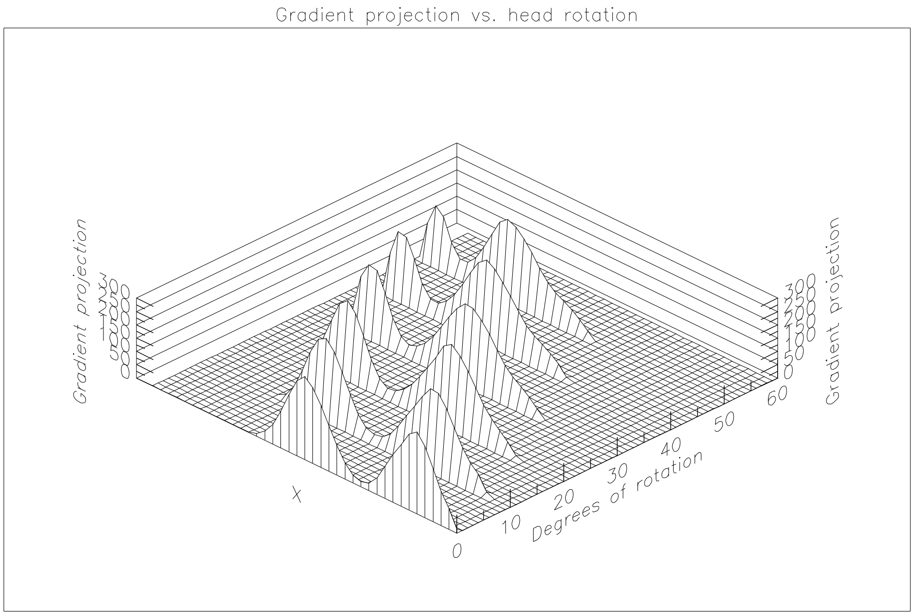

Fig. 9. The drawing reports the dependence of the gradient projection on the degrees of rotation around the vertical image axis. The projections are smoothed using a Gaussian kernel of $\sigma = 5$. Note the increasing asymmetry of the two peaks.

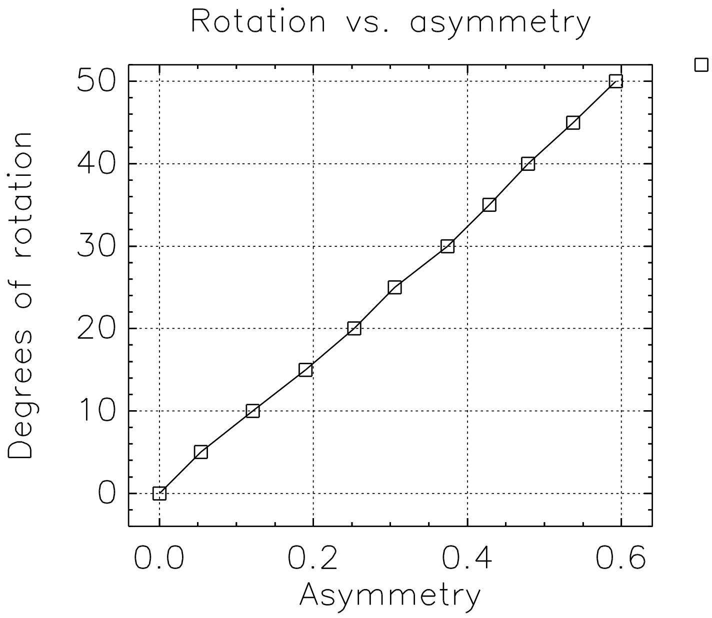

Fig. 10. The drawing reports the dependence of the asymmetry of the projection peaks on the degrees of rotation around the vertical image axis. The values are computed by averaging the data from three different people.

Fig. 11. The top row reports the test images while the bottom row shows the images generated using a simple 3D model and the rotation estimated using the approximately linear dependence of the gradient projection asymmetry on the rotation around the vertical image axis

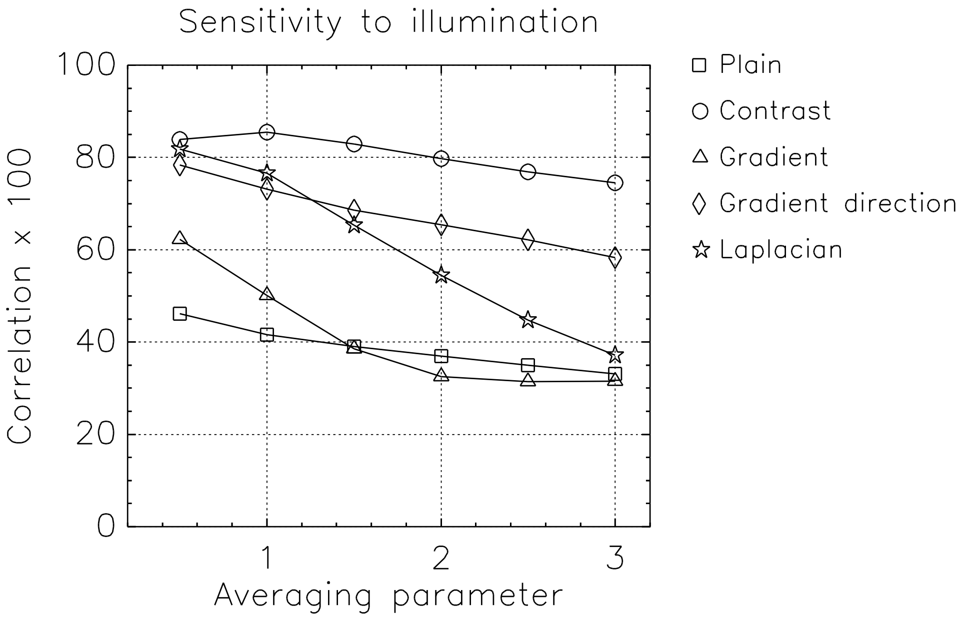

Fig. 12. Sensitivity to illumination of some common preprocessing operators. The horizontal axis represents the values of σ (see text for an explanation).