Head Pose And Light Source Estimation On Low Resolution Facial Images Using A Texture Based Approach

N.B. Kanters
Student Bachelor Electrical Engineering
University of Twente, Enschede, The Netherlands

***Abstract*—Face recognition is a biometric technique with the potential to identify non cooperating human beings in uncontrolled environments. Illumination conditions and head poses are unknown in these situations, causing the performance of state of the art face recognition techniques to drop significantly. An extensively investigated solution for this problem is to transform non frontal facial images into images with a frontal pose, increasing the performance of these FR algorithms. This requires accurate estimation of the head pose.**

This paper is about a head pose estimation algorithm based on a mathematical model of the human nose. Pixel intensity values, i.e. texture, of the nose region are calculated based on this model. The camera and the light source are modeled at various positions, resulting in a set of pixel vectors. Pixels in the nose region of geometrically normalized probe images are compared with these pixel vectors, after which the head pose and the light source position are estimated. Two different error measures were used. The algorithm shows promising results for head poses within the range of ±15° relative to the frontal view, but performs disappointingly for larger angles. Several improvements are suggested.

***Index Terms*—Face recognition, head pose estimation, Lambertian reflection**

I. Introduction

Face recognition has been an important research topic within the field of biometrics during the last decades. Contrary to techniques as iris and fingerprint scanning, face recognition has the potential to be successful without cooperative subjects and in uncontrolled environments. This is also called face recognition in the wild.

The most well-known application of face recognition in the wild is the recognition of criminals filmed by surveillance cameras. Images captured by these cameras suffer from varying illumination conditions and head poses, which complicates the recognition [1]. Another characteristic of images captured by surveillance cameras is their low resolution. Faces of persons to be recognized frequently consist of less than 32×32 pixels, which is why it is considered a Low Resolution Face Recognition (LR FR) problem [2].

Being able to estimate the head pose unlocks to the possibility of compensation or normalization. Viewpoint transformed FR approaches, for example, use estimated pose parameters to warp non-frontal probe images into a pose similar to the pose of a gallery image [3, 4].

In this paper, head pose estimation on LR facial images is discussed. Using a 2D model of the human nose, pixel intensity values are calculated for various head poses and (point) light source positions. The last is done in order to account for differences in illumination among probe images. The texture of the nose region of a probe image is compared with these synthesized pixels, after which the actual estimation of both head pose and light source position (abbreviated as light source estimation) is performed. This is done using a winner-takes-it-all approach. In order to simplify the problem, only rotations between ±30° around the vertical axis, yaw, are considered. Figure 1 shows which rotations determine the pose of a human head, or any other object, in real life. The research question on which this paper is based reads:

Is it possible to correctly estimate the head pose of low resolution facial images with varying illumination conditions by only using a mathematical model of the human nose?

The structure of this paper is as follows. Related research on (LR) head pose estimation is discussed in section II. The working principle of the proposed algorithm is explained in section III. This is followed by sections on the conducted experiments (IV) and their results (V). An answer on the research question is given in section VI, as well as recommendations for further research.

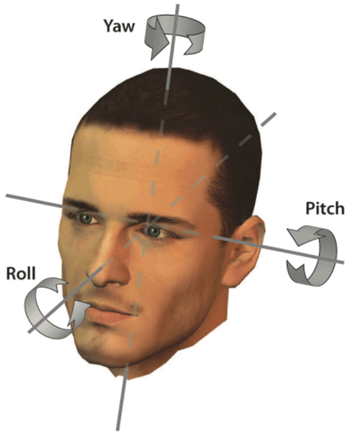


Fig. 1: The pose of a head is determined by rotations around three axes[5]

II. Related Research

A. Head Pose Estimation

Head pose estimation has been investigated extensively during the last decades. E. Murphy-Chutorian and M.M. Trivedi categorized current head pose estimation methods in 8 categories based on their fundamental approach [5]. The method most related to the one described in this paper is the 'Appearance Template Method'. X. Zhang *et al.* subdivided this category in 2 subcategories: holistic approaches and local approaches [1]. In holistic approaches, a probe image is compared with a set of gallery images, each labeled with a pose. The pose of the probe image is then assumed to be the same as the pose of the most similar gallery image. Misalignment of the facial images has a big influence on the outcome of the pose estimation, though. Local approaches like the algorithm proposed in this paper, consider small sub-regions of the face, instead of the entire face image. A disadvantage of both approaches is the need for a large number of gallery images.

Geometric approaches are the most intuitive. After detection of facial landmarks such as the eyes, nose and mouth, their positions are used to do an estimate on the head pose. An example is the work of A. Gee and R. Cipolla, who tried to estimate the pose by considering the location of the nose tip relative to the symmetry axis of the face [6]. The disadvantage of this, and other geometrical methods, is that it is very difficult to apply them to low resolution images.

J. Chen *et al.* used a 'Nonlinear Regression Method' to estimate the head pose of images of 10×10, 5×5 and 3×3 pixels [7]. After the extraction of HoG features from the facial images, SVR was applied. The 'mean ± standard deviation' of the absolute error for yaw was $9.9 \pm 12.4°$ in the 10×10 case.

Another frequently used approach is the implementation of Neural Networks. N. Gourier *et al.* trained an auto-associative network for each pose in a discrete set of poses [8]. The pose of a probe image was estimated by selecting the network with the highest score. Yaw was estimated correctly in 61.3% of the cases, and in 90% of the cases with a precision of 15°.

B. Image Transformation

Although not considered in this paper, some research on image transformations is presented as well. The main reason for head pose estimation is the transformation of non frontal images into a frontal view, so that already existing FR algorithms can be used. The work of C. Sanderson *et al.* is one of the many examples of such a transformation [9]. J.Y. Guillemaut *et al.* presented a complete face recognition system, including pose correction using an AAM based approach [10].

III. Algorithm Working Principle

A schematic overview of the proposed head pose estimation algorithm is depicted in figure 2. The first step is the derivation of a model of the human nose. This model is used to compose a set of light intensity profiles. These profiles are dependent on 2 parameters: head pose and illumination. Each intensity profile is then transformed into a set of 5 'pixel vectors': vectors containing the intensity values of a certain number of pixels. The texture extracted from the nose region of the probe image is compared with these pixel vectors. The parameter values belonging to the model-based vector most similar to the probe image texture are assigned to the probe image. These steps are explained in more detail in the following subsections.

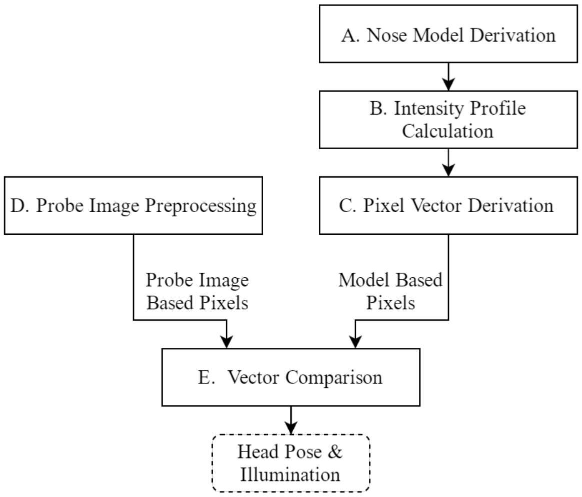
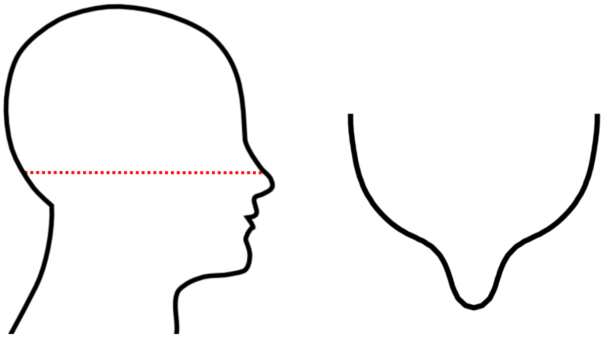


Fig. 2: Schematic overview of the head pose estimation algorithm

A. Nose Model Derivation

The 2D nose model was derived from a 3D model of a human head from the PV3D database. MeshLab was used to take a horizontal cross section at nose height (figure 3a). Only the frontal part of the head, i.e. the face, was used, since this part is relatively identical for most human beings, while the back of the head is mostly covered with hair. The useful part of the cross section is shown in 3b.

MATLAB R2015b was used for further processing of the model. A polynomial fit was performed on the data points of the cross section. The polynomial allows for derivative calculation at every desired point, which is necessary in order to find the points that have a direct line of sight with the camera and/or light source. A 14th order polynomial was considered to match the data well enough.



(a) Cross section position [11] (b) Cross section

Fig. 3: Cross Section of 3D model

B. Intensity Profile Calculation

The probe image characteristics to be estimated are head pose and light source position, meaning that these should both be included in the model. Intensity profiles were calculated for various values of these parameters, using the following assumptions:

1) The head is illuminated by a single point light source.
2) The light source is located infinitely far away from the head, meaning the light is collimated.
3) Light reflects diffusely on the skin of a human head.
4) The diffuse reflectivity is constant for every point on the skin located at the height of the cross section.

To simplify the calculations, the configuration of the nose model was fixed while the the camera and the light source were modeled at various positions, indicated by θ_C and θ_L respectively. For both parameters, a rotation of 0 degrees corresponds to a frontal view. Counterclockwise rotations were considered positive. This is visualized in figure 4.

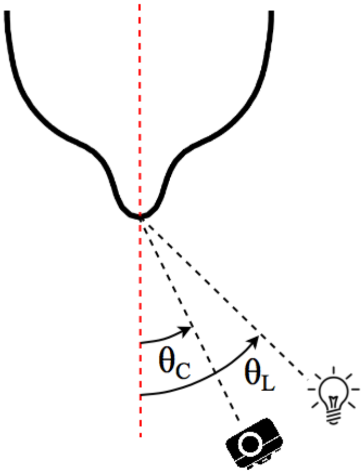


Fig. 4: Definitions of camera and light source angles

The intensity of the diffuse reflected light seen by the camera is given by the Lambertian reflectance model:

$$I_d = r_d I_s(\vec{l} \cdot \vec{s_n}) \tag{1}$$

where r_d is the diffuse reflection coefficient and I_s is the intensity of the point light source. $\vec{l}$ is the normalized vector pointing from a point on the surface to the point light source. $\vec{s_n}$ is the normalized vector normal to the surface. In a 2D case like this, the surface is replaced by a curve: the cross-section of the nose. I_d was normalized by setting both r_d and I_s to 1. The intensity profile $I_d(x)$ is obtained by calculating I_d for every point of the cross section and setting it to zero for points which cannot be reached by the light because of self-occlusion by the nose or cheek:

$$I_d(x) = r_d I_s(\vec{l}(x) \cdot \vec{s_n}(x)) \tag{2}$$

Figure 5 shows the influences of θ_C and θ_L on the intensity profile. The figure shows the profiles for angles of $-30°$, $0°$ and $30°$ for both parameters.

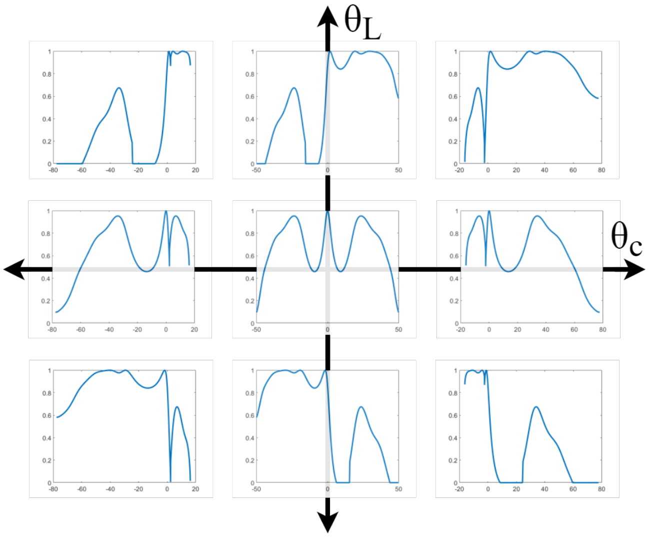


Fig. 5: Influences of θ_C and θ_L on the intensity profiles

C. Pixel Value Derivation

The next step of the algorithm is to convert the intensity profiles to sets of pixel vectors. To achieve a reliable comparison, the number of model-based pixels representing the nose region, $N_p \,\epsilon\, \mathbb{N}_1$, should be equal to the number of pixels representing the nose region in the probe image. The pixel intensities are calculated as in equations 3 to 7.

$$S = \frac{max(x) - min(x)}{N_p + 1} \tag{3}$$

$$i \in \{1, 2, 3, \ldots, N_p - 1, N_p\} \tag{4}$$

$$j \in \{1, 2, 3, 4, 5\} \tag{5}$$

$$l_{i,j} = min(x) + (i-1)S + \frac{(j-1)S}{4} \tag{6}$$

$$P_{j,i} = \frac{1}{S}\int_{l_{i,j}}^{l_{i,j}+S} I_d(x)dx \tag{7}$$

S is the width of the segments of $I_d(x)$. The average of $I_d(x)$ within one segment, calculated with the integral in equation 7, determines the intensity of 1 pixel. Since this integral cannot be calculated analytically, it was implemented as a summation. This causes small errors, but these are negligible, since the resolution of $I_d(x)$ was such that every pixel value was determined by at least 10 points. The parameter i is used to loop through the entire intensity profile. The shift introduced by $(j-1)S/4$ accounts for possible sub-pixel misalignments: the nose tip of a face in a probe image could either be captured in 1 or in 2 pixels, resulting in different pixel patterns. It was therefore decided to calculate the pixel intensities for 5 different start positions ranging from $min(x)$ to $min(x) + S$ in steps of $S/4$.

The result of these equations, P, is a $5 \times N_p$ matrix of which every row represents a pixel vector. Figure 6 shows the intensity profile for $\theta_C = \theta_L = 0°$ and its corresponding gray scale pixels for $N_p = 18$. The grid in the graph shows the intervals of for $j = 3$.

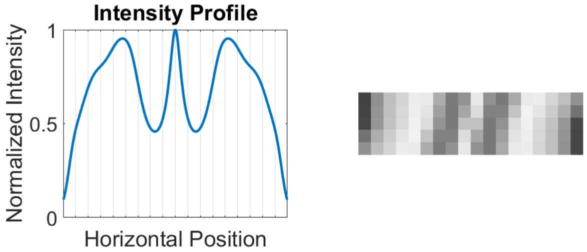


Fig. 6: Intensity profile and its corresponding gray scale pixels

Equations 8 and 9 show the separation of P in the 5 pixel vectors $\boldsymbol{r}_1$ upto $\boldsymbol{r}_5$. These row vectors are put in a column vector V. Calculating V for m values of θ_L and n values of θ_C results in a matrix U_{mb} containing all the model based pixel vectors, as shown in equation 10.

$$\boldsymbol{r}_\gamma = \{P_{\gamma,1}, P_{\gamma,2}, \ldots, P_{\gamma,N_p}\} \tag{8}$$

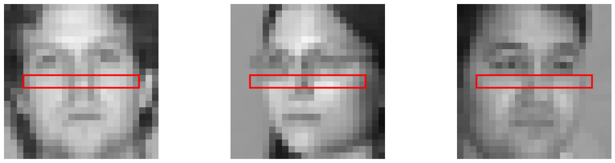
$$V = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_5 \end{bmatrix} \tag{9}$$

$$U_{mb} = \begin{array}{c} \\ \theta_{L_1} \\ \theta_{L_2} \\ \vdots \\ \theta_{L_m} \end{array} \begin{array}{c} \begin{array}{cccc} \theta_{C_1} & \theta_{C_2} & \ldots & \theta_{C_n} \end{array} \\ \begin{bmatrix} V_{1,1} & V_{1,2} & \ldots & V_{1,n} \\ V_{2,1} & V_{2,2} & \ldots & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m,1} & V_{m,2} & \ldots & V_{m,n} \end{bmatrix} \end{array} \tag{10}$$

D. Probe Image Preprocessing

In order to do an accurate estimation on the head pose and the light source position, the useful information, i.e. pixels representing the surroundings of the nose, has to be extracted from the probe images. Only faces successfully detected by a face detector were considered, meaning that the position of the nose was the same for all images. Figure 7 shows 3 examples of facial images[1] after face detection. The Region Of Interest is marked by the red rectangles. Its coordinates were determined heuristically to be as in equation 11, where x_{tl} and y_{tl} are the coordinates of the top left corner, and w and h the width and height respectively. W_{pi} and H_{pi} are measures for the resolution of the probe image. All these values are rounded to their nearest integer. The pixels in this region are averaged along the vertical dimension, resulting in a $1 \times w$ vector, called $\boldsymbol{v_{pi}}$.

$$[x_{tl}, y_{tl}, w, h] = \left[\frac{W_{pi}}{8}, \frac{11}{24}H_{pi}, \frac{3}{4}W_{pi}, \frac{1}{12}H_{pi}\right] \tag{11}$$



(a) $\theta_C = 0°$ (b) $\theta_C = 30°$ (c) $\theta_C = -15°$

Fig. 7: Region Of Interest for pose estimation

It can be seen that the circumvented pixels of the non-frontal images not only represent a face, but also a part of the background. This is harmful for the vector comparison, since the background is not included in the nose model. This could have been solved by decreasing the Region Of Interest, but this would result in a loss of useful information. It was therefore decided to account for it by making N_p dependent on θ_c, as shown in equation 12. This equation was determined heuristically.

$$N_p = round\left(\left(1 - \frac{|\theta_C|}{135}\right) \times w\right) \tag{12}$$

E. Vector Comparison

The final step of the algorithm is the actual head pose and light source estimation of a certain probe image. $\boldsymbol{v_{pi}}$ is compared with every vector in U_{mb} and the parameter values belonging to the vector with smallest error are assigned to the probe image. To prevent vector dimension mismatches, only the first N_p pixels of $\boldsymbol{v_{pi}}$ are compared with model based vectors belonging to values of $\theta_C < 0$, while only the last N_p pixels are used for values of $\theta_C > 0$. The vector containing the N_p useful pixels of $\boldsymbol{v}_{pi}$ is called $\boldsymbol{v}_u$.

To decrease the influence of brightness differences between the model based pixel vectors and $\boldsymbol{v}_u$, the model based vectors were adapted using the approach of G. Li *et al.* [12], which is shown in equation 13.

$$\boldsymbol{r}_\gamma = \frac{\sigma_{\boldsymbol{v}_u}}{\sigma_{\boldsymbol{r}_\gamma}}(\boldsymbol{r}_\gamma - \mu_{\boldsymbol{r}_\gamma}) + \mu_{\boldsymbol{v}_u} \tag{13}$$

$\mu_{\boldsymbol{r}_\gamma}$ and $\sigma_{\boldsymbol{r}_\gamma}$ are the mean and the standard deviation of the of the pixel values in $\boldsymbol{r}_\gamma$ respectively. $\mu_{\boldsymbol{v}_u}$ and $\sigma_{\boldsymbol{v}_u}$ are the mean and standard deviation of the pixels in $\boldsymbol{v}_u$.

The head pose estimation is a winner-takes-it-all process: the pixel vector with the lowest error 'wins'. Two error measures were used:

1) Euclidean Distance (ED)

$$E_{ed} = \frac{1}{N_p}\sum_{n=1}^{N_p}(\boldsymbol{r}_\gamma(n) - \boldsymbol{v}_u(n))^2 \tag{14}$$

2) Normalized Cross-Correlation (NCC)

$$E_{ncc} = \frac{1}{N_p}\sum_{n=1}^{N_p}\frac{(\boldsymbol{r}_\gamma(n) - \mu_{\boldsymbol{r}_\gamma})(\boldsymbol{v}_u(n) - \mu_{\boldsymbol{v}_u})}{\sigma_{\boldsymbol{v}_u}\sigma_{\boldsymbol{r}_\gamma}} \tag{15}$$

The normalized cross correlation is defined in such a way that the normalization of equation 13 has no effect, since the pixels are divided by their mean and standard deviation.

[1] Original images were taken from CMU Multi-PIE database

It is important to note that the Euclidean distance gives lower values for more similar vectors, while the normalized cross-correlation gives values between -1 and 1, where a higher value indicates more similarities. The benefit of the second method is that it is less sensitive to the exact pixel intensity values than the Euclidean distance, but more to the pixel patterns. This could appear to be a disadvantage as well, since an outlier in $\boldsymbol{v}_u$ messing up the pattern, causes the score to be unreliable.

This last part of the algorithm was implemented in two steps. After the errors of all vectors were calculated, the vector with the smallest error is taken out for each V in U_{mb}, so that each (θ_L, θ_C) coordinate contained 1 pixel vector. Out of these $m \times n$ vectors, the lowest one was chosen.

IV. Experiments

A. Database

Validation of the algorithm was done using facial images of the CMU Multi PIE database. This database contains more than 750.000 facial images of 337 different people under varying Poses, Illumination conditions and with different Expressions [13]. The images were taken using 15 different cameras of which 13 at head height. Both camera and flash light angles vary from $-90°$ to $90°$ relative to the frontal view, with intervals of $15°$. The images have an RGB format with a resolution of 640×480 pixels. To keep the experiments manageable, a small database of 1000 images was composed, containing 40 persons for each combination of 5 different poses and 5 different illumination conditions (both camera and flashlights located between $[-30°, 30°]$, with intervals of $15°$).

In order to simulate images of surveillance cameras, the images were preprocessed using the following steps. The result is shown in figure 8.

1) Face detection using the Viola and Jones object detector, built-in in the Computer Vision System Toolbox for MATLAB versions newer than R2012a
2) Cropping the image to the detected face region
3) Downsampling to a resolution of 24×24 pixels
4) RGB to gray scale conversion

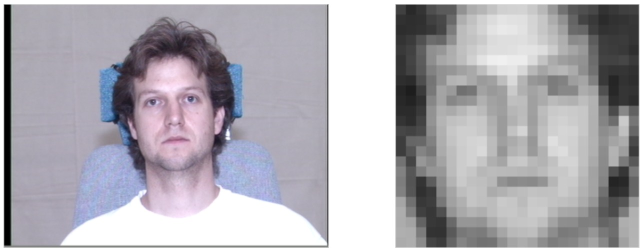


Fig. 8: Original and preprocessed face image

B. Setup

Table I gives an overview of the experiments that where conducted. The number of vectors in U_{mb} was varied by changing the range of θ_L. In experiment 2, the light source angle is used as an input parameter. Its value was set equal to the angle of the flashlight F°_{pi} in the probe image. In real life situations, especially outdoors and during daytime, the main light source is the sun, of which the position is known. Using a variable light source position in these cases is unnecessary and complicates the head pose estimation. The performance of both error measures, Euclidean Distance and Normalized Cross Correlation, is compared for both experiments.

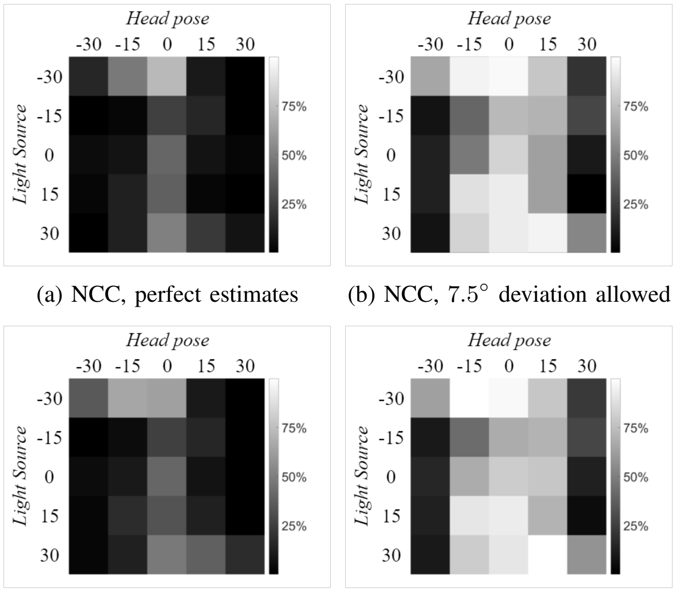
TABLE I: Overview of experiments

	θ_C		θ_L	
	Range	Interval	Range	Interval
1	$[-30°, 30°]$	$7.5°$	$[-30°, 30°]$	$7.5°$
2	$[-30°, 30°]$	$7.5°$	$[F^\circ_{pi}]$	-

V. Results & Discussion

A. Experiment 1

The results of experiment 1 are shown in figure 9. The percentage of correctly estimated head poses is shown for every probe image parameter combination (light source position and head pose). A brighter pixel represents a higher percentage, as visualized by the color bars. Figures 9a and 9b show results for the NCC method, while figure 9c and 9d show those for the Euclidean Distance. For figures 9a and 9c, the estimation was considered correct if the (θ_L, θ_C) coordinate of the lowest error corresponds exactly with the head pose and flash light characteristics of the probe image. For figures 9b and 9d, the (θ_L, θ_C) coordinate of the lowest error was allowed to be one interval off, in this case $7.5°$, for one or both of the characteristics.



(a) NCC, perfect estimates

(b) NCC, $7.5°$ deviation allowed

(c) ED, perfect estimates

(d) ED, $7.5°$ deviation allowed

Fig. 9: Algorithm performance, $7.5°$ intervals for θ_C and θ_L

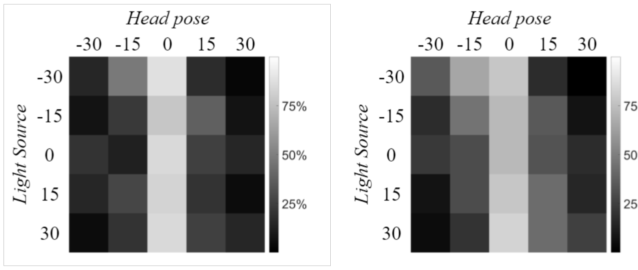
It can be seen that there is no significant difference between the two error measures. Besides that, it is clear that, assuming a deviation of 7.5° is acceptable, frontal images are estimated correctly in at least 70 % of the cases, with some positive outliers up to 100% in case of the ED method. In general, the performance drops for 'larger' head poses, i.e. larger angles with respect to the frontal view. The extremities of the diagonal are positive exceptions, since 50% to 65% is estimated exactly correct or with a deviation of 1 interval. This is most likely caused by the fact that the camera and the light source are modeled at the same positions for these coordinates, meaning that parts which cannot be reached by the light, and therefore result in dark pixels, are not captured by the camera. The effect of mistakes caused by the point light source assumption are therefore minimized.

Another observation is that all images in figure 9 are 180° point symmetric in their center. This was expected, since it means that the algorithm performance is equal for images with, for example, a head pose of 15° and a flash light angle of $-30°$ and images with a head pose of $-15°$ and a flash light angle of 30°.

Finally, when averaging over all light source conditions, it can be said that results are promising for head poses between $-15°$ and 15°. For larger angles, the performance drops significantly. Images with a smaller head pose resolution are required to determine these limits more precisely. These are not in the CMU Multi-PIE database.

B. Experiment 2

In the second experiment, the flash light angle was used as an input parameter for θ_L. Intervals for θ_C were the same as in experiment 1: 7.5°. Figure 10 shows the results for both error measures. Only estimates exactly equal to the actual probe image characteristics are considered correct.



(a) NCC, perfect estimates (b) ED, perfect estimates

Fig. 10: Algorithm performance, $\theta_L = F_{pi}^{\circ}$

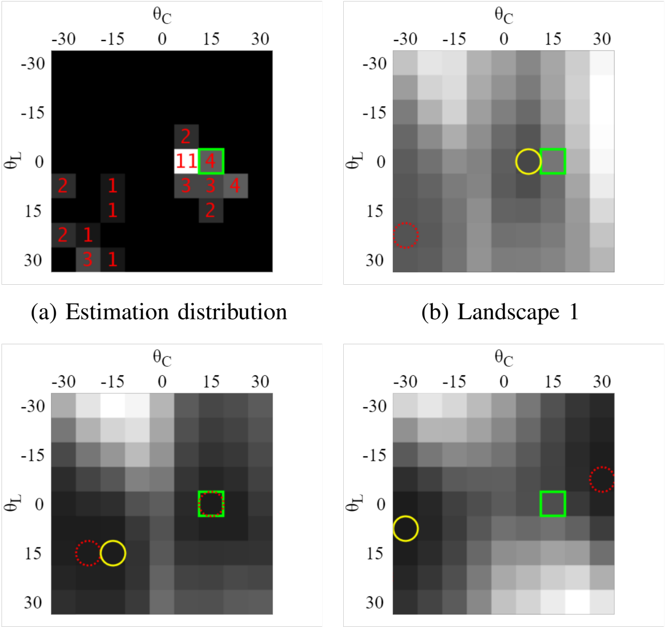
Frontal poses are estimated correctly in 77.5% to 87.5% of the cases when using the NCC error measure, and between 72.5% and 82.5% for the ED method. However, only 40 images per parameter combination were used. By repeating the experiment with more images, it can be verified whether the NCC method indeed outperforms the ED error measure.

The extremities of the antidiagonals show that the estimates for images with the corresponding characteristics are almost always incorrect. Performance figures for these images range between 0% and 5%. Since there were only 9 model based vectors to choose from for each image, 11% of the estimates would be correct in case the estimation would be performed randomly. This raises the expectation that the estimates follow a certain, incorrect, pattern. In the next subsection, the errors are analyzed in more detail.

C. Error analysis

Based on the experiments it can be said that the performance of the algorithm is disappointing, especially for angles larger than 15°. It was taken into account that the assumptions listed in section III-B would cause small errors, as well as the polynomial fit and the fact that the nose model was derived from another model. This subsection is aimed at showing that these are not the main sources for the low performance.

In order to determine whether there is a pattern in the wrong estimates, all the estimates belonging to one specific probe image parameter combination were considered. Figure 11 shows the estimation distribution belonging to images with a head pose of 15° and a flash light angle of 0°, as well as three representative error landscapes for images photographed under these conditions. The desired coordinate is marked with a green rectangle. The red numbers in the estimation distribution (figure 11a) sum up to 40. Each number shows the number of times that specific coordinate was estimated. The intensities of the underlying pixels serve as a clarification. The higher the number, the brighter the pixel. For the three error landscapes, brighter pixels represent a larger error, calculated using the Euclidean Distance. The red dotted circles indicate local minima in the error landscapes. The yellow circles show the lowest error of the entire landscape, and thus the estimated coordinate.



(a) Estimation distribution (b) Landscape 1

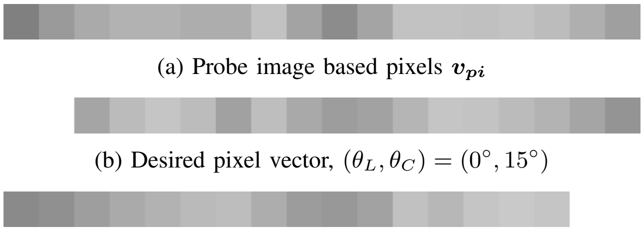
(c) Landscape 2 (d) Landscape 3

Fig. 11: Estimation distribution and three representative error landscapes. Desired coordinate $(\theta_L, \theta_C) = (0°, 15°)$

The error landscapes in figures 11b upto 11d all contain 2 local minima. Landscape 1 can be seen as the desired case:

There is a local minimum next to the desired coordinate, and, although there is another one, this minimum is also the global minimum, meaning the estimation was correct, assuming a deviation of 7.5° is allowed. The positions of the minima in landscape 2 are close to those of landscape 1. The 'correct minimum' is located at exactly the desired coordinate. This minimum is not the global minimum, though, meaning the estimation was incorrect. Landscape 3 again contains 2 minima, but both located at undesired coordinates. The estimation distribution shows that these 3 landscapes are indeed representative for all images with these characteristics. 27 out of the 40 estimates are located on or next to the desired coordinate. In most of the other cases, the global minimum appeared to be at, or close to, the positions of the undesired minima present in landscapes 1 and 2.

Figure 12 shows the pixel vectors corresponding to the landscape of figure 11c. There might be a lot to win, performance-wise, in cases like these. Figure 12a shows the probe image based pixels. The 2 left-most pixels are not part of the person's face, but of the background. Figure 12b shows the pixels corresponding to the desired coordinate. The pixels belonging to the global minimum are shown in figure 12c. The vectors are aligned in such a way that pixels that are compared with each other are all on the same vertical line.



(a) Probe image based pixels $\boldsymbol{v_{pi}}$

(b) Desired pixel vector, $(\theta_L, \theta_C) = (0°, 15°)$

(c) Lowest error pixel vector $(\theta_L, \theta_C) = (15°, -15°)$

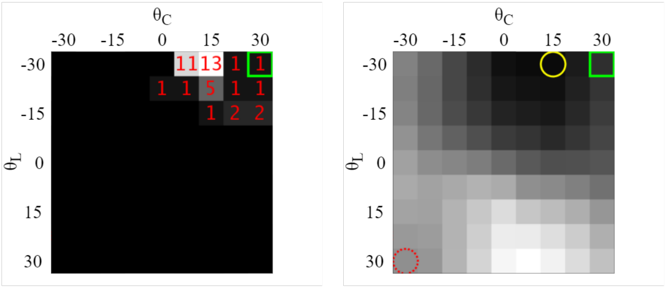
Fig. 12: Pixel vectors, desired $(\theta_L, \theta_C) = (0°, 15°)$

Both vectors in figures 12b and 12c show many similarities to the one in figure 12a, at least to the human eye. This is confirmed by their errors. The error of the desired pixel vector is only 0.7% higher than the one of the pixel vector with the lowest error, which causes the estimation to be incorrect.

Two causes for this mistake can be derived from these images. The most important one is the fact that a part of the probe image vector represents background instead of facial texture. These pixels are relatively dark with respect to the rest. Vectors corresponding to coordinates in the down-left corner appear to have this pattern as well, but these vectors are purely based on the nose model. By decreasing the width of the ROI, as shown in equation 11, in a way that only facial information is extracted from the probe images, this problem could be solved. The width of the nose model should be decreased accordingly.

The second cause of the mistakes is the scaling of the pixels. The width of the segments of the intensity profile which determine the pixel values is currently constant as shown in 3. This causes a problem, since the dark region in the middle of the desired pixel vector seems to be a little bit 'too wide'. This could be solved by making S dependent of x.

The same kind of images are shown in figures 13 and 14. The desired coordinate for these images was $(\theta_L, \theta_C) = (-30°, 30°)$. Figure 13a shows that all 40 estimates are in the correct corner of the coordinate system, but that the θ_C component of the coordinate of the global maximum is 15° too low. Figure 13b shows a typical example of the error landscapes of the images with these characteristics.

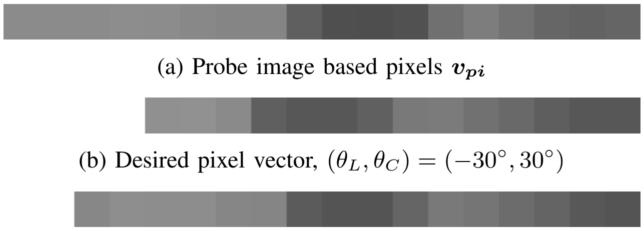


(a) Estimation distribution

(b) Error Landscape

Fig. 13: Estimation distribution and a representative error landscape. Desired coordinate $(\theta_L, \theta_C) = (-30°, 30°)$

The vectors in figure 14 show that the scaling, i.e. the value of N_p, is the main problem. The pattern of both vectors in figures 14b and 14c correspond quite well with the pattern of the vector in figure 14a. By squeezing the desired vector to 13 instead of 14 pixels and keeping the right-most pixels aligned, the matching of the patterns is improved. However, the results of both experiments show that the scaling in its current form results in an acceptable performance for frontal head poses. It can therefore be concluded that equation 12, which is used to calculated N_p, needs to be changed, most likely into a non-linear equation. The exact relation requires further research.



(a) Probe image based pixels $\boldsymbol{v_{pi}}$

(b) Desired pixel vector, $(\theta_L, \theta_C) = (-30°, 30°)$

(c) Lowest error pixel vector $(\theta_L, \theta_C) = (-30°, 15°)$

Fig. 14: Pixel vectors, desired $(\theta_L, \theta_C) = (-30°, 30°)$

VI. Conclusion

A head pose estimation algorithm based on a mathematical nose model was proposed in this paper. The algorithm was designed for gray scale, geometrically normalized, facial images with a low resolution. Pixels from the nose region of a probe image are compared with a set of pixel vectors derived from a nose model. The head pose estimation is based on the best match. Scores were calculated using two error measures.

The research question mentioned in section I can not be answered with a clear yes or no. Promising results were achieved for head poses between ±15°, but for larger angles the performance dropped significantly. The two main reasons for this are the fact that the ROI was too wide and that the scaling of the vectors was wrong, especially for head poses larger than 15° with respect to the frontal view.

A. Recommendations

Based on the previous subsection, it can be said that the revision of the equations used to calculate N_p and S is very important, as well as decreasing the width of the ROI of the probe images, and thus of the model itself. Besides that, the nose model could be derived for multiple heights. The current implementation averages pixels in the ROI of probe images in the vertical direction. This results in a loss of information. By comparing each pixel row from the ROI with a nose model of the corresponding height, all the available information is used.

References

[1] Xiaozheng Zhang and Yongsheng Gao. “Face recognition across pose: A review”. In: *Pattern Recognition* 42.11 (2009), pp. 2876–2896.

[2] Zhifei Wang et al. “Low-resolution face recognition: a review”. In: *The Visual Computer* 30.4 (2014), pp. 359–386.

[3] M. Hassaballah, S. Aly. “Face recognition: challenges, achievements and future directions”. In: *IET Computer Vision* 9.4 (2015), pp. 614–626. DOI: 10.1049/iet-cvi.2014.0084.

[4] Volker Blanz et al. “Face recognition based on frontal views generated from non-frontal images”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 454–461.

[5] E. Murphy-Chutorian and M. M. Trivedi. “Head Pose Estimation in Computer Vision: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.4 (Apr. 2009), pp. 607–626. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.106.

[6] Andrew Gee and Roberto Cipolla. “Determining the gaze of faces in images”. In: *Image and Vision Computing* 12.10 (1994), pp. 639–647.

[7] Jiawei Chen et al. “Estimating head pose orientation using extremely low resolution images”. In: *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE. 2016, pp. 65–68.

[8] Nicolas Gourier et al. “Head pose estimation on low resolution images”. In: *Multimodal Technologies for Perception of Humans*. Springer, 2006, pp. 270–280.

[9] Conrad Sanderson, Samy Bengio, and Yongsheng Gao. “On transforming statistical models for non-frontal face verification”. In: *Pattern recognition* 39.2 (2006), pp. 288–302.

[10] Jean-Yves Guillemaut et al. “General pose face recognition using frontal face model”. In: *Iberoamerican Congress on Pattern Recognition*. Springer. 2006, pp. 79–88.

[11] *Head Outline Clipart*. URL: http://www.clker.com/clipart-head-outline-1.html (visited on 06/09/2016).

[12] G. Li et al. “An Efficient Face Normalization Algorithm Based on Eyes Detection”. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2006, pp. 3843–3848. DOI: 10.1109/IROS.2006.281791.

[13] *The CMU Multi-PIE Face Database*. URL: http://www.multipie.org/ (visited on 06/15/2016).