Head pose estimation based on face symmetry analysis

Afifa Dahmane, Slimane Larabi, Ioan Marius Bilasco & Chabane Djeraba

Signal, Image and Video Processing

ISSN 1863-1703

SIViP
DOI 10.1007/s11760-014-0676-x

Springer

SIViP
DOI 10.1007/s11760-014-0676-x

ORIGINAL PAPER

Head pose estimation based on face symmetry analysis

Afifa Dahmane · Slimane Larabi · Ioan Marius Bilasco · Chabane Djeraba

Received: 11 September 2013 / Revised: 4 July 2014 / Accepted: 5 July 2014

Abstract This paper addresses the problem of head pose estimation in order to infer non-intrusive feedback from users about gaze attention. The proposed approach exploits the bilateral symmetry of the face. Size and orientation of the symmetrical area of the face is used to estimate roll and yaw poses by the mean of decision tree model. The approach does not need the location of interest points on face and presents robustness to partial occlusions. Tests were performed on different datasets (FacePix, CMU PIE, Boston University) and our approach coped with variability in illumination and expressions. Results demonstrate that the changes in the size of the regions that contain a bilateral symmetry provide accurate pose estimation.

Keywords Head pose estimation · Symmetry detection · Pattern recognition

1 Introduction

Head pose is often linked with visual gaze estimation and provides a coarse indication of the gaze in situations where either the system should be non-intrusive using only a regular camera, or when the eyes may be not detectable. In this context, a coarse head pose can give a good indication of the gaze attention.

Head pose estimation is a classic problem in computer vision and has been studied since 1994 [1]. Although major advances have been achieved in controlled environment, the problem is still open in unconstrained natural environment under variable lighting and human expression. Head pose estimation is widely used in many applications such as video conferencing, driver monitoring or human computer interaction. Moreover, for many pattern recognition applications, it is necessary to estimate coarse head pose to eliminate variations in pose for better accuracy (e.g., face recognition or facial expression analysis). Many approaches based on local facial features are proposed to deal with head pose estimation. However, the obvious difficulty for this local approaches lies in detecting outlying or missing features in situations where facial landmarks are obscured. Also, low resolution imagery make it difficult to precisely determine the feature locations.

This paper presents a method based on symmetry to estimate discrete head pose. We exploit the bilateral symmetry of the face to directly deduce two degrees of freedom for the head (yaw and roll). The symmetry is defined using global skin region, instead of local interest points. The proposed approach does not need the location of interest points on the face and can be deployed using low-cost and widely available hardware. Also, no initialization nor calibration are required. The estimated pose is coarse but sufficient to infer general gaze direction. The current work brings three main contributions:

– First, we develop a method for detecting the position of symmetry axis and its orientation in an image.
– Second, the roll angle is deduced from the inclination of the symmetry axis.
– Third, the yaw angle is calculated using the region which delimits symmetrical pixels.

A. Dahmane (✉) · S. Larabi
Computer Science Department, USTHB University, Algiers, Algeria
e-mail: fdahmane@usthb.dz

S. Larabi
e-mail: slarabi@usthb.dz

A. Dahmane · I. M. Bilasco · C. Djeraba
LIFL, USTL, University of Lille UMR CNRS, 8022 Lille, France
e-mail: marius.bilasco@lifl.fr

C. Djeraba
e-mail: chabane.djeraba@lifl.fr

Published online: 18 July 2014

Symmetrical region is defined by analyzing pixels intensity. The intensity of one pixel on the right side of the face is more similar to its mirror pixel than another pixel in the image. We have conducted experiments which indicate that the use of facial symmetry as a geometrical indicator for head pose is still reliable when local geometric features (such as eyes, nose or mouth) are missed due to occlusions or wrong detections. We give more insights about the method comparatively with our previous work and extend it to two degrees of freedom. Besides using public datasets (FacePix, CMU PIE and Boston University datasets [2–4]), we have also used web-cam captures in order to cover more situations such as completely occluded eye. Sample captures are available here: www.lifl.fr/~dahmane/VIDEOS.

The paper is organized as follows. We first review the related work on head pose estimation in Sect. 2. Then, we provide the methodology used for the estimation of the head pose using the symmetrical parts of the face in Sect. 3. In Sect. 4, we present and discuss the results of the head pose estimation evaluation process. Finally, we conclude and introduce the potential future work in Sect. 5.

2 Related work

In this section, we review the related work for head pose estimation regardless of the underlying descriptors and methodology. We analyze the existing methods in order to highlight advantages and disadvantages of each one. Then, we focus our attention on global approaches exploiting symmetry information. Even though the latter approaches are less popular, we strongly believe in the benefits of global symmetry for pose estimation.

Existing techniques for head pose estimation are summarized in [1] and can be categorized in six groups:

Model-based approaches include geometric and flexible model approaches. Geometric approaches use the location of facial features such as eyes, mouth and nose and geometrically determine the pose from their relative configuration [5,6]. Flexible model approaches use facial features to build a flexible model which fits to the image such that it conforms to the facial structure of each individual (AAM) [7]. However, accurate matching of a deformable face model to image sequences with large amounts of head movements is still a challenging task [8].

Classification-based approaches formulate the head pose estimation as a pattern classification problem. Most works have used various SVM classifiers [9,10]. Isarun and al. [11] uses random trees beside SVM. In [12], kernel principal component analysis (KPCA) is used to learn a nonlinear subspace for each range of view. Then a test face is classified into one of the facial views using Kernel Support Vector Classifier (KSVC). Also, classification is achieved in [13] using a set of randomized ferns and in [14] Naive Bayes classifier are applied to estimate head pose.

Regression-based approaches consider pose angles as regression values. Several regressors are possible such as Convex Regularized Sparse Regression (CRSR) [15] and Gaussian Progress Regression (GPR) [16]. Murad and al. [17] proposed a method based on Partial Least Squares (PLS) Regression to estimate the head pose. Support Vector Regressors (SVRs) are used to train Localized Gradient Orientation (LGO) histogram computed on detected facial region to estimate driver's head pose in [18]. Neural networks are one of the most used nonlinear regression tools for head pose estimation. Tia and al. [19], use multiple cameras and estimate head pose by fusing neural networks results from each camera.

Template Matching approaches compare images or filtered images to a set of training examples and find the most similar. In [20], author represents faces with templates that cover different poses. The input data is correlated with model templates to achieve face recognition finding the best match. Similarity to prototypes philosophy is adopted by authors in [21] in order to calculate the pose similarity ratio.

Manifold Embedding approaches produce a low dimensional representation of the original facial features and then learn a mapping from the low dimensional manifold to the angles. Biased Manifold Embedding for supervised manifold learning is proposed in [22]. The incorporation of continuous pose angle information into one or more stage of the manifold learning process such as Neighborhood Preserving Embedding (NPE) and Locality Preserving Projection (LPP) is studied in [23]. Dong [24] proposed Supervised Local Subspace Learning (SL2) to learn a local linear model where the mapping from the input data to the embedded space was learned using a Generalized Regression Neural Network (GRNN). In [25], author proposed the K-manifold clustering method, integrating manifold embedding and clustering.

Tracking approaches uses temporal information to improve the head pose estimation using the results of the head tracking [26]. In [11], a pedestrian tracker is applied to video to infer head pose labels from walking direction and automatically aggregate ground truth head pose labels. Ba and al. [27] aims recognition of people's visual focus of attention using a tracking system based on particle filtering techniques. KLT algorithm is used in [28] to track features over video frames in order to estimate 3D rotation matrix of the head.

Each approach has specific limitations. Identity of persons can hamper the pose discrimination. The effect of identity can cause more dissimilarity then the pose itself. It does for appearance-based approaches (template matching and manifold embedding methods). Those approaches suffer also from lighting and require high computational time. In contrast to model-based approaches which are fast and identity independent. The model-based approaches are, however, sensitive to

occlusion and usually require high resolution images. The difficulty lies in accurate detection of the facial features since all the facial features are required (as for example, the outer corner of both eyes hard to detect when wearing glasses or in the presence of long hair). High resolution imagery may not be available in many applications such as driver monitoring and e-learning systems. Also, model-based approaches [5,7] require frontal view to initialize the system.

A specific family of approaches that exploit global features of the face, reducing dependency on identity and avoiding initialization of frontal pose, is represented by solutions that exploit facial symmetry.

2.1 Symmetry-based approaches

The human perception of head pose is based upon two cues: the deviation of the head shape from bilateral symmetry, and the deviation of the nose orientation from the vertical [29]. Therefore, we presume that head pose is more related to the geometry of face images, and the symmetry of the face is a good indicator about the geometric configuration and the pose of the head. Despite the fact that the human face is not perfectly symmetrical, the facial symmetry of a person is significant and can be exploited. In [30], in order to detect possible regions containing a face in image, authors estimate the symmetry of the regions. The hypothesis is that the amount of symmetry can offer hints about head orientation.

Some works dealing with the head pose estimation through feature points use the symmetrical property of the head. For instance, facial symmetry has been used as a visual intent indicator in [31] for people with disabilities. The face pose (roll and yaw angles) are estimated from a single uncalibrated view in [32] where the symmetric structure of human face is exploited by taking the mirror image of a test face image as a virtual second view. Facial feature points of the test and its mirror image are matched against each other in order to evaluate the pose. In [33], authors introduce Gabor filters in order to enhance the symmetry computation process and estimate the yaw. Symmetry-based illumination model proposed in [34] is based on three features (the two eyes and the nose tip). For every combination of two eyes and a nose, head pose is computed using a weak geometry projection and internally calibrated camera. In the context of face recognition, in [35] authors use the bilateral symmetry of the face to deduce if the pose is frontal or not. Beside intensity images, 3D data can be used for head pose estimation [36,37].

Symmetry provides high-level knowledge about the geometry of face. We use the bilateral symmetry of the face to deal with the head pose estimation problem. We propose an approach to perform head pose estimation based on the symmetrical properties of the face.

3 Our approach

Our symmetry-based approach aims at being non-intrusive and do not require specific user collaboration. It has to be independent to user identity and can be deployed on still images as well as on videos. Our system use a geometrical model which is not based on specific feature points. We propose an approach that joins the effectiveness of both local and global methods. We select symmetrical areas on the face relative to skin pixels intensity and use the size of these areas and their orientation to estimate roll and yaw poses.

The proposed method (Fig. 1) first detects the face using Viola Jones algorithm [38]. Preprocessing (histogram equalization) is applied in order to reduce illumination influence. Then the symmetry axis is searched in the area of the face. This task is performed using a symmetry detection algorithm. Once the head and the symmetry axis are detected, we extract the features of the symmetry. We deduce the roll angle from the orientation of the symmetry axis and estimate the yaw by analyzing some characteristics of the symmetrical region. The method is summarized in Fig. 1. We can see that the symmetry detection allows the estimation of the roll angle and further the extraction of symmetric features.

In the following, we emphasize the correlation between symmetry and head pose by analyzing the symmetrical regions of the face. We detail the symmetry axis detection process and the characterization of the symmetry region. Then, we pass to the yaw estimation process by means of a decision tree classifier.

3.1 Analysis of symmetrical regions on face

When the face is in front of the camera, the symmetry between its two parts appears clearly (see Fig. 2) and the line

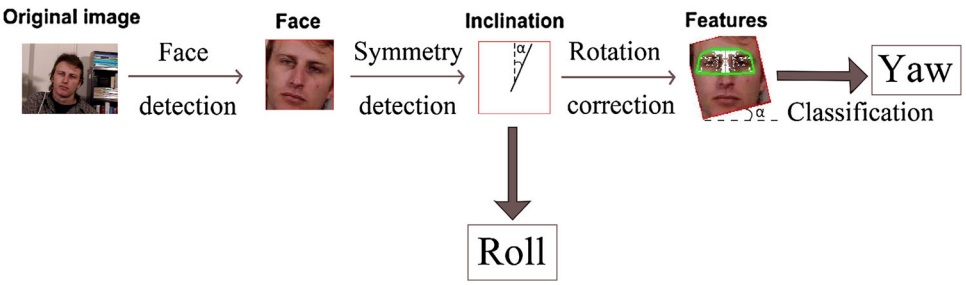

Fig. 1 Proposed approach

which passes between the two eyes and nose tip defines the symmetry axis. However, when the head performs a motion, for example, a yaw motion, this symmetry decreases.

We exploit the difference between the symmetries before and after the head rotation in order to infer measures which characterize the yaw movement.

Figure 2 shows the variation of the symmetrical region for various head yaw and roll poses. The symmetric pixels are superimposed in white on the images. First, for the yaw movement, we analyze the amount of symmetrical parts under various yaw angles (Fig. 2a using as support Fig. 3).

Let a and b two symmetrical points on the face. m is the middle of the segment $[ab]$. The projections of these points on the image plane are a_i, b_i, m_i. When the face is in front of the camera, the segments $[a_i m_i]$ and $[m_i b_i]$ are symmetrical with respect to m_i as shown in Fig. 3a. When the head performs a yaw motion, the feature points (a, b, m) are projected into (a'_i, b'_i, m'_i) (see Fig. 3b). Let ω' be the vanishing point associated to the direction of (a, b) in the image plane. Since the central projection preserves the cross ratio [39], the cross-ratios of (a, b, m, ∞) and $(a'_i, b'_i, m'_i, \omega')$ are equal. We obtain:

$$\frac{ma}{mb} = \frac{m'_i a'_i}{m'_i b'_i} \div \frac{\omega' a'_i}{\omega' b'_i} \tag{1}$$

As the two members of the Eq. 1 are equal to one (as m is the middle of $[ab]$), the point m'_i is not the middle of $a'_i b'_i$ and its position depends on the position of $a'_i b'_i$ relatively to ω'. Since m is the symmetry center of ab, the pixels of segment line $a'_i b'_i$ may satisfy a partial symmetry but in this case the symmetry center will not be the middle of $a'_i b'_i$. It will be m'_i and the symmetry concerns the segments $m'_i a'_i$ and $m'_i d'_i$ where d'_i is located between m'_i and b'_i so as $m'_i a'_i = m'_i d'_i$ (see Fig. 3b).

Thus, after a yaw motion, the symmetry on the image plane is partial. The symmetrical part of a segment linking two symmetrical points on the face, is smaller than the symmetrical part of the same segment before the movement.

Secondly, regarding the roll angle, we estimate that it corresponds to the angle of the symmetry axis (see Fig. 2b). We infer the pose angle from the inclination of the symmetry axis that we calculate in case of frontal view.

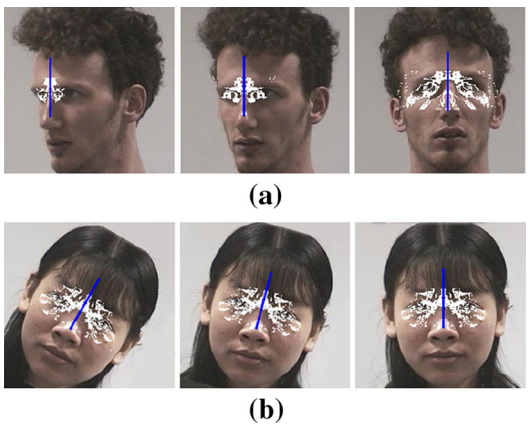

Fig. 2 **a** Variation in the size of the symmetrical region during yaw movement. **b** Variation in the angle of the symmetry axis during roll movement

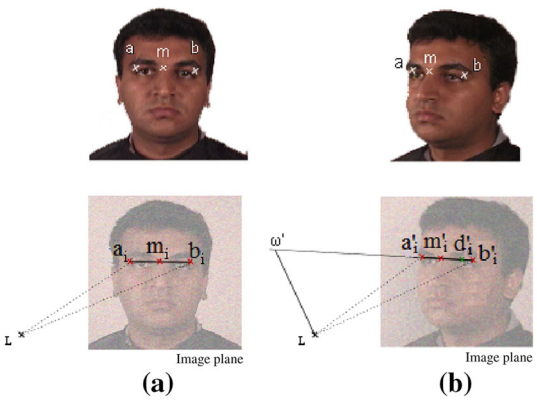

Fig. 3 **a** Projection of a segment (ab) when the face is in front of the camera, **b** Projection of the segment line after a yaw motion

3.2 Symmetry axis detection

We use pixels intensity to detect symmetry in the image. Therefore, illumination influences the detection and in some cases, causes errors. We apply preprocessing on images before starting the symmetry detection in order to improve the robustness. We use the RGB space which gives more significant information about skin color compared to gray scale and allows us to differentiate between face and background since one skin pixel is generally more similar to another skin pixel than to a background pixel. For this reason, we apply histogram equalization on each RGB color channels of the image in order to reduce illumination effect and then, we merge them back.

Our goal is to find the morphological symmetry of face, under different poses, provided that the desired symmetry does not disappear completely from the image (e.g., when yaw angle exceeds 45°). Our algorithm is based on Stentiford's Algorithm [40]. It was necessary to adapt the initial algorithm that highlights the symmetries in image regardless of what the image represents, a face or an object. After detecting the face, we consider an ellipse inside and we set our region of interest to the top half of the ellipse. This part of the face is chosen because upper part is more affected by head rotations. The change in the size of the symmetric region after a right/left rotation is greater in the region of the eyes than that of the mouth and presents less deformation than the lower part (talking, smiling).

The symmetry axis defined by the position P (one point of the axis) and the orientation α is computed by selecting the best one from those computed assuming that the orientation

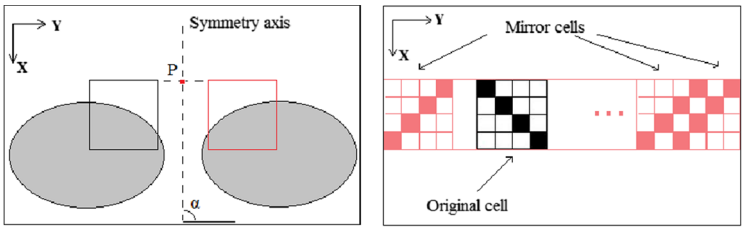

Fig. 4 **a** A local symmetry axis with one couple of symmetric cells. **b** Example of an original cell (*in black*) and all it's mirror cells relative to $\alpha = 90°$

α varies from α_{min} to α_{max}. It is chosen as the axis with greatest number of symmetrical pixels and the closest to the face center. The vote is performed as follow: we consider the distribution of the symmetry axes $A_{i\{P_i,\alpha_i\}}$ (one axis for each α). The axes are weighted by the number of the local symmetries which they satisfy. We take the m maximum of this distribution and vote for the axis $A_{\{P,\alpha\}}$ which minimize the distance to the face center C such that:

$$d(C, A_{\{P,\alpha\}}) = min\{d(C, A_{i\{P_i,\alpha_i\}})\} i \in [1, m] \quad (2)$$

Detect the symmetry relative to inclination: The region of interest in image is divided into small overlapping square blocks noted *cells*. The symmetrical cell of each non-homogeneous cell in the region of interest is determined relative to α. Local symmetries concerning the cells are computed. This local symmetry axes are perpendicular to the line joining two symmetrical cells (see Fig. 4a). Having the n symmetry axes $A_{i\{P_i,\alpha\}}, i = 1..n$, the best axis for a given α is selected based on the number of symmetrical pixels using the same vote mechanism as defined previously.

Define the local symmetries: Given the orientation α, for each original cell, we search a match from all its mirror cells relative to α (see Fig. 4b). The location of a mirror cell is calculated by reflecting the pixels of the original cell via Eq. 3. The coordinates of a pixel (x, y) in the reflected position are (x_i, y_i). We vary x_i along the width of the region of interest and obtain y_i.

$$y_i = y + ((\tan \alpha) \times (x_i - x)) \quad (3)$$

Mirror cells lie on the strip which passes through the original cell and that is inclined by an angle $\alpha + \pi/2$.

- Two matched cells are then considered symmetric.
- Two cells are matched if each pixel on the diagonal of the original cell matches its corresponding pixel on the mirror cell.
- A pixel matches another one if the intensity difference of the three channels does not exceed a given threshold ε.

The interval $[\alpha_{min}, \alpha_{max}]$ and the step of α influences the results. A small step gives more accuracy but takes more calculation time and requires a large amount of storage. We set the step according to the interval so that we do not obtain a high-dimensional distribution. Also, a big interval may provide symmetries which do not correspond to the bilateral symmetry searched. To cope with these issues, we set α_{max} to not exceed 135° and α_{min} not under 45° because the natural movement of the roll does not exceed 45° on each side. When analyzing video and taking into account the coherency of head movements, α_{min} and α_{max} can be set around various orientations.

The actions for detecting the symmetry axis position and orientation are summarized by Algorithm 1.

```
for α ← α_min to α_max do
    while ROI do
        Take a cell and search it's symmetric one
        if local symmetry exists then
            Save axis A_{i{P_i,α_i}}
        end
        Vote for the best axis relative to α
    end
end
Vote for the best image symmetry axis
```

Algorithm 1: Symmetry axis detection

3.3 Symmetry features

Once the symmetry axis is located, the image is rotated with respect to the axis inclination and symmetry features are extracted. For the given symmetry axis, and for each pixel, the symmetrical one is searched considered individually and not as being a part of a cell in order to define the region of symmetry without excluding homogeneous area. In this way, detection of the symmetrical region is not sensitive to pixel matching process since we use all the texture. If the difference of intensity between the two pixels is greater than a certain threshold, then they are not symmetric. The convex hull encompassing symmetrical pixels is used to extract features. Figure 5 illustrates symmetric pixels superimposed on the face. Vertical measurement of the convex hull is not kept as is not useful for yaw movement, only the width of the hull which contains symmetrical pixels and the mean distance of all symmetrical pixels to the axis of symmetry are used as features. Width is defined as the euclidean distance between the two most distant pixels.

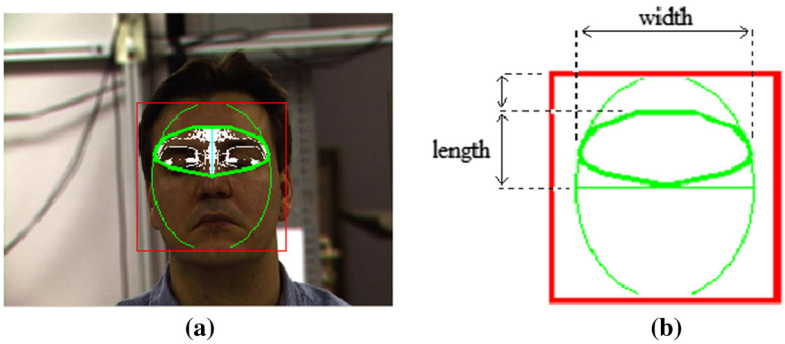

Fig. 5 Examples of extracting features. **a** Computing a convex hull which includes the symmetrical pixels. **b** Measures relating to the symmetrical region

3.4 Yaw estimation

3.4.1 Decision tree classifier

In order to determine the yaw motion, a decision tree classifier is trained using the features (relative width of the symmetrical region and the relative mean distance of symmetrical pixels) extracted from the symmetrical parts according to the amount of yaw motion. Each class of the classifier corresponds to a discrete pose. To increase performance of prediction, we used the Alternating decision tree which is based on boosting [41]. The tree alternates between prediction nodes and decision nodes. The root node is a prediction node and contains a value for each class. The prediction values are used as a measure of confidence in the prediction.

The set of head pose images used for learning represents the angles for which the symmetry axis is properly detected. The poses are discrete and vary from $-45°$ (left) to $+45°$ (right).

We construct the model from the feature vectors derived from images of several people recorded in different poses. Right and left poses with the same angle are gathered in the same class as they contain the same amount of symmetry and, therefore, the same information. Thus, to estimate $2 * n + 1$ discrete poses (n lateral right poses, n lateral left poses and 1 frontal pose), the classifier has $n + 1$ classes. For this, we use $2 * n + 1$ images per subject to represent the $2 * n + 1$ poses.

The root contains null values as prediction for the $n + 1$ classes. The first level contains decision nodes based on the values of the feature vector attributes, followed by prediction nodes for each class and so on until the leaves. The sum of the prediction values crossed when following all paths for which all decision node are true, is used to classify a given instance. The class which has the highest prediction value is the predicted class.

As we use a supervised classification approach, we first have to train the alternating decision tree classifier using the same number of images per person as the number of classes. With the constructed tree, we can predict the yaw for various test face images. Training and testing images do not have to be from the same dataset.

3.4.2 Left versus right poses

To differentiate between left and right poses, we use the difference in intensity between the skin and the background. Our assumption is that a pixel on the face is more similar to another pixel on the face than to a pixel on the background.

We take a pixel located on the symmetry axis to ensure that it is on the face. We compute the average intensity of the pixels surrounding and consider this value as a reference. If the symmetry axis is closer to the left contour (resp. right contour) of the face, then the face is oriented to the left (resp. right). We calculate two values: the difference between the reference value and the average intensity of pixels on the left side and the same difference for the reference value and the right side of the axis. If the difference is bigger on the left side (resp. right), we conclude that the face on the image is oriented to the left (resp. right).

With this method, we determine on which side the pose is oriented and this information is combined with the degree of orientation estimated by the decision tree in order to obtain the yaw head pose.

4 Experimental results and discussion

We first evaluated the approach using the FacePix [2] dataset which is ideal for the yaw motion. It consists of poses in the interval $\pm 90°$ at $1°$ increments. This allow us to form several class configurations as explained in Sect. (3.4.1) (e.g., 10 classes for 19 poses). Also, we tested our approach on the CMU PIE dataset [3] which gives more variability in term of illumination and expressions (e.g., eyes closed or smile). In addition to image datasets, we tested the video sequences of the Boston University (BU) dataset [4]. In BU dataset, subjects are doing free movements including yaw and roll variations. This allow us to estimate the roll (in-plane rotation) accuracy besides the yaw. Poses in the videos are predicted using the model built with the FacePix dataset. Video sequences are also recorded in the lab, to reproduce situations of partial occlusion not present in the available datasets. In all experiments, we used the same parameters,

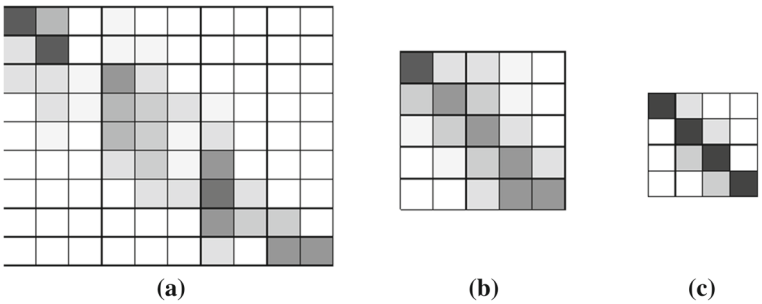

Fig. 6 Confusion matrix associated to: **a** 19 poses classifier with 5° step. **b** 9 poses classifier with 10° step. **c** 7 poses classifier with 15° step

the threshold used for matching cells $\varepsilon = 25$, the side of cells $s = 20$ and the number of axes which are part of the voting system $m = 3$. The interval of α is [85°, 95°] for FacePix and CMU PIE as limited roll is presented and [45°, 135°] for BU dataset as more natural movements and poses are presented. The results of our experiments are presented below.

4.1 Facepix dataset

We used the FacePix dataset [2] to build a head pose model and to evaluate it. The FacePix database consists of three sets of face images: variable pose, variable dark illumination and variable light illumination. The sets of variable illumination images have only frontal pose. This is why we used only the set of variable poses which is composed of 181 pose images of 30 different subjects. Among the 181 poses, we used poses varying from −45° to +45° because when exceeding this interval, the bilateral symmetry disappears from the image.

We tested several configurations, changing the number of classes each time. Figure 6 shows the confusion matrix for three classifiers : 19 discrete poses associated to the yaw angles from −45° to 45° with 5° step (10 classes), 9 discrete poses associated to the yaw angles from -40° to 40° with 10° step (5 classes) and 7 discrete poses associated to the yaw angles from −45° to 45° with 15° step (4 classes). One can see that the estimated pose is in the diagonal of the matrix. The 7 poses model had the higher classification rate but all confusion matrices show high values along the diagonal.

In order to evaluate the model, we split the data into 6 equal subsets and performed sixfold cross-validation. In each run, 5 subsets are used as the training set and the rest is used as a test set. The subjects in the training and test set are completely distinct since each subject is taken only once. On this dataset, we tested the sensitivity of the method to the symmetry axis detection accuracy. We annotated the position of the head and the position and the orientation of the symmetry axis in order to compare results in a semi-automatic and a fully automatic settings. A detailed description of the results with 7 poses is shown in Table 1.

When removing errors related to head detection and/or symmetry axis detection, the results outperform those of the completely automatic mode. The latest one are not much worse, since the classification accuracy reaches 79.6 % for the seven poses model.

Table 1 Classification rates and mean absolute errors (MAE) for FacePix dataset in semi and fully automatic modes

Data	Accuracy (%)	MAE (°)
Head and symmetry axis annotated	82.38	2.71
Head annotated and symmetry automatic	81.90	2.78
Head and symmetry detection automatic	79.63	3.14

4.2 CMU PIE dataset

The CMU Pose, Illumination and Expression dataset [3] presents different conditions of capturing images. It contains images of 68 subjects with a step of 22.5° between poses. In our experiments, we used the image set corresponding to variable expression (4 different expressions), the one recorded under variable lighting conditions (21 different flash orientations) and the set with subjects talking. Concerning the first set (Expression), we used images of poses between 45° and −45°. The challenge with this dataset is the variable lighting set. In this case, when there is an intense light source in a lateral side, the scene loses it's symmetry. We built a classifier for each set to study apart the robustness to expression and lighting variation. We calculated the classification rate for each classifier using sixfolds cross-validation. We also merged all images in one encompassing set.

Table 2 shows results for each set and those considering all images in one encompassing set. To achieve illumination invariance, the RGB histogram equalization is not sufficient. We applied a discrete cosine transform (DCT)-based normalization technique [42] to the full image. A number of DCT coefficients are truncated to minimize illumination variations since the variations mainly lie in the low frequency band. This truncation affect the matching process in the Expression set. The accuracy drop from 72.57 to 49.81 %. Unlike Talking and Lighting sets where DCT normalization did more good than bad, the illumination affects strongly the matching of symmetries than the noise added by the normalization. On the other hand, DCT normalization gives better results on

sets with a great number of learning images. In the CMU Expression set, for each person, each pose is represented by 3 or 4 images (neutral, blinked, smiling and for certain subject with glasses). However, in the Talking set, each pose has 60 images and in the Light set, 23 images are recorded for each pose. The large number of images used for learning offsets the loss due to the normalization and even allowed improvement of the accuracy for the Talking set.

4.3 Videos

We also test on videos as we aim to use the solution in real environment for having real pose-related feedback. We test our method on the video sequences of the Boston University head pose dataset [4]. We recall that the inclination of the symmetry axis corresponds to the roll angle in case of frontal view. The yaw and the roll are calculated over all the frames in order to compare with ground truth. In the experiments, we have used the alternating decision tree trained on FacePix dataset as it covers better the range of face poses than CMU PIE which has widely spaced poses (22.5° between poses). We ensure that the size of the face in the BU images is the same as in FacePix dataset. The best results for the yaw are obtained using the model built with 19 discrete poses and 5° step from the FacePix dataset, giving 5.24° mean absolute error (MAE), 6.80° root mean squared error (RMSE) and a standard deviation (STD) of 4.33°. Results for the yaw and the roll are shown in Table 3.

We exploit the temporal information contained in the video stream in order to reduce calculation time. We use the position and the orientation of the symmetry axis of a given frame to reduce the search interval in the next frame. We perform a coherency check every 10 frames (approximately one second).

Table 2 Results for the CMU PIE dataset

Data	Classification accuracy (%)	
	RGB Equalization	DCT
CMU Expression	72.57	49.81
CMU Talking	81.04	87.63
CMU Lighting	72.51	85.90
CMU PIE	72.48	82.26

Table 3 Results for the BU dataset

	RMSE (°)	MAE (°)	STD (°)
Roll	4.39	2.57	3.56
Yaw (FacePix model—5° intervall)	7.60	5.12	5.62
Yaw (FacePix model—15° intervall)	6.80	5.24	4.33

4.4 Resolution and occlusion

We also conducted experiments which indicate that the facial symmetry is a good geometrical indicator for head pose when the local geometric features (such as eyes, nose or mouth) are missed due to occlusions or wrong detections. When the head rotation exceeds 30° (in a left/right rotation), some feature points disappear from the image plane but partial symmetry still exists.

In order to measure the robustness of the approach, we generate low resolution images from the FacePix dataset where the head size was 80×80. We resize the head to generate two head image sets, the first is 40×40 pixels and the second 25×25 pixels. We succeeded in detecting the symmetric features, thing that cannot be done when relying on specific feature points. We built a 9 pose classifier for both sets using the parameters $\varepsilon = 25$, $s = 2$ et $m = 3$, $85° <= \alpha <= 95°$. The accuracy of the first classifier is 74.1 % and that of the second is 63.8 %. We can see that the accuracy drop from 79.6 % because the method is based on local symmetries and our algorithm is sensitive to symmetry axis calculation. On very low resolution images, the local symmetries are not enough relevant. But results are not very bad for heads which are only 25×25 pixels.

We also tested the system with web-cam in the laboratory simulating local partial occlusions. As the process does not need interest points, partially occluded faces can be processed since there is at least one couple of symmetrical pixels on the image. To do so, all the texture pixels in the region of interest, contribute to the demarcation of the symmetrical area. This can be seen in Fig. 7.

4.5 Summary and comparison with the state of the art

The main advantage of the method is that the calculation can start at any pose, without any initialisation, since the head and the symmetry axis are automatically detected for poses between −45° and +45°. Also, new face images can be classified easily using the pre-built model.

In video sequences where the head is performing free movements, wrong detections often occur. To resolve this problem, we exploit the continuity of movement. We exclude detections which are very far from the 3 previous frames considering them as wrong. We use instead an interpolated position of the head (see Fig. 8). The process is then, fully automatic but sensitive to the accuracy of head detection and symmetry axis calculation. The system is robust to changes in lighting condition, expression and also to identity information since the method is geometric. Besides, no specific points are needed to be detected on the face. So, closed eyes or partially occluded face give the same results as complete face.

We compared our results with others which used the same datasets. Tian et al. [19] obtained 85 % of good classifica-

Fig. 7 Sample frames from video sequences taken in lab

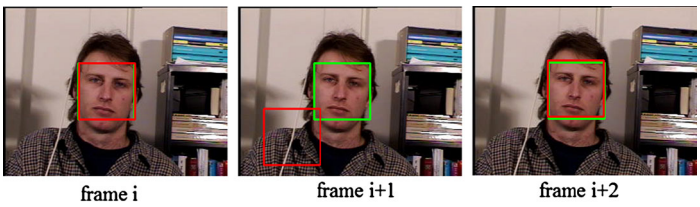

Fig. 8 Example of false head detection (frame i+1) and it's correction

Table 4 Comparison of the yaw results with the state of the art using FacePix dataset

Method	Resolution	MAE (°)	Acc (%)
Ji et al. [15] (Regression)	60 × 60	6.1	–
Liu et al. [25]* (Manifold clustering)	16 × 16	3.16	–
Vineeth et al. [22]* (Biased Isomap)	32 × 32	5.02	–
Vineeth et al. [22]* (Biased LLE)	32 × 32	2.11	–
Vineeth et al. [22]* (Biased LE)	32 × 32	1.44	–
Proposed (Symmetry classification)	80 × 80	3.14	79.6
	40 × 40	4.57	74.1
	25 × 25	6.73	63.8

* A significant drawback of manifold learning techniques is the lack of a projection matrix to treat new data points

Table 5 Comparison of the BU dataset results with the state of the art

		RMSE (°)	MAE (°)	STD (°)
Valenti et al. [26]	Yaw	6.10[a]	–	5.79[a]
	Roll	3.00[a]	–	2.82[a]
Morency et al. [43]	Yaw	–	4.97	–
	Roll	–	2.91	–
Proposed	Yaw	6.80	5.24	4.33
	Roll	4.39[b]	2.57[b]	3.56[b]

[a] Eye cues used, the pose is estimated only when eyes are detected
[b] The roll is estimated in case of frontal view

tion on CMU PIE dataset and 82 % for us. Tables 4 and 5 show results on FacePix and BU datasets expressed in MAE, RMSE, STD and classification accuracy (Acc). From these results, it is shown that our method provides comparable results on CMU PIE and BU datasets. On FacePix, manifold embedding methods give good results but there is no explicit solution for out-of-sample embedding in an LLE and LE manifold [1]. These methods are not automatic unlike ours. New data can be classified through a model of examples already built.

5 Conclusion

We presented a new approach to perform head pose estimation. We exploit bilateral symmetry of the face to deal with roll and yaw motions. The orientation of the symmetry axis indicates the roll angle of the head. The symmetrical region of the face with respect to this orientation provides us features such as the width of region which allow us to classify and then, to predict yaw angles. Symmetrical features may be extracted without the detection of special facial landmarks and no calibration nor initial frontal pose are required. The results obtained by our approach have been evaluated using public datasets and they outline the good performance of our algorithm with regard of the state of the art methods. In our future work, we explore new features which allow us to estimate combined yaw and pitch pose. We will also explore more advanced regression methods to achieve two degrees of freedom.

Acknowledgments This work was conducted in the context of the ITEA2 "Empathic Products" project, ITEA2 1105, and is supported by funding from DGCIS, France.

References

1. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: a survey. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **31**(4), 607–626 (2009)
2. Black, J., Gargesha, M., Kahol, K., Kuchi, P., Panchanathan, S.: A framework for performance evaluation of face recognition algorithms. In: ITCOM, Internet Multimedia Systems II, Boston (2002)
3. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression database. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1615–1618 (2003)
4. Valenti, R., Gevers, T.: Robustifying eye center localization by head pose cues. In: CVPR (2009)
5. Wang, J.-G., Sung, E.: Em enhancement of 3d head pose estimated by point at infinity. Image Vis. Comput. **25**(12), 1864–1874 (2007)
6. Pan, Y., Zhu, H., Ji, R.: 3-D Head Pose Estimation for Monocular Image. ser. Fuzzy Systems and Knowledge Discovery. Springer, Berlin (2005)
7. Baker, S., Matthews, I., Xiao, J., Gross, R., Kanade, T., Ishikawa, T.: Real-time non-rigid driver head tracking for driver mental state estimation. In: 11th World Congress on Intelligent Transportation Systems (2004)
8. Caunce, A., Taylor, C.J., Cootes, T.F.: Improved 3d model search for facial feature location and pose estimation in 2d images. In: BMVC (2010)
9. Huang, J., Shao, X., Wechsler, H.: Face pose discrimination using support vector machines (svm). In: ICPR (1998)
10. Dahmane, M., Meunier, J.: Object representation based on gabor wave vector binning: an application to human head pose detection. In: ICCV (2011)
11. Chamveha, I., Sugano, Y., Sugimura, D., Siriteerakul, T., Okabe, T., Sato, Y., Sugimoto, A.: Appearance-based head pose estimation with scene-specific adaptation. In: ICCV (2011)
12. Li, S., Fu, Q., Gu, L., Scholkopf, B., Cheng, Y., Zhang, H.: Kernel machine based learning for multi-view face detection and pose estimation. In: ICCV, vol. 2, pp. 674–679 (2001)
13. Benfold, B., Reid, I.: Colour invariant head pose classification in low resolution video. In: BMVC (2008)
14. Zhang, Z., Hu, Y., Liu, M., Huang, T.: Head pose estimation in seminar room using multi view face detectors. Multimodal Technol Percept. Hum. **4122**, 299–304 (2007)
15. Ji, H., Liu, R., Su, F., Su, Z., Tian, Y.: Robust head pose estimation via convex regularized sparse regression. In: ICIP (2011)
16. Ranganathan, A., Yang, M.-H.: Online sparse matrix gaussian process regression and vision applications. In: ECCV (2008)
17. Murad Al Haj, J.G., Davis, L.S.: On partial least squares in head pose estimation: how to simultaneously deal with misalignment. In: CVPR (2012)
18. Murphy-Chutorian, E., Doshi, A., Trivedi, M.: Head pose estimation for driver assistance systems: a robust algorithm and experimental evaluation. In: Intelligent Transportation Systems Conference, ITSC. IEEE, pp. 709–714 (2007)
19. li Tian, Y., Brown, L., Connell, J., Pankanti, S., Hampapur, A., Senior, A., Bolle, R.: Absolute head pose estimation from overhead wide-angle cameras. In: IEEE International Workshop on Analysis and Modeling of Faces and Gestures (2003)
20. Beymer, D.J.: Face recognition under varying pose. In: CVPR, pp. 756–761 (1994)
21. Jamie Sherrah, S.G., Ong, E.-J.: Understanding pose discrimination in similarity space. In: BMVC (1999)
22. Balasubramanian, V.N., Ye, J., Panchanathan, S.: Biased manifold embedding: a framework for person-independent head pose estimation. In: CVPR (2007)
23. BenAbdelkader, C.: Robust head pose estimation using supervised manifold learning. In: ECCV (2010)
24. Huang, D., Storer, M., la Torre, F.D., Bischof, H.: Supervised local subspace learning for continuous head pose estimation. In: CVPR (2011)
25. Liu, X., Lu, H., Li, W.: Multi-manifold modeling for head pose estimation. In: ICIP (2010)
26. Valenti, R., Sebe, N., Gevers, T.: Combining head pose and eye location information for gaze estimation. IEEE Trans. Image Process. **21**(2), 802–815 (2012)
27. Ba, S.O., Odobez, J.-M.: Multiperson visual focus of attention from head pose and meeting contextual cues. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 101–116 (2011)
28. Nabati, M., Behrad, A.: 3d head pose estimation and camera mouse implementation using a monocular video camera. Signal Image Video Process. **6**, 1–6 (2012)
29. Wilson, H.R., Wilkinson, F., Lin, L., Castillo, M.: Perception of head orientation. Vis. Res. **40**(5), 459–472 (2000)
30. Rowley, H.A., Baluja, S., Kanade, T.: Rotation invariant neural network-based face detection. In: CVPR (1998)
31. Luhandjula, T., Monacelli, E., Hamam, Y., van Wyk, B., Williams, Q.: Visual intention detection for wheelchair motion. In: International Symposium on Visual Computing (ISVC) pp. 407–416 (2009)
32. Vinod Pathangay, S.D., Greiner, T.: Symmetry-based face pose estimation from a single uncalibrated view. In: 8th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 1–8 (2008)
33. Ma, B., Li, A., Chai, X., Shan, S.: Head yaw estimation via symmetry of regions, In: FG, pp. 1–6 (2013)
34. Gruendig, M., Hellwich, O.: 3d head pose estimation with symmetry based illumination model in low resolution video. In: Lecture Notes in Computer Science 3175 Springer, Berlin, pp. 45–53 (2004)
35. Harguess, J., Gupta, S., Aggarwal, J.: 3d face recognition with the average-half-face. In: ICPR, pp. 1–4 (2008)
36. Hattori, K., Matsumori, S., Sato, Y.: Estimating pose of human face based on symmetry plane using range and intensity images. In: ICPR, vol. 2, pp. 1183–1187 (1998)
37. Gui, Z., Zhang, C.: 3d head pose estimation using non-rigid structure-from-motion and point correspondence. In: IEEE TENCON (2006)
38. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, pp. 511–518 (2001)
39. Coxeter, H.: Projective Geometry, ser. Fuzzy Systems and Knowledge Discovery, 2nd Revised edition. Springer, Berlin (2003)
40. Stentiford, F.: Attention based facial symmetry detection. In: Proceedings of ICAPR (2005)
41. Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., Hall, M.: Multiclass alternating decision trees. In: ECML, Springer, Berlin, pp. 161–172 (2001)
42. Chen, W., Er, M.J., Wu, S.: Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. IEEE Trans. Syst. Man Cybern. **36**, 458–466 (2006)
43. Morency, L.-P., Whitehill, J., Movellan, J.: Monocular head pose estimation using generalized adaptive view-based appearance model Image. Vision Comput. **28**, 754–761 (2010)