Precise 3D Pose Estimation of Human Faces

Ákos Pernek[1,2] and Levente Hajder[1]

[1] *Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u. 13-17, 1111-Budapest, Hungary*

[2] *Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Műegyetem rakpart 3-9, 1111-Budapest, Hungary*

Keywords: Structure from Motion, Symmetric Reconstruction, Non-rigid Reconstruction, Facial Element Detection, Eye Corner Detection.

Abstract: Robust human face recognition is one of the most important open tasks in computer vision. This study deals with a challenging subproblem of face recognition: the aim of the paper is to give a precise estimation for the 3D head pose. The main contribution of this study is a novel non-rigid Structure from Motion (SfM) algorithm which utilizes the fact that the human face is quasi-symmetric. The input of the proposed algorithm is a set of tracked feature points of the face. In order to increase the precision of the head pose estimation, we improved one of the best eye corner detectors and fused the results with the input set of feature points. The proposed methods were evaluated on real and synthetic face sequences. The real sequences were captured using regular (low-cost) web-cams.

1 INTRODUCTION

The shape and appearance modeling of the human face and the fitting of these models have raised significant attention in the computer vision community. Till the last few years, the state-of-the-art method used for facial feature alignment and tracking was the active appearance model (AAM) (Cootes et al., 1998; Matthews and Baker, 2004). The AAM builds a statistical shape (Cootes et al., 1992) and grey-level appearance model from a face database and synthesizes the complete face. Its shape and appearance parameters are refined based on the intensity differences of the synthesized face and the real image.

Recently, a new model class has been developed called the constrained local model (CLM) (Cristinacce and Cootes, 2006; Wang et al., 2008; Saragih et al., 2009). The CLM model is in several ways similar to the AAM, however, it learns the appearance variations of rectangular regions surrounding the points of the facial feature set.

Due to its promising performance, we utilize the CLM for facial feature tracking. Our C++ CLM implementation is mainly based on the paper (Saragih et al., 2009), however, it utilizes a 3D shape model.

The CLM (so as the AAM) requires a training data set to learn the shape and appearance variations. We use a basel face model (BFM) (P. Paysan and R. Knothe and B. Amberg and S. Romdhani and T. Vetter, 2009)-based face database for training data set. The BFM is a generative 3D shape and texture model which also provides the ground-truth head pose and the ground-truth 2D and 3D facial feature coordinates. Our training database consists of 10k synthetic faces of random shape and appearance. The 3D shape model or the so-called point distribution model (PDM) of the CLM were calculated from the 3D facial features according to (Cootes et al., 1992).

During our experiments we have identified that the BFM-based 3D CLM produces low performance at large head poses (above 30 degree). The CLM fitting in the eye regions showed instability. We propose here two novelties: (i) Since the precision of eye corner points are of high importance for many vision applications, we decided to replace the eye corner estimates of the CLM with that of our eye corner detector. (ii) We propose a novel non-rigid structure from motion (SfM) algorithm which utilize the fact that human face is quasi-symmetric (almost symmetric).

2 EYE CORNER DETECTION

One contribution of our paper is a 3D eye corner detector inspired by (Santos and Proença, 2011). The main idea of our method is that the 3D information increases the precision of eye corner detection. (In our case, it is available due to 3D CLM fitting.) We

Pernek Á. and Hajder L..
Precise 3D Pose Estimation of Human Faces.
DOI: 10.5220/0004741706180625
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 618-625
ISBN: 978-989-758-009-3

created a 3D eye model which we align with the 3D head pose and utilize to calculate 2D eye corner location estimates. These estimates are further developed to generate the expected values for a set of features (Santos and Proença, 2011) supporting the eye corner selection.

2.1 Related Work

The eye corner detection has a long history. Several methods have been developed in the past years. A promising method is described in (Santos and Proença, 2011). This method applies pre-processing steps on the eye region to reduce noise and increase robustness: a horizontal rank filter is utilized for eyelash removal and eye reflections are detected and reduced as described in (He et al., 2009). The method acquires the pupil, the eyebrow and the skin regions by intensity-based clustering and the final boundaries are calculated via region growing (Tan et al., 2010). It also performs sclera segmentation based on the histogram of the saturation channel of the eye image (Santos and Proença, 2011). The segmentation provides an estimate on the eye region and thus, the lower and upper eyelid contours can be estimated as well. One can fit an ellipse or as well as polynomial curves on these contours which provide useful information for the real eye corner locations. The method generates a set of eye corner candidates via the well-known Harris corner detector (Harris, C. and Stephens, M., 1988) and defines a set of decision features. These features are utilized to select the real eye corners from the set of candidates. The method is efficient and provides good results even on low resolution images.

2.2 Iris Localization

To localize the iris region, we propose to use the intensity based eye region clustering method of (Tan et al., 2010). However, we also propose a number of updates to it. Tan et al. orders the points of the eye region by intensity and assigns the lightest p_1% and the darkest p_2% of these points to the initial candidate skin and iris regions, respectively. The initial candidate regions are further refined by means of region growing. The method is repeated iteratively until all points of the eye region are clustered. The result is a set of eye regions: iris, eyebrow, skin, and possibly degenerate regions due to reflections, hair and glass parts. In order to make the clustering method robust, they apply the image pre-processing steps described in Sec. 2.1 as well.

Our choice for the parameter p_1 is 30% as suggested by (Tan et al., 2010). However, we adjust the parameter p_2 adaptively. We calculate the average intensity (i_{avg}) of the eye region (in the intensity-wise normalized image) and set the p_2 value to $i_d * i_{avg}$ where i_d is an empirically chosen scale factor of value $\frac{1}{12}$. The adaptive adjustment of p_2 showed higher stability during test executions on various faces than the fixed set-up.

Another improvement is that we use the method of (Jankó and Hajder, 2012) for iris detection. The method is robust and operates stable on eye images of various sources. We assign the central region of the fitted iris to the iris region to improve the clustering result.

The result of the iris detection and the iris center and the eye region clustering is shown in Figure 1. Note that we focus on the clustering of the iris region and thus, only the iris and the residual regions are displayed.

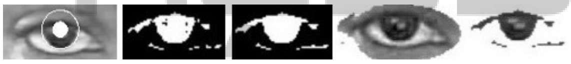

Figure 1: Iris and its center (of scale 0.4), initial/final iris, initial/final residual region (left to right).

2.3 Sclera Segmentation

The human sclera can be segmented by applying data quantization and histogram equalization on the saturation channel of the noise filtered eye region image (Santos and Proença, 2011). We adopt this method with some minor adaptations: we set the threshold for sclera segmentation as a function of the average intensity of the eye region (see Sec. 2.2). In our case, the scale factor of the average intensity is chosen as $\frac{1}{8}$.

We also limit the accepted dark regions to the ones which are neighboring to the iris. We have defined rectangular search regions at the left and the right side of the iris. Only the candidate sclera regions overlapping with these regions are accepted. The size and the location of the search regions are bound to the ellipse fitted on the iris edge (Jankó and Hajder, 2012). The sclera segmentation is displayed in Figure 2.

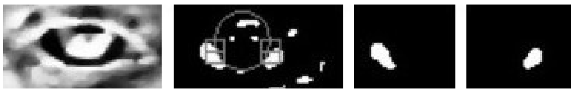

Figure 2: Homogenous sclera, candidate sclera regions and rectangular search windows, selected left and right side sclera segments (left to right).

2.4 Eyelid Contour Approximation

The next step of the eye corner detection is to approximate the eyelids. The curves of the upper and lower human eyelids intersect in the eye corners. Thus, the more precisely the eyelids are approximated, the more information we can have on the true locations of the eye corners.

The basis of the eyelid approximation is to create an eye mask. We create an initial estimate of this mask consisting of the iris and the sclera regions as described in Sections 2.2 and 2.3. This estimate is further refined by filling: the unclustered points which lay horizontally or vertically between two clustered points are attached to the mask. The filled mask is extended: we apply vertical edge detection on the eye image and try to expand the mask vertically till the first edge of the edge image. The extension is done within empirical limits derived from the eye shape, the current shape of the mask and the iris location (Jankó and Hajder, 2012).

The final eye mask is subject to contour detection. The eye mask region is scanned vertically and the up- and down most points of the detected contour points are classified as the points of the upper and lower eyelids, respectively.

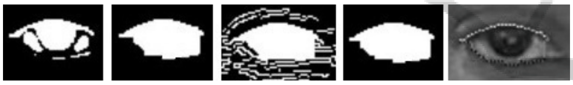

Figure 3: Eye mask, filled eye mask, vertical edge based extension, final eye mask, upper and lower eyelid contours (left to right).

2.5 Eye Corner Selection

We use the method of Harris and Stephens (Harris, C. and Stephens, M., 1988) to generate candidate eye corners as in (Santos and Proença, 2011). The Harris detector is applied only in the nasal and temporal eye corner regions (see Sec. 2.7). The detector is configured with low acceptance threshold ($\frac{1}{10}$ of the maximum feature response) so that it can generate a large set of corners. These corners are ordered in descending order by their Harris corner response and the first 25 corners are accepted. We constrain the acceptance with considerations of the Euclidean distance between selected eye corner candidates. A corner is not accepted as a candidate if one corner is already selected within its $1px$ neighborhood.

The nasal and the temporal eye corners are selected from these eye corner candidate sets. The decision is based on a set of decision features. These features are a subset of the ones described in (Santos and Proença, 2011): Harris pixel weight, internal angle, internal slope, relative distance, and, intersection of interpolated polynomials.

These decision features are utilized to discriminate false eye corner candidates. We convert them into probabilities indicating the goodness of an eye corner candidate. The goodness is defined as the deviation of the feature from its expected value. Finally, an aggregate score for each candidate is calculated with equally weighted probabilities except for the internal slope feature which we overweight in order to try selecting eye corners located under the major axis of the ellipse. One important deviation of our method from that of (Santos and Proença, 2011) is that we don't consider eye corner candidate pairs during the selection procedure. We found that the nasal eye corner is usually lower than the temporal one thus the line passing through them is not parallel to the major axis of the fitted ellipse.

2.6 3D Enhanced Eye Corner Detection

One major contribution of our paper is that our eye corner detector is 3D enhanced. A subset of the decision features (internal angle, internal slope and relative distance) in Sec. 2.5 requires the expected feature values in order to discriminate the false eye corner candidates. We define a 3D eye model and align it with the 3D head pose. We utilize the aligned model to calculate precise expected 2D eye corner locations and thus, expected features values as well.

Our 3D eye model consists of an ellipse modeling the one fitted on the eyelid contours and a set of parameters: p_1, p_2, p_3, p_4, and, b_a. Parameters p_1, p_2, p_3, and, p_4 denote the scalar projection of the eye corner positions w.r.t. ellipse center and the major and minor axes. Parameter b_a defines the bending angle: the expected temporal eye corner is rotated around the minor axis of the ellipse. Let us denote head yaw and pitch angles as: lr_a and ud_a, respectively (note that we do not model head roll). Assuming that the ellipse center is the origin of our coordinate system, the expected locations of the temporal and the nasal eye corners (of the right eye) can be written as: $c_t = (p_1 cos(lr_a - b_a)A, p_3 cos(ud_a)B)$ and $c_n = (p_2 cos(lr_a)A, p_4 cos(ud_a)B)$, respectively.

The ratio of the major A and minor B axes is a flexible parameter r_a and is unknown. However, it can be learnt from the first few images of a face video sequence (assuming frontal head pose).

In our framework the parameters p_1, p_2, p_3, p_4, and, b_a are chosen as -0.9, 0.9, -0.15, -0.5, and, $\frac{\pi}{12}$, respectively.

The eye model is visualized in Figure 4.

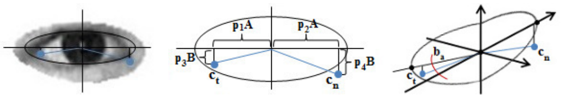

Figure 4: Eye corners and fitted ellipse, 2D eye model (b_a = 0), 3D eye model (left to right).

2.7 Enhanced Eye Corner Regions

Our method applies an elliptic mask in order to filter invalid eye corner candidates. We rotate this elliptic mask in accordance with the 3D head pose and we also shift the he rectangular eye corner regions vertically in accordance with the slope of the major axis of the ellipse (fitted on the eyelid contours). This allows us a better model for the possible location of the candidate eye corners (see Figure 5).

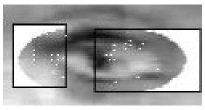

Figure 5: Rectangular eye corner regions masked by the 3D elliptic mask. The white dots denote the available eye corner candidates.

3 NON-RIGID STRUCTURE FROM MOTION

The other major contribution of our paper is a novel non-rigid and symmetric reconstruction algorithm which solves the structure from motion problem (SfM). Our proposed algorithm incorporates non-rigidity and symmetry of the object to reconstruct. The proposed method is applicable for both symmetric or quasi-symmetric (almost symmetric) objects.

This section summarizes the main aspects of the non-rigid reconstruction. The input of the reconstruction is P tracked feature points of a non-rigid object across F frames. (In our case, they are calculated by 3D CLM tracking and the proposed 3D eye corner detection method.)

Usually, the SfM-like problems are solved by matrix factorization. For rigid objects, the well-known solutions are based on the classical Tomasi-Kanade factorization (Tomasi, C. and Kanade, T., 1992). Our approach, similarly to the work of Tomasi and Kanade (Tomasi, C. and Kanade, T., 1992), assumes weak-perspective projection. We proposed an alternation-based method (Hajder et al., 2011; Pernek et al., 2008) in 2008 that divides the factorization method into subproblems that can be solved optimally. We extend our solution to the nonrigid case here.

3.1 Non-rigid Object Model

A rigid object in the SfM methods is usually modeled by its 3D vertices. We model the non-rigidity of the face by K so-called key (rigid) objects. The non-rigid shape of each frame is estimated as a linear combination of these key objects.

The non-rigid shape of an object at the j^{th} frame can be written as:

$$S^j = \sum_{i=1}^{K} w_i^j S_i \tag{1}$$

where w_i^j are the non-rigid weight components for the j^{th} frame and the k^{th} key object (k = $[1 \,..\, K]$) is written as:

$$S_k = \begin{bmatrix} X_{1,k} & X_{2,k} & \cdots & X_{P,k} \\ Y_{1,k} & Y_{2,k} & \cdots & Y_{P,k} \\ Z_{1,k} & Z_{2,k} & \cdots & Z_{P,k} \end{bmatrix} \tag{2}$$

3.2 Weak-perspective Projection Model

To estimate the key objects and their non-rigid weight components, the tracked $2D$ feature points has to be linked to the $3D$ shapes. This link is the projection model. Due to its simplicity, the weak-perspective projection is a good choice to express the relationship between the $3D$ shape and the tracked $2D$ feature points. It is applicable when the depth of the object is significantly smaller than the distance between the camera and the object center. Thus, the weak-perspective projection is applicable for webcam video sequences, which is in the center of our interest.

The weak-perspective projection equation is written as follows:

$$\begin{bmatrix} u_i^j \\ v_i^j \end{bmatrix} = q^j R^j \begin{bmatrix} X_i^j \\ Y_i^j \\ Z_i^j \end{bmatrix} + t^j \tag{3}$$

where q^j is the scale parameter, R^j is the 2 x 3 rotation matrix, t^j = $[u_0^j,\ v_0^j]^T$ is the 2 x 1 translation vector, $[u^j,\ v^j]^T$ are the projected $2D$ coordinates of the i^{th} $3D$ point $[X_i^j, Y_i^j, Z_i^j]$ of the j^{th} frame.

During non-rigid structure reconstruction, the q^j scale parameters can be accumulated in the non-rigid weight components. For this reason we introduce the notation $c_i^j = q^j w_i^j$. Utilizing this assumption, the weak-perspective projection for a non-rigid object in the j^{th} frame can be written as:

$$\begin{aligned} W^j = \begin{bmatrix} u_1^j & \cdots & u_P^j \\ v_1^j & \cdots & v_P^j \end{bmatrix} &= R^j S^j + t^j \\ &= R^j \left(\sum_{i=1}^{K} c_i^j S_i \right) + t^j \end{aligned} \tag{4}$$

where W^j is the so-called measurement matrix.

The projection equation can be reformulated as

$$W = MS = [R|t][S,1]^T \tag{5}$$

where W is the measurement matrix of all frames: $W = \left[{W^1}^T \dots {W^F}^T \right]^T$. R is the non-rigid motion matrix and t the translation vector of all frames:

$$M = \begin{bmatrix} c_1^1 R^1 & \cdots & c_K^1 R^1 \\ \vdots & \ddots & \vdots \\ c_1^F R^F & \cdots & c_K^F R^F \end{bmatrix} \quad t = \begin{bmatrix} t_1 \\ \vdots \\ t_F \end{bmatrix} \tag{6}$$

and M is the non-rigid motion matrix of all frames. and S is defined as a concatenation of the K key objects: $S = \begin{bmatrix} S_1^T & \dots & S_K^T & 1 \end{bmatrix}^T$

3.3 Optimization

Our proposed non-rigid reconstruction method minimizes the so-called re-projection error:

$$||W - MS||_F^2 \tag{7}$$

The key idea of the proposed method is that the parameters of the problem can be separated into independent groups, and the parameters in these groups can be estimated optimally in the least squares sense.

The parameters of the proposed algorithm are categorized into three groups: (i) camera parameters: rotation matrices (R^j) and translation parameters (t^j), (ii) key object weights (c_i^j), and (iii) key object parameters (S_k). These parameter groups can be calculated optimally in the least square sense. The method refines them in an alternating manner. Each step reduces the reprojection error and is proven to converge in accordance with (Pernek et al., 2008). The steps of the alternation are described here, the whole algorithm is overviewed in Alg. 1.

Rt-step. The Rt-step is very similar to the one proposed by Pernek et al. (Pernek et al., 2008). The camera parameters of the frames can be estimated one by one: they are independent of each other. If the j^{th} frame is considered, the optimal estimation can be given computing the optimal registration between the 3D vectors in matrices W and $\sum_{i=1}^{K} c_i^j S_i$. The optimal registration is described in (Arun et al., 1987). A very important remark is that the scale parameter cannot be computed in this step contrary to the rigid factorization proposed in (Pernek et al., 2008).

S-step. The cost function in Eq 7 depends linearly on the values of the structure matrix S. The optimal solution for S is[1] $S = M^\dagger W$. However, this is true only for non-symmetric points. We assume that many of the face feature points has a pair. If $s_{i,k}$ and $s_{j,k}$ are feature point pairs then $s_{i,k}^x = -s_{j,k}^x$, $s_{i,k}^y = s_{j,k}^y$, and $s_{i,k}^z = s_{j,k}^z$ if the plane of the symmetry is $x = 0$. ($s_{i,k}^x$, $s_{i,k}^y$, $s_{i,k}^z$ denotes the coordinates of the i^{th} point in key object k.) The corresponding parts of the cost function: $||W_i - [m_1, m_2, m_3][s_{i,x}, s_{i,y}, s_{i,z}]||$ and $\left|\left|W_i - [-m_1, m_2, m_3][s_{i,k}^x, s_{i,k}^y, s_{i,k}^z]\right|\right|$, where m_1, m_2, and m_3 are the columns of motion matrix M, and W_i and W_j the corresponding row pairs of measurement matrix W. The optimal estimation can be computed as

$$s_i = \begin{bmatrix} m_1 & m_2 & m_3 \\ -m_1 & m_2 & m_3 \end{bmatrix}^\dagger \begin{bmatrix} W_i \\ W_j \end{bmatrix} \tag{8}$$

$s_{i,x} = 0$ for non-symmetric points, thus, the linear estimation is simpler with respect to common rigid factorization since only two coordinates have to be calculated. Remark that S-step must be repeated for all key object.

c-step. The goal of the c-step is to compute parameters c_i^j optimally in the least squares sense if all the other parameters are known. Fortunately, this is a linear problem, the optimal solution can be easily obtained by solving an overdetermined one-parameter inhomogeneous linear system. (Hartley and Zisserman, 2003). Remark that the weight parameters for frame j must be calculated independently from those of other frames.

Algorithm 1: Non-rigid And Symmetric Reconstruction.

```
$k \leftarrow 0$
$R,t,c,S \leftarrow$ Initialize()
$R \leftarrow$ Complete(R)
$S \leftarrow$ MakeSymmetric(S)
$S \leftarrow$ CentralizeAndAlign(S)
repeat
    $k \leftarrow$ k + 1
    $S \leftarrow$ S-step(W,R,t,c)
    $c \leftarrow$ c-step(W,R,t,S)
    $(R,t) \leftarrow$ Rt-step(W,c,S)
    $W \leftarrow$ Complete(W,R,t,c,S)
until Error(W,R,w,S,t) $< \varepsilon$ or $k \geq k_{max}$
```

Completion. Due to the optimal estimation of the rotation matrix, an additional step must be included before every step of the algorithm as it is also carried out in (Pernek et al., 2008). The Rt-step yields 3 × 3 orthogonal matrices, but the matrices R^j used in non-rigid factorization are of size 2 × 3. Thus, the 2 × 3 matrix has to be completed with a third row: it is perpendicular to the first two rows, its length is the average of those. The completion should be done for the measurement matrix as well. Let r_3^j, w_3^j, and, t_3^j

[1] $\dagger$ denotes the Moore-Penrose pseudoinverse. In our case, $M^\dagger = \left(M^T M\right)^{-1} M^T$.

denote the third row of the completed rotation, measurement, and, translation at the j^{th} frame, respectively. The completion is written as:

$$w_3^j \leftarrow r_3^j \left(\sum_{i=1}^{K} c_i^j S_i \right) + t_3^j \qquad (9)$$

3.4 Initialization of Parameters

The proposed improvement is an iterative algorithm. If good initial parameters are set, the algorithm converges to the closest (local or global) minimum, because each step is optimal w.r.t. reprojection error defined in Eq. 5. One of the most important problem is to find a good starting point for the algorithm: camera parameters (rotation and translation), weight components, and, key objects.

We define the structure matrices of the K key objects w.r.t. the rigid structure as $S_1 \approx S_2 \cdots \approx S_K \approx S_{rig}$, where S_{rig} denotes the rigid structure. In our case S_{rig} is the mean shape of the 3D CLM's shape model. The approximation sign '$\approx$' means that a little random noise is added to the elements of S_i with respect to S_{rig}. This is necessary, otherwise the structure matrices remain equal during the optimization procedure. We set w_i^j weights to be equal to the weak-perspective scale of the rigid reconstruction. The initial rotation matrices R^j are estimated via calculating the optimal rotation (Arun et al., 1987) between W and S_{rig}.

The CLM based initialization is convenient for us, however, the initialization can be performed in many ways such as the ones written in (Pernek et al., 2008) or (Xiao et al., 2004).

We also enforce the symmetry of the initial key objects. We calculate the symmetry plane of them and relocate their points so that the single points lay on, the pair points are symmetrical to the symmetry plane. The plane of the symmetry is calculated as follows. The normal vector of the plane should be parallel to the vector between the point pairs, and the plane should contain the midpoint of point pairs. Therefore, the normal vector of the symmetry plane is estimated as the average of the vectors between the point pairs, and the position of the plane is calculated from the midpoints. Then the locations of the feature point of key objects are recalculated in order to fulfill the symmetricity constraint. (And the single points are projected to the symmetry plane.)

4 TEST EVALUATION

The current section shows the test evaluation of the 3D eye corner detection and the non-rigid and symmetric reconstruction.

For evaluation purposes we use a set of real and synthetic video sequences which contain motion sequences of the human face captured at a regular face - web camera distance. The subjects of the sequences perform a left-, a right-, an up-, and, a downward head movement of at most 30-40 degrees.

The synthetic sequences are based on the BFM (P. Paysan and R. Knothe and B. Amberg and S. Romdhani and T. Vetter, 2009)-based face database.

4.1 Empirical Evaluation

This section visualizes the results of the 3D eye corner detection on both real and synthetic (see Figure 6) video sequences. The section contains only empirical evaluation of the results. The sub-figures display the frontal face (first column) in big, and the right (middle column) and left (right column) eyes in small at different head poses.

The frontal face images show many details of our method: the black rectangles define the face and the eye regions of interest (ROI). The face ROIs are detected by the well-known Viola-Jones detector (Viola and Jones, 2001), however, they are truncated horizontally and vertically to cut insignificant regions such as upper forehead. The eye ROIs are calculated relatively to the truncated face ROIs. The blue rectangles show the detected (Viola and Jones, 2001) eye regions and the eye corner ROIs as well. The eye region detection is executed within the boundaries of the previously calculated eye ROIs. The eye corner ROIs are calculated within the detected eye regions with respect to the location and size of the iris. The red circles show the result of the iris detection (Jankó and Hajder, 2012) which is performed within the detected eye region. Blue polynomials around the eyes show the result of the polynomial fitting on the eyelid contours. The green markers show the points of the 3D CLM model. The yellow markers at eye corners display the result of the 3D eye corner detection.

The right and the left eye images of the sub-figures display the eyes at maximal left, right, up, and, down head poses in top-down order, respectively. The black markers show the selected eye corners. The grey markers show the available set of candidate eye corners.

The test executions show that the 3D eye corner detection works very well on our test sequences. The eye corner detection produces good results even for blurred images at extreme head poses.

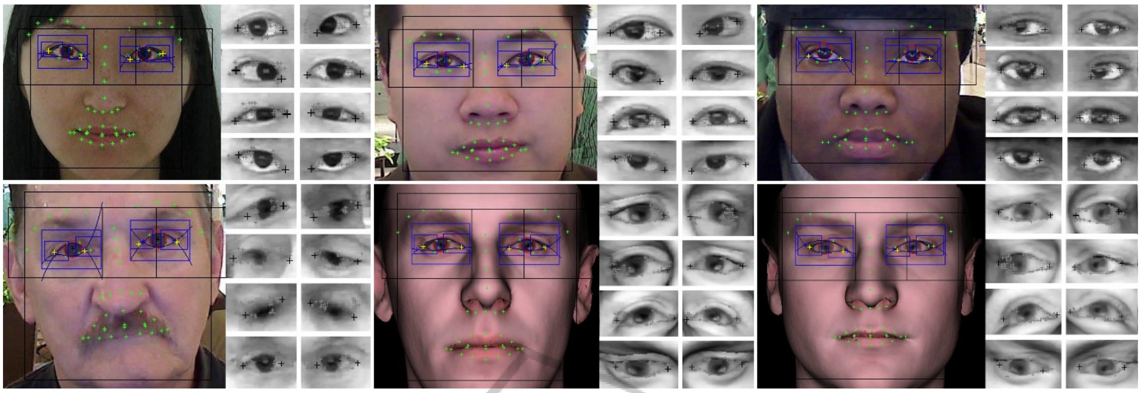

Figure 6: Real and synthetic test sequences.

4.2 2D/3D Eye Corner Detection

This sections evaluates the precision of the eye corners calculated by the 3D CLM model, our 3D eye corner detector and its 2D variant. In the latter case we simply fixed the (rotation) parameters of our 3D eye corner detector to zero in order to mimic continuous frontal head pose.

To measure the eye corner detection accuracy, we used 100 BFM-based video sequences . Thus, the ground-truth 2D eye corner coordinates were available during our tests.

The eye corner detection accuracy we calculated as the average least square error between the ground-truth and the calculated eye corners of each image of a sequence. The final results displayed in Table 1 show the average accuracy for all the sequences in pixels and the improvement percentage w.r.t the 3D CLM error.

Table 1: Comparison of the 3D CLM, and the 2D/3D eye corner (EC) detector.

Type	3DCLM	2DEC	3DEC
Accuracy	0.5214	0.4201	0.4163
Improve	0.0	19.42	20.15

The results show that the 3D eye corner detection method performs the best on the test sequence. It is also shown that both the 2D and the 3D eye corner detectors outperform the CLM method. This is due to the fact that our 3D CLM model is sensitive to extreme head pose and it tends to fail in the eye region. An illustration of the problem is displayed in Figure 7.

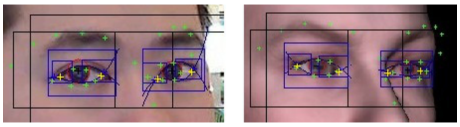

Figure 7: CLM fitting failure (green markers around eye and eyebrow regions) at extreme head poses.

4.3 Non-rigid Reconstruction

In this section we evaluate the accuracy of the non-rigid and symmetric reconstruction. For our measurements, we use the same synthetic database as in Section 4.2. The basis of the comparison is a special feature set. This feature set consists of the points tracked by our 3D CLM model. However, due to the eye region inaccuracy described in Section 4.2, we drop the eye points (two eye corners and four more points around the iris and eyelid contour intersections) and use the eye corners computed by our 3D eye corner detector.

The non-rigid reconstruction yields the refined cameras and the refined 2D and 3D feature coordinates of each image of a sequence. The head pose can be extracted from the cameras. We selected the head pose and the 2D and 3D error as an indicator of the reconstruction quality. The ground-truth head pose, 2D and 3D feature coordinates are acquired from the BFM.

We calculated the head pose error as the average least square error between the ground-truth head pose and the calculated head pose of each image of a sequence. The 2D and 3D error we define as the average registration error (Arun et al., 1987) of the centralized and normalized ground truth and the computed 2D and 3D point sets of each image of the sequence.

The compared methods are the 3D CLM, our non-rigid and symmetric reconstruction and its generic non-rigid variant (symmetry constraint not enforced).

The results displayed in Table 2 show the average accuracy for all the test sequences in degrees and the improvement percentage w.r.t the 3D CLM model. The generic (Gen) and the symmetric (Sym) reconstruction methods have been evaluated with different number of non-rigid components (K) as well.

It is seen that by optimizing a huge amount of parameters, lower reprojection error values can be

Table 2: Comparison of the 3D CLM, the symmetric and non-rigid and the generic non-rigid reconstruction.

Type	3DCLM	Gen (K=1)	Gen (K=5)	Gen (K=10)	Sym (K=1)	Sym (K=5)	Sym (K=10)
2D Err.	2.73162	2.72951	2.77952	2.78255	2.72853	2.72853	2.72853
2D Impr.	0.0	0.0772	-1.7535	-1.8644	0.1131	0.1131	0.1131
3D Err.	1.03933	0.89338	4.56524	2.50865	0.880928	0.880915	0.880910
3D Impr.	0.0	14.0427	-339.24	-141.37	15.2407	15.2420	15.2425
Pose Err.	0.3443	0.2756	0.5317	0.5974	0.2829	0.2807	0.2908
Pose Impr.	0.0	19.9535	-54.429	-73.5115	17.8332	18.4722	15.5387

reached, however, without the symmetry constraint this can yield an invalid solution. Our proposed symmetric method keeps stable even with a high number of non-rigid components (K).

One can also see that the head pose error of our proposed method outperforms the 3D CLM, however, the generic rigid reconstruction (Gen (K=1)) provides the best results. We believe that the rigid model can better fit to the CLM features due to the lack of the symmetry constraint.

On the other hand the best 3D registration errors are provided by our proposed method. It shows again that the symmetry constraint does not allow the reconstruction to converge toward a solution with less reprojection error, but with a deviated 3D structure.

The table also shows that the 2D registration is best by our proposed method, however, the gain is very little and the performance of the methods are basically similar.

5 CONCLUSIONS

It has been shown in this study that the precision of the human face pose estimation can be significantly enhanced if the symmetric (anatomical) property of the face is considered. The novelty of this paper is twofold: we have proposed here an improved eye corner detector as well as a novel non-rigid SfM algorithm for quasi-symmetric objects. The methods are validated on both real and rendered image sequences. The synthetic test were generated by the basel face model, therefore, ground truth data have been available for evaluating both our eye corner detector and non-rigid and symmetric SfM algorithms. The test results have convinced us that the proposed methods outperforms the compared ones and a precise head pose estimation is possible for real web-cam sequences even if the head is rotated by large angles.

REFERENCES

Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *PAMI*, 9(5):698–700.

Cootes, T., Taylor, C., Cooper, D. H., and Graham, J. (1992). Training models of shape from sets of examples. In *BMVC*, pages 9–18.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *PAMI*, pages 484–498. Springer.

Cristinacce, D. and Cootes, T. F. (2006). Feature detection and tracking with constrained local models. In *BMVC*, pages 929–938.

Hajder, L., Pernek, Á., and Kazó, C. (2011). Weak-perspective structure from motion by fast alternation. *The Visual Computer*, 27(5):387–399.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151.

Hartley, R. I. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

He, Z., Tan, T., Sun, Z., and Qiu, X. (2009). Towards accurate and fast iris segmentation for iris biometrics. *PAMI*, 31(9):1670–1684.

Jankó, Z. and Hajder, L. (2012). Improving human-computer interaction by gaze tracking. In *Cognitive Infocommunications*, pages 155–160.

Matthews, I. and Baker, S. (2004). Active appearance models revisited. *IJCV*, 60(2):135–164.

P. Paysan and R. Knothe and B. Amberg and S. Romdhani and T. Vetter (2009). A 3D Face Model for Pose and Illumination Invariant Face Recognition. *AVSS*.

Pernek, Á., Hajder, L., and Kazó, C. (2008). Metric reconstruction with missing data under weak perspective. In *BMVC*. British Machine Vision Association.

Santos, G. M. M. and Proença, H. (2011). A robust eye-corner detection method for real-world data. In *IJCB*, pages 1–7. IEEE.

Saragih, J. M., Lucey, S., and Cohn, J. (2009). Face alignment through subspace constrained mean-shifts. In *ICCV*.

Tan, T., He, Z., and Sun, Z. (2010). Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition. *IVC*, 28(2):223–230.

Tomasi, C. and Kanade, T. (1992). Shape and Motion from Image Streams under orthography: A factorization approach. *IJCV*, 9:137–154.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR*, 1:I–511–I–518 vol.1.

Wang, Y., Lucey, S., and Cohn, J. (2008). Enforcing convexity for improved alignment with constrained local models. In *CVPR*.

Xiao, J., Chai, J.-X., and Kanade, T. (2004). A Closed-Form Solution to Non-rigid Shape and Motion Recovery. In *ECCV*, pages 573–587.