Precise 3D Pose Estimation of Human Faces

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 discussed methods are evaluated on real and synthetic face sequences. The synthetic ones were generated by the Basel Face Model (BFM) while the real sequences were captured using regular (low-cost) web-cams.

1 INTRODUCTION

The shape and appearance modelling 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 for facial feature alignment and tracking was the so-called 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 iteratively based on the intensity differences of the synthesized and the real face.

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 3D shape model.

The CLM (so as the AAM) requires a training data set to learn the shape and appearance variations. We use the Basel Face Model (BFM) (P. Paysan and R. Knothe and B. Amberg and S. Romdhani and T. Vetter, 2009) to generate this 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). The classifiers of the individual features of the CLM has been taught from rectangular regions of size 11x11 centered at 2D facial features. The ratio of negative examples for the classifier generation was set to 5.

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 improvement is that the 3D information (provided by 3D CLM fitting) can increase the precision of eye corner detection. We created a 3D eye model which is rotated in accordance with the 3D head pose estimates. The rotated eye model is used to generate more accurate predictions for the true eye corner locations.

The next sections summarize our proposed

method and the main steps of the eye corner detection: image pre-processing, iris localisation, sclera segmentation, eyelid contour approximation, candidate eye corner set generation, and, 2D and 3D eye corner selection by decision features.

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 eye brow 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 Eye Pre-processing Steps

The eye regions of human face are prone to containing errors: reflections and occlusions can harden the image processing task. To achieve more robust results, these artifacts shall be handled.

One common problem is the occlusion caused by eyelashes. This can be reduced by filtering the eye region with a 1-D horizontal rank-p filter as described in (He et al., 2009). The 1-D rank-p filter is a nonlinear filter which aligns a sliding window of width L centered around the current image point. It orders the image intensities within this sliding window in a descending order and replaces the current pixel intensity with the p^{th} image intensity value of the ordered intensities. In our adaptation L=5 and p=2 parameters were selected. By setting p=2 the usually very dark eyelashes can be efficiently removed from the eye image as seen in Figure 1.

Figure 1: Eye before and after eyelash removal

Another common problem is that the eye images are prone to specular reflections. These reflections introduce several problems to eye processing. Intensity based eye region clustering tends to fail due the interruption of continuous eye region elements: iris, eye brow, skin (Tan et al., 2010). He et al. also reported problems of Adaboost-cascade learning (He et al., 2009) for human iris due to reflections. To reduce the effect of reflection we have adopted the reflection removal method of (He et al., 2009). Their proposed method classifies the top 5% brightest points of the eye region image as reflection. They apply a bilinear interpolation to all points of the reflection regions calculated from four so-called envelope points (P_l, P_r, P_t, P_d). These points are along the horizontal and vertical lines crossing through the current reflection point and the closest to the reflection region of this point (with respect to an adaptive separator of length L=2). The reflection removal is illustrated in Figure 2.

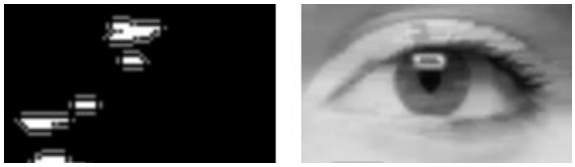

Figure 2: Reflection mask (left side): the white regions show the reflections, the grey lines show the envelope points; The eye after reflection removal (right side)

2.3 Iris Localisation

To localise the iris region, we propose to use the intensity based eye region clustering method of (Tan et al., 2010). However, we as well as 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. They calculate the standard deviation (d_R) and the average grey level (g_R) of each candidate region R and measures the distance of unclustered points P (with grey level g_P) from the region: $d = \frac{|g_P - g_R|}{d_R}$. If the distance is under a certain threshold T_R and the point P can be connected to the re-

gion R via eight-neighbour connectivity, the point is selected as a new point within the region. 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 preprocessing steps described in Sec. 2.2 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 normalised 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 robust method of (Jankó and Hajder, 2012) for iris detection. Tan et al. selects the iris region by semantical considerations, e.g. the usual shape of the iris region. The method of Jankó and Hajder convolves the image with a special convolution mask to find a rough initial estimate of the iris location. This estimate is further refined by optimizing an energy function created for iris detection. The result of the optimization is an ellipse which fits to the horizontal edges of the iris. The method is robust and operates stable on eye images of various sources. Note that we also utilize the fitted ellipse to explicitly expand the iris region: we add the iris center (derived from the scaling of the fitted ellipse with a factor of 0.4) to the iris region. This improves the clustering result in some cases when the iris region is poorly detected. The result of the iris detection and the iris center are visualized in Figure 3.

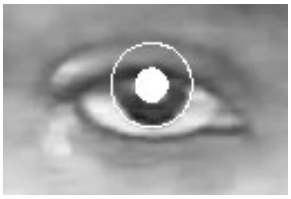

Figure 3: Iris detection and iris center

The result of the eye region clustering is shown in Figure 4. Note that we focus on the clustering of the iris region and thus, only the iris and the residual regions are displayed.

2.4 Sclera Segmentation

The saturation values of the human sclera are very low. Data quantization and histogram equalization can be applied on the saturation channel of the noise

Figure 4: Initial iris and residual region estimates (left side), final iris and residual region estimates (right side)

filtered (see Sec. 2.2) eye region image. In the resulting image the sclera is more homogenous and has significantly lower intensities than the other regions. Thus, it can be segmented by empirically set thresholds. We adopt the method of (Santos and Proença, 2011) for sclera segmentation, however, with some minor adaptations.

We set the threshold for the sclera segmentation as a function of the average intensity of the eye region (see Sec. 2.3). In this case, the scale factor of the average intensity is chosen as $\frac{1}{8}$.

Figure 5: Homogenous sclera in the histogram image

One issue we have identified with the above method is that homogenous and dark intensity regions of the histogram can occur outside of the sclera region. Thus, we limit the accepted dark regions to the ones which are neighbouring to iris. We have defined rectangular regions at the left and the right side of the iris. Only the candidate sclera regions are accepted which have an intersection with these rectangular regions. 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 6.

2.5 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

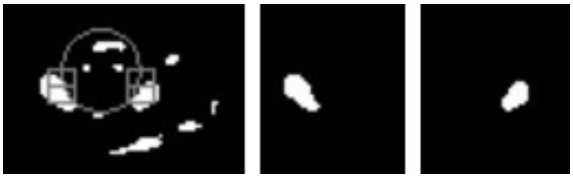

Figure 6: Candidate sclera regions and the rectangular search windows neighbouring the iris (first column). The selected left and right side sclera segments (second and third column)

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 2.3 and 2.4. 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).

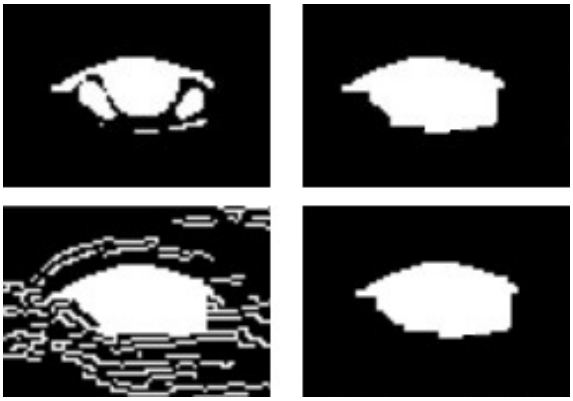

Figure 7: Eye mask (top-left), Filled eye mask (top-right), Edge based extension (bottom-left), Final eye mask (bottom-right)

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

2.6 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. 2.8). The detector is configured with low acceptance threshold ($1/10$ 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 eye corner if one corner is already selected within its $1px$ neighbourhood.

Figure 8: Upper and lower eyelid contours

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).

a, Harris pixel weight

The candidate eye corner points were generate by the Harris corner detector and thus, the Harris response is good indicator of the quality of a candidate point (Harris, C. and Stephens, M., 1988).

b, Internal angle

Let $e_c = (x_e, y_e)$ define the center, A and B define the major and the minor axes, and γ define the rotation of the ellipse (E as seen in Figure 9) fitted on the eyelid contours, respectively.

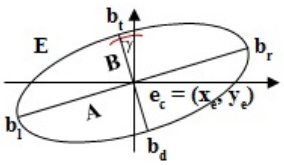

Figure 9: Ellipse model

The upper and lower points of the ellipse E alongside its minor axis are written:

$$\begin{aligned} b_t &= (x_e + sin(\gamma)B, y_e - cos(\gamma)B) \\ b_d &= (x_e - sin(\gamma)B, y_e + cos(\gamma)B) \end{aligned} \quad (1)$$

For each candidate eye corner points c_i let u and v denote the vectors $c_i - b_t$ and the $c_i - b_d$, respectively. The internal angle of the vectors u and v ($arccos\left(\frac{\langle u,v\rangle}{\|u\|\,\|v\|}\right)$) is a good indicator for the eye cor-

ner location (too small or too big angle indicates irrealistic location of the eye corner candidate). The internal angle feature is visualized in Figure 10.

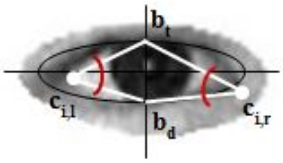

Figure 10: Ellipse internal angle feature

c, Internal slope
Let m_1 define the slope of the major axis of the ellipse E. Let $m_2 = \frac{y_e - y_i}{x_e - x_i}$ define the slope of the line connecting the candidate eye corner point $c_i = (x_i, y_i)$ and the ellipse center e_c. The angle between the slopes m_1 and m_2 can be written as: $arctan\left(\frac{m_2 - m_1}{1 + m_1\ m_2}\right)$. Our experiences show that both the nasal and the temporal eye corners lay in most cases under the major axis of the ellipse and the location of the nasal corner is lower. Thus, the internal slope usually defines a negative angle. The internal slope feature is visualized in Figure 11.

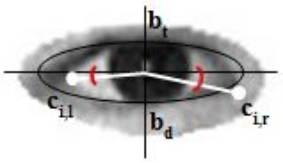

Figure 11: Ellipse internal slope feature

d, Relative distance
This feature considers the distance between the candidate point c_i and the ellipse center e_c divided by the length of the major axis A: $\frac{\sqrt{(x_i - x_e)^2 + (y_i - y_e)^2}}{A}$.

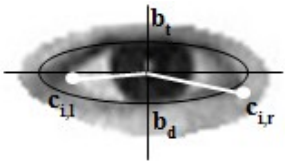

Figure 12: Ellipse distance feature

e, Intersection of interpolating polynomials The intersection of the polynomial curves fitted on the upper and the lower eyelid contours define the nasal and the temporal eye corners. We fitted second and third order polynomials on the upper and the lower eyelids as in (Santos and Proença, 2011), respectively.

We used the above described set of decision features to calculate an aggregate score for each candidate eye corner. The aggregate score is calculated

Figure 13: Ellipse polynomial intersection feature

with equal feature weights except for the internal slope feature which we overweight to tend to select 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. Santos and Proença state that the line passing through a high score nasal and temporal eye corner candidate pair shall have the same slope with the major axis of the fitted ellipse. In our case this consideration seemed not true and thus, we dropped it.

2.7 3D Enhanced Eye Corner Detection

One major contribution of our paper is that our eye corner detector is 3D enhanced. The decision features in Sec. 2.6 consider only 2D expectations on the eye corner locations. In our framework 3D information such as head pose is available due to the application of the 3D CLM model for facial feature tracking. Our expectation is that 3D information can raise the accuracy of eye corner detection. Thus, we defined a 3D eye model which we rotate in accordance with the 3D head pose and utilize it to calculate accurate expected values for the decision features.

For better understanding let's consider the 2D variant of the proposed eye model. It consists of an ellipse modelling the one fitted on the eyelid contours and a set of parameters c_1, c_2, c_3, and, c_4 which denote signed ratios controlling the relative distance of the expected eye corner locations to the ellipse center with respect to the length of the major and minor axes of the ellipse. The γ rotation parameter of the ellipse is assumed to be zero. 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: $c_t = (c_1 A, c_3 B)$ and $c_n = (c_2 A, c_4 B)$. For the left eye, the model has to be mirrored.

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).

The 2D eye model is visualized in Figure 14.

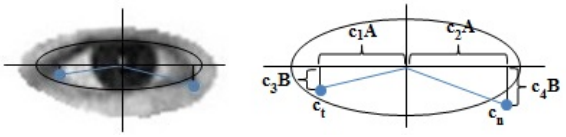

Figure 14: Eye corners and fitted ellipse (left side) and 2D eye model (right side)

The 3D eye model has close similarities with its 2D variant. One significant difference is that the 3D model is bent: the expected temporal eye corner is rotated around the minor axis of the ellipse (with bending angle: b_a) in order to model its greater depth compared to the nasal one. Let us denote head yaw and pitch angles as: lr_a and ud_a, respectively (Note that we do not model head roll). As the 3D eye model is aligned with the head pose by means of 3D rotations, the expected eye corner locations (of the right eye) are written as $c_t = (c_1 cos(lr_a - b_a)A, c_3 cos(ud_a)B)$ and $c_n = (c_2 cos(lr_a)A, c_4 cos(ud_a)B)$. For the left eye, the model has to be mirrored.

The following Figure 15 shows the details of the eye model.

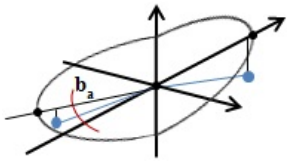

Figure 15: 3D eye model

In our framework the parameters c_1, c_2, c_3, c_4, and, b_a are chosen as -0.9, 0.9, -0.15, -0.5, and, $\frac{\pi}{12}$, respectively.

2.8 3D Enhanced Eye Corner Region

In this paper we apply an elliptic mask in order to better filter invalid eye corner candidates. We align this elliptic mask with the 3D head pose so that it adopts to the changing shape of the eye during head movements. This deformation of the mask is similar to the deformation of the 3D eye model as described in Sec. 2.7.

Please also note that the rectangular eye corner ROIs are slightly shifted vertically in accordance with the slope of the major axis of the ellipse fitted on the eyelid contours. This allows us the better model the ROIs for the candidate eye corners.

Our adaptive eye corner ROI generation is displayed in Figure 16.

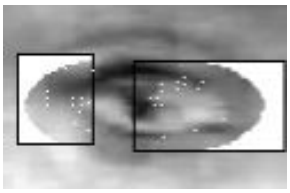

Figure 16: Rectangular eye ROIs masked by the 3D elliptic mask

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.

3.1 Related Work

The structure from motion (SfM) is a popular and wide area of computer vision. The aim of SfM is to estimate the camera parameters and the 3D structure from a 2D image sequence. Usually, it is solved by matrix factorization. The main idea is that the measurement matrix (2D coordinates of tracked points for all images of the sequence) can be factorized into rank 4 submatrices. The factorization result can be transformed into a metric reconstruction consisting of the real 3D structure and camera parameters.

The original factorization method for orthographic projection was published by Tomasi and Kanade (Tomasi, C. and Kanade, T., 1992). The method has later been extended to the weak-perspective, the para-perspective, and, the perspective cases (Weinshall and Tomasi, 1995; Poelman and Kanade, 1997; Sturm and Triggs, 1996).

Alternation based approaches were also developed for the factorization problem such as (Hajder et al., 2011; Pernek et al., 2008).

The factorization problem has also been extended to the non-rigid case. A common solution is to model the non-rigidity of an object by a linear combination of a number of rigid base structures (Torresani et al., 2001; Xiao et al., 2004; Brand and Bhotika, 2001).

Our proposed method is an alternating one similar to (Hajder et al., 2011; Pernek et al., 2008) and is using the non-rigid formulation of (Torresani et al., 2001). It is applicable under weak-perspective (and orthographic) projection models. The generic non-rigid reconstruction tends to converge to invalid so-

lutions as it optimizes a huge amount of parameters. Our proposed method incorporates the symmetry constraint which gives stability to the non-rigid reconstruction.

3.2 Non-rigid Object Model

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 cased calculated by 3D CLM tracking and 3D eye corner detection). The non-rigidity of the object is in most cases modelled via K key objects (Torresani et al., 2001; Xiao et al., 2004; Brand and Bhotika, 2001). It is expected that the non-rigid shape of each frame can be accurately estimated as a linear combination of its K key objects.

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

$$S^j = \sum_{i=1}^{K} w_i^j S_i \tag{2}$$

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

$$S_k = \begin{bmatrix} X_{1,k} & X_{2,k} & \cdots & X_{P,k} \\ Y_{1,k} & Y_{2,k} & \cdots & Y_{P,k} \\ Z_{1,k} & Z_{2,k} & \cdots & Z_{P,k} \end{bmatrix} \tag{3}$$

3.3 Weak Perspective Projection

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:

$$\begin{bmatrix} u_i^j \\ v_i^j \end{bmatrix} = q^j R^j \begin{bmatrix} X_i^j \\ Y_i^j \\ Z_i^j \end{bmatrix} + t^j \tag{4}$$

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 parameter can be accumulated in the non-rigid weight components w_i^j. Utilizing this assumption, the weak-perspective projection for a non-rigid object in the j^{th} frame can be written as:

$$\begin{aligned} W^j = \begin{bmatrix} u_1^j & \cdots & u_P^j \\ v_1^j & \cdots & v_P^j \end{bmatrix} &= R^j S^j + t^j \\ &= R^j \left(\sum_{i=1}^{K} w_i^j S_i \right) + t^j \end{aligned} \tag{5}$$

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

The projection equation can be reformulated as

$$W = MS \tag{6}$$

where W is the measurement matrix of all frames:

$$W = \begin{bmatrix} W^1 \\ \vdots \\ W^F \end{bmatrix} \tag{7}$$

and M is the non-rigid motion matrix for all frames:

$$M = \begin{bmatrix} w_1^1 R^1 & \cdots & w_K^1 R^1 & t_1 \\ \vdots & \ddots & \vdots & \vdots \\ w_1^F R^F & \cdots & w_K^F R^F & t_F \end{bmatrix} \tag{8}$$

and S is defined as a concatenation of the K key objects:

$$S = \begin{bmatrix} S_1 \\ \vdots \\ S_K \\ 1 \end{bmatrix} \tag{9}$$

3.4 Optimization

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

$$\|W - MS\|_F^2 \tag{10}$$

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. This is a well-known statement when rigid objects are reconstructed. Buchanan & Fitzgibbon (Buchanan and Fitzgibbon, 2005) discussed that the motion and structure parameters can be separated if affine projection is assumed.

The parameters of the proposed algorithm are categorized into the groups of 1. rotation matrices (R^j) and translation parameters (t^j), 2. key object weights

(w_i^j), and, 3. key object parameters (S_k) and. 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 in the following sub-sections.

3.5 Rt-step

The *Rt*-step is very similar to the one proposed by Pernek et al. (Pernek et al., 2008). The motion 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 $\sum_{i=1}^{K} w_i^j S_i$. 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).

3.6 w-step

The goal of the w-step is to compute parameters w_i^j optimally in the least squares sense. Let us group the weights corresponding to the j^{th} frame into vector w^j as $w^j = [w_1^j \cdots c_K^j]^T$. w^j is independent on w^i if $i \neq j$. For the j^{th} frame, the optimization problem can be rewritten as

$$min_{w^j} \parallel (W^j - t^j[1\dots1])^T(:) - \left[(R^jS_1)^T(:)\dots(R^jS_K)^T(:)\right] w^j \parallel_F^2 \tag{11}$$

where (:) denotes the column-wise vectorization operator.

This is a linear problem with respect to w^j. The optimal solution is obtained as follows:

$$w^j = \left[(R^jS_1)^T(:)\dots(R^jS_K)^T(:)\right]^\dagger \left(W^j - t^j[1\dots1])^T(:)\right) \tag{12}$$

where † denotes the Moore-Penrose pseudoinverse.

3.7 S-step

The aim of the S-step is to compute the K key objects (see Eq. 9). We have two assumptions on these key objects: 1. they are symmetrical and 2. a certain index identifies the same point within all of them (for example the index of the left eye corner of the left eye is the same for all the key objects).

We categorize the points of the symmetric key objects into two groups:

- pair points: they are symmetric to the symmetry plane of the object
- single points: laying on the symmetry plane of the object

We assume that we know the indices of the single points and the index pairs of the pair points prior to the application of our method.

From now on, let $sidx(i)$, $i = [1\ ..\ \#s]$ and $pidx(i,j)$, $i = [1\ ..\ \#p]$, $j = [1\ ..\ 2]$ denote the list of single and pair point indices, respectively where $\#s$ and $\#p$ are the number of the single and pair points, respectively. Furthermore, let $s_{k,l} = [s_{k,l,x}, s_{k,l,y}, s_{k,l,z}]^T$ denote the $3D$ coordinates of a single point of the k^{th} key object at the index l and let $r^{j,c}$ ($c = [1\ ..\ 3]$) denote the c^{th} columns of the rotation matrix of the j^{th} frame.

The S-step for single and pair points are explained in the next next sections.

3.7.1 S-step for single points

Using the notations introduced in Sec. 3.7, the non-rigid motion matrix (see Eq. 8) for the single points can be written as :

$$M_s = \begin{bmatrix} w_1^1[r^{1,1}, r^{1,2}, r^{1,3}] & \cdots & w_K^1[r^{1,1}, r^{1,2}, r^{1,3}] \\ \vdots & \ddots & \vdots \\ w_1^F[r^{F,1}, r^{F,2}, r^{F,3}] & \cdots & w_K^F[r^{F,1}, r^{F,2}, r^{F,3}] \end{bmatrix} \tag{13}$$

The optimal solution (in least square sense) for a single points is written as follows:

$$s = M_s^\dagger \left(w^{sidx(i)} - t[1\dots1]\right) \tag{14}$$

where $s = [s_{1,sidx(i)}^T \cdots s_{K,sidx(i)}^T]^T$ denote the refined 3D points of all key objects at index $sidx(i)$, and $w^{sidx(i)}$ is the corresponding column of the completed measurement matrix, t is composed of the 2D offset vectors as $t = [t^1; \dots; t^F]$, and † denotes the Moore-Penrose pseudoinverse.

As the single points lay on the symmetry plane of the centralized object, we set $s_{k,sidx(i),x}$ for all key objects to zero. This explicit modification slightly ruin the optimality of the S-step, however, it never caused a problem during our tests. The final refined single points at index $sidx(i)$ can be then written as:

$$s_{sidx(i)} = [(0, s_{1,sidx(i),y}, s_{1,sidx(i),z}) \cdots (0, s_{K,sidx(i),y}, s_{K,sidx(i),z})]^T \quad (15)$$

3.7.2 S-step for pair points

Assuming that our key objects are centralized and aligned (see Sec. 3.9), the pair points of the symmetric and centralized key objects differ only in the sign of their x-coordinates. Thus, the non-rigid motion matrix for the pair points can be formulated as:

$$M_p = \begin{bmatrix} w_1^1[-r^{1,1}, r^{1,2}, r^{1,3}] & \cdots & w_K^1[-r^{1,1}, r^{1,2}, r^{1,3}] \\ w_1^2[-r^{2,1}, r^{2,2}, r^{2,3}] & \cdots & w_K^2[-r^{2,1}, r^{2,2}, r^{2,3}] \\ \vdots & \ddots & \vdots \\ w_1^F[-r^{F,1}, r^{F,2}, r^{F,3}] & \cdots & w_K^F[-r^{F,1}, r^{F,2}, r^{F,3}] \\ w_1^1[r^{1,1}, r^{1,2}, r^{1,3}] & \cdots & w_K^1[r^{1,1}, r^{1,2}, r^{1,3}] \\ w_1^2[r^{2,1}, r^{2,2}, r^{2,3}] & \cdots & w_K^2[r^{2,1}, r^{2,2}, r^{2,3}] \\ \vdots & \ddots & \vdots \\ w_1^F[r^{F,1}, r^{F,2}, r^{F,3}] & \cdots & w_K^F[r^{F,1}, r^{F,2}, r^{F,3}] \end{bmatrix} \quad (16)$$

The optimal solution for a point pair is written as follows:

$$\begin{aligned} w^p &= [w^{pidx(i,1)}; w^{pidx(i,2)}] \\ t^p &= [t; t] \\ p &= M_p^{\dagger}(w^p - t^p[1 \ldots 1]) \end{aligned} \quad (17)$$

where $p = [s_{1,pidx(i,2)}^T \cdots s_{K,pidx(i,2)}^T]^T$ denotes the refined $3D$ point at index $pidx(i,2)$ for all key objects, w^p is the concatenation of the $pidx(i,1)^{th}$ and $pidx(i,2)^{th}$ columns of the completed measurement matrix, t is composed of the $2D$ offset vectors as $t = [t^1; \ldots; t^F]$, t^p is the concatenation of t with itself, and † denotes the Moore-Penrose pseudoinverse.

As the pair points of the centralized key objects differ only in sign of the x-coordinates, the points of $pidx(i,1)$ can derived from the points of $pidx(i,2)$ by negating the signs of x-coordinates: $s_{k,pidx(i,1)} = [-s_{k,pidx(i,2),x}, s_{k,pidx(i,2),y}, s_{k,pidx(i,2),z}]^T$. The final refined pair points at index $pidx(i,1)$ and $pidx(i,2)$ can be then written as:

$$\begin{aligned} s_{pidx(i,1)} = \\ [(-s_{1,pidx(i,1),x}, s_{1,pidx(i,1),y}, s_{1,pidx(i,1),z}) \cdots \\ (-s_{K,sidx(i,1),x}, s_{K,sidx(i,1),y}, s_{K,sidx(i,1),z})] \\ s_{pidx(i,2)} = \\ [(s_{1,pidx(i,2),x}, s_{1,pidx(i,2),y}, s_{1,pidx(i,2),z}) \cdots \\ (s_{K,sidx(i,2),x}, s_{K,sidx(i,2),y}, s_{K,sidx(i,2),z})] \end{aligned} \quad (18)$$

Algorithm 1 Non-rigid And Symmetric Reconstruction

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

3.8 Completion

Due to the optimal estimation of the rotation matrix, an additional step must be included before every step of the algorithm. 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 denote the third row of the completed rotation, measurement, and, translation at the j^{th} frame, respectively. The completion is written as:

$$w_3^j \leftarrow r_3^j \left(\sum_{i=1}^{K} w_i^j S_i \right) + t_3^j \quad (19)$$

3.9 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. 6. 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 planes of them and relocate their points so that the single points lay on, the pair points are symmetrical to the symmetry plane. We as well centralize and align the key objects. As a result of the alignment, the normal vectors of the symmetry planes shall be in the direction of $[1,0,0]$.

The symmetry plane of the k^{th} key object can be written as: $n_{k,1}x + n_{k,2}y + n_{k,3}z + d_k = 0$, where $n_k = [n_{k,1}, n_{k,2}, n_{k,3}]^T$ is the normal vector of the symmetry plane. The normal vector can be estimated from the pair points as:

$$n_k = \frac{\sum_{i=1}^{\#p} \left(s_{k,pidx(i,2)} - s_{k,pidx(i,1)}\right)}{\| \sum_{i=1}^{\#p} \left(s_{k,pidx(i,2)} - s_{k,pidx(i,1)}\right) \|_F^2} \tag{20}$$

The symmetry plane of the key object passes through the center of the k^{th} key object ($o_{c,k}$) and thus, the d_k parameter can be calculated (d_k = - $n_k^T o_{c,k}$) as well.

To re-normalize the point pairs, the intersections of the the symmetry plane and the lines passing through the point pairs ($s_{k,pidx(i,1)}$ and $s_{k,pidx(i,2)}$) are calculated. The distance of the points of a pair can be written as:

$$\begin{aligned} d_{k,pidx(i,1)} &= n_k^T s_{k,pidx(i,1)} + d_k \\ d_{k,pidx(i,2)} &= n_k^T s_{k,pidx(i,2)} + d_k \end{aligned} \tag{21}$$

And thus, the intersection point can be calculated:

$$i_{k,i} = s_{k,pidx(i,1)} + \left(s_{k,pidx(i,2)} - s_{k,pidx(i,1)}\right) \frac{|d_{k,pidx(i,1)}|}{|d_{k,pidx(i,1)}| + |d_{k,pidx(i,2)}|} \tag{22}$$

And the points of the point pairs can be re-normalized by positioning them perpendicularly to $i_{k,i}$ with the distance value: $d_{k,i}^{avg} = (|d_{k,pidx(i,1)}| + |d_{k,pidx(i,2)}|)/2.0$:

$$\begin{aligned} s_{k,pidx(i,1)}^{new} &= i_{k,i} - n_k d_{k,i}^{avg} \\ s_{k,pidx(i,2)}^{new} &= i_{k,i} + n_k d_{k,i}^{avg} \end{aligned} \tag{23}$$

The single points are re-normalized via setting their x-coordinates to zero (laying on the symmetry plane):

$$s_{k,sidx(i)}^{new} = \left[0, s_{k,sidx(i),y}, s_{k,sidx(i),z}\right]^T. \tag{24}$$

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 generated by Basel Face Model (BFM) (P. Paysan and R. Knothe and B. Amberg and S. Romdhani and T. Vetter, 2009).

See Figure 17 for a few images of a typical video sequence.

Figure 17: A video sequences at central, left, right, up, and, down head poses

4.1 Empirical Evaluation

This section visualizes the results of the 3D eye corner detection on both real and synthetic (see Fig. 18) video sequences. The section contains only empirical evaluation of the results. The figures referred above shows 6 test sequences which 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 display the eyes at maximal left, right, up, and, down headposes in top-down order, respectively. The black markers show the selected eye corners. The gray 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.

4.2 2D/3D Eye Corner Detection Evaluation

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 have generated 100 video sequences by the BFM as described in 4.1. 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 model.

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

Type	3DCLM	2DEC	3DEC
Accuracy	0.5214	0.4201	0.4163
Improvement	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 (this behavior of the CLM might be tuned if we use a more realistic face model than BFM). An illustration of the problem is displayed in Figure 19.

4.3 Non-rigid Reconstruction Evaluation

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.

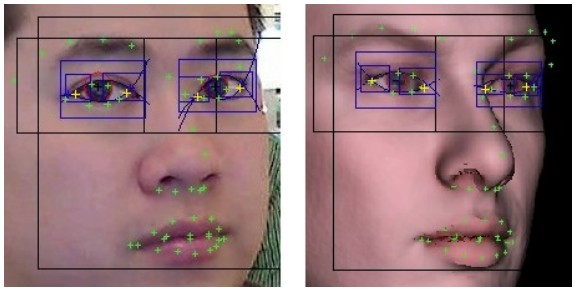

Figure 19: CLM error at extreme head pose

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). Instead of them, we 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 (as before).

The head pose error we calculated 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 es the average registration error (Arun et al., 1987) of the ground truth and the computed 2D and 3D point sets of each image of the sequence. Note that we centralize and normalize the ground truth 2D and 3D points sets so that the average distance of the points from the origin is $\sqrt{2}$.

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 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.

The test results shows the symmetric constraint is advantageous for the non-rigid reconstruction. The huge amount of parameters of the optimization can easily lead to lower reprojection error values, however, without the symmetric constraint this optimization can yield invalid solutions. Our proposed method keeps stable even with a high number of non-rigid components (K).

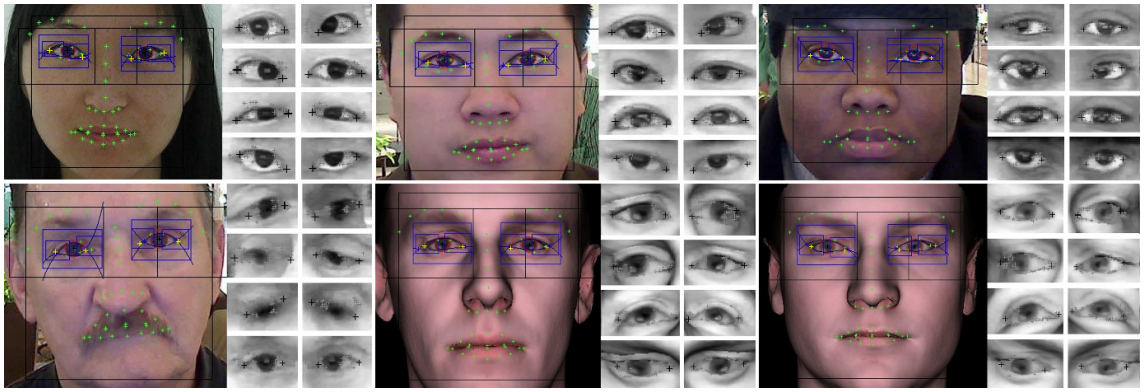

Figure 18: All real and synthetic test sequences

Table 2: Comparision of the 3D CLM, the symmetric and non-rigid and the generic non-rigid reconstruction. The K non-rigid parameters are displayed in the table.

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

One can also see that the head pose error of our proposed method outperforms the 3D CLM, however, the generic rigid reconstruction 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. Which means better fitting is not always the best if it converges to an invalid 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 algorithm. The test results have convinced us that the proposed methods outperforms the rival 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700.

Brand, M. and Bhotika, R. (2001). Flexible Flow for 3D Nonrigid Tracking and Shape Recovery. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 312–322.

Buchanan, A. M. and Fitzgibbon, A. W. (2005). Damped newton algorithms for matrix factorization with missing data. In *CVPR05*, volume 2, pages 316–322.

Cootes, T., Taylor, C., Cooper, D. H., and Graham, J. (1992). Training models of shape from sets of examples. In *In Proc. British Machine Vision Conference*, pages 9–18. Springer-Verlag.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 484–498. Springer.

Cristinacce, D. and Cootes, T. F. (2006). Feature detection and tracking with constrained local models. In Chantler, M. J., Fisher, R. B., and Trucco, E., editors, *BMVC*, pages 929–938. British Machine Vision Association.

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.

He, Z., Tan, T., Sun, Z., and Qiu, X. (2009). Towards accurate and fast iris segmentation for iris biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1670–1684.

Jankó, Z. and Hajder, L. (2012). Improving human-computer interaction by gaze tracking. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on*, pages 155–160.

Matthews, I. and Baker, S. (2004). Active appearance models revisited. *Int. J. Comput. Vision*, 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. *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS) for Security, Safety and Monitoring in Smart Environments*.

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

Poelman, C. J. and Kanade, T. (1997). A Paraperspective Factorization Method for Shape and Motion Recovery. *IEEE Trans. on PAMI*, 19(3):312–322.

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 *International Conference of Computer Vision (ICCV)*.

Sturm, P. and Triggs, B. (1996). A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. In *European Conference on Computer Vision*, volume 2, pages 709–720.

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

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

Torresani, L., Yang, D., Alexander, E., and Bregler, C. (2001). Tracking and Modelling Nonrigid Objects with Rank Constraints. In *IEEE Conf. on Computer Vision and Patter Recognition*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Weinshall, D. and Tomasi, C. (1995). Linear and Incremental Acquisition of Invariant Shape Models From Image Sequences. *IEEE Trans. on PAMI*, 17(5):512–517.

Xiao, J., Chai, J.-X., and Kanade, T. (2004). A Closed-Form Solution to Non-rigid Shape and Motion Recovery. In *ECCV (4)*, pages 573–587.