High Accuracy Head Pose Tracking Survey

Błażej Czupryński and Adam Strupczewski

Warsaw University of Technology, Poland

Abstract. Head pose estimation is recently a more and more popular area of research. For the last three decades new approaches have constantly been developed, and steadily better accuracy was achieved. Unsurprisingly, a very broad range of methods was explored - statistical, geometrical and tracking-based to name a few. This paper presents a brief summary of the evolution of head pose estimation and a glimpse at the current state-of-the-art in this field.

Keywords: Head pose estimation, 3D tracking, Face analysis, Optical flow, Tracking, Feature matching, Model matching.

1 Introduction

Head pose estimation is becoming a more and more popular topic in recent years. This is because it has a wide variety of uses in human-computer interaction. The development of smartphones, tablets and other mobile consumer electronics presents a large area for various applications of head pose tracking algorithms. Furthermore, head pose estimation is a very important element of eye gaze tracking when no infra-red sensors are used - that is only using a simple web camera. Some approaches [1] directly link head pose tracking to eye gaze tracking as a means of providing the initial field of view. Both head pose estimation and eye gaze tracking seem to be the future of computer vision.

Computer vision researchers have been exploring the topic of head pose estimation for the last 20 years. A lot of improvement has been made over the years, but still no single method is capable of estimating the head pose both accurately and robustly in all situations. There are generally two categories of head pose estimation algorithms: coarse head pose estimation, which tend to be more robust but not necessarily accurate, and fine head pose estimation, which demonstrate high accuracy. The focus of this paper are accurate methods, with other approaches presented as background information and not covered in much detail.

2 General Classification of Approaches

Over the past two decades a lot of different techniques have been tried. A good and fairly exhaustive overview is presented in [2]. Recently, considerable advances have been made in high accuracy tracking [3,4], which are covered in more detail

D. Ślęzak et al. (Eds.): AMT 2014, LNCS 8610, pp. 407–420, 2014.

in section 6 and 7. A lot of recent focus was also on approaches using a depth camera [5], which however is not the focus of this paper.

One way to divide the existing approaches could be by the importance of statistics and pre-learned models in comparison to the importance of geometrical measurements. This could lead to five groups:

- **Solely statistical approaches.** These include template based methods that compare new images to a set of head images with known poses [6], detector based methods that have many head detectors each tuned to a different pose [7], nonlinear regression methods that map the image data to a head pose measurement [8] and manifold embedding methods that seek low dimensional manifolds to model continuous variation in head pose [9].
- **Approaches using flexible models.** These include elastic graphs [10], active shape models [11] and active appearance models [12]. All methods attempt to fit a non-rigid model to the face in the image and derive the head pose from the model's parameters.
- **Geometrical approaches.** These methods first locate certain facial features and based on their relative locations determine the head pose [13].
- **Tracking approaches.** These track the face movement between consecutive frames and from this infer the pose changes. There are many variations among these methods - the tracked elements include features [14], models [15] or dynamic templates [16].
- **Hybrid approaches.** These approaches combine two or more of the previously mentioned approaches and tend to provide best robustness and accuracy. A good example is [17], where the authors combined a template database, real-time feature tracking and the Kalman filter to achieve very high head pose estimation precision.

The above groups will be covered in more detail in the following sections. The last two groups are the main point of focus of this article, as they allow the highest accuracy.

3 Statistical Approaches

The most simple statistical approach is using appearance templates and assigning the pose of the most similar template to the query. Normalized cross-correlation [18] or mean squared error over a sliding window [19] can be used for image comparison. Appearance tamplates have a number of advantages, most importantly: simplicity, easy template set expansion and independence of image resolution. However, they only allow to estimate a discrete set of poses corresponding to the pre-annotated database. They are also not very efficient, especially with large template sets. The biggest problem of the approach is the assumption that similarity in image space corresponds to similarity in pose space. This is often not true, as identity and illumination might influence the dissimilarity more than a change in pose. To resolve this problem, more sophisticated distance metrics have been proposed. Convolutions with a Laplacian-of-Gaussian

filter [20] emphasize facial contours, while convolutions with Gabor wavelets [6] emphasize directed features such as the nose (vertical) and mouth (horizontal). A recent development focused on mobile devices uses Hu moments calculated from facial pixels and their projection using the Fischer linear discriminant matrix to determine similarity with reference templates [21]. It is assumed however, that only five different poses are distinguished by the system.

A somewhat similar method to appearance templates is using arrays of detectors. One such approach using an ensemble of weak classifiers trained with Adaboost is described in [7]. Instead of matching the query to a set of templates, the query is run through a set of pre-trained detectors and the classifier with highest confidence determines the query's pose. Compared to appearance templates, detector arrays are trained to ignore the appearance variations and are sensitive to pose variations only. Furthermore, they solve the problem of face detection and localization - which remains a separate task in case of appearance templates. Big disadvantages of this scheme are the necessary scale of training, binary output of detectors and small accuracy - in practice at most 12 different detectors can be trained [2], which limits the pose estimation resolution to 12 states. Due to these constraints, detector arrays have not become very popular, and few papers propose usable systems. One usable approach is described in [7].

Another group of statistical approaches is constituted by nonlinear regression methods. These aim to find a mapping from the image space to a pose direction. If such a mapping existed given sample labeled data, a pose estimate could be found for any new data using the trained model. In practice such nonlinear regression is most often exploited using support vector regressors or neural networks. The first approach performs well if dimensionality can be reduced using PCA [22] or feature data extracted at facial feature points [23]. The second approach with neural networks can use either multi-layer perceptrons (MPL) [24], which in principle work similarly to detector arrays, or locally-linear maps (LLM), which first select a centroid for the query, and then perform linear regression for pose refinement [25]. These methods can work fast and be fairly accurate. The main disadvantages of nonlinear regression methods are the need for long, sophisticated training and the very big sensitivity to head localization. Inaccurate head localization will lead to a complete failure of these methods.

The last group of statistical approaches that should be mentioned are manifold embedding methods. The aim of these methods is to reduce the dimensionality of the input face image so that it can be placed on a low-dimensional manifold constrained only by allowable head poses. The placement then defines the head pose. Initially, PCA and LDA have been explored as dimensionality reduction techniques and pose-eigenspaces were created for projecting the input [9]. Alternatively, pose-eigenspaces could be substituted by SVMs, which have shown to perform best when combined with local Gabor binary patterns [26]. Different manifold embedding approaches perhaps perform even better. The most representative of these is probably isometric feature mapping [27]. Despite relatively good robustness, techniques using manifold embedding suffer for the same reason as appearance templates - they easily capture appearance and not only pose

variations. Furthermore, good models require a lot of training and ideally a complete set of poses from all people in the database [28].

All in all, statistical approaches demonstrate good robustness in pose estimation, but most often do not provide very high accuracy. The techniques which are capable of providing more than just a few discrete pose states require complicated training and are not guaranteed to work in cases that differ greatly from the training data. Despite their multiple advantages, statistical approaches do not seem to be the best way of estimating the head pose with high accuracy.

4 Flexible Model Approaches

Statistical methods treat head pose estimation as a specific signal processing task with the 2D image as input. Flexible models aim to fit a non-rigid model to certain facial features so as to uncover the facial structure in each case. Depending on how the process of fitting is performed and what the final result is, flexible models come in several forms. Early work in this area was related to Elastic Bunch Graphs. These deformable graphs of local feature points such as eyes, nose or lips could be first matched to labeled training images to establish a reference pose set. The same graph was later matched to the query image, and based on the mutual feature locations a best fit from the training data could be found [10]. An advantage of feature-domain comparisons instead of face image comparisons is a much stronger link to pose similarity. Unfortunately, similarly as appearance templates, these approaches only allow to distinguish between a discrete set of poses and become difficult with large amounts of training data.

Potentially much higher accuracy can be achieved by using active shape models (ASMs) and active appearance models (AAMs). The first technique involves fitting a specific shape to new data, where the fitted shape is a linear combination of some pre-trained eigenspace of shapes [29]. This allows to combine greedy local fitting with the constraint of a model. Adding appearance to the shape and combining the two proved to be a better solution when fitting these models to faces [30]. Once such a model, being originally a set of 2D points, is fitted to the face image, it is possible to infer the pose based on the relative locations of these points. One possibility is to use simple linear regression on these points [31]. Another possibility is to use a combined 2D and 3D Active Appearance Model, where an inherent 3D model is used to constrain the fitting of 2D points [12]. This allows to infer the head pose directly. Finally, it is possible to use structure from motion algorithms to infer the 3D locations of model points and directly calculate the pose [32]. In fact, this technique could even be used without flexible models by matching feature points detected on the face and using them for reconstruction. With increased processing power becoming available in recent years, a real-time implementation of this approach might be a promising line of research.

A more recent development is based on a combination of tree models and a shared pool of parts [33]. A set of facial landmarks is localized in the query image, thus simultaneously providing head detection, landmark localization and

pose estimation. The authors argue that their approach is capable of fitting the elastic models much better then AAMs and CLMs (Constrained Local Models). As the proposed system saturates previous benchmarks, a new face database was created for the purpose of evaluation - Annotated Faces in the Wild (AFW). The reported results are very impressive, but they all refer to coarse pose estimation - an error is assumed only when the detected pose deviates by more than 15 from the annotation.

To sum up, flexible models have a big and still largely unexplored potential to provide both good robustness and accuracy of head pose estimation. While their precision is limited by the feature fitting accuracy and so depends on the input image resolution and quality, they may be a good starting point for use with refining algorithms.

5 Geometric Approaches

Geometric approaches derive the head pose from the geometric configuration of facial features depending on how the face is positioned in relation to the silhouette of the head or how big the nose deviation is from bilateral symmetry. One group of approaches uses the outer corners of the eyes, the outer corners of the mouth and the nose position [34]. The center points between the eye corners and between the lip corners project a line. The distance of the nose from this line, as well as the angle, gives information about the pose. This is illustrated in Figure 1. A recent paper describes the implementation of this algorithm on contemporary hardware using a cascade of detectors for feature localization [35].

Another set of methods uses also the inner eye corners for measurements [36]. One of the most recent methods [13] uses the parallelism of lines between inner eye corners, outer eye corners and lip corners. A so called *vanishing point* can be calculated as the intersection of the eye line and the lip line. The head pose can

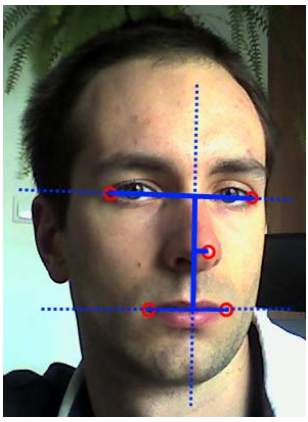

Fig. 1. Geometric approaches use relative feature positions

be estimated based on its location. Furthermore, the ratios of the lines can be used to infer the pitch of the head. Moreover, an EM algorithm with a Gaussian mixture model was proposed to account for the variation among different people.

Geometric approaches usually require few calculations and are simple. They are also capable of providing a highly accurate estimation, but their accuracy strongly depends on the accuracy of facial feature location estimation. At present, the most accurate detectors have an error of at least 1-2 pixels even with high quality images. This is a lower bound on the potential accuracy of geometric methods. In our experience the 2-pixel feature location error can cause a considerably large, 5-10 degree head pose angle estimation error for typical webcam face images. It means that when using a simple webcam, geometric methods are unable to provide an accurate head pose estimation because of limited feature detection accuracy. A further limitation of these methods is the ability to estimate the head pose only when all the facial features are visible and detectable.

6 Tracking Approaches

Tracking methods can be used to determine the head pose by accumulating estimations of the relative head movement between subsequent video frames. This approach assumes observing smooth motion of the head so it is applicable only when a continuous video stream containing the head motion is available. Tracking-based head pose estimation methods are capable of providing very accurate results, as they can calculate very small pose changes between successive frames and thus outperform other approaches [2]. The main disadvantage of these methods is the need of tracking initialization during algorithm start-up and after tracking gets lost. This means that it is necessary to use some different head pose estimation algorithm in the initialization step. To simplify this step, the tracking algorithm can assume that the head pose is frontal at initialization - which can be provided by a frontal face detection algorithm. Another drawback of tracking approaches is that they are very accurate only in the short-term. They are also sensitive to occlusions and illumination changes. Therefore, many extensions of tracking algorithms have been proposed to improve long-term stability and robustness, mainly in the presence of occlusions or large out-of-plane rotations [16,15,3].

6.1 Model-Based 3D Tracking

In ordinary 2D tracking algorithms the 3D motion of the object is modeled as a 2D transformation. To recover the full 3D head pose, which has six degrees of freedom (translations and rotations along three axes), a 3D tracking algorithm is required [37]. This can be achieved by utilizing a 3D shape model of an object. The translation and rotation of the model is estimated by analyzing the projected 2D images of the modeled object. The 2D translations can be unprojected from the image into the 3D scene to account for 3D motion. Several approaches have been proposed to solve this problem. [38,16,37,3,17].

The model used in tracking can be rigid or non-rigid (deformable). This paper focuses only on the rigid model approach, as non-rigid models are more useful for emotion recognition than high-accuracy head pose tracking. In general, the shape of the human head can be modeled either using a precise 3D model or approximated by a geometric primitive. The selection of the model type directly affects the accuracy and robustness of the tracker, but is something separate from the tracking algorithm itself. Approaches using both generic and user-specific precise head models have been proposed [39,38]. The main advantage of a precise face model is the capability to provide a very accurate pose estimation. However, this is only possible when combined with precise initialization to closely fit the model to the observed face. When the model is not well aligned to the face, tracking errors rise significantly. Moreover, the alignment usually degrades during tracking due to tracking error accumulation [17].

Algorithms based on approximating the head shape with geometrical primitives are more robust. The most simple one is a planar surface, but the recovery of out-of plane rotation in this case is inaccurate due to lack of depth in the model. From all geometric primitives, the best choice seems to be using a cylinder [16,15,3]. It possesses depth, is very robust to initialization errors and can work well despite appearance changes between different people.

6.2 3D Tracking under Perspective Projection

As mentioned in section 6.1, introducing a known 3D model into 2D tracking allows to recover the full 6-degree of freedom motion of the tracked object. A straightforward method of solving this problem was proposed in [3], based on [16]. Having a set of

- 2D image points at time $t-1$ along with their 3D coordinates in the real world given by a model,
- corresponding 2D image points at time t

it is possible to recover model pose at time t. If the model's pose at time $t-1$ is known, the updated pose at time t can be represented as the previous pose transformed by a motion vector $\mu = [t_x, t_y, t_z, \omega_x, \omega_y, \omega_z]$, which represents translation and rotation using twist representation. To describe correspondences between coordinates of the object in 3D space and their projections on the imaging plane of the camera, a simplified perspective projection camera model given only by the focal length f can be assumed. Considering a single tracked point, the locations p_{t-1}, p_t of this point are observed in the image at times $t-1$, t respectively. The 3D position P_{t-1} of this point at time $t-1$ is obtained by unprojecting p_{t-1} into 3D world coordinates using depth from the model, which has known orientation. The new 3D position estimated during tracking can be expressed as $P_t = M \cdot P_{t-1}$, where M is a transformation matrix based on vector μ:

$$M = \begin{bmatrix} 1 & -\omega_z & \omega_y & t_x \\ \omega_z & 1 & -\omega_x & t_y \\ -\omega_y & \omega_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

This situation is depicted in Figure 2. Given the transformation matrix M, the projection of P_t can be expressed using the previous position P_{t-1} and motion parametrized by vector μ:

$$p'_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} \frac{f}{z_t} = \begin{bmatrix} x_{t-1} - y_{t-1}\omega_z + z_{t-1}\omega_y + t_x \\ x_{t-1}\omega_y + y_{t-1}\omega_x - z_{t-1}\omega_x + t_y \end{bmatrix} \frac{f}{-x_{t-1}\omega_y + y_{t-1}\omega_x + z_{t-1} + t_z} \tag{2}$$

Now, two forms of the projection of point P_t are available. The first one is the observed location p_t and the second is p'_t, which was obtained by estimated motion of the previous location p_{t-1} based on 3D model and motion vector μ. Assuming N such point pairs have been collected, the goal is to compute motion vector μ, which minimizes the sum of distances between the observed points $p_{i,t}$ and the estimated points $p'_{i,t}$.

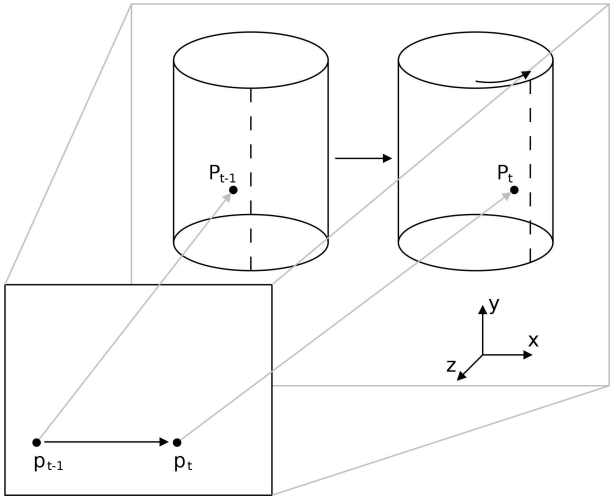

Fig. 2. Cylinder tracking under perspective projection

6.3 Finding Correspondences

For the presented algorithm to work, a set of corresponding point pairs from two images is necessary. One way to get it is by using a local feature point extraction algorithm. In [3] the authors use the SIFT algorithm, proposed in [40]. Feature

points are extracted independently for each frame and then matched by finding pairs with the most similar descriptors.

Although independent feature extraction and matching for each frame is possible, it not not the fastest method. A more efficient method is tracking feature points in consecutive frames. A technique of 3D pose estimation by applying optical flow to track 2D feature points and assuming a cylindrical head model was proposed in [16]. It uses a mesh of evenly distributed points as keypoints and works directly on image luminance. The algorithm operates on the same motion vector μ as introduced in section 6.2. Let the intensity of the image at point p and time t be denoted as $I(p,t)$. Let $F(p,\mu)$ be a function which maps point p into a new location p' using vector μ, according to the motion model described in section 6.2. This mapping function was already introduced in equation 2. The region containing all considered face pixels is denoted as Ω. Computing the motion vector between two frames based on luminance can be expressed as the minimization of the sum of luminance differences between the face image from previous frames and the current face image transformed by mapping function F:

$$\min\left(\sum_{p\in\Omega}\left(I\left(F\left(p,\mu\right),t\right)-I\left(p,t-1\right)\right)^2\right) \tag{3}$$

In [16] the motion vector μ is computed using the Lucas-Kanade method:

$$\mu=\left(\sum_{\Omega} w\left(I_pF_\mu\right)^T\left(I_pF_\mu\right)\right)^{-1}\sum_{\Omega} w\left(I_t\left(I_pF_\mu\right)^T\right) \tag{4}$$

where I_t and I_p are temporal and spatial image gradients, while w is a weight assigned to each point. F_μ denotes the partial differential of F with respect to μ at $\mu=0$:

$$F_\mu=\begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -\left(y^2+z^2\right) & xy & xz & 0 & z & -z \end{bmatrix}\frac{f}{z^2} \tag{5}$$

Motion is computed iteratively. After each iteration, the model is transformed using the computed motion vector μ and the weights for all points are updated.

A hybrid combination of tracking a mesh of points with feature point matching is also possible. The approach of [41] can be used for adjusting mesh points to good corner points within so called *cells*. Optical flow can be used to track these points and feature matching is not necessary, while at the same time the tracked points are not random. Although promising at first, assessing the impact of this modification on tracking accuracy is difficult, as there is no linear relationship between mesh tracking accuracy and the distinctiveness of mesh points.

6.4 Long-Term Stability

Tracking in a frame-to-frame fashion provides accurate pose estimation in the short-term, but it tends to lose accuracy over time due to drift error accumulation. To obtain the head pose in the actual frame, motion vectors have to

be accumulated starting from the beginning of tracking. Motion estimation errors are accumulated together with measurement errors. Thus, the drift error increases in time. Moreover, the frame-to-frame tracking is not able to recover after large errors, which occur in presence of occlusions or when the head is temporarily out of the camera's view.

In order to achieve long-term robustness and stability, the drift error has to be compensated and the algorithm has to be able to recover after tracking loss. This is usually done by introducing an alternative tracking algorithm capable of performing reinitialization. This auxiliary method periodically performs tracking to stored reference frames (templates) instead of the previous frame. Such compensation of errors greatly increases the pose estimation robustness and general performance in real-world videos. In [16] the initial frame with a specified pose is stored as a reference keyframe, and the tracking to this keyframe is called re-registration. Tracking to reference frames should be performed from time to time, in two cases:

- when normal frame-to-frame tracking is working for a long time and drift error has accumulated,
- when there is a large error in frame-to-frame tracking caused by abrupt movement, occlusion etc. and traditional tracking has failed

7 Hybrid Approaches

Tracking to a reference frame can be performed simultaneously with frame-to-frame tracking giving a head pose estimation algorithm which is robust and accurate at the same time. This approach was presented in [3], where the Kalman filter is applied to combine motion estimation of the frame-to-frame tracker and the reference frame tracker. The authors proposed a robust tracking system, where a head feature point database stores a set of reference templates. Each template is a set of SIFT points captured at a known head pose. The templates are extracted at certain head poses during the initial stage of tracking, where a frame-to-frame tracker is used to determine the head pose. The initial template is stored for a frontal pose. Additionally, it contains generic facial feature points from a Bayesian Tangent Shape Model [42], which is used for the initial alignment of the 3D head model. Once the template database is filled, successive input frames are tracked in parallel using previous frames and template frames from the database. SIFT points from the input frame are matched with all templates independently, and the template with the largest number of matches is used to track the model. Separate motion vectors are estimated using the previous frame. In both cases tracking is done using an algorithm described in section 6.2. Finally, the two motion vectors are combined using the Kalman filter.

Depending on the tracking errors, bigger confidence is assigned either to the continuous tracking method or to the template tracking method, giving the final resulting pose vector estimated by the filter. The biggest gain in using the Kalman filter is that the estimation is smooth regardless of abrupt temporary errors. If there is an occlusion and the frame-to-frame tracker suddenly gets

lost, the Kalman filter will use the estimate from the template tracker and such disturbances will be barely noticed.

A similar method is described in [17]. Here however, the authors propose to simultaneously use feature-based tracking and intensity-based tracking. The aim is to minimize the error from both methods. This is presented as a nonlinear optimization problem, with different weights assigned to the intensity tracking and feature tracking based on time and tracking conditions. The reported results are impressive and demonstrate better accuracy and robustness than each method alone, especially in demanding test sequences.

A more recent publication [43] attempts to use the Kalman filter to predict the head motion from previous motion, in order to support texture-based optical flow tracking. The authors claim that in comparison to pure texture-based tracking, a significant speed improvement can be achieved.

One of the most sophisticated hybrid tracking approaches is described in [4]. The system combines three types of tracking using linear regression:

- static person-independent pose estimation
- static user-dependent pose estimation based on reference templates
- differential motion-based estimation

As shown in section 8, the proposed hybrid approach is very competitive and achieves accuracy of head pose angle estimation close to the state-of-the-art. This is despite having a completely different architecture than most contemporary head pose estimation research.

8 Conclusion

To conclude it has to be noted that hybrid approaches provide the best accuracy and robustness. Table 1 shows a brief comparison of various tracking methods in terms of precision. For a reliable comparison, the head pose estimation methods were compared on the same dataset - Boston University dataset with uniform lighting. Most results are averaged for all 45 uniform lighting test sequences of the dataset, but some only for a subset of it - as mentioned in the remarks.

If only one method was to be chosen, then tracking based approaches are favored. The numerical comparison in Table 1 clearly shows their superiority [16,17]. The recent improvements of feature point estimation have allowed methods using them [35] to reach precision close to that of tracking methods.

It must be remembered though, that real-world situations and use cases place big importance on tracking robustness - something which is not necessarily shown with numbers. At the moment it seems that if computational resources are available, the most accurate and robust head pose estimation methods are hybrid tracking methods combining several tracking-based methods with filters [4]. For maximum robustness, combining tracking methods with a static head pose estimation method has emerged as a popular trend in research [16,3,17].

As for future improvement of tracking accuracy, a fusion of 3D model tracking and feature point based pose estimation seems to be a promising line of research.

Table 1. Head pose estimation accuracy on Boston University Dataset - as reported by authors

Method	Roll [deg]	Pitch [deg]	Yaw [deg]	Remarks
Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques [16]	1,4	3,2	3,8	Boston Uni (45 uniform) + Optotrack
Robust, Real-Time 3D Face Tracking from a Monocular View [17]	1,4	3,5	4,0	Boston Uni - uniform example, error values estimated from chart
Fasthpe: A recipe for quick head pose estimation [35]	3,03	5,27	3,91	Boston Uni (18 sequences)
Robust 3D Head Tracking by View-based Feature Point Registration [3]	3,44	4,22	2,07	Boston Uni (45 uniform)
Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation (GAVAM) [4]	3,67	4,97	2,91	Boston Uni (45 uniform)
Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models [15]	9,8	6,1	3,3	Boston Uni (45 uniform)

As shown in [3] matching features to templates can work very well. Considering the recent developments in real-time 3D reconstruction [44,45], it seems feasible to create and refine an online 3D head model and use it for continuous, highly accurate head pose tracking. If appropriately robust feature points can be chosen or developed for facial images, this approach could yield the most accurate head pose estimation yet.

References

1. Valenti, R., Sebe, N., Gevers, T.: Combining head pose and eye location information for gaze estimation. IEEE Transactions on Image Processing (2012)
2. Murphy-Chutorian, E., Trivedi, M.: Head pose estimation in computer vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2009)
3. Jang, J., Kanade, T.: Robust 3d head tracking by online feature registration. In: The IEEE International Conference on Automatic Face and Gesture Recognition (2008)
4. Morency, L., Whitehill, J., Movellan, J.: Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation. In: 8th IEEE International Conference on Automatic Face Gesture Recognition (2008)
5. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (2011)

6. Sherrah, J., Gong, S., Ong, E.J.: Face distributions in similarity space under varying head pose. Image and Vision Computing 19 (2001)
7. Viola, M., Jones, M., Viola, P.: Fast multi-view face detection. In: Proc. of Computer Vision and Pattern Recognition (2003)
8. Gourier, N., Hall, D., Crowley, J.: Estimating face orientation from robust detection of salient facial structures. In: FG Net Workshop on Visual Observation of Deictic Gestures (2004)
9. Srinivasan, S., Boyer, K.: Head pose estimation using view based eigenspaces. In: Proceedings of 16th International Conference on Pattern Recognition (2002)
10. Kruger, N., Potzsch, M., Malsburg, C.: Determination of face position and pose with a learned representation based on labelled graphs. Image and Vision Computing 15 (1997)
11. Lanitis, A., Taylor, C., Cootes, T., Ahmed, T.: Automatic interpretation of human faces and hand gestures using flexible models. In: International Workshop on Automatic Face- and Gesture-Recognition (1995)
12. Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real-time combined 2d+3d active appearance models. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004)
13. Wang, J., Sung, E.: EM enhancement of 3d head pose estimated by point at infinity. Image and Vision Computing 25 (2007)
14. Yao, P., Evans, G., Calway, A.: Using affine correspondence to estimate 3-d facial pose. In: Proceedings of International Conference on Image Processing (2001)
15. La Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000)
16. Xiao, J., Kanade, T., Cohn, J.: Robust full-motion recovery of head by dynamic templates and re-registration techniques. In: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (2002)
17. Liao, W., Fidaleo, D., Medioni, G.: Robust, real-time 3d face tracking from a monocular view. EURASIP Journal on Image and Video Processing (2010)
18. Beymer, D.: Face recognition under varying pose. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1994)
19. Niyogi, S., Freeman, W.: Example-based head tracking. In: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition (1996)
20. Gonzalez, R.C., Woods, R.E.: Digital Image Processing (2001)
21. Ren, J., Rahman, M., Kehtarnavaz, N., Estevez, L.: Real-time head pose estimation on mobile platforms. Journal of Systemics, Cybernetics and Informatics 8 (2010)
22. Li, Y., Gong, S., Sherrah, J., Liddell: Support vector machine based multi-view face detection and recognition. Image and Vision Computing 22 (2004)
23. Ma, Y., Konishi, Y., Kinoshita, K., Lao, S., Kawade, M.: Sparse bayesian regression for head pose estimation. In: 18th International Conference on Pattern Recognition (2006)
24. Zhao, L., Pingali, G., Carlbom, I.: Real-time head orientation estimation using neural networks. In: Proceedings of International Conference on Image Processing (2002)
25. Zhang, M., Li, K., Liu, Y.: Head pose estimation from low-resolution image with hough forest. In: 2010 Chinese Conference on Pattern Recognition (2010)
26. Ma, B., Zhang, W., Shan, S., Chen, X., Gao, W.: Robust head pose estimation using lgbp. In: 18th International Conference on Pattern Recognition (2006)

27. Raytchev, B., Yoda, I., Sakaue, K.: Head pose estimation by nonlinear manifold learning. In: Proceedings of the 17th International Conference on Pattern Recognition (2004)
28. Yan, S., Zhang, Z., Fu, Y., Hu, Y., Tu, J., Huang, T.: Learning a person-independent representation for precise 3D pose estimation. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 297–306. Springer, Heidelberg (2008)
29. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models-their training and application. Computer Vision and Image Understanding 61 (1995)
30. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Comput. Vision 60 (2004)
31. Cootes, T., Walker, K., Taylor, C.: View-based active appearance models. In: Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition (2000)
32. Gui, Z., Zhang, C.: 3d head pose estimation using non-rigid structure-from-motion and point correspondence. In: IEEE Region 10 Conference on TENCON (2006)
33. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012)
34. Gee, A., Cipolla, R.: Determining the gaze of faces in images. Image and Vision Computing 12 (1994)
35. Sapienza, M., Camilleri, K.: Fasthpe: A recipe for quick head pose estimation. In: Technical Report (2011)
36. Horprasert, T., Yacoob, Y., Davis, L.: Computing 3-d head orientation from a monocular image sequence. In: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (1996)
37. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects. Found. Trends. Comput. Graph. Vis (2005)
38. Malciu, M., Preteux, F.: A robust model-based approach for 3d head tracking in video sequences. In: Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition (2000)
39. Lu, L., Zhang, Z., Shum, H., Liu, Z., Chen, H.: Model- and exemplar-based robust head pose tracking under occlusion and varying expression. In: 2001 IEEE Conference on Computer Vision and Pattern Recognition (2001)
40. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60 (2004)
41. Matas, J., Vojir, T.: Robustifying the flock of trackers. In: 16th Computer Vision Winter Workshop, Mitterberg, Austria (2011)
42. Zhou, Y., Gu, L., Zhang, H.: Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2003)
43. Wang, Y., Gang, L.: Head pose estimation based on head tracking and the kalman filter. Physics Procedia (2011), 2011 International Conference on Physics Science and Technology
44. Stühmer, J., Gumhold, S., Cremers, D.: Real-time dense geometry from a handheld camera. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) DAGM 2010. LNCS, vol. 6376, pp. 11–20. Springer, Heidelberg (2010)
45. Wu, C.: Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision, pp. 127–134 (2013)