Head pose tracking from RGBD sensor based on direct motion estimation

Adam Strupczewski, Błażej Czupryński, Władysław Skarbek, Marek Kowalski and Jacek Naruniec

Warsaw University of Technology, Poland

Abstract. We propose to use a state-of-the-art visual odometry technique for the purpose of head pose estimation. We demonstrate that with small adaptation this algorithm allows to achieve more accurate head pose estimation from an RGBD sensor than all the methods published to date. We also propose a novel methodology to automatically assess the accuracy of a tracking algorithm without the need to manually label or otherwise annotate each image in a test sequence.

Index terms.
Head pose estimation, head pose tracking, camera pose estimation, RGBD camera, Kinect, visual odometry.

1 Introduction and related work

Head pose estimation is a very popular topic as it directly links computer vision to human-computer interaction. In recent years a lot of research has been conducted in this field. The focus of most contemporary approaches seem to be Active Appearance Model-like approaches such as [1,15] or hybrid approaches based on model tracking [9,13]. Despite impressive performance, we argue that the accuracy of these methods in terms of head pose estimation accuracy (rotation, translation) still leaves room for improvement. Most importantly, all the popular approaches assume that the RGB camera input is available, but do not consider depth input data. Recent developments in the field of depth sensors suggest that they will soon be deployed on a large scale, such as RGB cameras are today. Therefore, using this additional information for basic computer vision challenges such as head pose tracking, seems to be important.

We show how the method developed primarily be Kerl et al. [10,11] for camera pose estimation can be used for head pose tracking. We believe that this approach presents new opportunities and benefits compared to previous approaches. We demonstrate that this method has superior accuracy to the current state-of-the-art. Furthermore, we propose a generic scheme of adopting a camera pose estimation method to the head pose estimation scenario.

The contributions of this work are as follows:

1. A derivation on how to use the visual odometry algorithm proposed in [10] for head pose tracking
2. A new method to automatically evaluate the accuracy of a head pose tracking algorithm

1.1 Related work in head pose estimation

Head pose estimation is a very important topic in the field of human-computer interaction. It has many applications as a component of HCI systems such as eye gaze tracking or facial expression analysis, but also directly for controlling interface elements such as the cursor. The first vision-based approaches utilized a single RGB camera to estimate the head pose. Recently however, the focus of researchers has shifted to depth sensors. Starting with Microsoft Kinect, depth cameras have become increasingly available, which further encourages research in this area.

A good overview of head pose estimation using visible light cameras can be found in [16]. A follow-up of this study focused on high accuracy techniques was presented in [4]. The most accurate RGB based methods are related to tracking. The current state-of-the-art in this field are hybrid methods using a combination of static pose estimation and tracking, with the usage of reference templates [13,15].

The depth-based head pose tracking methods are all relatively new. One of the pioneering studies [14] infers pose from facial geometry. In order to make this work, facial landmarks such as the nose tip are first detected, and smoothed with a Kalman filter. The relative position of the facial landmarks allows to calculate the head pose. Despite high accuracy reported in the paper, it is questionable whether such can be achieved in real world scenarios under unconstrained head movements and varying lighting. Another statistical framework was later presented without the requirement of such high resolution input data [2]. The accuracy of the system seems to be good with a reported accuracy around 3 pixels related to manually tagged facial landmarks in images (in the presented set-up this is below 2mm). Other statistical head pose estimation approaches using depth cameras have also appeared [5,17], but they are only capable of performing coarse pose estimation, focused on robustness not accuracy.

A somewhat different type of approach has been proposed in [6]. The authors propose to perform depth based tracking and intensity based tracking separately, and later fuse them with an extended Kalman filter. A set of features is detected and tracked in the intensity domain, as well as in the pure depth image. The reported accuracy of around 3 degrees is quite impressive. In other work, [12] propose to track the head using an elaborated variant of ICP with head movement prediction. While being fast, the system seems less accurate than other related work. Last but not least, a new approach to using Active Appearance Models has been recently proposed [20]. It is an improved AAM tracking with linear 3D morphable model constraints and claims to have a point tracking accuracy of 3-6mm. It is a very good choice for tracking initialization, but does not reach the full potential of frame-to-frame tracking accuracy that we will present.

1.2 Related work in RGBD camera pose estimation

The traditional algorithm for RGBD pose estimation is the Iterative Closest Point (ICP) algorithm, which aligns two point clouds. One of the first popular

Simultaneous localization and mapping (SLAM) methods, Kinect Fusion [8], is based on this approach. ICP is however not very accurate, even when adopted in a coarse-to-fine scheme. A much more accurate method for small baselines has been recently proposed by Steinbrucker et al. [21]. Assuming that the scene geometry is known, the camera pose estimation can be denoted as the minimization of the following total intensity error

$$min_{\xi} \int_{\Omega} |I_1(\mathbf{x}) - I_2(\pi g_{\xi}(h\mathbf{x}))|^2 \, dx \tag{1}$$

where ξ represents the six degrees of freedom for rigid body motion, g_{ξ} is the rigid body motion (of the camera), $\mathbf{x}$ is a point in homogenous coordinates, $h\mathbf{x}$ is the 3D structure and π is projection. This can be solved by linearization and a Gauss-Newton approach. It turns out that on contemporary hardware such minimization, and in turn camera motion estimation, can be performed in real time even for high camera resolutions. Extensions to the basic idea include parallel minimization of the depth discrepancy as well as a probabilistic error model [11].

We use the described work in our derivation in section 2. We propose to solve a dual problem to the classical one : instead of estimating the camera pose we estimate the pose of the tracked object, in our case the human head.

2 Direct head pose tracking approach

We consider the problem of object pose tracking, i.e. the affine motion estimation using a single RGB-D camera. The camera can be used to find the instant motion with respect to its Cartesian coordinate system for selected objects which are detected in the image - in our case a human head. The below derivation is based on the work of [10,11,21] extending their presentation by the complete EM scheme including weight computation for Iteratively Reweighted Least-Squares (IRLS), and by the formulas (3), (5), (8), (9).

2.1 Spatial and pixel trajectories of tracked objects

Object affine motion can be approximated by discrete trajectories of its points represented in homogeneous Cartesian coordinates and defined recursively

$$\boldsymbol{X}_{t+1} \approx (I_4 + \hat{\boldsymbol{s}}_t) \tag{2}$$

where the twist vector $\boldsymbol{s}$ defines the instant motion matrix $\hat{\boldsymbol{s}}$. The cummulated rotation R_t and translation T_t can be recovered from the matrix$G_{t+1} = (I_4 + \hat{\boldsymbol{s}}_t)G_t,\ G_0 = I_4$, requiring identification of the twist $\boldsymbol{s}_t$ for each discrete time.

The spatial trajectories are viewed by the RGB-D camera as pixel trajectories. Having the depth function $D(x, y)$ and the intrinsic camera parameter matrix $K \in \mathbb{R}^{3\text{x}3}$, we can re-project the pixel (x_t, y_t) from the image frame I_t to the spatial point (X_t, Y_t, Z_t), find the point $(X_{t+1}, Y_{t+1}, Z_{t+1})$ after instant motion determined by the twist $\boldsymbol{s}_t = (\boldsymbol{a}_t, \boldsymbol{v}_t))$, and project onto the pixel

(x_{t+1}, y_{t+1}) into the image frame I_{t+1} (where $K' \in \mathbb{R}^{2\text{x}3}$ is the upper part of K):

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} = D_t(x_t, y_t) K^{-1} \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = K' \begin{bmatrix} \frac{X_t - (\boldsymbol{a}_t)_z Y_t + (\boldsymbol{a}_t)_y Z_t + (\boldsymbol{v}_t)_x}{-(\boldsymbol{a}_t)_y X_t + (\boldsymbol{a}_t)_x Y_t + Z_t + (\boldsymbol{v}_t)_z} \\ \frac{(\boldsymbol{a}_t)_z X_t + Y_t - (\boldsymbol{a}_t)_x Z_t + (\boldsymbol{v}_t)_y}{-(\boldsymbol{a}_t)_y X_t + (\boldsymbol{a}_t)_x Y_t + Z_t + (\boldsymbol{v}_t)_z} \end{bmatrix} \tag{3}$$

2.2 Probabilistic framework for object pose tracking

The twist vector $\boldsymbol{s}$ can be found by Iterated Re-Weighted Least Square method with weights derived from probabilistic optimization given residuals $\mathbf{r}_i$ (differences between warped and observed intensities / depths)

$$\boldsymbol{s}_{MAP} = \arg\min_{\boldsymbol{s}} \left[-\sum_{i=1}^{n} \log p(\mathbf{r}_i | \boldsymbol{s}) - \log p(\boldsymbol{s}) \right] \tag{4}$$

When d-variate Student t-distribution is assumed for d-dimensional residuals, the weights can be found by EM iterations. Let ν be fixed and initially $\mu^{(1)} = \mathbf{0}_d$, $\mathbf{\Sigma}^{(1)} = I_d$. Then at the iteration j, the parameters $w, \mu, \mathbf{\Sigma}$ are updated:

1. the i-th residual $\mathbf{r}_i$ gets the weight: $w_i^{(j)} \leftarrow \frac{\nu+d}{\nu+(\mathbf{r_i}-\mu^{(j)})^T \left(\mathbf{\Sigma}^{(j)}\right)^{-1} (\mathbf{r_i}-\mu^{(j)})}$
2. the mean value: $\mu^{(j+1)} \leftarrow \frac{\sum_{i=1}^{n} w_i^{(j)} \mathbf{r}_i}{\sum_{i=1}^{n} w_i^{(j)}}$
3. the covariance matrix: $\mathbf{\Sigma}^{(j+1)} \leftarrow \frac{\sum_{i=1}^{n} w_i^{(j)} (\mathbf{r_i}-\mu^{(j+1)})(\mathbf{r_i}-\mu^{(j+1)})^T}{\sum_{i=1}^{n} w_i^{(j)}}$

The stop condition occurs if both the mean vector and the weighted covariance matrix stabilize all their components. However, it should be noted that if the weight $w_i^{(j)}$ is small, the algorithm can be stopped earlier and thus speeded up. The optimization step can keep a linear form if the weights w_i satisfy the requirements $w_i \boldsymbol{A} \mathbf{r}_i \doteq \frac{\partial \log p(\mathbf{r_i}|\boldsymbol{s})}{\partial \mathbf{r_i}}$, for a certain matrix $\boldsymbol{A} \in \mathbb{R}^{\text{dxd}}$.

Assuming the normal prior probability distribution $p(\boldsymbol{s}) = \text{Norm}_{\boldsymbol{s}}(\boldsymbol{s}_{pred}, \mathbf{\Sigma}_{pred})$ and approximating the residual function $\mathbf{r}_i$ by its linear part we get the linear equation for the optimized step $\varDelta \boldsymbol{s}$:

$$\left[\mathbf{\Sigma}_{pred}^{-1} + \sum_{i=1}^{n} w_i \frac{\partial \mathbf{r}_i^T}{\partial \mathbf{s}} \varSigma^{-1} \frac{\partial \mathbf{r_i}}{\partial \mathbf{s^T}} \right] \varDelta \boldsymbol{s} = -\mathbf{\Sigma}_{pred}^{-1} (\boldsymbol{s}_0 - \boldsymbol{s}_{pred}) - \sum_{i=1}^{n} w_i \frac{\partial \mathbf{r}_i^T}{\partial \mathbf{s}} \varSigma^{-1} \mathbf{r}_i(\mathbf{s}_0) \tag{5}$$

2.3 Residual derivatives based on intensity and depth constancy

Since an RGBD camera produces 2D photometric data and 3D spatial data we can track the object relying on the depth D constancy along the 3D point trajectory and the luminance L constancy along the pixel trajectory. In this case the vectorial residual has the form:

$$\mathbf{r}_i(\boldsymbol{s}) = \begin{bmatrix} L'(x'_i, y'_i) - L(x_i, y_i) \\ D'(x'_i, y'_i) - Z' \end{bmatrix} \tag{6}$$

The twist derivative is the combination of two twist derivatives: luminance and depth

$$\frac{\partial \mathbf{r}_i^T}{\partial \mathbf{s}} = \left[\frac{\partial L'(x'_i, y'_i)}{\partial \mathbf{s}}, \left(\frac{\partial D'(x'_i, y'_i)}{\partial \mathbf{s}} - \frac{\partial Z'}{\partial \mathbf{s}} \right) \right] \tag{7}$$

The luminance derivative is a special case of the constancy formula for photometric attributes:

$$\frac{\partial L_i'^T}{\partial \mathbf{s}} = \left[\frac{\partial \mathbf{x}'}{\partial \mathbf{s}}, \frac{\partial \mathbf{y}'}{\partial \mathbf{s}} \right] \begin{bmatrix} \frac{\partial L'^T(x', y')}{\partial x'} \\ \frac{\partial L'^T(x', y')}{\partial y'} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{X}}_i \\ I_3 \end{bmatrix} \begin{bmatrix} Z' & 0 \\ 0 & Z' \\ -X' & -Y' \end{bmatrix} \frac{K^{*\,T}}{Z'^2} \begin{bmatrix} \frac{\partial L'^T(x', y')}{\partial x'} \\ \frac{\partial L'^T(x', y')}{\partial y'} \end{bmatrix} \tag{8}$$

where we define $K^* \doteq \begin{bmatrix} f_x & g_x \\ 0 & f_y \end{bmatrix}$. The depth is differentiated along the pixel trajectory and along the 3D point trajectory:

$$\frac{\partial \left[D_i'^T - Z_i'^T \right]}{\partial \mathbf{s}} = \begin{bmatrix} \hat{\boldsymbol{X}}_i \\ I_3 \end{bmatrix} \begin{bmatrix} Z' & 0 \\ 0 & Z' \\ -X' & -Y' \end{bmatrix} \frac{K^{*\,T}}{Z'^2} \begin{bmatrix} \frac{\partial D'^T(x', y')}{\partial x'} \\ \frac{\partial D'^T(x', y')}{\partial y'} \end{bmatrix} - [Y_i, -X_i, 0, 0, 0, 1]^T \tag{9}$$

Using the Gauss Newton algorithm and formulas 5 and 7 we can align two RGBD images by minimizing the photometric and geometric error. Because of using the probabilistic model and IRLS, the method is robust to small deviations and outliers. In order to perform head pose estimation relative to a fixed-position camera sensor, we propose to crop out the face image from the whole scene. Once only facial pixels are provided, the problem comes down to estimating rigid body motion as described above.

2.4 Head region extraction

In order to crop out the face, we use a face detection algorithm based on the boosting scheme. For feature extraction we use an extended Haar filters set along with HOG-like features. For weak classification we use logistic regression and simple probability estimations. The strong classifier is based on the Gentle Boosting scheme.

Our facial landmark detection method is based on a recently popular cascaded regression scheme [18], where starting from an initial face shape estimate S^0 the shape is refined in a fixed number of iterations. At each iteration t an increment that will refine the previous pose estimate S^{t-1} is found by regressing a set of features:

$$S^t = S^{t-1} + R^t \Phi^t(I, S^{t-1}), \tag{10}$$

where R^t is the regression matrix at iteration t and $\Phi^t(I, S^{t-1})^t$ is a vector of features extracted at landmarks S^{t-1} from image I. Figure 1 summarizes the whole process of facial feature alignment. Our method was trained using parts of the 300-W [19] and Multi-PIE [7] datasets.

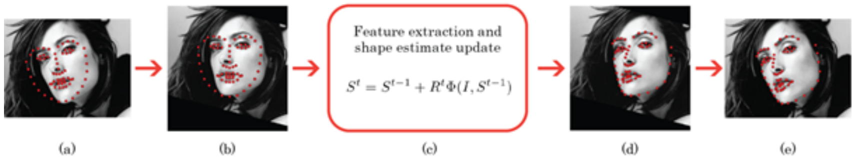
Feature extraction and shape estimate update
$S^t = S^{t-1} + R^t\Phi(I, S^{t-1})$
(a)
(b)
(c)
(d)
(e)

Fig. 1. The process of facial feature alignment. (a) The mean shape is aligned with the landmarks from the previous video frame. (b) The shape and image are transformed to a canonical scale and orientation. (c) Cascaded regression is performed for a predefined number of iterations. (d) The resulting shape. (e) An inverse transformation is applied to the shape to obtain its location in the original image.

3 Experiments

3.1 System implementation

We have developed our head pose tracking system on top of the dense visual odometry tracking algorithm as implemented by the authors of [10]. The original DVO tracker is designed to estimate the camera motion. As this is dual to the estimation of an object's pose relative to a motionless camera, we crop out face pixels from the whole image to get a consistent face pose estimation.

In order to provide the initial region of interest we run our face detection algorithm described in section 2.4 in the first frame. Once we have a coarse rectangular face area, depth-based segmentation can be performed. We begin with finding the mean head depth by averaging the depth values in a small window in the center of the face ROI. Next we perform depth thresholding to remove pixels far away from the mean head depth. The suitable depth range has been determined experimentally. During tracking the coarse region of interest is moved using estimated head motion. The depth thresholding is repeated in the same way for each subsequent frame. An example of coarse rectangular ROI and final segmented head is presented in figure 2. Once the head is segmented from the images the motion is computed using DVO tracker. The motion is estimated in a frame-to-frame manner and accumulated to obtain the transformation to the first frame.

3.2 Test methodology

Unfortunately, popular databases for head pose estimation, such as the Boston University dataset [3], do not suit our needs. We require a dataset where the RGB and depth stream are registered simultaneously and the user performs various head movements. Because of the lack of such datasets, we have decided to record our own. We have recorded ten sequences of five different people, each about one minute long. Each sequence begins with a frontal head pose, after which unconstrained, significant head movements are performed. Typically the sequences contain large head rotations along three axes (yaw, pitch, roll).

Recording the ground truth of a person's head pose is a very difficult task. Firstly, it requires specialized, expensive equipment such as magnetic trackers. Secondly, the accuracy of such a dataset is inherently limited by the accuracy of the equipment used. We propose a novel method to measure how well the head pose is estimated, which overcomes both of the mentioned limitations. We propose to estimate the motion of projected facial landmarks found in the input data. The algorithm can be outlined as follows.

For each image:

1. Estimate head pose in the image
2. Use the estimation to transform the head pose to the initial view
3. Calculate the projection of the transformed facial image
4. Calculate the distance between the projected coordinates of landmarks in the first and current image - this is proportional to the tracking accuracy

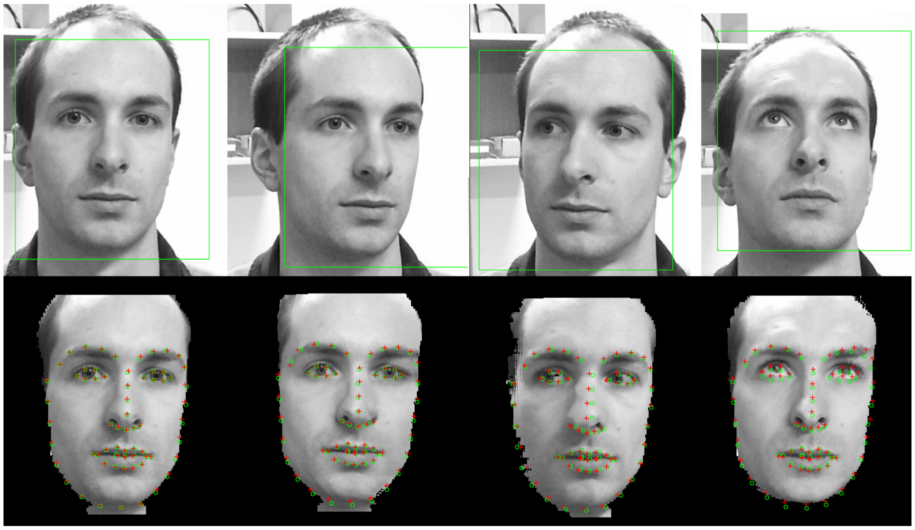

Fig. 2. Head pose dataset. Landmark locations from the reference frame are marked by green circles, those from the current warped frame are marked by red crosses.

As facial features we propose to use the eye inner and outer corners, and lip corners. The facial features can be tagged manually, but what is even more important, they can be determined automatically by a face landmark localisation algorithm. This is the biggest advantage of the proposed methodology - long sequences can be reliably evaluated without the necessity to perform tedious and error-prone manual tagging of each frame. Figure 2 shows the type of performed head movements in one of the test sequences (top), along with warped segmented faces using the described algorithm (bottom).

3.3 Results

The results on sequences that we have recorded are shown in table 1. The set of six chosen facial features (inner and outer eye corners, lip corners) was used to calculate the errors in the top row. For comparison, the bottom row shows the average errors for all points (68) detected by the facial landmark alignment algorithm. The average errors in pixels have been calculated for all the frames in each test sequence. We have used an RGB camera resolution of 1920x1080 pixels and recorded people who were about 1m away, which resulted in the face images having around 100 pixels between the eyes. For depth recording we have used the Kinect 2 sensor having 512x424 resolution in the image plane and 1mm depth resolution. The proposed algorithm works in real-time on a CPU.

Sequence	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	Seq 8	Seq 9	Seq 10
Subset error	3.78	3.48	2.77	4.4	3.37	4.15	2.76	3.69	4.38	4.62
Full set error	4.92	4.85	4.59	5.48	4.07	4.74	4.21	4.61	5.61	5.82

Table 1. Head pose estimation accuracy on ten sequences [pixels]

The measured errors of around 3-4 pixels demonstrate very high accuracy of the proposed head pose estimation method. Because the proposed tracking algorithm uses a probabilistic model and IRLS, it can handle outliers and non-rigid face motion by downvoting the pixels inconsistent with global motion.

We wish to point out, that the facial landmark localisation algorithm used in our experiments is sensitive to illumination and the inaccuracy of the facial landmark localization algorithm is a part of the measured errors in table 1. In our opinion, judging from a set of manually tagged images, the proposed head pose estimation algorithm has better accuracy than indicated by the measurements in table 1 by at least 1 pixel. This is better than the accuracy measured in [2].

Figure 3 shows the trajectory of head motion that was performed similarly in each test sequence. The trajectory is displayed as a virtual camera trajectory relative to a motionless head for better visualization. We would like to note that the trajectory finishes in roughly the same place as it started even though the motion is estimated in a frame-to-frame fashion. This proves that there is very small tracking drift, which is a strong advantage of the proposed algorithm.

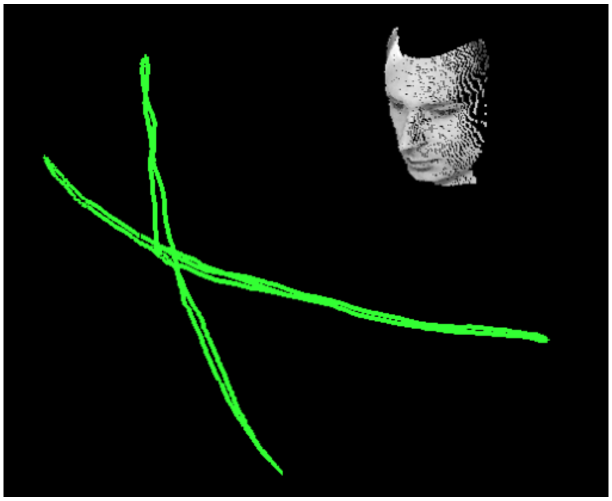

Fig. 3. Head pose trajectory presented as virtual camera trajectory

4 Conclusions

We have presented a novel, highly accurate approach to head pose tracking. Based on the obtained accuracy measurements, as well as observations of live performance when stabilizing the head image, we conclude that the presented method surpasses the current state-of-the-art in the field of RGBD head pose tracking. The real-time algorithms for camera motion estimation proposed in [10] can be easily adapted to the scenario of head pose tracking and used for various use cases such as eye gaze tracking or face frontalization. We have also presented a new head pose tracking accuracy evaluation methodology which can be easily used to assess algorithm performance.

A good direction for future work is exploring the possibility of using keyframes and thus removing accumulated tracking error from the proposed head pose estimation method.

References

1. T. Baltrušaitis, P. Robinson, and L. Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
2. Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3d deformable face tracking with a commodity depth camera. In *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ECCV'10, pages 229–242, Berlin, Heidelberg, 2010. Springer-Verlag.
3. M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(4):322–336, 2000.

4. B. Czuprynski and A. Strupczewski. High accuracy head pose tracking survey. In *Active Media Technology*, volume 8610 of *Lecture Notes in Computer Science*, pages 407–420. Springer International Publishing, 2014.
5. G. Fanelli, T. Weise, J. Gall, and L. Van Gool. Real time head pose estimation from consumer depth cameras. In *Pattern Recognition*, volume 6835 of *Lecture Notes in Computer Science*, pages 101–110. Springer Berlin Heidelberg, 2011.
6. O.S. Gedik and A.A. Alatan. Fusing 2d and 3d clues for 3d tracking using visual and range data. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1966–1973, July 2013.
7. R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–8, Sept 2008.
8. S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.
9. J. Jang and T. Kanade. Robust 3d head tracking by online feature registration. *The IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.
10. C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.
11. C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2013.
12. S. Li, K. Ngan, and L. Sheng. A head pose tracking system using rgb-d camera. pages 153–162, 2013.
13. W. Liao, D. Fidaleo, and G. Medioni. Robust, real-time 3d face tracking from a monocular view. *EURASIP Journal on Image and Video Processing*, 2010.
14. S. Malassiotis and M. G. Strintzis. Robust real-time 3d head pose estimation from range data. *Pattern Recognition*, 38(8):1153 – 1165, 2005.
15. L. Morency, J. Whitehill, and J. Movellan. Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation. In *Automatic Face Gesture Recognition, 8th IEEE International Conference on*, 2008.
16. E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2009.
17. P. Padeleris, X. Zabulis, and A.A. Argyros. Head pose estimation on depth data based on particle swarm optimization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Conference on*, pages 42–49, June 2012.
18. S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1685–1692, June 2014.
19. C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 896–903, June 2013.
20. N. Smolyanskiy, C. Huitema, L. Liang, and S. Eron Anderson. Real-time 3d face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32(11):860 – 869, 2014.
21. F. Steinbruecker, J. Sturm, and D. Cremers. Real-time visual odometry from dense rgb-d images. In *Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV)*, 2011.