Article

Highly Accurate and Fully Automatic 3–D Head Pose Estimation and Eye Gaze Estimation Using RGB-–D Sensors and 3D Morphable Models

Reza Shoja Ghiass [1,†] and Denis Laurendeau [2,†]

[1] Affiliation 1; reza.shoja@gmail.com
[2] Affiliation 2; Denis.Laurendeau@gel.ulaval.ca
* Correspondence: Denis.Laurendeau@gel.ulaval.ca; Tel.: +1-(418) 656-2131, ext. 2979
† Current address: 1665 Rue de l'Universite, Universite Laval, Quebec, QC, Canada, G1V 0A6

Version October 15, 2018 submitted to Sensors

Abstract: This work addresses the problem of automatic head pose estimation and its application in 3D gaze estimation using low quality RGB–D sensors without any subject cooperation or manual intervention. The previous works on 3D head pose estimation using RGB–D sensors require either an offline step for supervised learning or 3D head model construction which may require manual intervention or subject cooperation for complete head model reconstruction. In this paper, we propose a 3D pose estimator based on low quality depth data, which is not limited by any of the aforementioned steps. Instead, the proposed technique relies on modeling the subject's face in 3–D rather than the complete head, which in turn, relaxes all of the constraints with the previous works. The proposed method is robust, highly accurate and fully automatic. Moreover, it does not need any offline step. Unlike some of the previous works, the method only uses depth data for pose estimation. The experimental results on the Biwi head pose database confirm the efficiency of our algorithm in handling large pose variations and partial occlusion. We also evaluate the performance of our algorithm on IDIAP database for 3D head pose and eye gaze estimation.

Keywords: 3–D Morphable Models; 3–D Head Pose Estimation; 3–D Eye Gaze Estimation; Iterative Closest Point; RGB–D Sensors;

1. Introduction

Head pose estimation is a key step in understanding human behavior and can have different interpretations depending on the context. From the computer vision point of view, head pose estimation is the task of inferring the direction of head from digital images or range data compared to the imaging sensor coordinate system. In the literature, the head is assumed to be a rigid object with three degrees of freedom, i.e., the head pose estimation is expressed in terms of yaw, roll and pitch. Generally, the previous works on head pose estimation can be divided into two categories: (i) the methods based on 2D images, and (ii) depth data [1]. The pose estimators based on 2D images generally require some pre–processing steps to translate the pixel–based representation of the head into some direction cues. Several challenges such as camera distortion, projective geometry, lighting, changes in facial expression exist in 2D image–based head pose estimators. A comprehensive study of pose estimation is given in [1] and the reader can refer to this reference for more details on the literature.

Unlike the 2D pose estimators, the systems based on 3D range data or their combination with 2D images have demonstrated very good performance in the literature [2–7]). While most of the work on 3D pose estimation in the literature is based on non–consumer level sensors [8–10], recent advances in production of consumer level RGB–D sensors such as the Microsoft Kinect or the Asus Xtion has facilitated the design and implementation of real-time facial performance capture systems such as consumer-level 3D pose estimators, 3D face tracking systems, 3D facial expression capture systems and 3D eye gaze estimators. In this paper, we focus on the recent 3D pose estimators and tracking systems based on consumer level RGB–D sensors.

According to the literature [11], 3D head pose estimation is a key part of 3D eye gaze estimation. In other words, head pose estimation problem is highly correlated with the problem of gaze estimation. In this paper, we propose the design of reliable head pose estimation systems first and integrate it in a state of the art gaze estimation system next.

1.1. Related Work on 3D Pose Estimation using RGB–D sensors

The 3D head pose estimation systems can be divided into three categories: (i) statistical approaches, (ii) model based posed estimation methods, and (iii) facial feature based pose estimation techniques [12]. Each of these approaches comes with their specific limits and advantages. Statistical methods may need a large database for training a regressor. However, they can estimate the subject head pose on air, i.e., the system can estimate the head pose for each frame even in a shuffled video sequence. In contrast, model based approaches generally need an offline step for subject–specific head model reconstruction with significant subject cooperation. Next, a point cloud registration technique such as rigid/non–rigid ICP should be used to register the model with depth data. In other words, unlike the supervised learning based approaches, they are generally based on tracking. So, re–initialization becomes a challenge. Facial feature based pose estimation techniques try to track facial features or patches, which in turn, can help in calculation of pose using techniques such as PnP [13] or encoding the face 3D shape using view-invariant descriptors and infer head pose through matching [14].

To the best of our knowledge, one of the most important works on pose estimation using consumer level *RGB–D sensors* is the work of Fanelli et al. [2,3]. As the authors provide a ground truth data and a database for comparison, their work has become the gold standard for comparison in the literature. They work falls in the category of statistical approaches. In their work, the authors proposed a pose estimation system based on Random Forests. For the evaluation of their system, they acquired a database of 20 subjects which is called the Biwi head pose database. Next, they divided the database into a training and test set. Afterward, a commercial face tracker was used for annotation of the training set, i.e., a subject specific head model was constructed using the commercial system to match each person's identity and track the head in training depth frames. The commercial tracker measured a subject's 3D head locations and orientations, which in turn, were used to train their regression based system. Finally, some patches of fixed size from the region of the image containing the head as positives samples, and from outside the head region as negatives were randomly selected for training the system. A major limitation of this system was that it required an offline training phase with subject cooperation. Moreover, the performance of the system in the testing phase was subject to the output of the commercial head tracker in the training phase. In [3] the authors continued their previous work [2] by creating a dataset of synthetic depth images of heads, and extracting the positive patches from the synthetic data, while using the original depth data to to extract negative patches. A drawback of this system was the limited number of synthetic models and negative patches for performing a regression task, without learning subject's own head [3]. [15] proposed a system based on cascaded tree classifiers with higher accuracies than Fanelli et al. [9] proposed a 3D face tracker based on particle filters. The main idea in their system was the combination of depth and 2–D image data in the observation model of the particle filter.

With the main intention of designing a gaze estimator, Funes and Odobez [7,16] proposed the first model based pose estimator by building a subject–specific model based face tracker using Iterative Closest Point (ICP) and 3D Morphable Models. Their system was not only able to estimate the pose, but was also able to track the face and stabilize it. A major limitation of their method was the offline step for subject specific 3D head model reconstruction. For this purpose, they manually placed landmarks

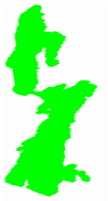


Figure 1. Depth data obtained from the first frame and visualized from profile. It consists of both facial part and spurious data.

(eye corners, eyebrows, mouth corners) on RGB image of the subject, and consequently added an extra term to the cost function in their ICP formulation. In other words, their ICP formulation was supported by a manual term. Moreover, the user had to cooperate with the system and turn the head from left to right. Recently, the authors have proposed a more recent version of their system in the work of [12] without the need for manual intervention.

1.2. Contribution of the proposed

Unlike [2,3,7,16] our proposed system does not require any commercial system to learn a subject's head nor any offline step. A key contribution of our approach is proposing a method to automatically learn a subject's 3D face rather than the entire 3D head. As a consequence, we no longer need subject's cooperation (i.e., turning the head from left to right) which is important in previous works for model based pose estimation systems. In addition, unlike [7] our system does not require any manual intervention for model reconstruction. Instead, we rely on Haar features and boosting for facial feature detection, which in turn, can be used for face model construction. Note that we use only one RGB frame for model reconstruction. The tracking step is based on depth frames only. After learning a subject's face, the pose estimation task is performed by a fully automatic, user non–cooperative and generic ICP formulation without any manual term. Our ICP formulation is robustified with Tukey functions in tracking mode. Thanks to the Tukey functions, our method successfully tracks a subject face in challenging scenarios. The outline of the paper is as follows: The method details are explained in section 2. Afterward, the experimental results are discussed in section 3. Finally, the conclusions are drawn in section 4.

2. Method Details

Our method consists of four key steps:(i) Geometry processing of a generic face model and the first depth frame, (ii) generic face model initialization (i.e., model positioning at the location of the head), (iii) subject–specific face model construction by morphing the initialized generic model, and (iv) tracking the face in the next depth frames using the subject–specific face model. In our proposed system, we only model the face of the subject rather than the entire head, which in turn, helps us to design a very robust, accurate and non–cooperative head tracking and pose estimation system. To accomplish this goal, a generic model is positioned on the subject's face in depth data. Next, it learns the subject's face and finally starts to track it. Both *positioning* a generic model on subject's face and *tracking* it in depth data are accomplished using an ICP based technique. However, the ICP registration technique which serves for positioning faces a major challenge: the generic model is a model of a complete head and not only the face. On the other hand, the depth data contains not only the face of the subject as well as the other parts such as the torso or the background (Fig. 1). Note that a major difficulty with ICP is its sensitivity to outliers and missing data between two 3D point clouds. To tackle this problem in model initialization, we perform geometry processing which is explained next.

2.1. Geometry processing

In this step, the goal is trimming the depth data and the generic model in order to remove spurious data and outliers. Note that we perform this step on the first depth frame only. The reason is that we

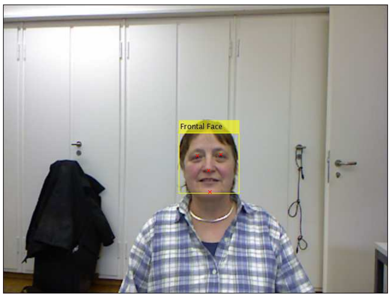


Figure 2. Face and facial features detection from the first RGB frame of a subject

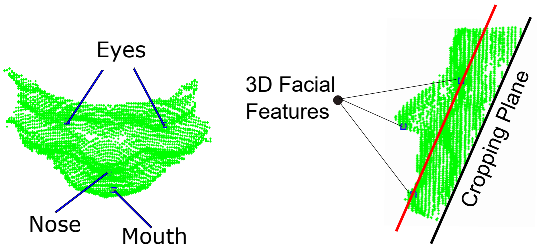


Figure 3. Trimming the first depth frame: (left) upper view, (right) profile view

should initialize (position) the model at the position of the subject's head, before tracking starts. For this purpose, we capture the entire environment using the Kinect. Next, we filter out the spurious point cloud and just keep the region of interest in the first depth frame, i.e., the facial surface. For this purpose, the first depth frame is automatically trimmed in order to discard the residual data. To this end, we need to automatically detect and localize the facial features (i.e., eyes, nose, and mouth) on the point cloud in order to determine the way the depth data should be trimmed. Detection of the facial features from a noisy depth frame directly is a challenge. Fortunately, the Kinect provide us the first RGB frame. So, the face and facial features are detected on the first RGB frame by first using Haar features and boosting ([17]). Fig. 2 demonstrates an example of the face and facial feature detection. As some false detections may occur, the next step is to reject them automatically. This is accomplished by utilizing the prior knowledge about the structure of a face and the relative positions of eyes, nose and mouth on a detected face.

After the features are detected on the first RGB frame, their 3*D* loci are determined on the first depth frame through back projection using Kinect calibration data. In order to trim the depth data, a 3D plane passing through the 3D coordinates of the eyes and mouth is defined and shifted by an offset equal to the distance between the left and right eye. The shifted plane is called *the cropping plane*. Next, the depth data beneath the plane is discarded. Fig. 3 shows the 3D loci of the facial features on the corresponding depth data trimmed by the cropping plane.

Once the subject's face is captured and trimmed in 3D, the next step is to construct a model which simulates the subjects face (rather than the complete head). The type of 3D model we use to simulate the subject's identity is a family of Active Appearance Models (AAMs) called 3D Morphable models. Using these models, a subject's 3D head scan can be reconstructed by adding a set of weighted principal components (PCs) to the mean shape (the mean shape is the mean of all of the 200 subject's face scans in the database). For instance, we focus on the mean shape of the model. Much like trimming the depth data, the mean shape of the 3D Morphable model is trimmed in order to facilitate the procedure of subject specific model construction through registration. In this context, a plane similar to that of Fig. 3(b) is fitted to the model's mean shape. Once the mean shape is trimmed, it should be scaled to the size of the subject's face in 3D space. To this end, the model is scaled so the distance between the left and right eyes of the model and that of the subject's face scan (i.e., the first depth frame) becomes equal. Fig. 4 demonstrates the mean shape of model, $\mathbf{m}$, before and after trimming.

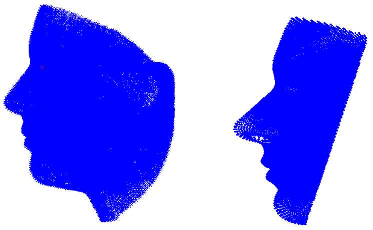


Figure 4. Trimming the 3D Morphable model mean shape: (left) before trimming, and (right) after trimming

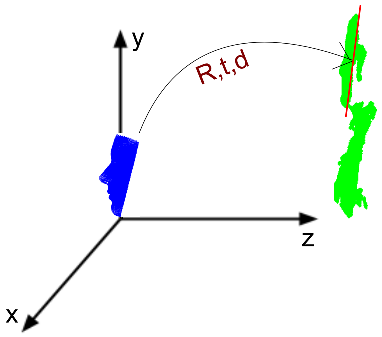


Figure 5. Positioning the model on the face. The blue is the trimmed model, while the green is the subject scanned by the Kinect.

2.2. Generic Model Positioning

After processing both model and depth data, the trimmed model is positioned on the face of the subject using rigid ICP. Fig. 5 demonstrates this step for the model and first depth frame. After initializing the generic model on the subject's face, the model is ready to morph and learn the subject's face (Sec. 2.3) and track it afterward(Sec. 2.4).

2.3. Learning and Modeling the Subject's Face

Capturing the subject's facial shape variations via morphing the mean shape is the main objective of this step. This problem can be considered as finding the weights of shape PCs in the 3D Morphable model, where each weight describes the contribution of its corresponding PC in simulating a subject's face. Much like the generic ICP problem, this part also can be described by minimization of a cost function. So, we can unify both ICP and PCA terms into a unique equation and reformulate a more generic ICP problem through minimizing the following energy function ([18]):

$$\begin{aligned} \mathbf{E}(Z,\mathbf{d},\mathbf{R},\mathbf{t}) &= \omega_1 \mathrm{E}_{match} + \omega_2 \mathrm{E}_{rigid} + +\omega_3 \mathrm{E}_{model} \\ \mathrm{E}_{match} &= \sum_{i=1}^{n} (\mathbf{N}_i^T(\mathbf{z}_i - C_Y(\mathbf{z}_i)))^2 \\ \mathrm{E}_{rigid} &= \sum_{i=1}^{n} \parallel \mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t}) \parallel_2^2 \\ \mathrm{E}_{model} &= \sum_{i=1}^{n} \parallel \mathbf{z}_i - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i) \parallel_2^2 \end{aligned} \qquad (1)$$

where Y is the target surface in R^3, X is the source surface, and Z is a deformed version of X which should be aligned with Y. Notice also that $C_y(\mathbf{z}_i)$ is the closest point in the target surface to the point $\mathbf{z}_i$ ($i = 1, 2, ..., n$, where n is the number of points in source). In this equation, the first term is the point-to-plane matching error, the second term is the point-to-point matching error, while the third term is the model error (for more details about these error the reader is referred to [18]). The energy function can be minimized by linearizing Eq. 1 and iteratively solving the following linear system:

$$\begin{aligned}\underset{Z_i^{t+1},\mathbf{d},\widetilde{R},\widetilde{t}}{\operatorname{argmin}} \sum_{i=1}^{n} \omega_1(\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - C_Y(\mathbf{z}_i)^t))^2 + \\ \omega_2 \parallel \mathbf{z}_i^{t+1} - (\widetilde{R}(\mathbf{R}\mathbf{x}_i + \mathbf{t}) + \widetilde{t}) \parallel_2^2 + \\ \omega_3 \parallel \mathbf{z}_i^{t+1} - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i) \parallel_2^2\end{aligned} \quad (2)$$

where t is the number of iterations, $\mathbf{z}_i^0 = x_i$, d contains the weights of PCs, and R̃ and t̃ are the linear updates which we obtain for the rotation (**R**) and translation (**t**) matrices at each iteration. Notice that n_i is the normal to the surface at point $C_Y(\mathbf{z}_i)^t$, i.e., point to plane matching error. For more details, the reader is referred to the tutorial by [18].

2.4. 3D Head Tracking and Pose Estimation

Once the model is constructed from the first depth frame, the pose (orientation alone) of the head can be calculated directly from the rotation matrix, R, in terms of roll, pitch and yaw [19]. In Sec. 2.3, the rotation matrix corresponding to the first depth frame of the subject was obtained during model construction. A question arises here: How can one obtain the rotation matrices for the next depth frames? Indeed, this question is addressed by 3D registration of the form of Eq. 1 with some differences. The first difference is that we no longer need to capture the subject's face variations, d, because it is calculated once for the entire procedure. So, the E_{model} term is dropped from Eq. 1. The other difference is that we no longer need to trim the next depth frames. The reason is that the model is already fitted to the first depth frame during model construction (see Fig. 5) and we expect the system to work in tracking mode. In tracking mode, head displacement in the next frame compared to the current frame is small and the model displacement should be very small compared to the initialization mode. So, instead of trimming the next depth frames, one can take advantage of registration using *Tukey functions*, which will filter out bad correspondences with large distances. The pose estimation procedure for the next depth frames is as follows: for the second depth frame, the model rotation and translation increments are calculated relative to that of the first depth frame. Next, the rotation and translation matrices for the second depth frame are obtained by applying the updates to the rotation and translation matrices in the first depth frame. This procedure is continued for the next frames. For each frame, the head pose can be directly calculated from the rotation matrix in terms of pitch, yaw and roll.

Robustness of Registration to Outliers

As mentioned, partial overlap between source and target and outliers in the data are the most challenging problems in registration through ICP [20]. Two types of outliers exist: (i) outliers in the source point cloud and (ii) outliers in the target point cloud. Discarding unreliable correspondences between the source and the target is the most common way to handle this problem. In Sec. 2.2, this goal was accomplished by trimming both model and depth data in the first depth frame. However, for the 3D face tracking mode the same method can not be used. The reason is that the initialization modality is based on detection of facial features. Applying facial feature detection for each frame can decrease the frame rate at which the system operates. On the other hand, a limit of our method is that the system can not start from an extreme pose, as the facial feature detection algorithms will fail. Fortunately, as the model is already positioned onto the face of the subject, we no longer need to trim the upcoming depth frames to perform tracking using ICP. Instead, we use Tukey functions to robustify the ICP. Tukey functions assign less weight to the bad correspondences and decrease or remove their effect on the energy function.

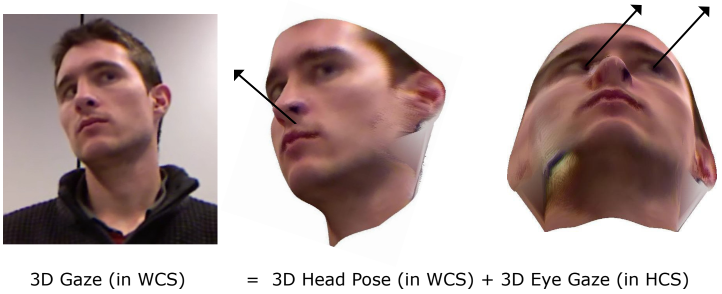


Figure 6. Applying the face texture in (a) to the subject specific model. The subject is chosen from IDIAP database. The figure in (b) is the pose free model visualized from down side of the subject, the figure in (c) is the same pose free model visualized from right side, the figure in (d) is the same pose free model visualized from left side, while the figure in (e) is the TPFM visualized from frontal view

Robustness of Registration to extreme pose

A question may arise at this point: Can the method handle the case of a face with extreme pose where most facial parts cannot be sensed by the Kinect? In this case the model should be registered with a partial point cloud of subject's face. This leads to increasing the number of points in the source (trimmed model) without good correspondences in the target (partial point cloud of face). As a result, such points will form bad correspondences with relatively large Euclidean distance. Fortunately, we also address this problem by using robust functions in the tracking mode, as robust functions discard/decrease the effect of such bad correspondences in the energy function. To clarify this, notice that bad correspondences inherently produce large Euclidean distances, while this is not the case for good correspondences. On other other hand, narrow robust functions act as low pass filters and discard the bad correspondences.

Robustness of Registration to Facial Expression Changes

As we use rigid ICP, facial expression changes may be considered as a challenging factor. In this context, Funes and Odobez [7] used a mask and only consider the upper part of the face in the rigid registration part of their system. Notice that we do not use such a mask. The reason is that, most of the time, the subject may not show *significant* facial expression changes (such as laughing or opening the mouth). On the other hand, relying on more data of the face may result in a more robust registration task. The problem becomes more challenging if we consider that self–occlusion may occur on the upper part of the face. Due to this reasons, we prefer not to use a mask. Instead, we rely on the robust Tukey functions to improve the robustness in the case of facial expression changes.

2.5. Gaze Estimation

In this paper, we will study the impact of the proposed 3D head pose estimator on the appearance–based 3D gaze estimation systems of [16]. Inspired by a series of works in the literature, we use the two supervised learning methods to calculate the gaze direction on the head coordinate system (we will refer to this vector as the Gaze-on-Head vector or GoH in the remainder of this paper): (i) A K–NN based approach and (ii) Adaptive Linear Regression (ALR) [7,16,21].

2.5.1. Head Pose Stabilization

This step is a pre–processing step for gaze estimation. As soon as the head pose is calculated, the texture of the corresponding RGB frame can be back–projected to a pose free 3D head model (Fig. 5). The 3D gaze vector is calculated on this pose free head model next, and the 3D head pose is added back to the calculated 3D gaze vector to bring the 3D gaze vector back to the world coordinate system (See Fig. 6).

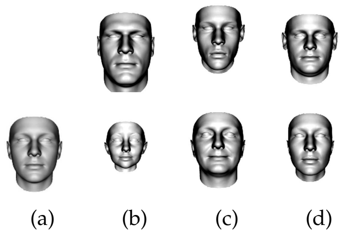


Figure 7. The mean [a] and the first [b], second [c] and third [d] principal components (visualized: ± 5 standard deviation) of the shape model. The images are taken from the database website

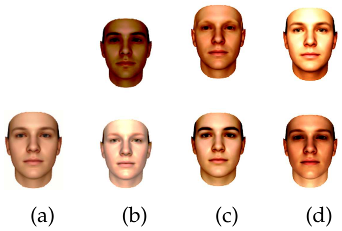


Figure 8. The mean [a] and the first [b], second [c] and third [d] principal components (visualized: ± 5 standard deviation) of the texture model. The images are taken from the database website.

3. Experimental evaluation

In this section we report our empirical evaluation. We start by describing the data sets used in our experiments, follow with an explanation of the evaluation protocol, and finish with a report of the results and their discussion.

3.1. Databases

3.1.1. 3D Basel Face Model (BFM)

The 3D Basel Face Model (BFM) is a Morphable model calculated from registered 3D scans of 100 male and 100 female faces. The model geometry consists of 53,490 3D vertices connected by 160,470 triangles. The model is given by the followings:

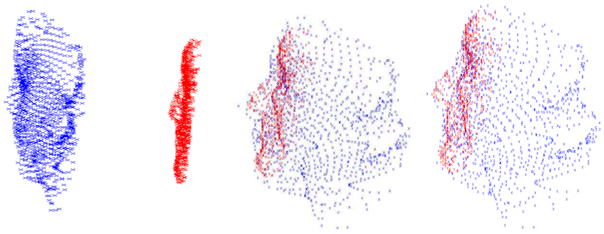
- The mean shape
- 199 principal components (PCs) of shape obtained by applying PCA on 200 subjects facial shape in the database
- The variance of shape
- The mesh topology
- The mean texture
- 199 principal components (PCs) of texture obtained by applying PCA on 200 subjects facial texture in the database
- The texture variance

Figs. 7 and 8 demonstrate the mean and the first, second and third principal components (visualized: ± 5 standard deviation) of the shape and texture model respectively.

Any unknown face can be explained as a linear combination of the principal components and the mean shape/texture. In this paper, we only use the shape data set (i.e, shape principal components together with mean shape) for the construction of a subject's specific face model (i.e, the head trackers).

3.1.2. Biwi Kinect Head Pose Database

We used the Biwi Kinect Head Pose Database ([2,3]) to evaluate the effectiveness of our method. There are reasons for this choice. Firstly, to the best of our knowledge, it is the only RGB–D database



(a) a (b) b (c) c

Figure 9. Registration procedure to capture subject's variations, d, together with transformation matrices R and t: (a) before model initialization, (b) after model initialization and before capturing subject's face variations, and (c) after the initialization is accomplished (ready for tracking).

for pose estimation reported in the literature. Secondly, it provides ground truth data for comparison, and we wanted to make our results directly comparable to not only those of Fanelli *et al.* ([2,3]), but also to the recent works which have used this database. The dataset contains over 15000 depth frames and RGB image of 20 people, six females and fourteen males, where four people were recorded twice. The head pose ranges through about 75 degrees yaw and 60 degrees pitch. The ground truth for head rotation is also provided by a commercial software.

3.1.3. EYEDIAP Database

We used EYEDIAP gaze database [22] to evaluate the effectiveness of *gaze estimation* part of our method. There are several reasons for this choice. Firstly, to our best knowledge it is the only Kinect based database for gaze estimation in the literature. Secondly, it provides ground truth data for comparison of gaze estimation (bot not pose) results. In addition, we wanted to make our results statistically comparable to the work of Funes and Odobez [7,16]. The dataset contains over 4450 depth frames and RGB image of 16 people among them 14 subjects participated in a screen based gaze estimation scenario. Each session itself is divided into two other session where the subject was asked to keep the head stationary or moving.

3.2. Evaluation methodology

3.2.1. Subject Specific Model Construction

We evaluated the proposed algorithm in a setting in which the first RGB frame and the first depth frame were used for learning in an unsupervised context, while the other depth frames were used for testing. Fig. 9 demonstrates the registration procedure. In this figure, the blue point cloud is the (down sampled) mean shape [1] of the BASEL data, while the red point cloud is the trimmed depth scan of the first subject in the Biwi database (the subject in Fig. 2). We want to register the two shapes with each other and, at the same time, capture the variation of the subject's face by minimizing Eq. 1.

3.2.2. Pose Estimation and Tracking

After the subject's specific model is constructed from the first (trimmed) depth frame, we drop the model term from the energy function and continue the registration of the model and depth data. The pose estimation for each frame can be directly calculated from the rotation matrix, R, obtained from registration. Fig. 10(a) shows a sample where the model (red) is registered with the depth data (blue). The model is superimposed on the corresponding RGB frame through the Kinect calibration data in Fig. 10(b) for a better visualization.

[1] Notice that trimming the model is not shown here, but it is considered in calculations.

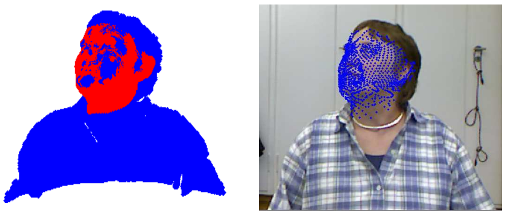


Figure 10. An instant of the tracking mode: the model (red) tracks the depth data (blue) and it is back–projected to the RGB frame

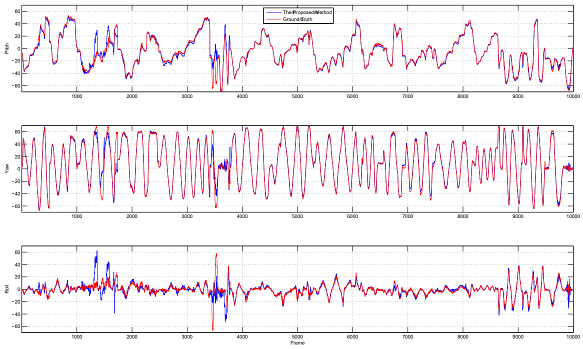


Figure 11. The experimental results of the proposed facial pose estimator on Biwi head pose database compared to the ground truth.

Fig. 11 shows the result of pose estimation in terms of yaw, depth, and roll for 10000 frames from the Biwi head pose database compared to the ground truth.

A summary of the key evaluation results and method features of the proposed algorithm compared to two previous works is shown in Table 1. Four criteria were considered to compare the systems according to Pauly:

- Accuracy compared to the ground truth data
- Performance in terms of time and memory efficiency
- Robustness to occlusions, poor lighting and fast motion
- Usability in real scenarios, i.e., user–specific training, system calibration and manual intervention need to be kept to a minimum.

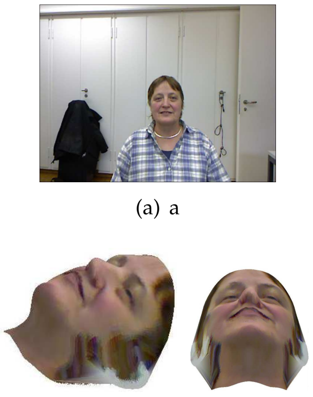
On one hand our system demonstrates better results than [2,3] in terms of average error and standard deviation. Both systems proposed by Fanelli *et al.* show slightly better performance than our system *only* in terms of standard deviation of roll. Moreover, our system is generic and the training phase is performed with a single RGB/depth frame. On the other hand, the systems of Fanelli *et al.* can work on a frame by frame basis, while our system can only work in tracking mode (i.e., the subject's head motion in successive frames should be small). Notice that both systems of Fanelli *et al.* need a training phase supported by a commercial face tracker, while we propose a new face tracker in this work. As both systems of Fanelli *et al.* require a training phase based on positive and negative patches cropped from a database of 20 subjects, the generic aspect of their system is an issue. We also compared our system to the other non–model based approaches of [9,13,15,24], and the results show the effectiveness of the proposed system. The only comparable system is the model–based work of [12] which shows very good precision too.

3.2.3. Face Stabilization and Gaze Estimation

Using the calibration matrices of the Kinect, it is possible to apply the texture from the RGB frames to the constructed 3D model. [7,16] used this property for the first time to accomplish gaze estimation by manual intervention in ICP registration. This idea can still be investigated further in the future in

Table 1. A summary of the key evaluation results and method features of the proposed algorithm, and the two previous works based on supervised learning Fanelli *et al.* [2,3]. Notice that we do not compare the results with those of Funes and Odobez [7] for face pose estimation due to lack of details in yaw, roll and pitch. Legend: ● very good; ◑ good; ○ weak.

	Pose Estimation Error			Specifications		
	Pitch	Yaw	Roll	Accuracy	Robustness	Usability
Our Proposed Method	0.1±6.7°	0.25±8.7°	0.26±9.3°	●	●	◑
1st report Fanelli *et al.*	8.5±9.9°	8.9±13.0°	7.9±8.3°	◑	●	◑
2st report Fanelli *et al.*	5.2±7.7°	6.6±12.6°	6.0±7.1°	◑	●	◑
Tulyakov *et al.*	N/A	3.18±5.3°	N/A	●	●	◑
Baltrusaitis *et al.*	5.10°	6.29°	11.29°	◑	●	◑
Rekik *et al.*	4.32±2.65°	5.13±3.33°	5.24±3.33°	◑	●	◑
Papazov *et al.*	2.5±7.4°	3±9.6°	3.8±16°	◑	●	◑
Martin *et al.*	2.54°	2.57°	3.62°	●	●	◑
Yu *et al.*	1.7°	2.5°	2.3°	●	●	◑



(a) a

(b) b

(c) c

Figure 12. Applying the face texture from the RGB frame in (a) to the subject's specific model. The figure in (b) is the model visualized from side view, while the figure in (c) is the same model visualized from down view

order to have a fully automatic gaze estimation system. Fig. 12(a) shows the RGB frame of a subject, while Figs. 12(a) and (b) shows two different views of the same RGB frame warped to the subject's specific 3D model (in the facial area) using our proposed method. Note that the artifacts in perimeter of the 3D head model is due to the cropping plane.

A summary of the key evaluation results and method features of the proposed algorithm compared to two previous works are shown in Tabs. 2 to 5. Three criteria were considered to compare the systems:

Table 2. A summary of the key evaluation results and method features of the proposed algorithm, and the previous work when Adaptive Linear Regression (ALR) is used and the subjects keep the head stationary: ● very good; ◑ good; ○ weak.

	Gaze Estimation Error		Specifications		
	Left Eye	Right Eye	Accuracy	Robustness	Usability
Our Proposed Method	7.55°	6.89°	◑	●	●
Funes and Odobez Method	9.73°	10.5°	◑	◑	◑

Table 3. A summary of the key evaluation results and method features of the proposed algorithm, and the previous work when K–NN is used and the subject keeps the head stationary: ● very good; ◑ good; ○ weak.

	Gaze Estimation Error		Specifications		
	Left Eye	Right Eye	Accuracy	Robustness	Usability
Our Proposed Method	8.83°	6.49°	◑	●	●
Funes and Odobez Method	10.23°	9.56°	◑	◑	◑

Table 4. A summary of the key evaluation results and method features of the proposed algorithm, and the previous work when Adaptive Linear Regression (ALR) is used and the subjects have free head motion: ● very good; ◑ good; ○ weak.

	Gaze Estimation Error		Specifications		
	Left Eye	Right Eye	Accuracy	Robustness	Usability
Our Proposed Method	9.78°	9.49°	◑	●	●
Funes and Odobez Method	15.57°	14.2°	◑	◑	◑

- Accuracy compared to the ground truth data
- Robustness to occlusions, bad lighting and fast motions
- Usability in real scenarios, i.e., user–specific training, system calibration and manual intervention need to be kept to a minimum.

Our system demonstrates better results than Funes and Odobez in terms of average gaze estimation error. One possible reason for this can be the high precession of our pose estimation system which performs almost like a commercial state of the art pose estimator, while the pose estimator of Kenneth and Odobez shows slight deviation from our precise pose estimator, which result in a non precise texture warping on head model, which in turn can affect the gaze estimation process.

Table 5. A summary of the key evaluation results and method features of the proposed algorithm, and the previous work when K–NN is used and the subjects have free head motion: ● very good; ◑ good; ○ weak.

	Gaze Estimation Error		Specifications		
	Left Eye	Right Eye	Accuracy	Robustness	Usability
Our Proposed Method	9.03°	8.86°	◑	●	●
Funes and Odobez Method	17.97°	14.63°	◑	◑	◑

4. Conclusion

This work addressed the problem of automatic facial pose and gaze estimation without subject cooperation or manual intervention using low quality depth data provided by the Microsoft Kinect. The previous works on pose estimation using the Kinect were based on supervised learning or require manual intervention. In this work, we proposed a 3D pose estimator based on low quality depth data. The proposed method is generic and fully automatic. The experimental results on the Biwi head pose database confirm the efficiency of our algorithm in handling large head pose variations and partial occlusion. Our results also confirmed that model based approaches outperform the other approaches in terms of precision. We also evaluated the performance of our algorithm on the IDIAP database for 3D head pose and eye gaze estimation and we obtained promising results.

5. Acknowledgement

The research presented in the paper was funded by grant F506-FSA of the Auto21 Networks of Centers of Excellence Program of Canada.

1. Murphy-Chutorian, E.; Trivedi, M.M. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **2009**, *31*, 607–626.
2. Fanelli, G.; Weise, T.; Gall, J.; Van Gool, L. Real time head pose estimation from consumer depth cameras. In *Pattern Recognition*; Springer, 2011; pp. 101–110.
3. Fanelli, G.; Dantone, M.; Gall, J.; Fossati, A.; Van Gool, L. Random Forests for Real Time 3D Face Analysis. *Int. J. Comput. Vision* **2013**, *101*, 437–458.
4. Breitenstein, M.D.; Kuettel, D.; Weise, T.; Van Gool, L.; Pfister, H. Real-time face pose estimation from single range images **2008**. pp. 1–8.
5. Fanelli, G.; Gall, J.; Van Gool, L. Real time head pose estimation with random regression forests **2011**. pp. 617–624.
6. Seemann, E.; Nickel, K.; Stiefelhagen, R. Head pose estimation using stereo vision for human-robot interaction **2004**. pp. 626–631.
7. Funes Mora, K.A.; Odobez, J. Gaze estimation from multimodal Kinect data **2012**. pp. 25–30.
8. Morency, L.P. 3D Constrained Local Model for Rigid and Non-rigid Facial Tracking. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE Computer Society: Washington, DC, USA, 2012; CVPR '12, pp. 2610–2617.
9. Rekik, A.; Ben-Hamadou, A.; Mahdi, W. 3D Face Pose Tracking using Low Quality Depth Cameras. VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications, 2013, Vol. 2.

10. Cai, Q.; Gallup, D.; Zhang, C.; Zhang, Z. 3D Deformable Face Tracking with a Commodity Depth Camera. Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III; Springer-Verlag: Berlin, Heidelberg, 2010; ECCV'10, pp. 229–242.
11. Hansen, D.; Ji, Q. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **2010**, *32*, 478–500.
12. Yu, Y.; Mora, K.A.F.; Odobez, J.M. Robust and Accurate 3D Head Pose Estimation through 3DMM and Online Head Model Reconstruction. 2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017), 2017, pp. 711–718. doi:10.1109/FG.2017.90.
13. Baltrusaitis, T.; Robinson, P.; Morency, L.P. 3D Constrained Local Model for rigid and non-rigid facial tracking. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2610–2617. doi:10.1109/CVPR.2012.6247980.
14. Papazov, C.; Marks, T.K.; Jones, M. Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4722–4730. doi:10.1109/CVPR.2015.7299104.
15. Tulyakov, S.; Vieriu, R.L.; Semeniuta, S.; Sebe, N. Robust Real-Time Extreme Head Pose Estimation. 2014 22nd International Conference on Pattern Recognition, 2014, pp. 2263–2268. doi:10.1109/ICPR.2014.393.
16. Funes-Mora, K.A.; Odobez, J.M. Gaze estimation in the 3d space using rgb-d sensors. *International Journal of Computer Vision* **2016**, *118*, 194–216.
17. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features **2001**. *1*, I–511.
18. Bouaziz, S.; Pauly, M. Dynamic 2d/3d registration for the kinect **2013**. p. 21.
19. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/node103.html.
20. Bouaziz, S.; Tagliasacchi, A.; Pauly, M. Sparse iterative closest point **2013**. *32*, 113–123.
21. Lu, F.; Sugano, Y.; Okabe, T.; Sato, Y. Inferring human gaze from appearance via adaptive linear regression **2011**. pp. 153–160.
22. Funes Mora, K.A.; Monay, F.; Odobez, J.M. EYEDIAP Database: Data Description and Gaze Tracking Evaluation Benchmarks. Idiap-RR Idiap-RR-08-2014, Idiap, 2014.
23. Pauly, M. Realtime Performance-Based Facial Avatars for Immersive Gameplay **2013**. pp. 23:23–23:28. doi:10.1145/2522628.2541252.
24. Martin, M.; v. d. Camp, F.; Stiefelhagen, R. Real Time Head Model Creation and Head Pose Estimation on Consumer Depth Cameras. 2014 2nd International Conference on 3D Vision, 2014, Vol. 1, pp. 641–648. doi:10.1109/3DV.2014.54.