3D Face Pose Tracking using Low Quality Depth Cameras

Ahmed Rekik[1], Achraf Ben-Hamadou[2] and Walid Mahdi[1]

[1]*Sfax University, Multimedia InfoRmation systems and Advanced Computing Laboratory (MIRACL),*
Pôle Technologique de Sfax, Route de Tunis Km 10, BP 242, 3021 Sfax, Tunisia
[2]*Paris-Est University, LIGM (UMR CNRS), Center for Visual Computing,*
Ecole des Ponts ParisTech, 6-8 Av. Blaise Pascal, 77455 Marne-la-Vallée, France

Keywords: 3D Face Tracking, RGB-D Cameras, Visibility Constraint, Photo-consistency, Particle Filter.

Abstract: This paper presents a new method for 3D face pose tracking in color image and depth data acquired by RGB-D (*i.e.,* color and depth) cameras (e.g., Microsoft Kinect, Canesta, *etc.*). The method is based on a particle filter formalism. Its main contribution lies in the combination of depth and image data to face the poor signal-to-noise ratio of low quality RGB-D cameras. Moreover, we consider a visibility constraint to handle partial occlusions of the face. We demonstrate the accuracy and the robustness of our method by performing a set of experiments on the Biwi Kinect head pose database.

1 INTRODUCTION

3D face pose tracking is becoming an important task for many research domains in computer vision like Human-Computer Interaction and face analysis (Weise et al., 2011; Cai et al., 2010; Maurel et al., 2008) and recognition (Kim et al., 2008). Indeed, these research fields have dramatically increased these very last years. This arises particularly from the ubiquity of vision systems in our day life (*i.e.,* webcams in laptops, smart-phones, *etc.*) and lately from the arrival of low-cost RGB-D cameras, such as Microsoft Kinect and Canesta. Such new cameras allow for synchronously capturing a color image and a depth map of the scene with a rate of about 30 acquisitions per second.These cameras provide a lower quality and much more noisy data that bulky 3D scanners. However, they are efficient in several domains like gesture recognition and video gaming. Kinect is a good example. Nowadays, many applications use Kinect-like cameras. For example, (Weise et al., 2011) try to customize avatars using Kinect data and (Ramey et al., 2011) use Kinect cameras to interact with machines (*e.g.,* robots and computers).

This paper aims at developing a new method for 3D face pose tracking in color and depth images acquired from Kinect like cameras using a particle filter formalism. Our method is robust to the poor signal-to-noise ratio of such cameras. The main idea is to combine depth and image data in the *observation model* of the particle filter. Moreover, we handle partial occlusions of the face by integrating a *visibility constraint* in the observation model.

This paper is organized as follows. In the next section, we present the previous work related to 3D face pose tracking using color images and/or depth maps. Then, we detail our tracking method in section 3. Finally, section 4 presents the experiments and the results obtained for the evaluation of our method.

2 RELATED WORK

Several research works have been proposed in the literature for face pose estimation and tracking. These can be broadly categorized into 2D image or depth data based approaches.

The first category gathers approaches that use 2D images to estimate the face pose. It refers to the Appearance and Feature based methods. While Appearance-based methods attempt to use holistic facial appearance (Morency et al., 2003), Feature-based methods rely on the localization of specific facial features and suppose that some of these are visible in all poses (Yang and Zhang, 2002; Matsumoto and Zelinsky, 2000). In general, these methods suffer from partial occlusions and are sensitive to the accuracy of feature detection methods.

Rekik A., Ben-Hamadou A. and Mahdi W..
3D Face Pose Tracking using Low Quality Depth Cameras.
DOI: 10.5220/0004220202230228
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 223-228
ISBN: 978-989-8565-48-8

Depth data based approach, however, rely only on depth data to estimate the 3D face pose. Weise *et al.* (Weise et al., 2011) use Iterative Closest Point (ICP) with point-plane constraints and a temporal filter to track the head pose in each frame. In (Fanelli et al., 2011), Fanelli *et al.* present a system for estimating the orientation of the head from depth data only. This approach is based on discriminative random regression forests, given their capability to handle large training datasets. Another approach is presented in (Breitenstein et al., 2008) where a shape signature is first used to identify the nose tip in depth images. Then, several face pose hypotheses (pinned to the localized nose tip) are generated and evaluated to choose the best pose among them. These methods are very sensitive to highly noisy depth data. Indeed, it is difficult to distinguish the face regions in highly noisy data.

Recently, (Cai et al., 2010) have used both depth and appearance cues for tracking the head pose including facial deformations. This idea behind the combination of 2D images and depth data is to overcome the poor signal-to-noise ratio of low-cost RGB-D cameras. Their approach relies on detecting and tracking facial features in 2D images. Assuming that the RGB-D camera is already calibrated, one can find the corresponding 3D coordinates of these detected features. Finally, a generic deformable 3D face is fitted to the obtained 3D points. Nonetheless, this method does not handle partial occlusions of the face. Moreover, like Feature based methods, it is sensitive to the accuracy of feature detection and tracking algorithms. In the same vein, Seemann *et al.* (Seemann et al., 2004) present a face pose estimation method based on a trained neural networks to compute the head pose from grayscale and disparity maps. Similar to the method proposed by (Cai et al., 2010), this method does not handle partial occlusions of the face.

This paper presents a new Appearance-Based method for 3D face pose tracking in sequences of image and depth data. To cope with noisy depth maps provided by the RGB-D cameras, we use both depth and image data in the observation model of the particle filter. Unlike (Cai et al., 2010), our method does not rely on tracking 2D features in the images to estimate the face pose. Instead, we have used the whole visible texture of the face. In this way, the method is less sensitive to the quality of the feature detection and tracking in images. Moreover, our method handles the case of facial partial occlusions by introducing a visibility constraint.

3 3D FACE POSE TRACKING METHOD

Our tracking method is based on the Particle filter formalism which is a Bayesian sequential importance sampling technique. It recursively approximates the posterior distribution using a set of N weighted *particles* (samples). In the case of 3D face pose tracking, particles stand for 3D pose hypotheses (*i.e.,* 3D position and orientation of the face in the scene). For a given frame t, we denote $X_t = \{\boldsymbol{x}_t^i\}_{i=1}^N$ the set of particles and $\boldsymbol{x}_t^i \in \mathbb{R}^6$ is the i-th generated particle that involves the 6 Degrees of Freedom (*i.e.,* 3 translations and 3 rotation angles) of a 3D rigid transformation.

The general framework of the particle filter is to throw an important number N of particles to populate the parameter space, each one representing a 3D face pose. The *observation model* allows for computing a weight w_t^i for each particle $\boldsymbol{x}_t^i$ according to the similarity of the particle to the *reference model* and using the observed data $\boldsymbol{y}_t$ in frame t (*i.e.,* color and depth images). Thus, the posterior distribution $p(\boldsymbol{x}_t^i \mid y_t)$ is approximated by the set of the weighted particles $\{\boldsymbol{x}_t^i, w_t^i\}$ with $i \in \{1, \ldots, N\}$.
The *transition model* allows for propagating particles between two consecutive frames. Indeed, at a current frame t, particles $\hat{x}_{t-1}$ are assumed to approximate the previous posterior. To approximate the current posterior, each particle is propagated using a transition model approximating the process function:

$$x_t = \hat{x}_{t-1} + u_t, \tag{1}$$

where u_t is a random vector having a normal distribution $\mathcal{N}(0, \Sigma)$ and the covariance matrix Σ is set experimentally according to the prior knowledge on the application. For instance, in Human-Computer Interaction, the head displacements between two consecutive frames are small and limited. Therefore, the values of Σ entries may be small as well.

In last section, the general framework of our method is presented. In the next section, the reference model and the observation model which are our main contributions of this paper will be detailed.

3.1 Reference Model

To create the reference model $\boldsymbol{M}_{ref}$, we use a modified version of the Candide 3D face model (see Figure 1(a)). The original Candide model (Ahlberg, 2001) is modified in order to remove the maximum of non-rigid part (*e.g.,* mouth and chin parts). We consider only parts that present the minimum of animation and expression (see Figure 1(b)). Our reference model is defined as a set of K ($K = 93$) vertices $\mathcal{V}$

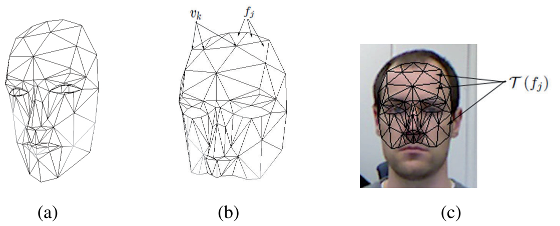


Figure 1: Reference model initialization. (a) Original Candide model; (b) Extracted rigid part of the Candide model; (c) Extracted texture of the reference 3D face model by projecting the the fitted face model onto the color image.

and a set of J ($J = 100$) facets $\mathcal{F}$. Each vertex $\boldsymbol{v}_k \in \mathcal{V}$ is a point in $\mathbb{R}^3$, and each facet $f_j \in \mathcal{F}$ is a triplet of vertices.

The tracking process needs to initialize the reference model. This initialization step consists of fitting the Candide deformable model to the user's face and extract its texture. Assuming that a neutral face is available in the first frame, we detect the face in the color image using the well-known Viola and Jones method (Viola et al., 2003), then, we apply a facial point detection algorithm (Valstar et al., 2010) on the detected face to provide L=22 landmark points l_r such as eye corners, the nose tip, *etc.*.Given the calibration data between the color and the depth sensors of the camera, we project each landmark point l_r to the depth map to compute its corresponding 3D point l'_r. Each of these landmark points has a corresponding vertex in the 3D face model. In a similar manner as (Lu and Jain, 2006), to align the deformable face model to the user's face, we minimize a cost function $\mathcal{C}_{init}$ which is the euclidean distance between the extracted 3D landmark points and their corresponding points in the 3D face model:

$$\mathcal{C}_{init} = \frac{1}{L}\sum_{r=1}^{L} \|\boldsymbol{v}_r - l'_r\|^2. \tag{2}$$

In equation (2), we abuse slightly the notation and denote $\boldsymbol{v}_r$ the vertex of the 3D face model that corresponds to the landmark point l'_r. This notation belongs only to this equation and will not be used in the rest of the article. We use BFGS quasi-Newton optimization method to solve the shape and pose parameters of the 3D face model by minimizing the cost function $\mathcal{C}_{init}$. Thereby, the shape of the reference model is adapted to the shape of the user's face. Afterwards, the texture $\mathcal{T}(f_j)$ of each facet in the reference model is obtained by projecting the face model on the color image (see Figure 1(c)). Thus, we end up with a reference model $\boldsymbol{M}_{ref}$ corresponding to the shape and the texture of the user's face.

3.2 Observation Model

The observation model allows the filter to evaluate the generated particles $\mathcal{X}_t$ according to the observed data $\boldsymbol{y}_t$ and the obtained reference model $\boldsymbol{M}_{ref}$. In other words, this evaluation allows for weighting each particle proportionally to the probability $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t^i)$ of the current measurement $\boldsymbol{y}_t$ given the particle $\boldsymbol{x}_t^i$. An appearance model A_t^i is first generated from each particle $\boldsymbol{x}_t^i$. The appearance model A_t^i consists of a set of K vertices $\mathcal{V}\left(\boldsymbol{x}_t^i\right)$ and a set of J facets $\mathcal{F}\left(\boldsymbol{x}_t^i\right)$. The coordinates of each vertex $\boldsymbol{v}_{k,t}^i \in \mathcal{V}\left(\boldsymbol{x}_t^i\right)$ are computed as follows:

$$\boldsymbol{v}_{k,t}^i = \mathbf{R}_t^i \boldsymbol{v}_k + \mathbf{t}_t^i, \tag{3}$$

where $\mathbf{R}_t^i$ and $\mathbf{t}_t^i$ are respectively the 3×3 rotation matrix and the translation vector generated in a standard way from the six parameters of the particle $\boldsymbol{x}_t^i$. The texture $\mathcal{T}(f_{j,t}^i)$ of each facet $f_{j,t}^i$ in the appearance model is defined as a set of pixels in the triangle given by the projection of $f_{j,t}^i$ in the color image. An annoying situation that occurs very often in face tracking is when the person's face is partially hidden by another object (*e.g.,* hand, *etc.*). Consequently, the texture $\mathcal{T}(f_{j,t}^i)$ may be affected by the texture of the object in the foreground, and as a results, the evaluation of the particle $\boldsymbol{x}_t^i$ becomes inaccurate. To avoid such situation, we introduce in our filter a *visibility constraint* which states that a given pixel $p_m \in \mathcal{T}(f_{j,t}^i)$ is considered invisible if an external object exists between the 3D face model (obtained by equation (3)) and the RGB-D camera. Lets q the corresponding 3D point of p_m in the depth map. We define a binary function $\delta(p_m)$ that returns 1 if p_m is visible, 0 otherwise:

$$\delta(p_m) = \begin{cases} 1, & if\ \ d < \varepsilon, \\ 0, & otherwise, \end{cases} \tag{4}$$

where d is the euclidean distance between q and its corresponding 3D point locate on the facet $f_{j,t}^i$. The pixel p_m is considered visible if the distance d is lower than a threshold[1] ε. Only pixels with $\delta(p_m)$ equals 1 are used in the evaluation of particles.

The particle evaluation of a given particle $\boldsymbol{x}_t^i$ depends on two energies. The first energy $E_{3D}\left(A_t^i, \mathcal{P}_t\right)$ measures the superimposition of the 3D face model (*i.e.,* corresponding to the particle $\boldsymbol{x}_t^i$) on the 3D point cloud[2] $\mathcal{P}_t$ acquired by the depth sensor of the RGB-D camera. The second energy is a photo-consistency energy denoted by $E_{ph}\left(A_t^i, \boldsymbol{M}_{ref}\right)$. It indicates the similarity between textures of the reference model $\boldsymbol{M}_{ref}$

[1] The threshold ε is fixed experimentally to 10 *mm* in our setup.

[2] Given the calibration data of the RGB-D camera, one can obtain the point cloud form the depth map acquired by the camera.

and the appearance model A_t^i. The combination of these two energies is given by:

$$w_t^i = \alpha\ exp\left(-E_{3D}\left(A_t^i,\mathcal{P}_t\right)\right) + (1-\alpha)\ exp\left(E_{ph}\left(A_t^i,\boldsymbol{M}_{ref}\right)\right), \quad (5)$$

where $\alpha \in [0,1]$ is weighting scalar. We remember that the weights w_t^i are used to select the best particle. In the next two sections, we define the 3D and photo-consistency energies.

3.2.1 3D Energy

The 3D energy indicates the closeness of an appearance model A_t^i (corresponding to the particle x_t^i to evaluate) to the point cloud $\mathcal{P}_t$ acquired at a time t and compares their shapes. Given the calibration data of the RGB-D camera, we can define a set of K corresponding points $\{(\boldsymbol{v}_{k,t}^i, \boldsymbol{p}_{k,t}^i)\}_{k=1}^K$ between the vertices $\mathcal{V}\left(\boldsymbol{x}_t^i\right)$ forming the appearance model A_t^i and the point cloud $\mathcal{P}_t$, where $\boldsymbol{p}_{k,t}^i \in \mathcal{P}_t$, $\boldsymbol{v}_{k,t}^i \in \mathcal{V}\left(\boldsymbol{x}_t^i\right)$, and $\boldsymbol{p}_{k,t}^i$ corresponds to the closest 3D point to $\boldsymbol{v}_{k,t}^i$ in the point cloud $\mathcal{P}_t$.

The more the appearance model A_t^i is close to the point cloud $\mathcal{P}_t$, the more the euclidean distance $d_1\left(A_t^i,\mathcal{P}_t\right)$ tends towards 0:

$$d_1\left(A_t^i,\mathcal{P}_t\right) = \frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{v}_{k,t}^i - \boldsymbol{p}_{k,t}^i\|^2. \quad (6)$$

Similar to (Cai et al., 2010), we use the point to plane distance as well. Let $\boldsymbol{n}_{k,t}^i$ be the surface normal of point $\boldsymbol{v}_{k,t}^i$. The point to plane distance reads:

$$d_2\left(A_t^i,\mathcal{P}_t\right) = \frac{1}{K}\sum_{k=1}^{K}\left(\left(\boldsymbol{n}_{k,t}^i\right)^T\left(\boldsymbol{v}_{k,t}^i - \boldsymbol{p}_{k,t}^i\right)\right)^2. \quad (7)$$

These two distance measures defined in equations (6) and (7) are combined following equation (8) to give the final formula of the 3D energy $E_{3D}\left(A_t^i,\mathcal{P}_t\right)$.

$$E_{3D}\left(A_t^i,\mathcal{P}_t\right) = \frac{1}{2}\left(d_1\left(A_t^i,\mathcal{P}_t\right) + d_2\left(A_t^i,\mathcal{P}_t\right)\right) \quad (8)$$

3.2.2 Photo-consistency Energy

The photo-consistency energy $E_{ph}\left(A_t^i,\boldsymbol{M}_{ref}\right)$ is defined as the normalized cross-correlation between the texture of the reference model $\boldsymbol{M}_{ref}$ and the texture of the appearance model A_t^i. The photo-consistency energy is computed as the average of the normalized cross-correlation between each two corresponding facet texture $\mathcal{T}\left(f_j\right)$ and $\mathcal{T}(f_{j,t}^i)$:

$$E_{ph}\left(A_t^i,\boldsymbol{M}_{ref}\right) = \frac{1}{J}\sum_{j=1}^{J} NCC\left(\mathcal{T}(f_j),\mathcal{T}(f_{j,t}^i)\right). \quad (9)$$

We employ a barycentric warping scheme to find pixel correspondences between the texture of facets $\mathcal{T}(f_j)$ and $\mathcal{T}(f_{j,t}^i)$. This is needed to compute the normalized cross-correlation.

4 EXPERIMENTS AND RESULTS

This section details the experiments performed to evaluate our face pose tracking method. We first assess the interplay between the 3D and the photo-consistency energies. This first assessment allows us to demonstrate the importance of each of these energies and by the way to tune the weighting parameter α of equation (5). Then, we evaluate the accuracy of the 3D pose estimation using the Biwi Kinect Head pose database (Fanelli et al., 2011) which is provided with ground truth data. Finally, we demonstrate the importance of the visibility constraint in our 3D face tracking method.

Before detailing the evaluation, we will start by introducing the Biwi database and the parameters of our tracking method used during the evaluation. Then, we will present the experiments and the obtained results.

Biwi Kinect Head Pose Database. (Cai et al., 2010) evaluate the accuracy of their tracker by considering 2D errors only. It is basically the mean Euclidean distance between manually annotated reference points (*e.g.,* eye corners, *etc.*) in 2D images and their corresponding ones estimated by the tracker and back-projected on the images. We believe that this process of manual annotation lacks precision and repeatability, which makes the evaluation unreliable for 3D pose estimation. We rather choose to perform the evaluation of our method mainly with the Biwi Kinect Head Pose Database (Fanelli et al., 2011) which contains 24 sequences of 24 different persons. In each sequence, a person rotates and translates his face in different orientations. For each frame in the sequences, depth and color images are provided as well as ground truth face poses (3 translations in *mm* and 3 rotation angles in degree).

The evaluation of our tracking method using the Biwi database is done as follows. For a given sequence from the database, we apply our tracking method. Then, we compare the obtained 3D face poses to the ground truth ones. We define a position error (*i.e.,* distance between our face positions and the ground truth ones) and three rotation errors which are the difference between the obtained angles (*i.e.,* yaw, pitch, and roll) and the ground truth angles.

Parameters of the Tracking Method. Our tracking method is mainly designed for Human-Computer Interaction applications. Generally in these applications, the head displacements between two consecutive frames are limited. As a result, we experimentally set Σ entries to small values (*i.e.,* 4 *mm* for translations and 5° for rotations). To populate the parameter space, the number of particles is set to 200. Moreover, as it will be described later on, the weighting parameter α is experimentally fixed to 0.8.

4.1 Photo-consistency vs 3D

To show the interplay between the two energies involved in our tracker, we apply for a same sequence three configuration of our method, namely, using depth data only ($\alpha = 1$), using 2D images only (*i.e.,* $\alpha = 0$), and using both depth and image data (*i.e.,* $\alpha = 0.8$). In Figure2, we show the variation of the position and orientation errors with different values of the weighting parameter α. We can see that the optimal combination of the 3D and photo-consistency energies is given by $\alpha = 0.8$.

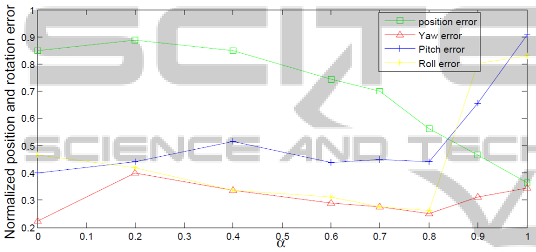


Figure 2: Variation of the tracking errors with different values of α. Here all errors are normalized between 0 and 1 so they can be shown in a same scale. The tracking method generates the best results with α equals 0.8. We can notice that the position error continues to decrease even after $\alpha = 0.8$. However, the face pose is considered wrong because of the rotation errors.

4.2 Accuracy Evaluation

We apply our tracking method on each sequence of the Biwi database. Then, we compare the obtained 3D face poses to the ground truth ones by considering the position and orientation errors. The mean and the standard deviation of the obtained errors are summarized in Table 1.

Table 1: Mean and standard deviation of the position and angle errors obtained for all sequences of the Biwi database.

	mean error	standard deviation
Position error	5.1 *mm*	±8 *mm*
Yaw error	5.13°	±3.33°
Pitch error	4.32°	±2.65°
Roll error	5.24°	±3.33°

These quantitative results show the accuracy of our face pose tracking method. In comparison to (Fanelli et al., 2011), we have better results. Indeed, Fanelli *et al.* have 14.7 *mm* for the position mean error and 9.2°, 8.5°, and 8° as mean errors respectively on yaw, pitch, and roll angles.

4.3 Visibility Constraint Importance

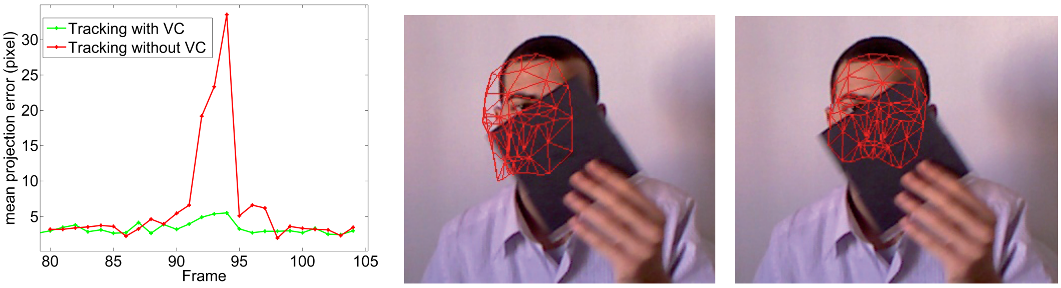
In addition to these evaluations, we demonstrate the robustness of our method against partial occlusions. The Biwi database doesn't include this kind of situation. Thus, we acquire our own sequences in which a person intentionally passes an object or his hand to make a partial occlusion of his face. Then, we apply our tracking method twice: first with the visibility constraint, second without visibility constraint. Finally, the evaluation is performed as follows. We have manually labelled visible control points around the eyes and the nose in every frame. Then, we computed the average of the Euclidean distance between the labelled control points and the 2D projection of their corresponding in the obtained 3D face model. Figure 3 shows the results of the tracking for a test data sample. The evolution of these error measures along the sequence is shown in Figure3(a). Even if this evaluation is done using 2D errors, we notice that the visibility constraint dramatically improves the quality of out tracker in case of partial occlusions. Figures 3(b) and 3(c) show the visual difference between the estimated poses, respectively, with and without using the visibility constraint.

5 CONCLUSIONS

This paper presents a new approach for 3D face pose tracking using color and depth data from low-quality RGB-D cameras. Our approach is based on the particle filter formalism. The particle evaluation model is based on combining image and 3D data.

We have performed a quantitative evaluation of the proposed method on the Biwi Kinect Head Pose database, and we have demonstrated the importance of the interplay between the 3D and photo-consistency energy computed to evaluate particles. Future work, will extend our tracker to handle action and expression deformation of the face.

Moreover, we intend to perform a GPU implementation of our method to evaluate simultaneously all generated particles in each frame. Indeed, the actual implementation of our method requires about 1 second to estimate the face 3D pose for a new frame. The GPU implementation can make the method faster and we can rich a real time processing. The GPU implementation allows also to consider a larger number of particles and to deal with more severe head displacements.



(a) (b) (c)

Figure 3: Tracking results in case of partial occlusions (occlusions occurred between frames 85 and 100). (a) Evolution of the error along the sequence. (b) Visual result of the face pose estimation without visibility (VC) constraint for frame 94. (c) Visual result of the face pose estimation using the visibility constraint for same frame. We notice a significant improvement of the face pose estimation when the visibility constraint is used.

REFERENCES

Ahlberg, J. (2001). Candide-3 - an updated parameterised face. Technical report.

Breitenstein, M. D., Küttel, D., Weise, T., Gool, L. J. V., and Pfister, H. (2008). Real-time face pose estimation from single range images. In *"Proceedings of IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8.

Cai, Q., Gallup, D., Zhang, C., and Zhang, Z. (2010). 3d deformable face tracking with a commodity depth camera. In *European Conference on Computer Vision*, pages 229–242.

Fanelli, G., Weise, T., Gall, J., and Gool, L. V. (2011). Real time head pose estimation from consumer depth cameras. In *Proceedings of the 33rd international conference on Pattern recognition*, pages 101–110.

Kim, M., Kumar, S., Pavlovic, V., and Rowley, H. (2008). Face tracking and recognition with visual constraints in real-world videos. In *"Proceedings of IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8.

Lu, X. and Jain, A. K. (2006). Deformation modeling for robust 3d face matching. In *Proceedings of IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1377–1383.

Matsumoto, Y. and Zelinsky, A. (2000). An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In *International Conference on Automatic Face and Gesture Recognition*, pages 499–505.

Maurel, P., McGonigal, A., Keriven, R., and Chauvel, P. (2008). 3D model fitting for facial expression analysis under uncontrolled imaging conditions. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4.

Morency, L.-P., Sundberg, P., and Darrell, T. (2003). Pose estimation using 3d view-based eigenspaces. In *In Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 45–52.

Ramey, A., González-Pacheco, V., and Salichs, M. A. (2011). Integration of a low-cost rgb-d sensor in a social robot for gesture recognition. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 229–230.

Seemann, E., Nickel, K., and Stiefelhagen, R. (2004). Head pose estimation using stereo vision for human-robot interaction. In *International Conference on Automatic Face and Gesture Recognition*, pages 626 – 631.

Valstar, M. F., Martinez, B., Binefa, X., and Pantic, M. (2010). Facial point detection using boosted regression and graph models. In *Proceedings of IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 2729–2736.

Viola, M., Jones, M. J., and Viola, P. (2003). Fast multi-view face detection. In *Proc. of Computer Vision and Pattern Recognition*.

Weise, T., Bouaziz, S., Li, H., and Pauly, M. (2011). Real-time performance-based facial animation. *ACM SIGGRAPH 2011*, 30(4):77:1–77:10.

Yang, R. and Zhang, Z. (2002). Model-based head pose tracking with stereovision. In *Automatic Face and Gesture Recognition*, pages 255–260.