Head pose estimation on depth data based on Particle Swarm Optimization

Pashalis Padeleris, Xenophon Zabulis and Antonis A. Argyros
Institute of Computer Science - FORTH
Heraklion, Crete, Greece
{padeler,zabulis,argyros}@ics.forth.gr

Abstract

We propose a method for human head pose estimation based on images acquired by a depth camera. During an initialization phase, a reference depth image of a human subject is obtained. At run time, the method searches the 6-dimensional pose space to find a pose from which the head appears identical to the reference view. This search is formulated as an optimization problem whose objective function quantifies the discrepancy of the depth measurements between the hypothesized views to the reference view. The method is demonstrated in several data sets including ones with known ground truth and comparatively evaluated with respect to state of the art methods. The obtained experimental results show that the proposed method outperforms existing methods in accuracy and tolerance to occlusions. Additionally, compared to the state of the art, it handles head pose estimation in a wider range of head poses.

1. Introduction

Head pose estimation is a special problem of human posture recognition. The ability to solve it accurately and robustly is of particular interest, because the head pose of a person conveys important information on its behavior and intentions. In this work, we investigate how head pose estimation can be performed based on 3D structure information provided by depth cameras. Head pose estimation can benefit from color information. The importance of color information is acknowledged and its fusion with depth information is left for future work.

The proposed approach is outlined in Fig. 1. The top right image shows a reference range image of a human subject in a frontal posture, obtained by a depth camera at initialization. During operation, a surface model $\mathcal{M}$ is reconstructed from the image D of the same camera. Range images of the model are rendered from candidate viewpoints, upon a hypothetical surface patch $\mathcal{W}$ (middle-right). These images may be incomplete due to sensor noise and occlusions. To estimate head pose, the proposed method searches for a view at which the rendered image matches the reference (bottom-right). This registration is cast as an optimization problem that is solved through Particle Swarm Optimization (PSO) [9]. To increase the computational performance of the method, the rendering of range images is performed on the GPU.

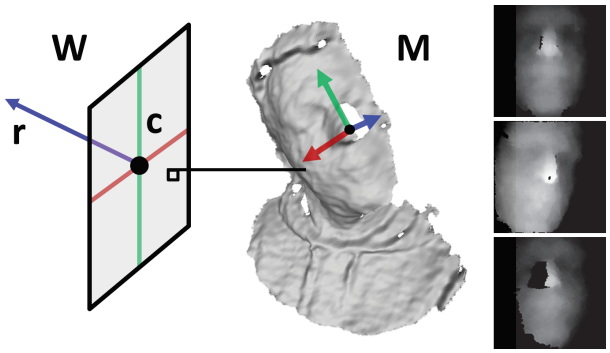

Figure 1. Method overview. The proposed method renders range images of the head, $\mathcal{M}$, as reconstructed by a depth camera, from different views and evaluates their similarity to a frontal reference range view T (top-right) acquired at pose $\mathcal{P}_0$. These images are formed upon a hypothetical surface patch $\mathcal{W}$, centered at $\vec{c}$ with normal $\vec{r}$. The image corresponding to the current pose of $\mathcal{W}$ is the middle-right image. The pose optimizing the above similarity yields the estimation result and is shown superimposed on $\mathcal{M}$; its corresponding range image is shown on the bottom-right.

The remainder of this paper is organized as follows. In Sec. 2 related work is reviewed. In Sec. 3, the proposed head pose estimation method is presented in detail. In Sec. 4, experiments which evaluate the accuracy, performance and usability of the approach are presented. Finally, Sec. 5 summarizes the main conclusions of this work.

2. Related work

A review of appearance based head pose estimation methods can be found in [5]. The improvement in performance obtained by the addition of 3D information as obtained by depth cameras or stereo has been multiply cited in

the literature. This section is focused on these methods.

Several methods that utilize 3D information to achieve head pose estimation are based on recognition of rigid landmark formations on the face. In [28], matching of landmarks in a stereo pair enables their triangulation. In [29], a user's face is robustly tracked based on stereo-reconstructed facial feature matching. The employed features are manually identified at initialization and semantically annotated through a model. In [20], multiple features are tracked from multiple cameras. In [12], pose is estimated through nasal ridge and facial symmetry detection, assuming unoccluded facial views. The method in [4] uses a color and a depth camera to robustly track the 6D head pose rather that obtain its absolute estimate, based on deformable matching of facial features in the color image. The corresponding 3D locations of these matches in the depth image are utilized in tracking, but as intensity information is originally employed the method is prone to illumination artifacts. Such methods exhibit reduced accuracy if features are occluded, which is often the case at oblique views.

Several methods are based on nosetip detection, as this is a size-dominant feature visible from a wide range of viewpoints. In [24], a coarse estimate is provided by fitting a plane to the 3D points around the nosetip and an ellipse to the facial contour. The offline method presented in [19] employs a spherical representation around the nosetip to estimate head pose. In [3], a set of precomputed reference range views of a synthetic 3D face model, centered at detected candidate nosetips, are compared with the facial depth map and the best match yields head pose. Precision is proportional to the number of reference views. The proposed work includes a similar operation, but does not rely on the accuracy of nosetip detection and does not discretize the space of head poses.

Another class of methods use a global representation of the head and do not rely on the result of a few landmark detections. In [10], a coarse estimate is obtained based on the partial derivatives of the depth map at the region of the (detected) head. In [2], color and depth information are fused in an iterative method to obtain head pose relative to to the previous frame. However, the method is limited to mainly frontal poses where a face can be detected in the color image. In [21], a neural network classifies color image pairs and the corresponding, stereo-acquired depth-maps into pose estimations. In [13], consecutive passive-stereo scans of the face are registered using ICP to a 3D face model obtained through active-stereo, but the method lacks accuracy during facial expressions and oblique angles, where the face is partially reconstructed. The method in [30] uses a face detector upon the textured visual hull of a person. Illumination changes in the scene are expected to influence the visual hull estimation and, thus, the acccuracy of head pose estimation.

The methods in [7, 8] are the most efficient and accurate representatives of this class of methods. They estimate head pose through random forests based classification. As such, they require extended training. Because of their accuracy, these methods are used as the reference in a comparative evaluation with the method proposed in this paper. This evaluation demonstrates that the proposed method exhibits increased accuracy and robustness.

The *proposed method* utilizes all available head related information of a depth image instead of relying on a few landmarks. As opposed to other global methods, it does not require training or semantic information, but only a simple initialization process that is typically restricted to the first frame of a sequence. Unlike [3], which also compares reference range views with the sensed image, it does not resort to exhaustive search nor does it rely on landmarks to determine such views. Instead, it uses a continuous search within the pose domain which is also optimized to conserve computational time. Furthermore, to the best of our knowledge, this is the first work to explore the application of evolutionary optimization techniques to the problem of human head pose estimation. Most importantly, the proposed approach proves to be more accurate and tolerant to sensor noise and occlusions while, at the same time, operates at a wider range of head poses compared to state of the art methods.

3. Head pose estimation

The proposed algorithm renders auxiliary range images of the reconstructed head from candidate poses and attempts to find the most similar to a single reference view that is obtained at initialization. An objective function is optimized (minimized) to find the 6 pose parameters that render the most similar view. This function is the cumulative depth discrepancy between corresponding pixels in the candidate and the reference views.

Typically, the reference view is selected to be frontal due to the richness of the facial structure. Nevertheless, there is no inherent limitation in selecting any other view as the reference one.

3.1. Rendering head pose hypotheses

Given a candidate head pose $\mathcal{P} = \{R, \vec{c}\}$, head range image ι is formed as follows (see Fig. 1). Let $\mathcal{W}$ be a hypothetical rectangular patch upon the XY-plane, centered at the origin. Let also its intrinsic axes $\vec{e}_x$, $\vec{e}_y$ be aligned with xx' and yy'. Transformation $R\vec{x} + \vec{c}$, where $\vec{x}$ a point upon $\mathcal{W}$, brings $\mathcal{W}$ in a relative pose to the reconstruction of the head. The orthogonal projection of the surface $\mathcal{M}$ upon $\mathcal{W}$ forms ι, where each pixel reads the distance of $\mathcal{W}$ from the surface point that is projected upon it. The resolution of ι is determined by the parameterization of hypothetical points upon $\mathcal{W}$; each such point $\vec{x}$ corresponds to a pixel $\vec{p}$ in ι.

Surface $\mathcal{M}$ is represented as a mesh of triangles which are established through the neighborhood relationships of pixels in D. Image ι is formed by projecting the triangles of $\mathcal{M}$ upon $\mathcal{W}$. During this projection, the ids of the triangles that project upon each pixel are collected. The value of ι at $\vec{p}$ is the distance of $\vec{x}$ to the point inside the triangle corresponding to $\vec{p}$. This point is found as the intersection of the triangle with the line that passes through pixel $\vec{x}$ and is oriented as the normal of $\mathcal{W}$, $\vec{r} = R \cdot [0\,0\,1]^T$. Since multiple triangles may project upon $\vec{p}$, the closest one is selected so that ι respects visibility constraints. The above rendering is parallelized in the GPU using the framework in [11]. In all experiments, the dimensions of $\mathcal{W}$ were $160\,mm \times 160\,mm$ and a $1\,mm$ parameterization of $\vec{x}$ yielded 160×160 pixel views.

At initialization, the user is prompted to face frontally and leveled with the ground plane, towards a predefined direction in space. By convention, this is the reference pose $\mathcal{P}_0 = \{R_0, \vec{c}_0\}$, where $\vec{c}_0 = [0\,0\,0]^T$ and R_0 is the 3×3 identity matrix. The head is detected in D and its center is considered as $\vec{c}_0$. This center is not any special landmark point, but used only as a reference for pose transformations. Head detection is achieved through face detection [26] applied to the RGB image of our RGBD sensor. In the absence of intensity/color information, depth-based approaches to head detection can be adopted, such as those presented in [18, 22], or [8].

Rendering the range view from $\mathcal{P}_0$ yields reference image T. The method's result is encoded as a rotation and translation relative to $\mathcal{P}_0$.

3.2. Evaluating head pose hypotheses

The objective function o is defined for a candidate pose $\mathcal{P}_k = \{R_k, \vec{c}_k\}$ as the dissimilarity of the corresponding range image ι_k to T, where both images are rendered in the same resolution (k enumerates candidate poses). As both T and ι_k may exhibit pixels with null depth measurements, a mask image S of the same dimensions is used, in which a pixel is set to 1 if the corresponding pixels in T and ι_k are both valid and to 0 otherwise. The dissimilarity is quantified as the Sum of Squared Differences (SSD) of pairwise pixel differences for pixels indicated as valid by S, normalized for their number:

$$o(\mathcal{P}_k) = \frac{\sum_i \sum_j S(i,j) \cdot [\iota_k(i,j) - T(i,j)]^2}{\sum_i \sum_j S(i,j)}. \quad (1)$$

Mask S filters out points where depth observations or hypotheses are missing, i.e. due to lack of visibility, noise, facial expressions, etc. In addition, at very oblique poses or during severe occlusions a small portion of the subject's head is reconstructed and subsequently most of ι_k's pixels are invalid. We choose not to trust the output of the objective function when more than the $1/3$ of ι_k is invalid.

To explore the behavior of function o regarding continuity and local minima, we have exhaustively computed it. this investigation demonstrated that the objective function is continuous in the vicinity of the global maximum, but exhibits several local minima further away from it. As a result, optimization methods that assume a smooth, continuous and unimodal objective function are expected to fail.

Henceforth, the pose estimate at frame t is noted $\mathcal{P}_t$, the corresponding value of o as $s_t = o(\mathcal{P}_t)$, and its corresponding range image as ι_t.

3.3. Particle Swarm Optimization

The optimization (i.e., minimization) of the objective function (Eq.(1)) has been performed based on Particle Swarm Optimization (PSO) [9] which has been demonstrated to be a very effective and efficient method for solving other vision optimization problems such as background modeling parameters estimation [25] and hand articulation tracking [14, 15, 16, 17]. PSO is an evolutionary algorithm that achieves optimization based on the collective behavior of a set of particles that evolve in runs called generations. The rules that govern the behavior of particles emulate "social interaction". A population of particles is essentially a set of points in the parameter space of the objective function to be optimized.

Canonical PSO, the simplest of PSO variants, has several attractive properties. More specifically, it only depends on very few parameters, does not assume knowledge of the derivatives of the objective function and requires a relatively low number of objective function evaluations [1].

Every particle holds its current position (current candidate solution, set of parameters) in a vector x_t and its current velocity in a vector v_t. Moreover, each particle i stores in vector p_i the position at which it achieved, up to the current generation t, the best value of the objective function. Finally, the swarm as a whole, stores in vector p_g the best position encountered across all particles of the swarm. p_g is broadcasted to the entire swarm, so that every particle is aware of the global optimum. The update equations in every generation t to re-estimate each particle's velocity and position are

$$v_t = K(v_{t-1} + c_1 r_1 (p_i - x_{t-1}) + c_2 r_2 (p_g - x_{t-1})) \quad (2)$$

and

$$x_t = x_{t-1} + v_t, \quad (3)$$

where K is a constant *constriction factor* [6]. In Eqs. (2), c_1 is called the *cognitive component*, c_2 is termed the *social component* and r_1, r_2 are random samples of a uniform distribution in the range $[0..1]$. Finally, $c_1 + c_2 > 4$ must hold [6]. In all performed experiments the values $c_1 = 2.8$, $c_2 = 1.3$ and $K = \frac{2}{\left|2 - \psi - \sqrt{\psi^2 - 4\psi}\right|}$ with $\psi = c_1 + c_2$ were used.

Method	Location (mm)	Yaw (°)	Pitch (°)	Roll (°)	Accuracy (%)
[8]	14.50 (22.10)	9.10 (13.60)	8.50 (9.90)	8.00 (8.30)	79.0
Our run on [8]	5.21 (2.77)	2.38 (1.80)	2.97 (2.16)	2.75 (2.09)	78.7
This work DE	2.76 (1.79)	1.08 (1.04)	1.26 (1.11)	1.72 (1.69)	88.6
This work PSO	2.78 (1.24)	1.00 (1.05)	1.14 (1.09)	1.60 (1.69)	91.4

Table 1. Head pose accuracy comparison using the [8] dataset. Table shows mean error (and standard deviation) of head pose errors and the percentage of successful detections, for a given angle accuracy threshold (see text).

In our problem formulation, the rotation component of candidate poses is parameterized in terms of yaw (θ), pitch (ϕ), and roll (ω) angles, correspondingly yielding $R = R_x(\theta) \cdot R_y(\phi) \cdot R_z(\omega)$ for each parameter combination. Translation is parameterized by the XYZ coordinates of the face center $\vec{c}$. Particles are initialized at a normal distribution around the center of the search range with their velocities set to zero. Each dimension of the multidimensional parameter space is bounded in some range. If, during the position update, a velocity component forces the particle to move to a point outside the bounded search space, this component is zeroed and the particle does not perform any move at the corresponding dimension. This is the only constraint employed on velocities. In case that the head pose needs to be continuously tracked in a sequence instead of being estimated in a single frame, temporal continuity can be exploited. More specifically, the solution over frame t is used to restrict the search space for the initial population at frame $t+1$. In related head pose tracking experiments, the search range (or the domain throughout which particle positions were initialized) extended $\pm 15\,mm$ and $\pm 10°$ around the estimate of the previous frame.

3.4. Detecting and treating head pose estimation failures

Pose estimation failures are detected by considering the score of the objective function after optimization. More specifically, if the score of the objective function at frame t is below threshold τ_s then the head pose estimation is considered inaccurate. This may be due to an erroneous estimation or even because the person is absent from the scene. In such cases, the last valid pose is used to determine the center of the search range for all subsequent frames until a valid pose estimate is computed again. If the tracking is lost for more than a number n of frames, then the bootstrapping procedure is attempted until a head is detected. In all experiments, $n = 10$ and $\tau_s = 0.15$.

4. Experiments

The experimental evaluation of the proposed method was based on a prototype implementation running on a conventional PC equipped with *NVidia GeForce GTX 580* $1.56\,GHz$ GPU. All reported experiments considered 25 particles running in 40 generations. Under this configuration, the implemented method runs at $10\,fps$. As each particle is independent, we parallelize its respective computation in the GPU.

The methods in [24, 10, 2, 13, 19] do not provide a metric evaluation of their accuracy and, thus, we cannot quantitatively compare with them. From those, the method reported in [19] seems to be the most accurate and robust one, but operates offline as tens of seconds are required to process a frame.

The comparative evaluation of our method has been performed with respect to the methods reported in [8, 7, 3] as they are more recent and report state of the art accuracy. More specifically, we compare the proposed method to [8] as that method is shown to be more accurate than [7] and [3]. In the experiments we employed the implementation of [8] that is provided by the authors. Table 1, Table 2 and Fig. 3 report data from the aforementioned papers.

4.1. Ground truth experiments

In order to evaluate the proposed method experimentally, we employed two publicly available datasets annotated with ground truth. Both exhibit large variability in terms of facial expressions and person appearances (persons wearing glasses, hats, a variety of hairstyles etc.). The first dataset [8] contains more than $15 \cdot 10^3$ images of 20 persons and was obtained with the *Kinect* sensor. Head poses range in $\pm 75°$ yaw and $\pm 60°$ pitch, but only cover mild roll rotations of $\pm 20°$. The second dataset [3] is acquired using the depth sensor presented in [27]. It contains more than 10^3 range images of 20 persons. Head poses cover about $\pm 90°$ yaw and $\pm 45°$ pitch rotations. Roll rotations were not performed in this dataset. The two datasets differ significantly with respect to the noise in the depth measurements. More specifically, the first dataset contains more significant noise than the second one which was obtained by a higher quality depth sensor.

In the experiments, the first frame of each sequence was used to construct the reference head pose. As in the evaluation of both [8] and [3], we, also, consider a head pose estimate to be a "miss" when the L2 norm of 3 pose angles is greater than $\tau_a = 10°$ and the distance error is greater than $10\,mm$.

Table 1 compares the accuracy of the proposed method

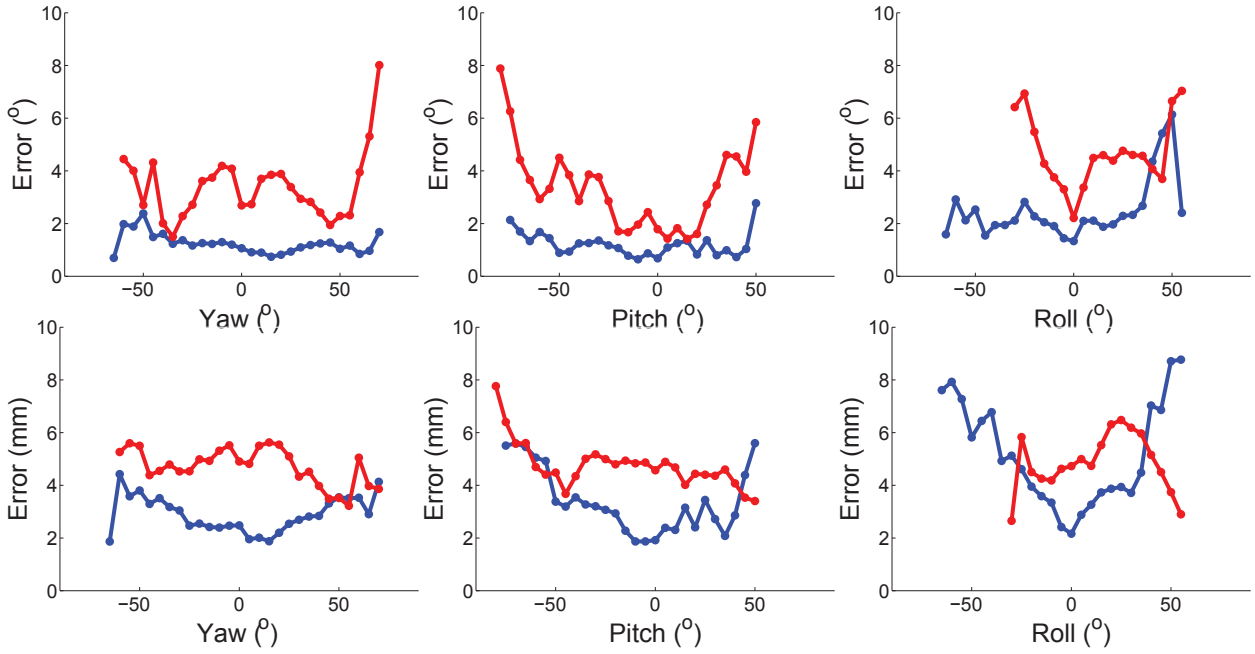

Figure 2. Comparison of estimation error as a function of actual head pose obliqueness, for this method (blue) and [8] (red).

with that of [8]. The 2^{nd} column measures errors in head localization, while columns from 3 to 5 measure errors on the three rotational components of pose. In the table, the 6^{th} column is the percentage of successful estimations for these values of τ_a. In the 1^{st} row, the results from the respective work are copied. The method in [8] uses a 5-fold cross-validation to avoid evaluation biases, as the training data was produced from the same dataset that the evaluation was run upon. In our experiments, we used the whole dataset of [8], without knowing which were the training and which were the test images. As we wish to compare the two methods in more detail (see below), we executed this method on the whole dataset albeit that, in this way, the method is favored. Naturally, the results (shown in the 2^{nd} row) are more accurate than those reported in [8]. Even so, the proposed approach performed better in all measures. The results of this test are reported in the 4^{th} row of Table 1 and are also plotted in Figs. 2 and 3.

The last two rows show the results obtained by two different evolutionary optimization methods. The last row reports results obtained by PSO (Sec. 3.3). In the penultimate row we use the same overall framework but use the Differential Evolution (DE) algorithm [23] as the optimization engine. DE run for the same number of particles and generations as PSO. As it can be verified, PSO slightly outperforms DE. Still, PSO is the preferred optimization technique because, unlike DE, it has a genuinely data parallel computational structure that permits a very efficient GPU implementation.

Table 2 compares the proposed method with the methods in [7, 3] in the same way as above in the dataset made available by [3]. The findings of this comparison also show that the proposed method provides better estimates of head pose compared to [7, 3].

Another significant aspect of a head pose estimation algorithm is the range of head poses that can be reliably estimated. In Fig. 2, the error of estimates is plot as a function of actual (ground truth) head obliqueness, in terms of yaw, pitch, and roll. Both angular and distance errors are provided. We observe that the proposed work retains rotational errors at a low level (i.e. $< 3°$) for a wider range of poses, with the exception of head location during steep roll rotations. However, these poses exhibited higher errors in head rotation and contained yaw and pitch components, for which the proposed method exhibits higher overall accuracy. Furthermore, the results plotted in this graph for [8] were obtained by using the training set as the test set, too.

In Fig. 3, the success ratio of the proposed work is plot[1] as a function of the values of thresholds τ_h, τ_a that determine whether a pose estimate was a hit or a miss, according to its disparity from the corresponding ground truth data. Thus, the graphs plot the predicted success rate as a function of the requested estimation accuracy. We observe that the proposed method systematically outperforms [8]. The only exception is for very large errors ($> 22\,mm$) in head localization, where the estimation is, anyway, overly inaccurate. Still, in this case, the rotational component of pose is more accurate for our method. During system operation, we con-

[1]This ratio is called "accuracy" in the vertical axes, for comparison with previous works. We refer to the same quantity as success ratio.

Method	Location (mm)	Yaw (°)	Pitch (°)	Accuracy (%)
[7]	13.40 (21.10)	5.70 (15.20)	5.10 (4.90)	90.4
[3]	9.00 (14.00)	6.10 (10.30)	4.20 (3.90)	80.8
This work PSO	7.05 (6.46)	1.62 (1.59)	2.05 (1.87)	90.1

Table 2. Head pose accuracy comparison using the [3] dataset. Table shows mean error (and standard deviation) of head pose errors and the percentage successful detections, for a given angle accuracy threshold (see text).

sider such cases as a missed detection as they typically correspond to an output of the objective function greater that τ_s.

4.2. Qualitative evaluation

The qualitative evaluation of the behavior of the proposed approach in sequences of depth images has been based on the datasets described above, but also in further challenging datasets that have been acquired at our laboratory. In Fig. 4 indicative results are shown, superimposed on the color image of the sensor. It is noted that RGB data are not used in pose estimation. Frames (i) to (v) and (x) in this figure were acquired in our laboratory, while the rest are taken from the dataset in [8].

For the data collected at our laboratory, we used the depth camera of a *Kinect* sensor. Figure details contain the reference view T (right) and the aligned range image ι_t (left), for the proposed method. Besides the superimposition of the results, the accuracy of the proposed method can be judged by considering the level of alignment between T and ι_t.

The frames presented in Fig. 4 show the performance of the proposed method in characteristic situations. In frame (i), a subject makes a facial expression by opening his mouth, producing a hole in ι_t. In frames (ii) to (v) the methods are tested in the presence of occlusions. We observed the proposed method to retain its accuracy for faces occluded up to $1/3$ of the reference view. In frame (iv) we test the methods for the case where, other than the nose, head or hand parts appear intensely protruding in the depth map. In such cases, methods (i.e. [24, 19, 3]) that assume that the nosetip is the most protruding head region often fail. In addition, in frames (v), (vi), and $(viii)$ we test the methods for increasing facial variability induced by eyeglasses and hats that were not worn by the subject during the acquisition of the reference pose T. In frames (vi) to $(viii)$, we test the methods for very oblique poses that avail small pixel support due to self-occlusions of the subject's face. In frame (ix), we compare estimates at a steep roll pose. We have noticed that [8] tends to fail in cases of steep roll, perhaps because is was trained on a dataset with only a few such samples. The proposed approach is invariant to this effect.

In frames (x) to (xii), we show cases of failure, where our method provides an inaccurate pose estimate. In frame

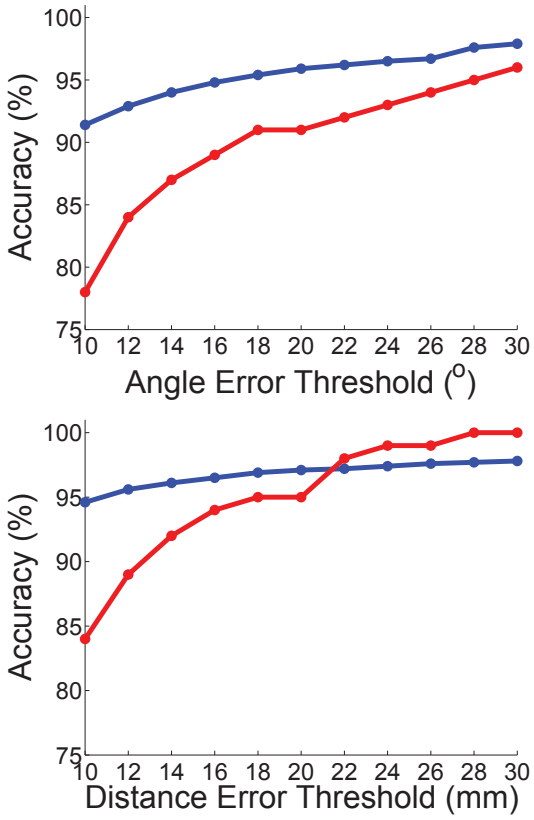

Figure 3. Success ratio as a function of required accuracy for this work (blue) and [8] (red) in the [8] dataset.

(x), this is due to the combined effect of obliqueness and increased distance, while, in frames (xi) and (xii), due to the extremely oblique head pose of the subject. In (x) the method in [8] does not yield a result, but in (xii) the latter method is more accurate. Finally, both methods are inaccurate for frame (xii).

The results of the proposed method on the employed datasets are also presented in the video accompanying this paper submission.

5. Summary

We proposed a method that estimates the head pose based on depth data. The head pose estimation problem has been formulated as an optimization problem that has been

Figure 4. Qualitative evaluation. Superimposed are the results of the proposed method (blue) and [8] (red) for characteristic situations (see text).

solved based on Particle Swarm Optimization. The quantitative evaluation of the proposed method against standard datasets annotated with ground truth demonstrates that the proposed method outperforms state of the art methods in terms of robustness and accuracy and that it operates successfully in a wider range of poses than pertinent methods. The experimental results also demonstrate that this claim is valid for depth data provided by different depth sensors and

different noise characteristics. The proposed method has also been shown to cope well with occlusions and in variability in facial size/distance, subject expressions, and view obliqueness. We attribute the increased accuracy to the fact that the full depth information is utilized in pose estimation. This is in contrast to methods that achieve head pose estimation through the registration of a few facial landmarks.

Currently, our method operates in $10\,fps$. Although GPU processing is used, our implementation results in several calls to the GPU. Thus, significant time proportion is spent at GPU communication and thread initialization operations. A careful GPU exploitation is expected to improve considerably the computational performance of the method, making it a good candidate for head pose estimation in applications where real time operation is a critical issue.

Acknowledgments

This work was partially supported by the IST-FP7-288917 project DALi, the IST-FP7-288146 project HOBBIT, and by the FORTH-ICS internal RTD Programme "Ambient Intelligence and Smart Environments".

References

[1] P. Angeline. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. *Evolutionary Programming VII, LNCS*, 1447:601–610, 1998. 3

[2] A. Bleiweiss and M. Werman. Robust head pose estimation by fusing time-of- flight depth and color, 2010. 2, 4

[3] M. Breitenstein, D. Kuttel, T. Weise, L. V. Gool, and H. Pfister. Real-time face pose estimation from single range images. In *CVPR*, pages 1–8, 2008. 2, 4, 5, 6

[4] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3D deformable face tracking with a commodity depth camera. In *ECCV*, pages 229–242, 2010. 2

[5] E. Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *PAMI*, 31(4):607–626, 2009. 1

[6] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002. 3

[7] G. Fanelli, J. Gall, and L. V. Gool. Real time head pose estimation with random regression forests. In *CVPR*, pages 617–624, 2011. 2, 4, 5, 6

[8] G. Fanelli, T. Weise, J. Gall, and L. V. Gool. Real time head pose estimation from consumer depth cameras. In *DAGM*, 2011. 2, 3, 4, 5, 6, 7

[9] J. Kennedy, R. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, 2001. 1, 3

[10] F. Kondori, S. Yousefi, H. Li, S. Sonning, and S. Sonning. 3D head pose estimation using the kinect. In *WCSP*, pages 1–4, 2011. 2, 4

[11] N. Kyriazis, I. Oikonomidis, and A. A. Argyros. A gpu-powered computational framework for efficient 3d model-based vision. Technical Report TR420, ICS-FORTH, Jul. 2011. 3

[12] S. Malassiotis and M. Strintzis. Robust real-time 3D head pose estimation from range data. *Pattern Recognition*, 38:1153–1165, 2005. 2

[13] R. Niese, A. Al-hamadi, and B. Michaelis. A novel method for 3D face detection and normalization. *Journal of Multimedia*, pages 1–12, 2007. 2, 4

[14] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Markerless and Efficient 26-DOF Hand Pose Recovery. In *ACCV*, pages 744–757. Springer, 2010. 3

[15] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient Model-based 3D Tracking of Hand Articulations using Kinect. In *BMVC*, Dundee, UK, Aug. 2011. 3

[16] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full DOF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints. In *ICCV*, pages 2088–2095. IEEE, Nov. 2011. 3

[17] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*. IEEE, June 2012. 3

[18] C. Papazov and D. Burschka. Stochastic global optimization for robust point set registration. *Computer Vision and Image Understanding*, 115(12):1598–1609, 2011. 3

[19] N. Pears, T. Heseltine, and M. Romero. From 3D point clouds to pose-normalised depth maps. *IJCV*, 89(2-3):152–176, 2010. 2, 4, 6

[20] R. Ruddarraju, A. Haro, and I. Essa. Fast multiple camera head pose tracking. In *Vision Interface*, 2003. 2

[21] E. Seemann, K. Nickel, and R. Stiefelhagen. Head pose estimation using stereo vision for human-robot interaction. In *FGR*, pages 626–631, 2004. 2

[22] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011. 3

[23] R. Storn and K. Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, 1997. 5

[24] Y. Tu, C. Zeng, C. Yeh, S. Huang, T. Cheng, and M. Ouhyoung. Real-time head pose estimation using depth map for avatar control. *CVGIP*, 2011. 2, 4, 6

[25] K. Tzevanidis and A. Argyros. Unsupervised learning of background modeling parameters in multicamera systems. *CVIU*, 115(1):105 – 116, 2011. 3

[26] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 3

[27] T. Weise, B. Leibe, and L. V. Gool. Fast 3d scanning with automatic motion compensation. In *CVPR*, 2007. 4

[28] M. Xu and T. Akatsuka. Detecting head pose from stereo image sequence for active face recognition. In *FGR*, pages 82–87, 1998. 2

[29] R. Yang and Z. Zhang. Model-based head pose tracking with stereovision. In *FGR*, pages 255–260, 2002. 2

[30] X. Zabulis, T. Sarmis, and A. A. Argyros. 3D head pose estimation from multiple distant views. In *BMVC*, 2009. 2