REAL-TIME HEAD AND HAND TRACKING BASED ON 2.5D DATA

Xavier Suau *Josep R. Casas* *Javier Ruiz-Hidalgo*

Universitat Politècnica de Catalunya (UPC)
Email: {xavier.suau, josep.ramon.casas, j.ruiz}@upc.edu

ABSTRACT

A novel real-time algorithm for head and hand tracking is proposed in this paper. This approach is based on 2.5D data from a range camera, which is exploited to resolve ambiguities and overlaps. Experimental results show high robustness against partial occlusions and fast movements. The estimated positions are fairly stable, allowing the extraction of accurate trajectories which may be used for gesture classification purposes.

***Index Terms*—** head tracking, hand tracking, time-of-flight camera, gesture recognition, body tracking

1. INTRODUCTION

Gesture recognition technologies are being widely applied to countless applications. One may notice a global tendency to replace traditional control devices with device-less vision-based solutions. In a first step towards this goal, tactile interfaces are already in the market and allow the suppression of traditional input devices such as keypads, mouse, etc. Indeed, the mid-term objective is to obtain device-less, but also marker-less, gesture recognition systems which allow users to interact as naturally as possible, providing a truly immersive experience.

Tracking of body parts is a key aspect for many recognition systems. For example, being able to track the head and hands of a person, may help navigating through menus in a TV set. Indeed, a visual feed-back may be provided to the user showing how gestures are being interpreted by the system. Furthermore, body tracking provides spatio-temporal information, such as trajectories of points of interest or joint angles of a body model. Such features may be exploited in a gesture classification step, delivering a more robust recognition than classification based on raw image features.

Classically, vision-based tracking solutions are separated in 2D and 3D based. Proposals based on 2D images can hardly track complex movements involving motion in the three axes. On the other hand, 3D based systems offer a very precise tracking, but costly and non-portable setups are required, as the SmartRoom presented by Neumann *et al.* [1].

Recently, new families of sensors have appeared as suitable trade-off solutions between 2D and 3D approaches. As an example, cameras which are based on the time-of-flight (TOF) principle, as explained by Kolb *et al.* [2], are able to deliver a depth estimation of the recorded scene. Since such information contains depth, but restricted to a given viewpoint, we call it 2.5D information. Thus, 2.5D is a simplified 3D (x, y, z) surface representation that contains at most one depth value (z direction) for every point in the (x, y) plane. The main advantage of TOF cameras is that complex multi-view setups are no longer needed to obtain depth information.

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 248138. This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación, under project TEC2010-18094.

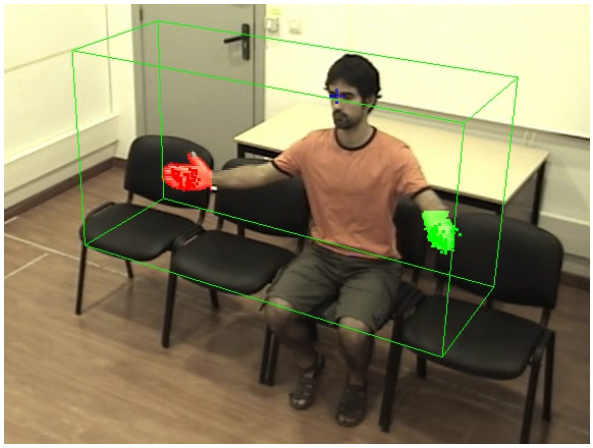

Fig. 1. Snapshot of the proposed head+hand tracking system output. In this sequence, the head (blue cross) and both hands (green and red blobs) are being tracked. Movement is restricted to the hand workspace (green box), which is attached to the estimated head position. Results are presented on a lateral view for visualization purposes.

In the literature, Lehment *et al.* [3] propose a model-based annealing particle filter approach on data coming from a TOF camera. Bevilacqua *et al.* [4] track multiple persons in a crowded scene with a TOF zenital camera. Bleiweiss and Werman [5] propose to combine color and 2.5D depth to improve image segmentation, which is consequently used for tracking.

Focusing on head tracking, Haker *et al.* [6] compute principal curvatures on 2.5D data as features to estimate the position of the head. Bohme *et al.* [7] improve Haker's proposal. Nichau and Blanz [8] present an algorithm to detect the tip of the nose in 2.5D data by comparing the extracted silhouette to an average head profile template. Malassiotis *et al.* [9] estimate head pose by fitting an ellipse to the 2.5D data and detecting the nose tip, which helps determining head orientation.

In this paper we propose a novel approach based on 2.5D data (an SR4000 TOF camera is used), which concatenates head tracking and hand tracking (see a final result in Figure 1). Head position is estimated depending on the distance between the user and the camera. Such estimation is robust against partial occlusions and fast movements, and helps in defining a region where hands are likely to be found. Hands are detected and tracked in such region.

The remaining of this paper explains the main parts of the tracking algorithm, summarized in the block diagram in Figure 2. The

978-1-61284-350-6/11/$26.00 ©2011 IEEE

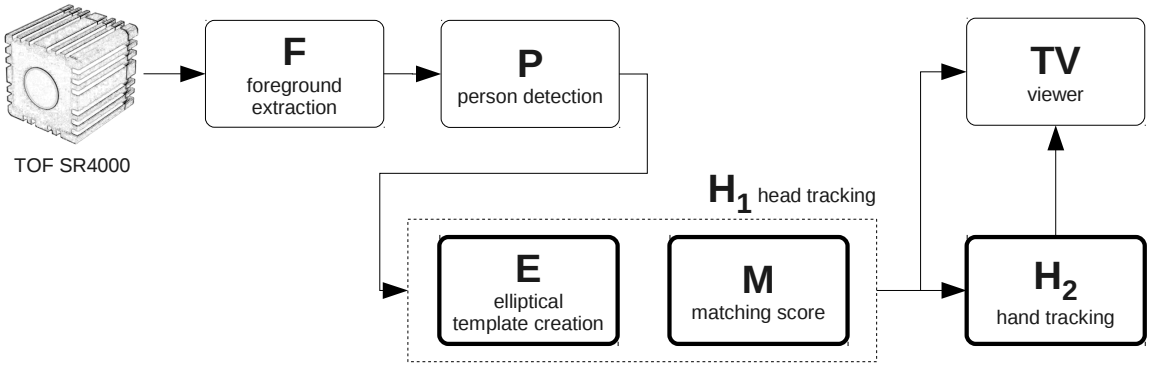

Fig. 2. Synthetized block diagram of the proposed head+hand tracking system, from the capture with the SR4000 TOF camera to the final visualization on a TV set.

head tracking algorithm is explained in Section 3, followed by the hand tracking proposal in Section 4. Experimantal results on accuracy and speed are shown in Section 5. Conclusions are drawn in the final Section, as well as some future ideas to improve the proposed approach.

2. PRELIMINARY STEPS

Before starting the head+hand tracking strictly speaking, some preliminary processing is required (blocks **F** and **P** in Figure 2).

2.1. (F) Foreground extraction

A foreground mask $\mathcal{F}$ is extracted from the 2.5D depth estimation images. In this paper, a simple and fast $\mathcal{F}$ extractor is used, modelling each pixel's value with a gaussian distribution [10]. The results may be appreciated in Figures 3 (third column), 4 and 7.

2.2. (P) Person detection

Some morphological operators are applied to $\mathcal{F}$ in order to detect when a person enters or exits the scene. More precisely, a morphological open-close is applied to $\mathcal{F}$, obtaining $\tilde{\mathcal{F}} = (\mathcal{F} \circ B) \bullet B$. Consequently, the extracted foreground mask $\mathcal{F}$ results filtered in $\tilde{\mathcal{F}}$ with a certain size threshold. If one or more areas remain in $\tilde{\mathcal{F}}$, the largest one is selected as a person. The centroid of this area is computed and used for tracking initialization, as explained in Section 3.

3. HEAD TRACKING

Many persons at different distances from the camera may coincide in the same scene. Such persons may enter, exit and freely move around the camera field-of-view. Therefore, some aspects have to be taken into account when undertaking the head tracking problem.

Occlusions Partial and total occlusions are a common problem of (single viewpoint) visual analysis systems, since moving objects may overlap.

Apparent head size User's heads may be placed at different depth levels, resulting in different head sizes when projected onto the image plane.

In order to overcome such problems, we propose to firstly estimate the size of the head on the image **(E)**, then estimate its position **(M)**, which corresponds to the head tracking step $\mathbf{H_1}$ in Figure 2.

3.1. (E) Head size estimation

Depth-invariant elliptical shape of heads is exploited thanks to depth estimations from the TOF camera. Heads may present many different sizes on the recorded images depending on the depth level where they are placed. Nevertheless, most of people's heads are likely to have an elliptical shape, no matter the distance they are away from the camera.

Indeed, we assume that people will stand-up or be seated, but not laying down or other poses. Therefore, the depth of the whole body is likely to be similar to the depth of the head. By applying the morphological operators explained in Section 2.2 to the foreground mask of the image, we are able to detect the number of persons in the scene, as well as their centroids C_i. Thus, we assume that head's depth $d_{H_i} \approx d_{C_i}$.

An elliptical mask of the size of a regular adult head (about 17×23 cm) is placed at the calculated d_{H_i} depth level and projected onto the camera image plane, obtaining an ellipse of the apparent head size (H_x, H_y) in pixels (see Figure 4). Such ellipse, which is called *template* or $(\mathcal{E})$ is then used to find a head position estimate.

3.2. (M) Head position estimation

With the aim of finding the image zone which better matches the elliptical shape, a matching score between the template ellipse $(\mathcal{E})$ and the global foreground mask $(\mathcal{F})$ is calculated at every pixel position (m, n) of the image. Such matching is performed within a rectangular search zone (see Section 3.3), by shifting the template ellipse across the image plane. The matching score is calculated according to conditions C^k presented in (1), where $b = background$ and $w = foreground$. Conditions are checked at every pixel position $(u, v) \in \mathcal{E}$ of the template, which is itself centered at (m, n). When a condition C^k is satisfied, $C^k = 1$, otherwise $C^k = 0$. The final matching score for the pixel (m, n) is calculated as the sum of all the scores obtained on the template pixels, as shown in Equation (1).

$$
\begin{array}{lllllll}
C^1_{u,v}: & (\mathcal{E}_{u,v} = b) & \wedge & (\mathcal{F}_{u,v} = b) & & \\
C^2_{u,v}: & (\mathcal{E}_{u,v} = w) & \wedge & (\mathcal{F}_{u,v} = w) & \wedge & (|d_{u,v} - d_{H_i}| < d_{max}) \\
C^3_{u,v}: & (\mathcal{E}_{u,v} = b) & \wedge & (\mathcal{F}_{u,v} = w) & \wedge & (|d_{u,v} - d_{H_i}| > d_{max})
\end{array}
$$

$$
\mathcal{M}_{m,n} = \sum_{\forall (u,v) \in \mathcal{E}} (C^1_{u,v} + C^2_{u,v} + C^3_{u,v}) \qquad (1)
$$

The pixel (m, n) with a higher score $\mathcal{M}^H = \max\{\mathcal{M}_{m,n}\}$ is selected, by simple max-pulling, as the best head position estimation

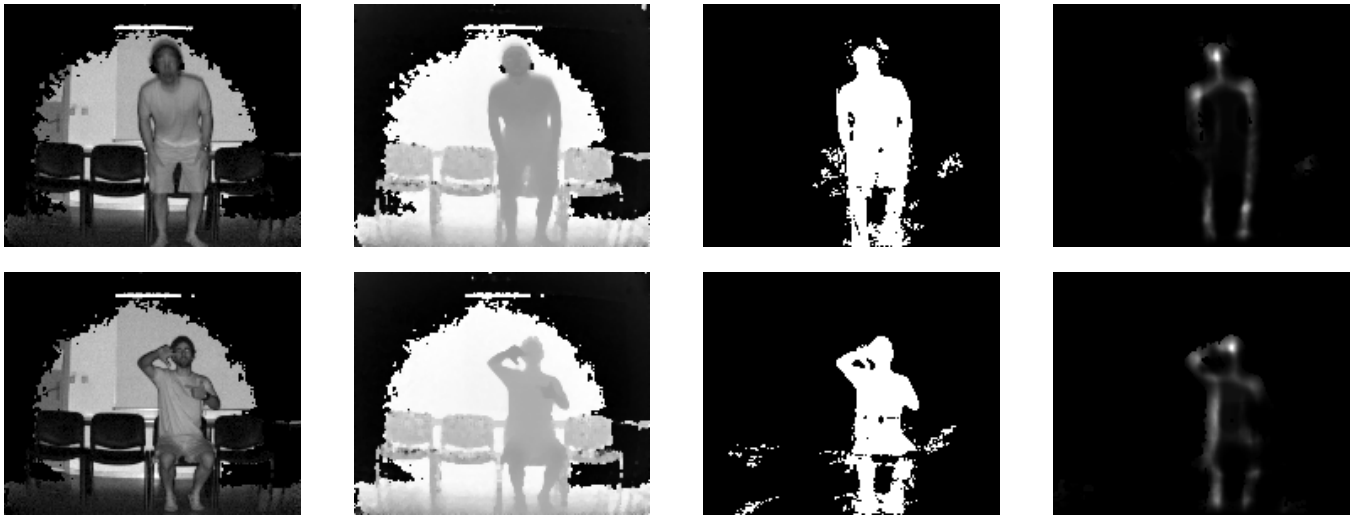

Fig. 3. Head position estimation in two video frames (each row). From left column to right: IR amplitude, TOF 2.5D depth estimation, raw foreground mask and the obtained head matching score. The whitest zone is chosen as the most likely head position.

$\hat{p}_H$ in a given search zone. Two examples of the proposed solution are presented in Figure 3.

Conditions C^2 and C^3 provide robustness against partial occlusions. Indeed, the elliptical shape of the head would be very polluted by occlusions, since we are working on a foreground mask $\mathcal{F}$ which does not take depth into account. By means of conditions C^2 and C^3, depth is incorporated to the matching score, not taking into account those pixels which are not consistent with the calculated head depth d_{H_i}. Remark that a depth threshold d_{max} is used to decide whether a depth value is consistent or not. The way d_{max} is defined and updated is explained in Section 3.3.

Furthermore, such approach is robust against horizontal head rotation and slight lateral head tilt, since the elliptical shape is still recognizable.

3.3. Search zone resizing

In order to increase the robustness of the presented head estimation algorithm we propose to resize the search zone where the matching score $\mathcal{M}_{m,n}$ is calculated.

The position and size of such search zone is adapted to the head position variance, and also to the confidence on the estimation. More precisely, the new search zone size is calculated as a function of last matching score $\mathcal{M}^H$, the head size estimation (H_x, H_y), and the spatial variance of the estimation σ. The latter is calculated over the previous L frames, obtaining σ_x and σ_y for each axis. As for the matching score, it is normalized by the best achievable score for the current template $\mathcal{M}^{max}$ such that $\bar{\mathcal{M}} = \frac{\mathcal{M}^H}{\mathcal{M}^{max}} \in [0, 1]$.

The new rectangular search zone is centered at the last head position estimation, while the rectangle sides R_x and R_y are resized according to Equation (2).

$$\begin{aligned} R_x &= \sigma_x + (1+\mu)\cdot H_x \\ R_y &= \sigma_y + (1+\mu)\cdot H_y \end{aligned} \qquad with \qquad \mu = e^{\frac{-\bar{\mathcal{M}}}{1-\bar{\mathcal{M}}}} \tag{2}$$

Such resizing is effective against fast head movements, since the search zone will be adapted to the variance of the estimations. For example, horizontal movements will enlarge the search zone along the horizontal axis, as shown in Figure 7.(c).

Furthermore, including the matching score in Equation (2) makes the system robust against bad estimations, making the search zone slightly larger in case a better match appears close to the previous zone.

Note that when the head estimation is stable and confident, the search zone is about the size of the head (H_x, H_y).

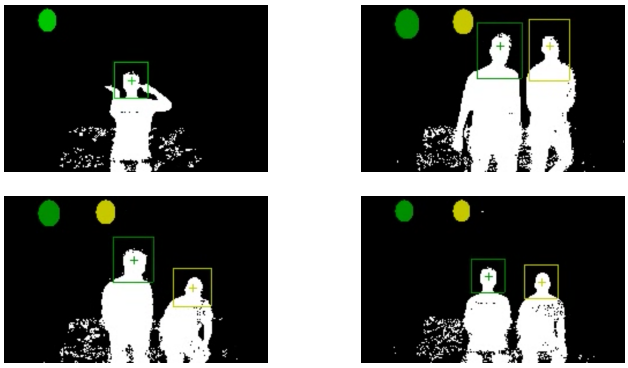

Fig. 4. Head tracking snapshots from our experiments . Head position is estimated by shape matching with an ellipse which is continuously resized depending on the distance between the camera and the person. Ellipses being currently used are presented at the upper-left corner of the image. The rectangles in the image correspond to the current search zones.

4. HAND TRACKING

Hands are probably one of the most difficult parts of the body to track, but also one of the most important targets for many applications. Gesture recognition is tightly related to hand tracking, most of the information being obtained from hand movement. An accurate

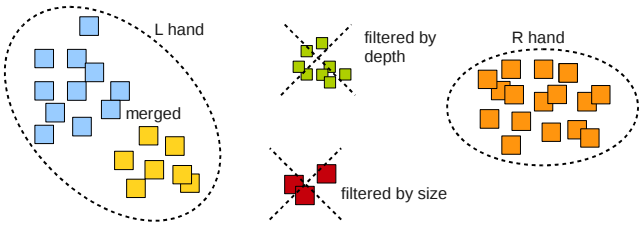

Fig. 5. Example of cluster merging and filtering for hand detection. Color represents candidate clusters obtained with a kd-tree structure. The red cluster is too small. The blue and yellow clusters are merged since the Hausdorff distance between them is small enough. Three clusters remain, but the green one is filtered since it is placed further in depth (represented with smaller squares). Thus, the remaining two clusters are labeled as R and L hands.

and robust hand tracking system is desirable to face complex gesture recognition.

The hand tracking system proposed in this section (block **H₂** in Figure 2) relies on the robust head estimation presented in Section 3. Hands are supposed to be *active* (performing gestures) in a zone placed in front of the body. Following this basic assumption, a hand workspace Ω is defined as a 3D box of size $140 \times 70 \times 60\, cm$, hands being supposed to lay within it when moving. Ω is attached to the head position $\hat{p}_H$ so that Ω follows the user's head at every time instant, as shown in Figure 1.

Hands are to be detected among the 2.5D points in Ω. Dense clusters are searched by means of a kd-tree structure, which allows fast neighbor queries among 3D point clouds [11, 12]. A list of candidate clusters is obtained and filtered according to the following sequential criteria:

Merging Two clusters are merged as a single cluster if the Hausdorff distance between them is smaller than a given distance threshold δ_{min}.

Size filtering The resulting merged clusters are filtered by size (number of points in cluster), keeping the largest ones. A size threshold s_{min} is set to determine what clusters are accepted as hand candidates.

Depth filtering Clusters that fulfill the previous criteria are sorted by depth. Those clusters placed closer to the camera are selected, knowing that a maximum of two clusters may be chosen.

Thresholds δ_{min} and s_{min} should be tuned depending on the type of camera and scene. A graphical illustration of these criteria is shown in Figure 5. The number of detected hands depends on the number of clusters that pass the merging and filtering steps, resulting in two, one or no hands being detected in Ω. For example, two hands are being detected in Figure 1 .

In addition, spatio-temporal coherence is included in the hand tracking scheme, which increases its robustness. New hand estimations are compared to the previous ones by means of Hausdorff distance measurements, in order to provide coherence to tracking and avoid right-left hand shifting. Furthermore, when only one hand is being detected, it is labeled (right / left) depending on the previous estimations and its relative position in Ω. For example, a cluster placed further right in Ω probably corresponds to the right hand.

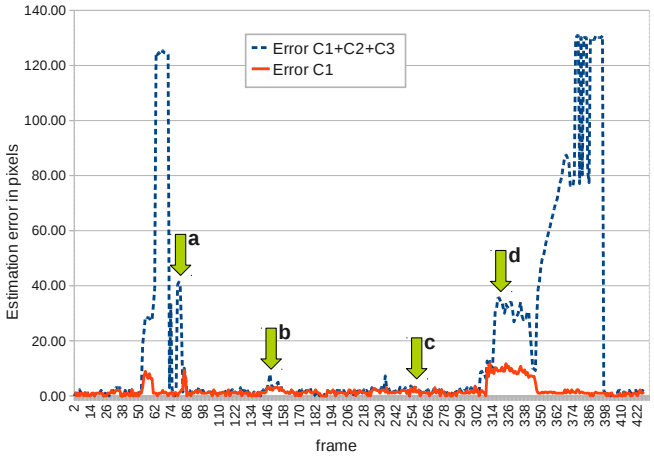

Fig. 6. Error between the obtained head estimations and the ground-truth. $C^1 + C^2 + C^3$ error does not go above 10 pixels, which is about the head radius. The C^1 version loses target twice, a reset of the algorithm being needed (frames 74 and 398). The labeled arrows in Figure 6 correspond to the frames shown in Figure 7.

5. EXPERIMENTAL RESULTS

5.1. Head tracking results

Given the recent interest in 2.5D cameras and the variety of environments and scenes where they may be used, there does not exist a common dataset to which compare our results.

Therefore, in order to evaluate the performance of the proposed head tracking, we analyze the convenience of conditions C^2 and C^3 in Equation (1), and how robustness is increased by exploiting 2.5D data. We compare the proposed algorithm with a reduced version which only verifies condition C^1. This way, the contribution of C^2 and C^3 is shown. The estimation error between a ground-truth (marked by hand) head position g^H and the estimated head position is calculated as $\varepsilon = |g^H - \hat{p}^H|$, and presented in Figure 6 for the two versions of the algorithm. Note that, even if C^1 plays the role of 2D method, 2.5D data is used for the initial head size estimation. Both methods run on the same foreground extraction, obtained from the 2.5D depth images as explained in Section 2.1.

Figure 6 shows the estimation error of a sequence which contains two persons and three main events. Around frame 50, a single person waves hands before his head, hiding it. At frame 135, he performs a gesture with both hands which partially cover the head zone. At frame 300 the second person enters the scene and walks behind the first person. These events are illustrated in Figure 7.

The proposed $C^1 + C^2 + C^3$ algorithm is able to track the first user's head despite these three polluting events, while the C^1 versions loses the target when the shape of the head is changed (hands moving, persons walking, etc.).

5.2. Head+Hands tracking results

As stated in Section 4, hand tracking depends on the robustness of head tracking. A set of gestures have been considered to analyze the performance of hand tracking, including gestures with one or two hands such as waving, pointing or separating hands apart (see Table 1). Marked hand positions have been extracted by hand from the 2.5D images. Thanks to the depth estimation, these 3D ground-truth

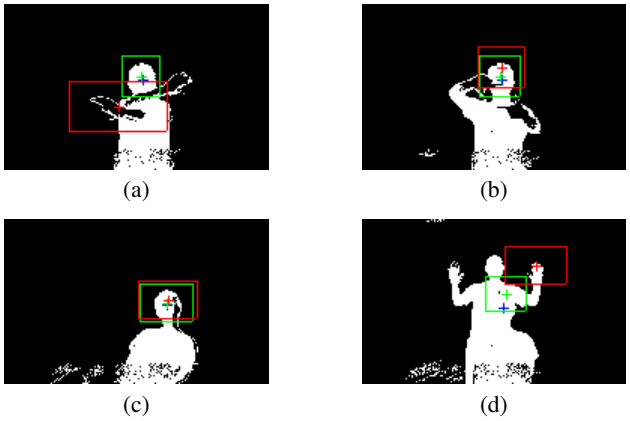

Fig. 7. Head tracking snapshots of the sequence presented in Figure 6. In red, the C^1 version, in green the $C^1 + C^2 + C^3$ and in blue the ground-truth position. Note how the $C^1 + C^2 + C^3$ is more robust in these adverse situations.

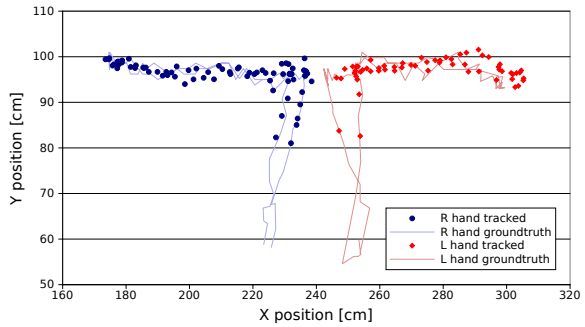

Fig. 8. Ground-truth and estimated trajectories of the (R)ight and (L)eft hands. The estimated hand positions are fairly close to the reference ground-truth positions. Only the XY projection of these 3D trajectories is shown.

trajectories may be extracted from the 2D marked points. Such 3D trajectories are used for comparison hereafter.

In Figure 8 the marked and estimated trajectories of both hands are presented corresponding to a gesture where each hand moves horizontally, like a slow hand clap (only the frontal projection of these 3D trajectories is shown for visual clarity). The error between the estimated and ground-truth trajectories is shown in Figure 9.

Hands are detected from the moment that they enter Ω (frame 7). During the first 10 frames, both hands are touching each other, misleading the tracker which only *sees* one hand. However, once they are separated for a few cm, the tracker is able to detect and track both hands. It should be emphasized that both ground-truth and estimated trajectories are polluted by noise from the depth estimation. In addition, ground-truth trajectories have been extracted by hand, selecting a reasonable hand center which may vary in some cm along frames. Despite these adverse noisy conditions, the hand estimation error rarely goes above $10\,cm$.

Table 1 summarizes the average error for different gestures, some of them performed with one hand. The average error is higher for fast movements such as the *circle* and the *up-down* gestures, resulting in about $6\,cm$ of error. For the other gestures, the error is of about $3\,cm$, which is fairly adequate given the size of a hand.

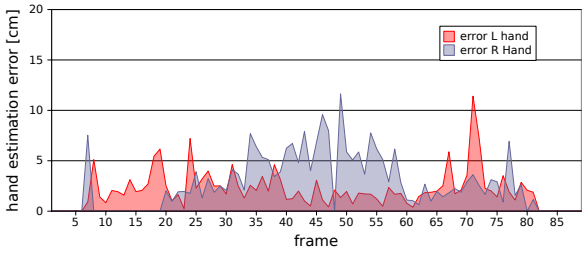

Fig. 9. Hand estimation error, calculated as the Euclidean distance between the estimated and ground-truth 3D positions. The maximum error is about $10\,cm$ for this sequence.

Gesture	# frames	error R hand	error L hand
push	30	2.62 cm	-
circle	30	6.61 cm	-
roll	35	2.86 cm	-
hand up-down	115	5.87 cm	-
separate hands	75	2.36 cm	3.80 cm

Table 1. Hand estimation error for different one-handed and two-handed gestures, computed as the average error during the performance of the gestures. For example, the errors for *separate hands* gesture are calculated from the errors in Figure 9.

5.3. Execution speed

The main applications of the proposed head+hand tracking algorithm require real-time processing. In order to evaluate the real-time performance of our proposal, execution speed experiments have been carried out on a Intel Xeon 3GHz CPU. Two magnitudes are studied:

- T^P Processing time composed of the head + hand tracking algorithms stand-alone.
- T^O Overall processing time including 2.5D data reading from disk. Such data, which has been captured with the TOF camera, consists of a QCIF 16-bit depth image, a QCIF foreground mask and a QCIF$\times$3 64-bit matrix containing the real-world 3D positions delivered by the camera software.

Experiments are run in sequences where the head and both hands are detected, which is the worst case in terms of computational load. After several experiments with different two-handed gestures we obtain a processing frame rate $f^P = \frac{1}{T^P} \approx 68\,fps$, which is fast enough for real-time applications. As for the overall time including reading from disk, an overall frame rate $f^O = \frac{1}{T^O} \approx 25\,fps$ is obtained. These frame rates very encouraging and open the door to further improvements and refinements of the algorithm, since a high percentage of CPU resources is not currently used.

When dividing the overall processing time into tasks, it may be noticed that the head tracking task takes 34.7% of the total while hand tracking is much faster (2.65%) and reading from disk occupies a 62.65% of CPU time (Figure 10. It may be taken into account that the reading task does not affect CPU when the system works live, with the camera delivering 2.5D information at about $20 - 30\,fps$.

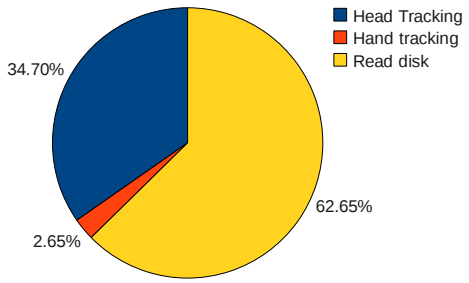

Fig. 10. Percentage of CPU occupancy for the main three processing tasks.

6. CONCLUSION

Real-time head and hands tracking is a crucial issue for many gesture recognition systems. In this paper we have proposed a fast algorithm for head and hands tracking based on 2.5D data coming from a range camera.

The proposed algorithm performs head tracking in a first step. With this aim, an elliptical head template is adapted to the depth level where the person is placed. Such template is projected onto the camera image plane where a matching score is calculated, in order to find the best head position estimation by simple max-pulling. Tracking robustness is increased with an adaptive resizing of the search zone where the matching score is computed.

The hand tracking algorithm fully relies on the head position estimation. A box-shaped 3D zone where hands are likely to be placed is attached to this head estimation, allowing the user to freely move in the scene. Hands are detected and tracked as 2.5D data blobs, which are filtered and merged in order to increase robustness and trajectory accuracy.

Experimental results show that the contribution of 2.5D data makes head tracking more robust than using only 2D data. Indeed, situations with partial occlusions and clutter are resolved with an error smaller than the head radius. Therefore, tracking does not lose target thanks to the inclusion of 2.5D data.

As for hand tracking, results show that hands are tracked with an average error of between $2\,cm$ and $6\,cm$ depending on the gesture being performed. Fast gestures are tracked worse than slow and smooth ones. Two-handed gestures are also tracked, with an average error smaller than $4\,cm$ per hand.

The presented system executes at $68\,fps$ on a standard CPU, which is fast enough for real-time, since current range cameras have a frame rate of about $20 - 30\,fps$.

Many improvements to this proposal are foreseen. Exploiting hand estimation to improve head estimation and vice-versa (cross feedback) is currently under study. Other setup proposals are considered, such as including a color camera, which will contribute with color and disparity information. Finally, fitting an upper-body model to these estimates is also considered. Such approach will help increasing the temporal consistency of the tracking, and also including inverse kinematics techniques to be able to perform a more complete and accurate tracking, as well as to recognize more complex gestures.

7. REFERENCES

[1] Joachim Neumann, Josep R Casas, Dušan Macho, and Javier Ruiz Hidalgo, "Integration of audiovisual sensors and technologies in a smart room," *Personal and Ubiquitous Computing*, vol. 13, no. 1, pp. 15–23, 2007.

[2] A Kolb, E Barth, R Koch, and R Larsen, "Time-of-Flight Cameras in Computer Graphics," *Computer Graphics Forum*, vol. 29, no. 1, pp. 141–159, 2010.

[3] Nicolas H. Lehment, Moritz Kaiser, Dejan Arsic, and Gerhard Rigoll, "Cue-Independent Extending Inverse Kinematics For Robust Pose Estimation in 3D Point Clouds," in *International Conference on Image Processing (ICIP)*, 2010, p. to appear.

[4] Alessandro Bevilacqua, Luigi Stefano, and Pietro Azzari, "People Tracking Using a Time-of-Flight Depth Sensor," *2006 IEEE International Conference on Video and Signal Based Surveillance*, pp. 89–89, 2006.

[5] Amit Bleiweiss and Michael Werman, "Fusing Time-of-Flight Depth and Color for Real-Time Segmentation and Tracking," in *Proceedings of the DAGM 2009 Workshop on Dynamic 3D Imaging*. 2009, vol. 5742, p. 69, Springer.

[6] Martin Haker, Martin Bohme, Thomas Martinetz, and Erhardt Barth, "Geometric Invariants for Facial Feature Tracking with 3D TOF Cameras," *Circuits and Systems ISSCS 2007*, , no. July, pp. 3–6, 2007.

[7] Martin Bohme, Martin Haker, Thomas Martinetz, and Erhardt Barth, "Head tracking with combined face and nose detection," *2009 International Symposium on Signals Circuits and Systems*, , no. July, pp. 1–4, 2009.

[8] Marco Nichau and Volker Blanz, "Pose-insensitive nose detection in TOF-scans," *Computer Vision and Image Understanding*, vol. 114, no. 12, pp. 1346–1352, Dec. 2010.

[9] Sotiris Malassiotis and Michael G Strintzis, "Robust real-time 3D head pose estimation from range data," *Pattern Recognition*, vol. 38, no. 8, pp. 1153–1165, Aug. 2005.

[10] C Stauffer and W E L Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*, vol. 2, no. c, pp. 246–252, 1999.

[11] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.

[12] Songrit Maneewongvatana and David M. Mount, "Its okay to be skinny, if your friends are fat," in *4th Annual CGC Workshop on Comptutational Geometry*, 1999, number October, pp. 1–8.