REAL TIME 3D HEAD POSE ESTIMATION: RECENT ACHIEVEMENTS AND FUTURE CHALLENGES

Gabriele Fanelli, Juergen Gall, Luc Van Gool

Computer Vision Laboratory - ETH Zurich

ABSTRACT

Most automatic face recognition algorithms try to normalize facial images in order to remove variations caused by anything but the identity of the person. Lighting conditions being less problematic since the introduction of reliable and affordable depth sensors, head pose is the other great source of undesired variations in facial images. In this paper, we describe recent state-of-the-art methods for real time head pose estimation from depth data, present available databases, and discuss open problems to be addressed by future research.

***Index Terms*—** Head pose estimation, depth, 3D, random forests, real time.

1. INTRODUCTION

Automatic face recognition systems have developed greatly in the last decades. However, we are far from the perfect algorithm and huge efforts are being put into developing methods robust to the great amount of variations which can appear between two images of the same subject. One main cause of variation is head pose; for this reason, robust and accurate systems for head pose estimation, and its consequent normalization, are highly demanded.

Automatic analysis of head movements is an active field of research, however, most of the methods in the literature focus on standard images or videos as input, facing challenges like illumination changes which are yet to be overcome. A recent survey on head pose estimation methods from 2D imagery can be found in [1].

Recently, depth sensors have become both affordable (*e.g.*, MS Kinect, ASUS Xtion) and accurate (*e.g.*, [2]). The additional depth information proves key for overcoming many of the problems inherent to 2D video data and recent works demonstrate its importance for the head pose estimation problem [3, 4, 5, 6]. In this paper, the state-of-the-art methods in head pose estimation from range data are presented, and insights are given for the challenges to be faced in the future.

2. PREVIOUS WORK

Recently, computer vision research has devoted increasing efforts to head movements analysis leveraging the newly available depth cue. However, most of the methods do tracking rather than estimation [7, 8, 9, 10]. Even though the results can be impressive, tracking usually requires initialization, may suffer from drifting and can lose track if the head motion is too fast or occlusion too severe.

Some recent works try to solve the problem on a frame-by-frame basis, leading to algorithms which result in really robust head pose analysis systems. Such methods can be divided into two main groups, depending on whether they rely on the detection of some specific facial features, *e.g.*, the nose, or not.

2.1. Features-based approaches

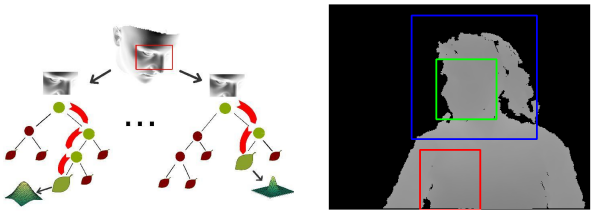
Examples of features-based methods include the work of Lu and Jain [11], where nose position hypotheses are generated based on directional maxima. Chang et al. [12] use curvature information to find eye cavities, nose saddle and nose tip, similarly to [13]. These methods are not real time and in general cannot handle large pose variations.

Breitenstein et al. [5] presented a robust, real time system able to handle large pose changes. They use a geometric descriptor to find nose hypotheses in high quality range scans. Such hypothesis are then all evaluated, comparing the input data with a large number of renderings of a generic face template, finally choosing the orientation minimizing a specific error function. The system achieves real time performance thanks to parallel computation on the GPU, a requirement often clashing with portability and/or power consumption constraints. Even though the system is robust to partial occlusions, these cannot include the nose. The same authors extended their system to use lower quality depth images from a pair or stereo cameras in [6], though the main shortcomings of the original method remain.

2.2. Feature-less approaches

Instead of relying on some specific facial feature, which might become occluded, it is possible to use the whole facial image to estimate the head pose; after all, humans don't need to see someone's nose to guess where (s)he is facing.

Our recent works [3, 4] directly learn a regression between depth images and probabilities in the head pose space,



(a) (b)

Table 1. Left: Example of a random forest for head pose estimation from depth data. Right: Example training patches extracted from an annotated depth image.

i.e., the 3D head location plus the three Euler rotation angles. Such an approach lets every image region vote for the head pose, therefore it is not constrained to a certain area of the face to be visible. In addition, no initialization is needed and partial occlusions are handled, even of the nose.

Random forests [14] have recently enjoyed a notable success in various branches of computer vision: From object detection and action recognition on 2D images and videos [15], to real time body pose estimation from depth images [16].

We use random regression forests to map depth image patches to probabilistic votes in a continuous head pose space, both using high quality range scans [3] and lower quality depth images acquired with a commercial sensor (*e.g.*, Kinect) [4]. The speed of random forests makes it possible to achieve real time performance on standard processors, and the desired trade-off between accuracy and computation cost can be easily found by sampling more or less patches at test time. Random forests are very powerful in learning complex mappings from large training sets, therefore collecting training data is a key issue.

In the following, we describe in details our approach for head pose estimation based on random forests, present two valuable databases available for research purposes, and discuss future challenges of head pose estimation from 3D data.

3. RANDOM FORESTS FOR HEAD POSE ESTIMATION

Regression trees map complex input spaces into continuous, simpler spaces. A tree splits a difficult problem into smaller ones, solvable with simple predictors, achieving highly non-linear mappings. Each node in a tree performs a test, the results of which directs a data sample towards one or the other child node. The tests are optimized in order to cluster the training data as to allow good predictions using simple models, which are computed and stored at the leaves.

Random forests are collections of trees trained on a randomly sub-sampled portion of the training dataset to prevent over-fitting [14]. An example regression forest for head pose estimation is shown in Figure 2.2(a).

Training a forest is a supervised learning problem: In our setup, depth patches are annotated with class label (whether the patch was extracted from the head region or not) and a vector containing the offset between the 3D point falling on the patch's center and the head center location, plus the Euler rotation angles describing the head orientation. We randomly select positive and negative patches as show in Figure 2.2(b) and construct the tree following the standard random forest procedure [14].

Training a tree boils down to selecting, for each non-leaf node starting from the root, a binary test out of a large sample of randomly generated tests, in order to split the current set of patches according to the desired optimization function. The data is then split and the process iterates until a leaf is created when the maximum depth is reached or less than a small number of patches are left. Given the training patches left, leaves store the ratio of positive versus negative patches and the multivariate Gaussian distribution computed from the positive patches' labels.

In [3], we assumed the face to be the prominent object in the scan and all patches to be positive. In that case, the optimization function only tends to maximize the tree's regression power, *i.e.*, splitting the patches so that the variance of their labels is minimized. When, instead, the forest needs to additionally classify patches into head/not head, as it is more realistically the case in most application scenarios, the optimization function should be a mixture of regression and classification measures. As demonstrated in [4], different strategies for combining such measures are possible and in fact lead to similar results.

Given a test image, patches are densely sampled and sent through all trees in the forest. At each node, the patches are evaluated according to the stored binary test and passed either to the right or left child until a leaf node is reached. Arrived at a leaf, a patch is classified and a vote cast for the pose parameters based on the stored distribution.

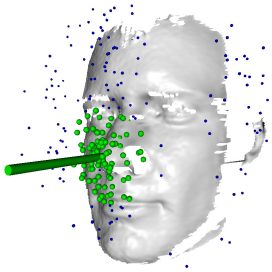


Fig. 1. Selected votes (in green) after clustering and mean-shift. The outliers (blue) are not taken into account for the final estimate.

To remove outliers, the votes are first clustered in 3D, thus roughly localizing the heads present in the scene, and then a few mean-shift iterations are performed. All votes falling outside the (spherical) kernel are discarded, a process exempli-

	Head localization error	Yaw error	Pitch error	Direction estimation accuracy
3D scanner	$13.4 \pm 21.1mm$	$5.7 \pm 15.2\,^\circ$	$5.1 \pm 4.9\,^\circ$	90.4%
Kinect	$14.6 \pm 22.3mm$	$8.9 \pm 13.0\,^\circ$	$8.5 \pm 9.9\,^\circ$	80.0%

Table 2. Mean and standard deviations of the angular errors for the head pose algorithm applied to high quality 3D scans and low quality Kinect data. The last column shows the percentage of images where the angular error was below 10 degrees.

fied in Fig. 1, with the green spheres being the selected votes. The sum of the remaining random variables is a new Gaussian, representing our final estimate of the output parameters (the mean) and a measure of its uncertainty (the covariance).

4. DATABASES

Breitenstein et al. [5] collected a dataset of over 10k annotated range scans of heads. The subjects, both males and females, with and without glasses, were recorded using the scanner of [2] while turning their heads around, trying to span all possible yaw and pitch rotation angles they could. The scans were automatically annotated, tracking each sequence with a personalized template and the ICP algorithm.

The system of [3] was trained on 50k depth images synthetically generated using the Basel Face Model [17], a linear model capable of different identities (but not expressions). We rendered the faces under random perturbations of the PCA parameters and arbitrary, large rotations. The Basel Face Model is available to the research community, so it is possible to recreate a similar training set.

For the work presented in [4], where depth images coming from low-quality consumer cameras were used, we collected a new database of real head movements, using a Kinect. We recorded 24 sequences of 20 people, sitting about one meter away from the sensor, trying to span all possible head rotations. We processed the data off-line with the template-based head tracker provided by `faceshift.com` [8]; the process is fully automatic and the mean translation and rotation errors of the annotations were around 1 mm and 1 degree respectively. The resulting dataset contains about 15K frames (both depth and rgb images), all annotated with head center locations and rotation angles, ranging between around $\pm 75\,^\circ$ for yaw, $\pm 60\,^\circ$ for pitch, and $\pm 50\,^\circ$ for roll. Fig. 4 shows some frames from the database.

(a) Depth images (b) RGB images

Fig. 2. Example frames from the Biwi Kinect Head Pose Database, containing both depth images (a) and rgb images (b) of 20 different subjects.

Both the ETH Face Pose Range Image Data Set [5], and the Biwi Kinect Head Pose Database [4] are freely available for download [1].

5. RESULTS

Table 2 shows the average accuracy for the approaches summarized in Section 3, when tested on the above databases, depending on whether they were applied to high resolution range scans [3] or to low resolution, noisy Kinect data [4]. It can be noted that, even though the Kinect depth images are of a much lower quality, the errors are still acceptable. Fig. 3 shows the head pose estimation system running in real time on a standard laptop. Source code for the demo can be found at `www.vision.ee.ethz.ch/gfanelli`.

6. CHALLENGES

We have summarized recent work on 3D, real time head pose estimation. The use of the new depth cue allowed for great improvements in performance, compared to previous methods relying on 2D images.

Even though depth sensors can solve much of the ambiguities inherent of standard video (*e.g.*, lighting changes) and even if their prices recently dropped with the launching of Microsoft Kinect, their distribution is still limited. Moreover, the most successful type of depth sensors are based on infrared structured light, which only works indoor. For outdoor scenarios, other technologies are needed, such as time of flight cameras or stereo setups; while the former still provide rather low resolution depth images, the latter can produce very noisy reconstructions. However, we believe depth sensors will improve in the future, both in portability and accuracy.

Future work on head pose estimation could use color images in addition to depth data, as an RGB camera is available in the most common devices, *e.g.*, MS Kinect and Asus Xtion. Moreover, a desirable addition to the current systems, in order to ease the pose normalization procedure required by recognition algorithms, would be the extension to localizing more facial feature points like eyes or mouth corners.

The methods presented above only work for rather controlled settings. In particular, challenges are posed by long-

[1] http://www.vision.ee.ethz.ch/datasets/

Fig. 3. The system of [4] running in real time, on a standard laptop.

haired subjects and by heads occluding each other. To improve learning-based methods, new, more realistic training databases are required, which should be large enough to cover all possible scenarios to be expected at test time, *e.g.*, overhead cameras placed in crowded spaces. Synthesizing training data is an option, and databases like the one of [16] could be generated; however, covering all possible variations in hair styles and head-wears might be problematic.

Even when training data could be synthesized, there would still be the need for large, realistic testing databases. Annotating such datasets is a difficult task, as tracking algorithms cannot handle many real-life scenarios and manual annotation is both expensive and error-prone.

Acknowledgments The authors acknowledge financial support from the EU projects RADHAR (FP7-ICT-248873) and TANGO (FP7-ICT-249858) and from the SNF project Vision-supported Speech-based Human Machine Interaction (200021-130224).

7. REFERENCES

[1] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *TPAMI*, vol. 31, no. 4, pp. 607–626, 2009.

[2] T. Weise, B. Leibe, and L. Van Gool, "Fast 3d scanning with automatic motion compensation," in *CVPR*, 2007.

[3] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *CVPR*, 2011.

[4] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *DAGM*, 2011.

[5] M. D. Breitenstein, D. Kuettel, T. Weise, L. Van Gool, and H. Pfister, "Real-time face pose estimation from single range images," in *CVPR*, 2008.

[6] M. D. Breitenstein, J. Jensen, C. Hoilund, T. B. Moeslund, and L. Van Gool, "Head pose estimation from passive stereo images," in *SCIA*, 2009.

[7] T. Weise, H. Li, L. Van Gool, , and M. Pauly, "Face/off: Live facial puppetry," in *SCA*, 2009.

[8] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Real-time performance-based facial animation," *SIGGRAPH*, 2011.

[9] M. Breidt, H. Buelthoff, and C. Curio, "Robust semantic analysis by synthesis of 3d facial motion," in *AFGR*, 2011.

[10] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, "3d deformable face tracking with a commodity depth camera," in *ECCV*, 2010.

[11] X. Lu and A.K. Jain, "Automatic feature extraction for multiview 3d face recognition.," in *AFGR*, 2006.

[12] K. I. Chang, K. W. Bowyer, and P. J. Flynn, "Multiple nose region matching for 3d face recognition under varying facial expression," *TPAMI*, vol. 28, no. 10, pp. 1695–1700, 2006.

[13] Y. Sun and L. Yin, "Automatic pose estimation of 3d facial models," in *ICPR*, 2008.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *TPAMI*, vol. 33, no. 11, pp. 2188–2202, November 2011.

[16] 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.

[17] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," in *AVSS*, 2009.