Real-time Kinect Head Pose Estimation

Gabriele Fanelli Juergen Gall Luc Van Gool
Computer Vision Laboratory
ETH Zuerich, Switzerland
{fanelli,gall,vangool}@vision.ee.ethz.ch

Abstract

We present a system for estimating the location and orientation of a person's head from depth data acquired with a Kinect or similar consumer depth cameras. Our approach is based on discriminative random regression forests which simultaneously classify image regions into whether they belong to the head region or not and cast probabilistic votes in a continuous space of head poses, defined as the 3D position of the nose and the Euler rotation angles.

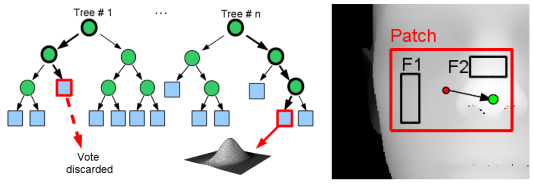
1. Introduction

Despite recent advances, people still interact with machines mainly through devices like keyboards and mice. Besides the interpretation of full body movements, as done by the Kinect, future, more natural interfaces will need to take also head motion into account, as much of human-human communication goes through face and head movements.

In particular, head pose estimation is a key component of many desirable applications, from identity recognition to driver's drowsiness detection. Most the works in the literature [6] use standard imagery, facing challenges like lighting changes and texture-less facial regions; many of such problems can now be solved thanks to the introduction of the Kinect and other affordable depth sensors. Yet, 3D data has mainly been used for face tracking [7], leaving open issues of drift and (re-)initialization. Frame-by-frame estimation, on the other hand, provides increased robustness.

We define 3D head pose estimation as the localization of the nose tip and the determination of the head orientation encoded as Euler angles. Most 3D methods use geometry to localize the nose tip and are thus sensitive to its occlusion. In [2] and [3], we instead introduced a voting framework where different depth patches contribute to the estimation. We use random forests (RFs) [1], following their recent success in many computer vision tasks: from object detection and action recognition [4] to real-time human pose estimation [5]. RFs are fast at both training and testing, lend themselves to parallelization and are inherently multi-class. The method we demonstrate estimates the desired, continuous parameters directly from low resolution depth images acquired with a Kinect [3]. Our system works in real-time, without manual initialization, and does not rely on specific features to be visible. In our demonstrator, we show that it works for unseen faces and that it handles large pose changes, variations in facial hair, and partial occlusions due to glasses, hands, or missing parts in the 3D reconstruction.

2. Method



(a) (b)

Figure 1. (a): Example random forest for head pose estimation. Test samples ending in positive leaves retrieve a Gaussian distribution used for voting in the head pose space. (b): Example binary test defined within the depth patch.

Random forests [1] are collections of decision trees, each trained on a randomly sampled subset of the available data. In our system, a tree is built from a set of annotated patches, randomly extracted from the training images. Annotation includes a binary class label c (head/not head) and real-valued quantities describing the head pose θ (*i.e.*, the offset vector from the 3D patch center and the nose tip and the three Euler rotation angles). Each tree is built recursively by selecting a binary test $\phi(\mathcal{I}) \rightarrow \{0,1\}$ at each node, which sends each patch either to the left or right child based on its appearance $\mathcal{I}$. Our binary tests $\phi_{F_1,F_2,\tau}(\mathcal{I})$ are defined as:

$$|F_1|^{-1} \sum_{\boldsymbol{q} \in F_1} \mathcal{I}(\boldsymbol{q}) - |F_2|^{-1} \sum_{\boldsymbol{q} \in F_2} \mathcal{I}(\boldsymbol{q}) > \tau, \qquad (1)$$

where F_1 and F_2 are two rectangles defined within the patch, and τ is a threshold. An example test is shown in

Fig. 1 (a), with the red patch containing the two regions F_1 and F_2 defining the test (in black); the arrow is the offset between the 3D patch center (in red) and the ground truth nose location. The optimal test for a node is chosen from a large pool of randomly generated binary tests, as the one which maximizes the information gain of the split $IG(\phi)$:

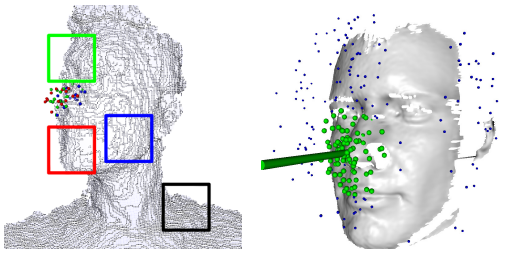
$$IG(\phi) = \mathcal{H}(\mathcal{P}) - \sum_{i \in \{L,R\}} w_i \mathcal{H}(\mathcal{P}_i(\phi)), \qquad (2)$$

where w_i is the ratio of patches sent to each child node. $\mathcal{H}(\mathcal{P})$ is a measure of the patch cluster $\mathcal{P}$ related to the labels' entropy, taking both classification (dividing head patches from the rest of the body) and regression (clustering together patches with similar votes in the head pose space) into account. In [3], we evaluated different strategies for jointly solving the classification and regression problems at hand, all achieving comparable results. The process continues with the left and the right child using the corresponding training sets until a leaf is created when either the maximum tree depth is reached, or less than a minimum number of training samples are left.

For each leaf, the probabilities of belonging to a head $p(c = 1|\mathcal{P})$ and the distributions of the continuous head pose parameters $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \overline{\boldsymbol{\theta}}, \boldsymbol{\Sigma})$ (we assume multivariate Gaussians) are stored. The distributions are estimated from the training patches that arrive at the leaf. When presented with a test image, patches are sampled (a stride controls the sampling's density) and sent down through all trees. Each patch is guided by the binary tests until a leaf, where the probability $p(c = 1|\mathcal{P})$ tells whether the patch belongs to a head or not. We only consider leaves with a high probability and use the stored distributions for estimating $\boldsymbol{\theta}$, as exemplified by Fig. 2. The votes are then clustered, and the clusters further refined by mean shift in order to remove outliers, as shown in Fig. 2(b). A cluster with a large number of votes is declared as a head and the votes averaged, resulting in a Gaussian which encodes the head pose estimate (mean) and a measure of its uncertainty (covariance).

3. Evaluation

To train and test our system, we recorded a database with a Kinect: 24 sequences of 20 subjects recorded while sitting about 1 meter away from the sensor. All subjects rotated their heads around freely and we labeled the sequences with the nose position and head orientation using the person-specific head tracker of [7]. The resulting Biwi Kinect Head Pose Database contains head rotations in the range of around $\pm 75°$ for yaw, $\pm 60°$ for pitch, and $\pm 50°$ for roll. We performed a 4-fold, subject-independent cross-validation, using as measure $\mathcal{H}$ the exponential weighting scheme proposed in [3], with



(a) (b)

Figure 2. (a) Example votes. The colored patches are classified as positives and cast votes for the nose position (correspond spheres). The black patch is classified as negative and does not vote. (b) Example test image: the green spheres represent the votes selected after outliers (blue spheres) are filtered out. The green cylinder stretches from the estimate of the nose center in the face direction.

Figure 3. Example frames from a video of the system running in real time on a standard laptop.

$\lambda = 5$. For a forest of 20 trees and a stride of 10, the nose localization error was $13.0 \pm 26.2mm$ and the direction estimation error $4.1 \pm 8.3°$, for a processing time of about 19 ms. Videos, database, and sample source code are available at www.vision.ee.ethz.ch/~gfanelli/head_pose/head_forest.html.

References

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 1

[2] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1

[3] G. Fanelli, T. Weise, J. Gall, and L. Van Gool. Real time head pose estimation from consumer depth cameras. In *German Association for Pattern Recognition*, 2011. 1, 2

[4] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 2011. 1

[5] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. *ICCV*, 2011. 1

[6] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009. 1

[7] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2011. 1, 2