Real Time Head Pose Estimation from Consumer Depth Cameras

Gabriele Fanelli[1], Thibaut Weise[2], Juergen Gall[1] and Luc Van Gool[1,3]

[1]ETH Zurich, Switzerland [2]EPFL Lausanne, Switzerland [3]KU Leuven, Belgium
{fanelli,gall,vangool}@vision.ee.ethz.ch, thibaut.weise@epfl.ch

Abstract. We present a system for estimating location and orientation of a person's head, from depth data acquired by a low quality device. Our approach is based on discriminative random regression forests: ensembles of random trees trained by splitting each node so as to simultaneously reduce the entropy of the class labels distribution and the variance of the head position and orientation. We evaluate three different approaches to jointly take classification and regression performance into account during training. For evaluation, we acquired a new dataset and propose a method for its automatic annotation.

1 Introduction

Head pose estimation is a key element of human behavior analysis. For this reason, many applications would benefit from automatic and robust head pose estimation systems. While 2D video presents ambiguities hard to resolve in real time, systems relying on 3D data have shown very good results [5, 10]. Such approaches, however, use bulky 3D scanners like [22] and are not useful for consumer products or mobile applications like robots. Today, cheap depth cameras exist, even though they provide much lower quality data.

We present an approach for real time 3D head pose estimation robust to the poor signal-to-noise ratio of current consumer depth cameras. The method is inspired by the recent work of [10] that uses random regression forests [9] to estimate the 3D head pose in real time from high quality depth data. It basically learns a mapping between simple depth features and real-valued parameters such as 3D head position and rotation angles. The system achieves very good performance and is robust to occlusions but it assumes that the face is the sole object in the field of view. We extend the regression forests such that they discriminate depth patches that belong to a head (*classification*) and use only those patches to predict the pose (*regression*), jointly solving the classification and regression problems. In our experiments, we evaluate several schemes that can be used to optimize both the discriminative power as well as the regression accuracy of such a random forest. In order to deal with the characteristic noise level of the sensor, we cannot rely on synthetic data as in [10], but we have to acquire real training examples, i.e., faces captured with a similar sensor. We therefore recorded several subjects and their head movements, annotating the data by tracking each sequence using a personalized template.

Our system works on a frame-by-frame basis, needs no initialization, and runs in real time. In our experiments, we show that it can handle large pose changes and variations such as facial hair and partial occlusions.

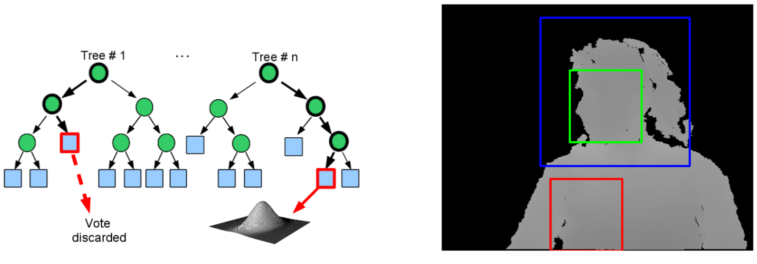
2 Related Work

The literature contains several works on head pose estimation, which can be conveniently divided depending on whether they use 2D images or depth data.

Among the algorithms based on 2D images, we can further distinguish between appearance-based methods, which analyze the whole face region, and feature-based methods, which rely on the localization of specific facial features, e.g., the eyes. Examples of appearance-based methods are [13] and [17], where the head pose space is discretized and separate detectors are learned for each segment. Statistical generative models, e.g., active appearance models [8] and their variations [7, 19, 2], are very popular in the face analysis field, but are rarely employed for head pose estimation. Feature-based methods are limited by their need to either have the same facial features visible across different poses, or define pose-dependent features [24, 16]. In general, all 2D image-based methods suffer from several problems, in particular changes in illumination and identity, and rather textureless regions of the face.

With the recent increasing availability of depth-sensing technologies, a few notable works have shown the usefulness of the depth for solving the problem of head pose estimation, either as unique cue [5, 10], or in combination with 2D image data [6, 20]. Breitenstein et al. [5] developed a real time system capable of handling large head pose variations. Using high quality depth data, the method relies on the assumption that the nose is visible. Real time performance is achieved by using the parallel processing power of a GPU. The approach proposed in [10] also relies on high quality depth data, but uses random regression forests [9] to estimate the head pose, reaching real time performance without the aid of parallel computations on the GPU and without assuming any particular facial feature to be visible. While both [10] and [5] consider the case where the head is the only object present in the field of view, we deal with depth images where other parts of the body might be visible and therefore need to discriminate which image patches belong to the head and which don't.

Random forests [4] and their variants are very popular in computer vision [18, 11, 9, 14, 12] for their capability of handling large training sets, fast execution time, and high generalization power. In [18, 11], random forests have been combined with the concept of Hough transform for object detection and action recognition. These methods use two objective functions for optimizing the classification and the Hough voting properties of the random forests. While Gall et al. [11] randomly select which measure to optimize at each node of the trees, Okada [18] proposes a joint objective function defined as a weighted sum of the classification and regression measures. In this work, we evaluate several schemes for integrating two different objective functions including linear weighting [18] and random selection [11].



(a) (b)

Fig. 1. Simple example of Discriminative Regression Forest a): A patch is sent down to two trees, ending up in a non-head leaf in the first case, thus not producing a vote, and in a head leaf in the second case, extracting the multivariate Gaussian distribution stored at the leaf. In b), one training depth image is shown. The blue bounding box enclosing the head specifies where to sample positive (green - inside) and negative patches (red - outside).

3 Discriminative Random Regression Forests for Head Pose Estimation

Decision trees [3] are powerful tools capable of splitting a hard problem into simpler ones, solvable with trivial predictors, and thus achieving highly non-linear mappings. Each node in a tree performs a test, the result of which directs a data sample towards one of the children nodes. The tests at the nodes are chosen in order to cluster the training data as to allow good predictions using simple models. Such models are computed and stored at the leaves, based on the clusters of annotated data which reach them during training.

Forests of randomly trained trees generalize much better and are less sensitive to overfitting than decision trees taken separately [4]. Randomness is introduced in the training process, either in the set of training examples provided to each tree, in the set of tests available for optimization at each node, or in both.

When the task at hand involves both classification and regression, we call Discriminative Random Regression Forests (DRRF) an ensemble of trees which allows to simultaneously separate test data into whether they represent part of the object of interest and, only in the positive cases, vote for the desired real valued variables. A simple DRRF is shown in Figure 1(a): The tests at the nodes lead a sample to a leaf, where it is classified. Only if classified positively, the sample retrieves a Gaussian distribution computed at training time and stored at the leaf, which is used to cast a vote in a multidimensional continuous space.

Our goal is to estimate the 3D position of a head and its orientation from low-quality depth images acquired using a commercial, low-cost sensor. Unlike in [10], the head is not the only part of the person visible in the image, therefore the need to classify image patches before letting them vote for the head pose.

3.1 Training

Assuming a set of depth images is available, together with labels indicating head locations and orientations, we randomly select patches of fixed size from the region of the image containing the head as positives samples, and from outside the head region as negatives. Figure 1(b) shows one of the training images we used (acquisition and annotation is explained in Section 4), with the head region marked in blue, and examples of a positive and negative patch drawn in green, respectively red.

A tree T in the forest $\mathcal{T} = \{T_t\}$ is constructed from the set of patches $\{\mathcal{P}_i = (\mathcal{I}_i, c_i, \boldsymbol{\theta}_i)\}$ sampled from the training images. $\mathcal{I}_i$ are the depth patches and $c_i \in \{0,1\}$ are the class labels. The vector $\boldsymbol{\theta}_i = \{\theta_x, \theta_y, \theta_z, \theta_{ya}, \theta_{pi}, \theta_{ro}\}$ contains the offset between the 3D point falling on the patch's center and the head center location, and the Euler rotation angles describing the head orientation.

As in [10], we define the binary test at a non-leaf node as $t_{F_1,F_2,\tau}(\mathcal{I})$:

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

where F_1 and F_2 are rectangular, asymmetric regions defined within the patch and τ is a threshold. Such tests can be efficiently evaluated using integral images.

During training, for each non-leaf node starting from the root, we generate a large pool of binary tests $\{t^k\}$ by randomly choosing F_1, F_2, and τ. The test which maximizes a specific optimization function is picked; the data is then split using the selected test and the process iterates until a leaf is created when either the maximum tree depth is reached, or less than a certain number of patches are left. Leaves store two kinds of information: The ratio of positive patches that reached them during training $p\big(c = 1|\,\mathcal{P}\big)$ and the multivariate Gaussian distribution computed from the pose parameters of the positive patches.

For the problem at hand, we need trees able to both classify a patch as belonging to a head or not and cast precise votes into the spaces spanned by 3D head locations and orientations. This is the main difference with [10], where the face is assumed to cover most of the image and thus only a regression measure is used. We thus evaluate the goodness of a split using a classification measure $U_C\big(\{\mathcal{P}\,|t^k\}\big)$ and a regression measure $U_R\big(\{\mathcal{P}\,|t^k\}\big)$: The former tends to separate the patches at each node seeking to maximize the discriminative power of the tree, the latter favors regression accuracy.

Similar to [11], we employ a classification measure which, when maximized, tends to separate the patches so that class uncertainty for a split is minimized:

$$U_C\big(\{\mathcal{P}\,|t^k\}\big) = \frac{|\mathcal{P}_L| \cdot \sum_c p\big(c|\,\mathcal{P}_L\big) ln\big(p\big(c|\,\mathcal{P}_L\big)\big) + |\mathcal{P}_R| \cdot \sum_c p\big(c|\,\mathcal{P}_R\big) ln\big(p\big(c|\,\mathcal{P}_R\big)\big)}{|\mathcal{P}_L| + |\mathcal{P}_R|}, \tag{2}$$

where $p\big(c|\,\mathcal{P}\big)$ is the ratio of patches belonging to class $c \in \{0,1\}$ in the set $\mathcal{P}$.

For what concerns regression, we use the information gain defined by [9]:

$$U_R\big(\{\mathcal{P}\,|t^k\}\big) = H(\mathcal{P}) - (w_L H(\mathcal{P}_L) + w_R H(\mathcal{P}_R)), \tag{3}$$

where $H(\mathcal{P})$ is the differential entropy of the set $\mathcal{P}$ and $w_{i=L,R}$ is the ratio of patches sent to each child node.

Our labels (the vectors $\boldsymbol{\theta}$) are modeled as realizations of a multivariate Gaussian, i.e., $p(\boldsymbol{\theta}|L) = \mathcal{N}(\boldsymbol{\theta}; \overline{\boldsymbol{\theta}}, \boldsymbol{\Sigma})$. Moreover, as in [10], we assume the covariance matrix to be block-diagonal, i.e., we allow covariance only among offset vectors and among head rotation angles, but not between the two. For these reasons, we can rewrite eq. (3) as:

$$U_R\big(\,\{\mathcal{P}\,|t^k\,\}\,\big) = \log\left(|\boldsymbol{\Sigma}^v| + |\boldsymbol{\Sigma}^a|\right) - \sum_{i=\{L,R\}} w_i \log\left(|\boldsymbol{\Sigma}_i^v| + |\boldsymbol{\Sigma}_i^a|\right), \tag{4}$$

where $\boldsymbol{\Sigma}^v$ and $\boldsymbol{\Sigma}^a$ are the covariance matrices of the offsets and rotation angles (the two diagonal blocks in $\boldsymbol{\Sigma}$). Maximizing Eq. (4) minimizes the determinants of these covariance matrices, thus decreasing regression uncertainty.

The two measures (2) and (4) can be combined in different ways, and we investigate three different approaches. While the method [11] randomly chooses between classification and regression at each node, the method [18] uses a weighted sum of the two measures, defined as:

$$\arg\max_k \Big(U_C + \alpha \max\big(p\big(c = 1|\,\mathcal{P}\big) - t_p, 0\big)\, U_R\Big). \tag{5}$$

In the above equation, $p\big(c = 1|\,\mathcal{P}\big)$ represents the ratio of positive samples contained in the set, or purity, t_p is an activation threshold, and α a constant weight. When maximizing (5), the optimization is steered by the classification term alone until the purity of positive patches reaches the threshold t_p. From that point on, the regression term starts to play an ever important role.

We propose a third way to combine the two measures by removing the activation threshold from (5) and using as weight an exponential function:

$$\arg\max_k \Big(U_C + (1.0 - e^{-\frac{d}{\lambda}})U_R\Big), \tag{6}$$

where d is the depth of the node. In this way, the regression measure is given increasingly higher weight as we descend towards the leaves, with the parameter λ specifying the steepness of the change.

3.2 Head pose estimation

For estimating the head pose from a depth image, we densely extract patches from the image and pass them through the forest. The tests at the nodes guide each patch all the way to a leaf L, but not all leaves are to be considered for regression; only if $p\big(c = 1|\,\mathcal{P}\big) = 1$ and $trace\,(\boldsymbol{\Sigma}) < max_v$, with max_v an empirical value for the maximum allowed variance, the Gaussian $p(\boldsymbol{\theta})$ is taken into account. As in [10], a stride in the sampling of the patches can be introducted in order to find the desired compromise between speed and accuracy of the estimate. To be able to handle multiple heads and remove outliers, we perform a

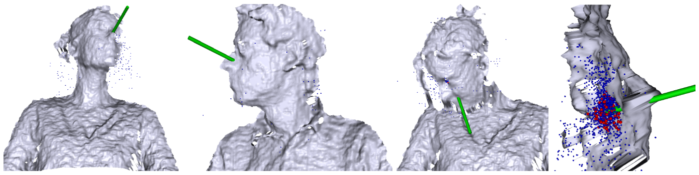


Fig. 2. Some head pose estimation results. Starting from the left, two successfull estimations, one failure, and one image with the camera placed on the side, showing the single votes. In particular, the blue, smaller spheres are all votes returned by the forest, while the larger, red spheres are the votes selected for the final estimation.

bottom-up clustering step: All votes within a certain distance to each other (the average head diameter) are grouped, resulting in big clusters around the heads present in the image. We subsequently run 10 mean shift iterations (using a spherical kernel with a fraction of the average head diameter as radius), in order to better localize the centroid of the clusters. Then, similarly to [9], we select only a percentage of the remaining votes, starting from the ones with smallest uncertainty: if more votes than a threshold are left, we declare a head detected. The Gaussians left at this point are summed, giving us another multivariate Gaussian distribution whose mean is the estimate of the head pose and whose covariance represents its confidence.

Figure 2 shows some processed frames. The green cylinder encodes both the estimated head center and direction of the face. The first two images show success cases, the third one is a failure case, while the last one shows a scan from a side view, revealing the colored votes clustering around the head center. The small blue spheres are all the votes returned by the forest (the means of the Gaussians stored at the leaves reached by the test patches), while the larger, red spheres represent the votes which were selected to produce the final result.

4 Data Acquisition and Labeling

For training and testing our algorithms, we acquired a database of head poses captured with a Kinect sensor. The dataset contains 24 sequences of 20 different people (14 men and 6 women, 4 wearing glasses) recorded while sitting about 1 meter away from the sensor. The subjects were asked to rotate their heads trying to span all possible ranges of angles their head is capable of. Because the depth data needs to be labeled with the 3D head pose of the users for training and evaluation, we processed the data off-line with a template-based head tracker, as illustrated in Fig. 3. To build the template, each user was asked to turn the head left and right starting from the frontal position. The face was detected using [21] and the scans registered and integrated into one 3D point cloud as described by [23]. A 3D morphable model [2] with subsequent graph-based non-rigid ICP [15] was used to adapt a generic face template to the point cloud. This resulted in a template representing the shape of the head. Thanks

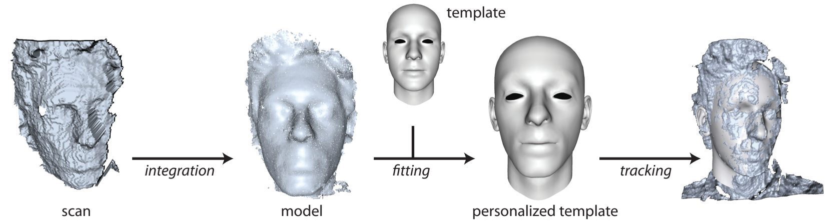


Fig. 3. Automatic pose labeling: A user turns the head in front of the depth sensor, the scans are integrated into a point cloud model and a generic template is fit to it. The personalized template is used for accurate rigid tracking.

to such personalized template, each subject's sequence of head rotations was tracked using ICP [1], resulting in a pose estimate for each frame. Although this method does not provide perfect estimates of the pose, we found that the mean translation and rotation errors were around 1 mm and 1 degree respectively. Note that the personalized face model is only needed for processing the training data, our head pose estimation system does not assume any initialization phase.

The final database contains roughly 15K frames, annotated with head center locations and rotation angles. The rotations of the heads range between around $\pm 75\,^\circ$ for yaw, $\pm 60\,^\circ$ for pitch, and $\pm 50\,^\circ$ for roll.

5 Experiments

For evaluation, we divided the database into a training and test set of respectively 18 and 2 subjects. In order to compare the weighting schemes described in Section 3.1, we trained each forest using exactly the same patches. We fixed the following parameters: patch size (100x100 pixels), maximum size of the subpatches F_1 and F_2 (40x40), maximum tree depth (15), minimum number of patches required for a split (20), number of tests generated at each node (20K), and number of positive and negative patches to be extracted from each image (10). Depending on the method used to combine the classification and regression measures, additional parameters might be needed. For the linear weighting approach, we set the α and t_p as suggested by [18], namely to 1.0 and 0.8. In the interleaved setting [11], each measure is chosen with uniform probability, except at the two lowest depth levels of the trees where the regression measure is used. For the exponential weighting function based on the tree depth, we used λ equal to 2, 5, and 10. For testing, we use the following settings: a 5 pixels stride, maximum leaf variance $max_v = 1500$, radius of the spherical kernel for clustering r_c equal to the average head diameter, and mean shift kernel radius $r_{ms} = r_c/6$.

Results are plotted in Fig. 4 and Fig. 5. All experiments were conducted by building 7 trees, each on 3000 sample images. In Fig. 4(a), the accuracy of all

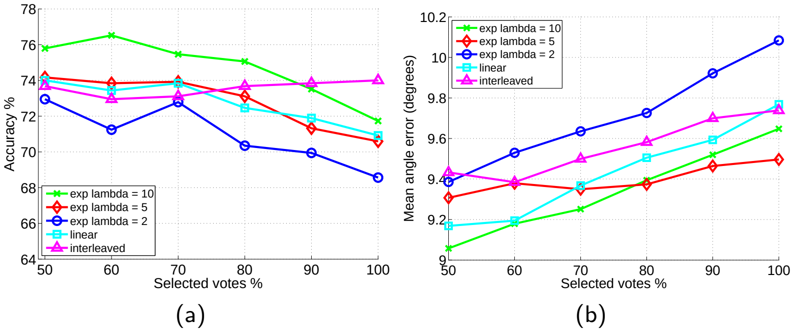


Fig. 4. Accuracy (a) of the tested methods as a function of the percentage of votes selected for each cluster; success is defined when the head estimation error is below $10mm$ and the thresholds for the direction estimation error is set to 15 degrees. The plots in (b) show the average angle errors again as a function of the percentage of selected votes. It can be noted that the evaluated methods perform rather similarly and the differences are small.

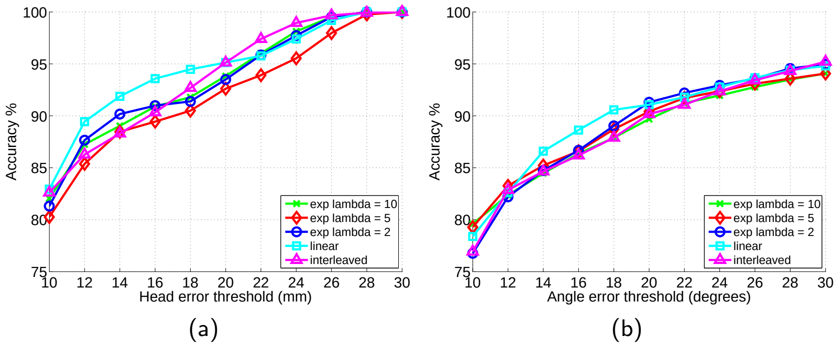


Fig. 5. Accuracy of the head center estimation error (a), respectively of the angle error (b) of the tested methods. The curves are plotted for different values of the threshold defining success. All methods show similar performance.

methods changes as function of the percentage of leaves to be retained during the last step of the regression, as explained in Section 3.2. Success means that the detected head center was within $10mm$ from the ground truth location, and the angle error (L2 norm of the Euler angles) below $10\,^{\circ}$. All methods appear to behave similarly, but we note a slightly higher accuracy for an exponential weight and a 60% of the votes retained. Fig. 4(b) shows the average angular error of the estimate, again plotted with respect to the percentage of retained votes. Again, the differences between the weighting schemes are very small, as can be seen also in the plots of Figs. 5 (a) and (b), showing the accuracy of the head center estimation error, respectively of the angle error, for different values of the threshold defining success.

Stride	Head error	Yaw error	Pitch error	Roll error	Missed detections	Time
4	$14.7 \pm 22.5mm$	$9.2 \pm 13.7\,^\circ$	$8.5 \pm 10.1\,^\circ$	$8.0 \pm 8.3\,^\circ$	1.0%	$87.5ms$
6	$14.5 \pm 22.1mm$	$9.1 \pm 13.6\,^\circ$	$8.5 \pm 9.9\,^\circ$	$8.0 \pm 8.3\,^\circ$	1.5%	$24.6ms$
8	$14.1 \pm 20.2mm$	$9.0 \pm 13.2\,^\circ$	$8.4 \pm 9.6\,^\circ$	$8.0 \pm 8.3\,^\circ$	2.1%	$11.8ms$
10	$14.6 \pm 22.3mm$	$8.9 \pm 13.0\,^\circ$	$8.5 \pm 9.9\,^\circ$	$7.9 \pm 8.3\,^\circ$	2.3%	$7.7ms$

Table 1. Mean and standard deviation of the errors for the 3D head localization task and the individual rotation angles as a function of the stride parameter, together with missed detection rates and average processing time. The values are computed by 5-fold cross validation on the entire dataset.

As a last experiment, we chose the exponentially decreasing weighting of the measures, defined by Equation (6), with λ set to 5. We then ran a 5-fold cross-validation on the full dataset. We trained 7 trees for each fold, each on 3000 depth images. The results are given in Table 1, where mean and standard deviation of the head localization, yaw, pitch and roll errors are shown together with the percentage of missed detections and the average time necessary to process an image, depending on the stride parameter. It can be noted that the system performs beyond real time already for a stride of 6 (needing only 25ms to process a frame on a 2.67GHz Intel Core i7 CPU), still maintaining a small number of wrong detections and low errors.

6 Conclusions

We presented a system for real time head detection and head pose estimation from low quality depth data captured with a cheap device. We use a discriminative random regression forest, which classifies depth image patches between head and the rest of the body and which performs a regression in the continuous spaces of head positions and orientations. The trees making up the forest are trained in order to jointly optimize their classification and regression power by maximizing two separate measures. Two existing methods were presented for combining such measures and a third weighting scheme was introduced which favors the regression measure as an exponential function of the node depth. In our experiments, we compared the proposed methods and observed similar performances in terms of accuracy. In order to train and test our algorithms, we collected and labelled a new dataset using a Kinect sensor, containing several subjects and large variations in head rotations.

7 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).

References

1. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE TPAMI 14(2), 239–256 (1992)
2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH '99. pp. 187–194 (1999)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA (1984)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Breitenstein, M.D., Kuettel, D., Weise, T., Van Gool, L., Pfister, H.: Real-time face pose estimation from single range images. In: CVPR. pp. 1–8 (2008)
6. Cai, Q., Gallup, D., Zhang, C., Zhang, Z.: 3d deformable face tracking with a commodity depth camera. In: ECCV. pp. 229–242 (2010)
7. Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: View-based active appearance models. Image and Vision Computing 20(9-10), 657 – 664 (2002)
8. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE TPAMI 23, 681–685 (2001)
9. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in ct studies. In: Recognition techniques and applications in medical imaging. pp. 106–117 (2010)
10. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: CVPR. pp. 617–624 (2011)
11. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE TPAMI (2011)
12. Huang, C., Ding, X., Fang, C.: Head pose estimation based on random forests for multiclass classification. In: ICPR. pp. 934–937 (2010)
13. Jones, M., Viola, P.: Fast multi-view face detection. Tech. Rep. TR2003-096, Mitsubishi Electric Research Laboratories (2003)
14. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE TPAMI 28, 1465–1479 (2006)
15. Li, H., Adams, B., Guibas, L.J., Pauly, M.: Robust single-view geometry and motion reconstruction. ACM Trans. Graph. 28(5) (2009)
16. Matsumoto, Y., Zelinsky, A.: An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In: Aut. Face and Gesture Rec. pp. 499–504 (2000)
17. Morency, L.P., Sundberg, P., Darrell, T.: Pose estimation using 3d view-based eigenspaces. In: Aut. Face and Gesture Rec. pp. 45–52 (2003)
18. Okada, R.: Discriminative generalized hough transform for object dectection. In: ICCV. pp. 2000–2005 (2009)
19. Ramnath, K., Koterba, S., Xiao, J., Hu, C., Matthews, I., Baker, S., Cohn, J., Kanade, T.: Multi-view aam fitting and construction. IJCV 76, 183–204 (2008)
20. Seemann, E., Nickel, K., Stiefelhagen, R.: Head pose estimation using stereo vision for human-robot interaction. Aut. Face and Gesture Rec. pp. 626–631 (2004)
21. Viola, P., Jones, M.: Robust real-time face detection. IJCV 57(2), 137–154 (2004)
22. Weise, T., Leibe, B., Van Gool, L.: Fast 3d scanning with automatic motion compensation. In: CVPR. pp. 1–8 (2007)
23. Weise, T., Wismer, T., Leibe, B., Van Gool, L.: In-hand scanning with online loop closure. In: 3DIM. pp. 1630–1637 (2009)
24. Yang, R., Zhang, Z.: Model-based head pose tracking with stereovision. In: Aut. Face and Gesture Rec. pp. 255–260 (2002)