Continous Head Pose Estimation using Random Regression Forests

Pierre-Luc Bacon
McGill University
plbacon@cim.mcgill.ca

Abstract

Head pose is a rich visual cue that finds great interest in the field of human robot interaction (HRI) and for video surveillance applications. Previous attempts at solving this problem have often proposed solutions formulated in a classification setting. Furthermore, strong assumptions on illumination and scale in an occlusion-free environment have usually been made. We propose a regression solution to head pose estimation based on random regression forests trained over SURF features. We believe that slight improvements would suffice to make it even more robust under unconstrained environments.

1. Introduction

The problem of face detection and recognition has received a considerable amount of attention early in the field [18]. As for the problem of head pose estimation, it stems directly from this early work but aims at producing an estimate of the direction in which a person is looking at. Whether this problem should be treated as a multi-class classification problem or as a regression one, there is no definite answer applicable to every possible application [9]. However, for the class of applications outlined below, we believe that a regression solution is more desirable.

Successful appearance-based models have been proposed with some relative degree of robustness against different illumination and viewpoints conditions [20]. A severe limitation of many of these techniques however lies in their inability to cope with partial occlusions or changes in scale. For those methods that consider head pose estimation jointly with tracking, this concern of face occlusion is certainly diminished. However, these hybrid methods are still not immune to variations in scale and depend on the quality of the initial estimate about the location of the head in the image.

[4] addresses some of these problems by learning a *codebook* over a set of anatomical regions described by augmented SIFT [14] features in the *Object Constring Invariant* (OCI) framework [15] By computing these local features, the algorithm avoids scaling and illumination issues. With many keypoints detected over the candidate face, the Bayesian inference process applied over the learned codebook can still yield useful estimates. In practice however, the algorithm shows some weaknesses in its ability to estimate head poses that are far from the fronto-parallel axis. Furthermore, the proposed solution does not produce continuous pose estimates.

Some recent work made for depth cameras shows the power of random regression forests to solve head pose estimation in a continuous fashion. [5] uses a Random Forest (RF) [2]: an ensemble of decision trees constructed using different randomization strategies. The resulting ensemble also do not tend to overfit the data; a phenomenon otherwise expected with decision trees. As for prior work in RF for 2D cameras, both [10] and [12] provide a solution framed in a multi-class classification setting but base the inference process on a different feature definition. This paper applies the theory from [5] to revisit [10, 12] and obtain continuous head pose estimation.

Although the canonical application domain for head pose estimation is mainly in video surveillance, it has direct outreaches in *human robot interaction* (HRI) for the development of visual attention models. The ability for a robot to follow the gaze direction of an interacting human would allow to infer the target object of attention in the environment and establish *joint attention*.

2. Random Regression Forests

Random Forests (RF) improve upon the standard decision trees and avoid the overfitting that they are usually prone too by introducing randomness at various levels (variable selection, node tests) during the construction of the ensemble. [2] shows that RF can act not only as a powerful bias reduction technique but can also lower the variance by increasing the number of trees: a by-product of the law of large numbers.

A random forest consists of a set of trees $\mathcal{T} = \{T_t\}$ whose outputs are aggregated into a single estimate. At each node of a tree is installed a simple binary test which defines the path taken by a given instance. The apparent simplicity

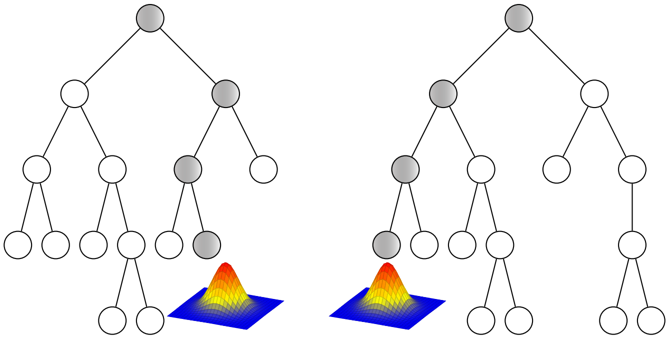


Figure 1. Hypothetical random forest showing two of its trees. The leaves each store a model for a multivariate normal distribution obtained from the patches that arrived to them during training.

of these tests is however not constraining the applicability of RF to non-linear problems. Binary tests are selected from a pool of randomly generated tests in such as a way as to maximize some measure of the quality of a split. For the proposed approach, the information gain is being used.

2.1. Training

The feature vector definition used in this paper find its origins in [11] and had been later reused in [12]. It consists of the raw pixel intensities taken in a fixed neighbourhood around salient keypoints in an image. If necessary, this definition could also be extended to multiple feature channels including the components of the keypoint descriptors themselves or values of the Laplacian of the image. Because of its fast processing, SURF [1] features were chosen (as opposed to [11, 12] but it can be easily conceived that SIFT [14] could produce comparable results.

Given the training set consisting of labeled images of heads with varying yaw and pitch angles, the SURF keypoints are first computed and an image patch is extracted around each of them. Each tree is then grown independently over a randomized subset of the extracted image patches. At each node, a large set of possible binary tests is generated. Following the binary test definition of [11, 12] or [5, 3] for 3D data, the difference (eq. 1) between the values of two randomly chosen pixels within a patch for a given channel is compared against some threshold to decide whether a test sample should go left or right in the tree.

$$C(m_1, m_2) = \begin{cases} \text{go left} & m_1 - m_2 > \tau \\ \text{go right} & \text{otherwise} \end{cases} \tag{1}$$

In order to minimize the effect of noise in the evaluation of a test, Gaussian smoothing is applied over the extracted image patches. Values for m_1, m_2 and τ are generated uniformly at random and the candidate splits that they induce

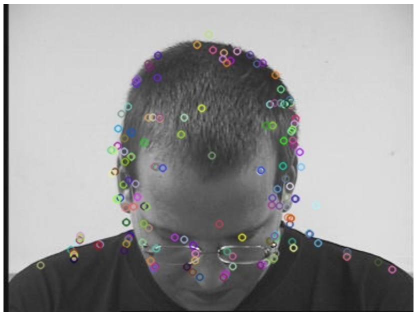


Figure 2. Example of SURF keypoints detected in an image. We take a fixed-size neighbourhood of pixels around every keypoint and let all the patches sink down to the leaves in all the trees.

are evaluated based on a measure of the information gain (eq. 2). Using the same notation as in [5]:

$$IG = H(\mathcal{P}) - (w_L H(\mathcal{P}_L) + w_R H(\mathcal{P}_{\mathcal{R}})) \tag{2}$$

We define the information gain of a binary test as the difference between the entropy of the set of patches at a parent node $\mathcal{P}$ and the sum of the entropies within each candidate partitions $\mathcal{P}_R, \mathcal{P}_L$ at the children nodes. We are thus essentially seeking for the test which reduces the most the uncertainty in the random variable θ. In other words, the confidence in θ at $\mathcal{P}_R$ and $\mathcal{P}_L$ together should be as large as possible (low entropy). Because the sizes for the candidate patches vary between the parent node and its children, we also need to account for it in the expression of the information gain. We thus define the weighting factors w_L and w_R as $w_i = \frac{|\mathcal{P}|}{|\mathcal{P}_\rangle|}$. A simple choice for modelling the representative pose for a set of patches that arrived a leaf is to evaluate their empirical mean and covariance, effectively obtaining $\theta \sim \mathcal{N}(\theta, \bar{\theta}, \Sigma)$ as a bivariate normal distribution for θ over the pitch and yaw angles. This formulation not only allows us to get a measure on the uncertainty of an estimate through the covariance matrix Σ, but also has for consequence that both the pitch and yaw angles can be evaluated with the same tree. Another option would have been to maintain two separate forests: one for each angle. The information gain of equation 2 can thus be expressed as:

$$IG = \log|\Sigma(P)| - \Sigma_{i \in L,R} w_i \log|\Sigma(P_i)| \tag{3}$$

Where Σ is the 2×2 covariance matrix over the pitch and yaw angles. Trees will be grown until either the set of samples remaining at a node is too small or if the maximum depth as been reached. When a leaf is created, the empirical mean and covariance of the samples that it received is computed for latter retrieval during testing.

2.2. Testing

From the set of trees obtained during training, the pose estimation procedure consists in dropping every image patch extracted from the test image one by one into each tree. The binary tests installed at the nodes will guide the samples down a branch to a leaf where the parameters for the stored bivariate normal distribution will be fetched.

A single pose estimate for the image patch is then obtained by taking the average over the estimates from each leaf. In order to eliminate most some of the noisy estimates, a threshold on the maximum value for the determinant of Σ can be set.

From the set of all estimates from the extracted image patches, a final value is obtained after once again taking the average over all estimates. Possible improvements for better detecting outliers are discussed further in this paper.

3. Experiments

In order to assess the accuracy of our RF algorithm in the presence of uninformative image patches (such as those found in the image background or around the shoulders) we used two different approaches: one which uses all the SURF keypoints in the image and the other which only keeps those within the face region.

The Pointing'04 head pose database [8] was used for this experiment. It consists of 2790 labeled images of 15 different people with varying pose in the pitch and yaw angles. Two series of 93 images for each person is provided and includes those subjects wearing glasses. Even though the scene parameters were controlled precisely (as opposed to an unconstrained database such as in [4] for example) we do not think that it would affect negatively the generalization capabilities of the head pose estimator. This assumption can be motivated by the fact our keypoint-based approach offers more robustness in case of occlusion and partial information can be recovered from the remaining observable features.

3.1. Using Raw Images

The number of SURF keypoints was not a fixed parameter across images but on average, about two hundred of them were found for every training instance. From the raw Pointing'04 database, 511104 image patches were extracted around the SURF keypoints resulting in about 400Mb of data. Because of the lack of suitable software packages for RF in a regression setting, we had to write our own implementation and optimize it according to this magnitude of data. We decided to base our implementation on some of the C++ code developed in [7] which also used Random Regression Forests but coupled with some object detection algorithm. This also allowed us to implement binary tests of the form described in equation 1. This is a distinctive characteristic of this algorithm with respect to the original RF formulation of [2].

Considering the scale of the images in our database, we have decided to set a window of 16×16 pixels around every keypoint. Prior to patch extraction, the image is also convolved with a Gaussian filter to diminish the effect of noise for the pixel-based binary tests. Because of memory usage and training time concerns, the maximum tree depth was set to 15. Similarly, it was deemed necessary to impose the minimum number of samples at a node to be at least 5. Practically, it also turned out that values lower than this would often lead to numerical issues when computing the information gain.

An important parameter of the RF is the size of the random tests pool generated at each node. We have decided to retain a similar methodology as in [7] for test generation and selection. The pixel locations m_1 and m_2 are first selected at random within the window and the difference between the pixel intensities at these locations is then computed for every image patch being currently evaluated at a node. The minimum and maximum values of those pixel differences are then computed so that we can restrict the range of possible values chosen at random for the threshold τ. For this experiment, we used 100 iterations for choosing different locations and 10 values for τ at each of them, producing a set of random tests of size 1000. We note here that the authors of [7] chose to set the number of possible tests as the number of training instances available at the root, which in our case corresponds to 511104. Such a setting leads to extremely slow training and is unnecessary large for our problem.

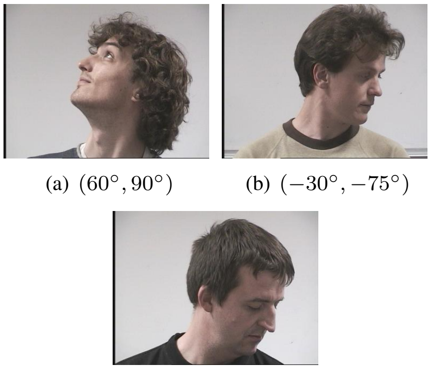
Under the conditions described previously, we have been able to train 15 trees on all the extracted image patches within about 30 mins on a quad-core Intel Xeon @ 2.66 Ghz. For the sake of comparison, we have also tried to train 15 trees with 1000 random tests at each node on the complete training set using the `randomForest` software package for R [13]. However, it seems that the current implementation prohibits memory usage of the magnitude dealt with in this paper. We thus had to reduce the number of training instances down to 25000.

Since neither of our own implementation or `randomForest` [13] led to satisfactory results under the settings described above, we implemented face detection as a pre-processing step. We described this experiment in the next section.

3.2. With Pre-Processing for Face Detection

Expecting similar results as in [17] where SIFT features are extracted directly from the raw image without any pre-filtering, we discovered that the proposed algorithm would benefit from a pre-processing phase that would extract the head contained in an image. This way, the SURF keypoints computed within this region of interest might reflect more of the statistical regularities that we seek to capture with the random forest.

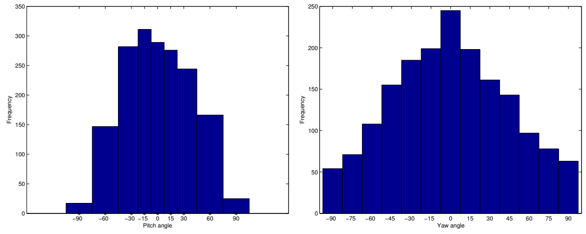
We made use of the Viola-Jones cascade classifier to detect faces because of its wide availability in the OpenCV



(a) $(60°, 90°)$ (b) $(-30°, -75°)$

(c) $(30°, 30°)$

Figure 3. Some of the training instances that the face detector failed to identify. The built-in Viola-Jones [19] classifier from OpenCV was used for this task.



(a) Pitch (b) Yaw

Figure 4. Distribution over the pitch and yaw angles for the heads detected by the Viola-Jones classifier.

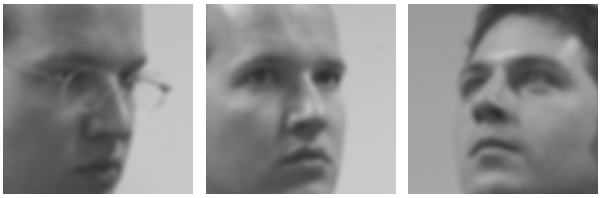
library. However, a major limitation of this built-in classifier is in its specialization for frontal faces. A constraint which is not without consequences on the quality of the resulting dataset. Figure 3 shows examples of faces which could not be detected by the classifier that we used. We can also see in 4 that the recognition rate for the Viola-Jones classifier drops as the pitch and yaw angles are approaching 90°. Clearly, frontal poses are over-represented in the training set and can only induce bias in the accuracy of the regression.

As opposed to the training dataset obtained directly from the images in the database, the number of training instances goes does down drastically from 511104 to 67214 image patches: a drop which is quite noticeable as well in terms of training time.

4. Results

Parameters	Pitch (MAE)	Yaw (MAE)
20 trees, 85 tests per node	9.3270	11.5815
100 trees, 85 tests per node	9.3268	11.6253

Table 1. Mean Absolute Error (MAE) on a subset of the Pointing'04 database where the face can be detected by a Viola-Jones face detector. The distribution of the detected head poses is shown in figure 4.



(a) $(-14°, -41°)$ (b) $(11°, -14°)$ (c) $(19°, 6°)$

Figure 5. Estimated head poses. The ground truth for the above images is: $(-15°, -45°)$, $(15°, -15°)$ and $(30°, 30°)$.

By extracting image patches around SURF keypoints located in the face region, we have been able to obtain mean absolute error (MAE) measures (table 1) which are comparable to those reported in [12]. The accuracy for the case where keypoints were taken indistinctly over the image was so low that we decided not to include them here. Based on the MAE results (obtained from the RF implementation of [13]), it seems that increasing the number of trees (at the expense of training and testing time) from 20 to 100 does not lead to an appreciable decrease in the error.

5. Discussion and Future Work

If the regression accuracy reported here compares well against [12], a legitimate critique could be formulated about the omission of proper cross-validation. Furthermore the estimation results of table 1 are for only 1003 images from 2790 in the database and correspond to those for which the face detection algorithm succeeded on. Therefore, no only did our method induced bias during training as shown in figure 4 but it also had a similar effect during testing. It is thus very likely that the results reported in [12] could not be compared directly against ours if in their case the face detection succeeded on every instances of the head database (something which is not stated clearly).

In designing this algorithm, we have tried to use theoretical tools that could make it more robust in unconstrained environments. However, this robustness remains to be determined on a suitable image database such as the one collected in [4]. Since for this paper we were only interested in the problem of head pose estimation, detection capabilities of the face classifier still remains the main limiting factor in the overall system. An interesting extension to this work might be to set both problems of head detection and pose estimation together in same RF formulation as in [6].

From the results of [4] and [15], there are good reasons to believe that the Viola-Jones face classifier used for this experiment could be easily outperformed by the OCI (Object Class Invariant) face detector used in the aforementioned papers. In this framework, the geometrical relationship between a set of scale-invariant features (such as SIFT [14]) with respect to an *invariant* of the image (a line segment on the nose) is inferred from their statistical occurrence in natural images. With every keypoints in the OCI model comes a specification of its location, scale and orientation in the image as well as some description of the pixel intensities in the local neighbourhood. The fact this algorithm seeks to relate a given keypoint to the nose location has several useful implications. One of them is that informative keypoints should all agree on the nose location (our OCI): an idea shared with [5] but used differently. Based on a measure of mutual information [15] can also reduce redundancy among features and only retain the most informative with respect to a given object class. By taking into account the spatial likelihood of keypoints, OCI also distinguishes itself from the class of spatially unstructured object detectors: a constraint that [17] also leverages. Instead of extracting patches around every SURF keypoints, we could thus make use of OCI to detect and localize faces while only retaining those features which are the most informative. The random regression forest algorithm described in this paper could then be used in the same way over these filtered keypoints.

In the manifold embedding technique of [17] the 128-d SURF descriptors are directly included in the definition of the features. It might be interesting to see if such information could be beneficial to our application. Similarly, the appearance component of the OCI description of a keypoint could also contribute to the feature definition. This question of feature selection also comes with the one of test selection. Is the form of the binary test in 1 appropriate for our task ? Could we reformulate it more generally in a way which is closer to [2] ?

The OCI approach of [4] to head pose estimation is built upon a set of tools suitable for uncontrolled environments. However, the authors report some weaknesses when the algorithm tries to estimate poses as more of the anatomical regions become unobservable. It would be interesting to evaluate how an hybrid OCI+RF approach would compare under these conditions.

SURF detection already provides some degree of robustness against variation in scale. However the algorithm that we propose here uses a fixe size neighbourhood around keypoints. In an unconstrained environment we would thus have to either rescale the detected face area or take into account the scale of the observed keypoints in a more principled way. The scale component in the OCI definition of a keypoint might be used for that.

6. Conclusion

We have presented an algorithm for estimating head poses from 2D images using random regression forests. We made use of the computational efficient SURF [1] features because their scale-invariance property suitable for the class of applications meant to operate in unconstrained environments. By using the regression formulation of Random Forests [2], we have been able to estimate the head pose in the pitch and yaw angles for images from the Pointing'04 [8] head database. Since our approach does not offer a solution to head detection, we used a Viola-Jones [19] face detector as a pre-processing step for extracting fixed-size image patches around each of the SURF keypoints. This face detection and localization step had a great positive impact on the quality of our results. However, since the current implementation of our face detector handles only frontal faces, it is a limiting factor in its applicability to unconstrained environments. A proposal for a more robust system using the Object Class Invariant (OCI) framework [15] has been presented and might constitute future work. Our choice of using Random Forests for regression has the advantage of being parallelizable very naturally and makes our algorithm already suitable in a realtime context.

7. Code availability

The implementation of the algorithm described in this paper is available online at `http://github.com/pierrelux/rf_pose` under the same license as [7].

References

[1] H. Bay, E. Andreas, T. Tinne, and G. Luc J. Van. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(September):346–359, 2008.

[2] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. 10.1023/A:1010933404324.

[3] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. In *Proceedings of the 2010 international MICCAI conference on Medical computer vision: recognition techniques and applications in medical imaging*, MCV'10, pages 106–117, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] M. Demirkus, B. Oreshkin, J. J. Clark, and T. Arbel. Spatial and probabilistic codebook template based head pose estimation from unconstrained environments. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 573 –576, 2011.

[5] G. Fanelli, J. Gall, and L. Van Gool. Real-Time Head Pose Estimation Using Random Regression Forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624, 2011.

[6] G. Fanelli, T. Weise, J. Gall, and L. V. Gool. Real Time Head Pose Estimation from Consumer Depth Cameras. In R. Mester and M. Felsberg, editors, *Pattern Recognition*, volume 6835 of *Lecture Notes in Computer Science*, pages 101–110. Springer, 2011.

[7] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *In Proceedings IEEE Conference Computer Vision and Pattern Recognition*, 2009.

[8] N. Gourier and J. L. Crowley. Estimating Face Orientation from Robust Detection of Salient Facial Features. In *Proceedings of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures*, 2004.

[9] G. Guo, Y. Fu, and C. Dyer. Head pose estimation: Classification or regression? *Pattern Recognition, 2008.*, 2008.

[10] C. Huang, X. Ding, and C. Fang. Head Pose Estimation Based on Random Forests for Multiclass Classification. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 934–937. Ieee, Aug. 2010.

[11] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1465 –1479, 2006.

[12] Y. Li, S. Wang, and X. Ding. Person-independent head pose estimation based on random forest regression. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1521–1524, 2010.

[13] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, Nov. 2004.

[15] M. Toews and T. Arbel. Detection over viewpoint via the object class invariant. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 765–768. Ieee, 2006.

[16] M. Toews and T. Arbel. Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1567–81, Sept. 2009.

[17] M. Torki and A. Elgammal. Regression from Local Features for Viewpoint and Pose Estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2603 –2610, 2011.

[18] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE Comput. Sco. Press, 1991.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 – I–518 vol.1, 2001.

[20] Y. Wu and K. Toyama. Wide-range, person-and illumination-insensitive head orientation estimation. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 183–188, 2000.