

3D Facial Pose Estimation by Image Retrieval

Nemanja Grujić, Slobodan Ilić
Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{Nemanja.Grujic,Slobodan.Ilic}@tu-berlin.de

Vincent Lepetit, Pascal Fua
CVLAB, EPFL
CH-1015 Lausanne, Switzerland
{Vincent.Lepetit,Pascal.Fua}@epfl.ch

Abstract

We propose an approach to 3D facial pose estimation that, unlike most state-of-the-art techniques, can handle arbitrary poses, including extreme out-of-plane rotations, background clutter and facial expressions.

It relies on a large database of registered face images of different people viewed from several perspectives. We use a powerful image retrieval technique to match the input image against database ones, which returns the most similar 3D pose. This 3D pose can then be refined using matches between input image and database images.

We will show qualitative and quantitative results using images of people who do not appear in image database.

1. Introduction

While it is crucial for the automation of many applications such as 3D face tracking, Human-Machine interfaces, Driver Attention Monitoring, or non-frontal Facial Recognition, estimating an accurate 3D pose for a human face in the absence of prior knowledge remains largely unexplored. Available methods rely either on the face appearance or on the facial features. The face appearance methods are usually based on face detection or on statistical appearance models. Frontal face detection algorithms provide only the 2D position and scale of frontal faces [16, 19, 25, 27], while multi-view face detection algorithms provide a qualitative 3D pose [20, 22, 28] such as “frontal” or “profile,” which is not sufficient for these applications that require the six degrees-of-freedom pose. Active appearance models [2, 6, 26] align generic statistical face model when close to the given position, what makes them more suitable for tracking than direct pose detection. Feature based methods [9, 15, 17, 23] look for a facial landmarks, i.e characteristic key-points in the image and align labeled face models to them. The main problem is to detect a head pose invariant facial key-points.

We focus here on the 3D pose estimation of human faces in monocular images, where the lighting environment is un-

known and the background may be cluttered. Because it is sufficient for many applications, we restrict ourselves to the case where at most one face is present but do not require to know its owner’s identity beforehand. As depicted by Fig. 1, this is very challenging because the appearance of human faces may vary drastically under such general conditions. So far, no satisfactory solution has been proposed and that is what our technique aims for. Our approach starts with a large labeled database [17] of different people viewed from several perspectives and under identical lighting conditions. To introduce 3D face pose information we register a generic 3-D face model to each of database images. We use powerful image retrieval technique based on vocabulary tree [18] to match the input image—the one of the novel face image which may deform due to facial expressions—against database ones. This gives us images in similar 3D poses to the face in the input image. Since the database images are registered, we refine the 3D pose by matching feature points detected on the face between the input image and the retrieved ones. Note that the feature points are not facial landmarks, but rather generic interest points such as corners [13] or regions [14].

Fig. 1 shows examples where we handle a wide range of poses, even extreme out of plane rotations. Furthermore, we performed quantitative tests that showed an average angular deviation of 13.7° from the labels attached to the ground truth. Given that those labels are quantized in 15° increments and therefore not particularly accurate, this is as much as can be expected from such data. Furthermore, a 15° angular error is hardly noticeable by a human observer.

In the remainder of the paper, we first discuss the state-of-the-art methods for face pose estimation. Then we describe the basic idea behind vocabulary tree followed by the description of our approach to face pose estimation by image retrieval and 3D pose refinement. Finally, we present the results and conclude.

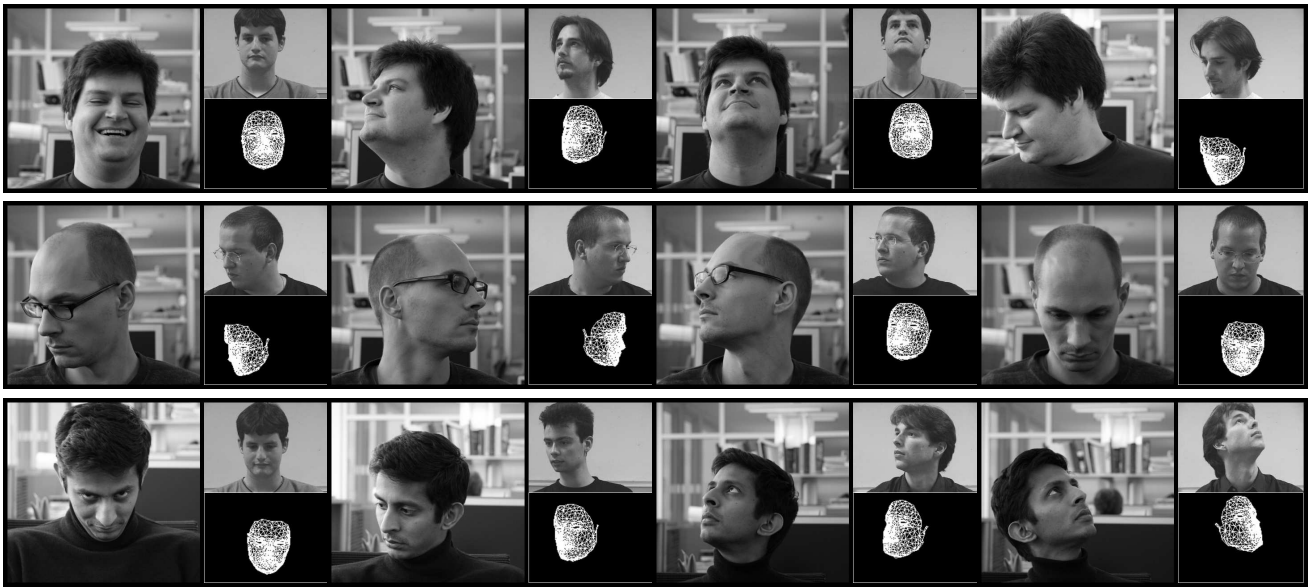


Figure 1. The examples of face pose estimation showing that wide range of poses, including those of extreme out of plane rotations, can be handled. The left image corresponds to the input query image, while on the right is a database image in the most similar pose and the accompanying 3D model.

2. Related work

Face pose estimation has often been formulated as a face detection problem. Modern techniques rely on face/non-face classifiers trained on large sets of features some of which are face specific. There are many frontal face detection algorithms [16, 19, 25, 27] that work remarkably well under real world conditions and in real time. By contrast, current detectors designed to work for arbitrary head poses [8, 12, 20, 22, 29] are more brittle. They can easily fail not only when the faces are strongly tilted due to up-down head rotation or large out-of-plane rotations, but also when the lighting conditions are difficult, the face deforms due to facial expressions, and when the background is cluttered. Furthermore, they do not usually yield an accurate 3D pose as we do but rather a rough estimate based on the poses of the images they have been trained with.

More specifically, frontal face detectors such as those based on neural networks [20] or cascade classifiers [27] have been extended to multi-view face detection [19, 28]. The first of these extensions [19] is only invariant to in plane face rotations while the second [28] can also handle out of plane rotations and is based on classifiers trained for each pose. The in-plane rotations are broken into 12 different rotation classes and out of plane rotations are classified as “left” or “right,” which is relatively coarse. More recently, efficient real-time implementations [4, 12] based of FloatBoost and Vector boosting algorithms have relied on a coarse-to-fine strategy. First, the out-of-plane rotation is detected at the top level, which has been trained on a range

of examples covering the whole rotation space from -90° to 90° degrees. Then, the pose is refined using an additional set of orientation-specific classifiers.

The alternative approaches for 2D and 3D face pose estimation are using generic face models and facial landmarks or the combination of two. Active Appearance Models(AAM) [2, 6, 26] are used to align the model shape and texture to the face. The assumption is that the model has to be initialized close to the input face position. These approaches are more suitable for tracking when the current face pose is not far from the one in the previous frame. Instead relying on the model texture Lie and Kanade [10] build their model around sparse 3D points and the view-based patches associated with every point. This model is more generic and does not make assumptions on illumination, thus is very robust in different lightening conditions. However, it also starts in the proximity of the given pose which facilitates face landmark detection. Feature based methods [9, 15, 17, 23] look for a facial landmarks, i.e characteristic key-points in the image and align labeled face models to them. Some of them use oriented Gabor filters to detect facial features. The main problem is that they are not head pose invariant so they are learned for a particular pose using, for example, principle component analysis(PCA).

The methods discussed above provide either only a 2D location or a qualitative 3D pose expressed in terms of two rotation angles identical to those of the training images and no 3D location. Furthermore, the extreme head rotations and the full range possible poses are never handled. By contrast, combining a powerful image retrieval technique and

an image database containing many registered poses yields 3D estimates that are sufficiently good to be further refined into a precise 3D pose.

We propose an entirely different approach to 3D face pose estimation that is based on fast image retrieval technique. Unlike the image retrieval methods [5, 18, 24] where the object in the input image is contained into the database, sometimes recorded from different perspective and lightening, our input face image is not contained in the image face database. We developed a specific method which rather retrieves the face images in the most similar pose. Having associated a generic 3D face model to each face image of the database and imposing geometric constrains between input image and face images in retrieved poses, we can estimate the actual 3D head pose.

3. Pose Estimation by Image Retrieval

We use a large database of registered face images where each image has associate pose label. Our approach to 3D facial pose estimation involves retrieval of an approximate pose followed by refinement of that pose. We start by using feature points to match the input image against a database of registered images to retrieve an ordered set of possible pose labels. We then refine them by selecting an image that is most geometrically consistent with the input image in terms of matches between facial features. To increase our algorithm’s performance when the background is cluttered, we optionally perform a preprocessing step that suppresses background features by using a simple classifier to distinguish facial from non facial feature points. In the remainder of this section, we discuss these steps individually.

3.1. Pose Retrieval

For our experiments, we use the INRIA head pose database [17], which includes 2790 images from 15 people. Poses include up-down head rotations with vertical angles values -90, -60, -30, -15, 0, +15, +30, +60, and +90 degrees and out-of-plane head rotations with horizontal angles -90, -75, -60, -45, -30, -15, 0, +15, +30, +45, +60, +75, and +90 degrees. This results in 93 poses per person, each of whom was asked to look at markers on the wall in a calibrated setting and photographed twice. Those wearing glasses were photographed once with them and once without.

Given an input image, our approach to finding the closest pose in that database is inspired by a *vocabulary tree* technique [18] that we briefly review below in its original form before describing how we customized it for our purposes.

3.1.1 Vocabulary Tree for Image Retrieval

The method of [18] consists of three steps: Building a tree structure called a vocabulary tree, inserting images in the database, and querying.

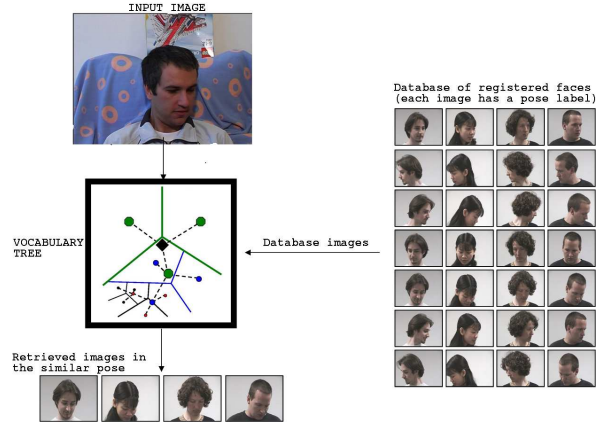


Figure 2. The process of pose estimation using image retrieval. The part of the figure showing vocabulary tree is courtesy of [18] and the database images are courtesy of [17].

The vocabulary tree is an efficient structure that makes image retrieval very fast. It is built in the process of unsupervised training by clustering the training set of feature descriptors using hierarchical k -means, where k is also the branching factor of the tree. Initially, a k -means is applied to training data producing k clusters of feature descriptors. The process is recursively applied to each cluster, thus further dividing the feature descriptor space in finer and finer clusters. The clustering stops when the maximal level of the tree L is reached or when the training data is exhausted.

Inserting an image in the database mainly means computing a score vector \mathbf{d} for the image based on the vocabulary tree. The dimension of \mathbf{d} is the number of nodes in the vocabulary tree, and each coordinate $\mathbf{d}(i)$ is computed as $\mathbf{d}(i) = n_i w_i$, where n_i is the number of features that belong to the cluster corresponding to node i , and w_i is a weight computed to give more importance to more discriminant nodes. Weight is computed as $w_i = \ln \frac{N}{N_i}$ where N is the total number of images in the database, and N_i is the number of images in the database with at least one feature point belonging to node i 's cluster, corresponding to TF-IDF [21] from information retrieval.

Finally, querying is performed by computing a similar vector score \mathbf{q} for the query image, and searching for nearest neighbors in the image database based on the L_1 -norm of the normalized score vectors

$$D(\mathbf{q}, \mathbf{d}) = \left\| \frac{\mathbf{q}}{\|\mathbf{q}\|_1} - \frac{\mathbf{d}}{\|\mathbf{d}\|_1} \right\|_1. \quad (1)$$

The nearest neighbors are the database images most likely to represent the same object as the query image.

3.1.2 Vocabulary Tree for Pose Estimation

As shown in Fig. 2, given an input image containing a human face, we use the image retrieval technique described

above to find the database images being in similar pose to the query image. We use MSER regions [14] as image features and compute SIFT descriptors [13] for them. Since our goal is pose estimation, we removed the invariance of SIFT to rotation by skipping the canonical orientation estimation. We tried three different strategies to find the pose p of the query image using the closest database images and their associated poses.

Let \mathbf{q} be the query image score vector, \mathbf{d}_j^i be the score vector for the image of the j^{th} person in the i^{th} pose, and D be the distance measure as defined in Eq. (1). We tried following scoring techniques:

- *Best image first.* We simply take p to be the pose of the closest database image. We write:

$$p = \operatorname{argmin}_{i=1..N_{ps}} \min_{j=1..N_{pr}} D(\mathbf{q}, \mathbf{d}_j^i),$$

where N_{pr} is the number of different persons in the database and N_{ps} is the number of poses.

- *Post accumulation simple voting.* For each possible pose i , the distances of all images in that pose are summed. The pose p with the smallest sum is adopted as the pose estimate, and we write:

$$p = \operatorname{argmin}_{i=1..N_{ps}} \sum_{j=1..N_{pr}} D(\mathbf{q}, \mathbf{d}_j^i).$$

- *Post accumulation cross voting.* For each possible pose i , the distances of all images are summed and weighted according to the similarity of their pose and pose i . The pose i with the smallest sum is adopted as the pose estimate:

$$p = \operatorname{argmin}_{i=1..N_{ps}} \sum_{j=1..N_{pr}} \sum_{k=1..N_{ps}} D(\mathbf{q}, \mathbf{d}_j^k) \cdot W(i, k),$$

where the weight $W(i, k)$ is the distance measure between poses i and k , and must have lower values for more similar poses. In practice we simply use the dot product between the orientation vectors.

In practice, the output is not a single pose computed above, but the ordered set of poses with the smallest sums of distance measures D . The three methods can be implemented very efficiently thanks to the vocabulary tree, and will be compared in Section 4.

3.2. Background Noise Suppression

If we simply applied the method described above on an arbitrary image, a cluttered background could produce many spurious features and degrade the performances of image retrieval and, thus, of the pose estimation. We remove some of the background by cropping the largest skin

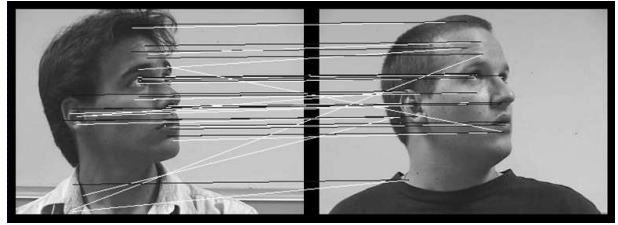


Figure 3. Feature matching. The white lines are matches before geometric constraints and RANSAC, while the black lines are matches after RANSAC. Note that the number of outliers is efficiently suppressed.

color region and using it as a query image. This helps but does not suffice in general because spurious features may remain, especially if the background contains objects whose color is close to skin color.

As a preprocessing step, we therefore aim to classify features in the query image as background or face features. Only the face features will then be used to compute the query image vector \mathbf{q} . Many classification algorithms could be used, such as K-Nearest Neighbor [11], Support Vector Machines [3] or Neural Networks [1]. However we found that the vocabulary tree itself could be used for this purpose.

Once it has been built, we consider two sets of training images, a set of face images and a set of background images in which no faces are present. The features extracted from these images are dropped down the tree to estimate for each leaf L_i the posterior probabilities

$$\begin{aligned} P_{\text{face}}(L_i) &= P(\mathbf{k} \text{ falls in } L_i \mid \mathbf{k} \text{ lies on face}) \text{ and} \\ P_{\text{bg}}(L_i) &= P(\mathbf{k} \text{ falls in } L_i \mid \mathbf{k} \text{ lies on background}) \end{aligned}$$

where \mathbf{k} is an image feature. At run-time, a feature \mathbf{k} from the query image is classified by comparing the posterior probabilities for the leaf it falls into. If $P_{\text{bg}}(L) > P_{\text{face}}(L)$, where L is the leaf \mathbf{k} falls into, we treat it as a background feature and do not use it to compute the query image vector \mathbf{q} .

3.3. Pose Refinement

The output of the image retrieval is an ordered list of best poses. For each pose there is a number of registered images corresponding to different people in the database. However, we still do not know where the face is in the image and which person corresponds the best to the person in the input image. Moreover, some retrieved images may be outliers. We therefore enforce global geometric constraints between the input image and the retrieved ones to obtain the complete face pose despite outliers.

To do that, we simply match features between the retrieved images and the input image as shown on Fig. 3, and robustly compute a global 2D motion from the matches using RANSAC [7]. Since the two images are already very

similar up to a translation, we look for a motion as a composition between a 2D translation and scales along the image axes. We repeat this process for each of the retrieved images. The image with the most inliers is the image of a person in the pose being the most geometrically consistent with the input image.

4. Results

In this section we present quantitative and qualitative results using input images both from the INRIA Pointing database [17] and from other sources.

4.1. Quantitative Experiments

Here we quantify the influence of some of our implementation choices, such as the tree depth and the choice of scoring technique, as discussed in Section 3.1.2. We also demonstrate the improvement that pose refinement brings about.

To this end, we used 1860 out of the 2790 images of our database—corresponding to 10 people in 93 poses photographed twice—for training purposes and the remaining 930 for testing purposes. Recall from Section 3.1 that by construction of the database a pose label is associated to each image [17]. However, because people position themselves differently when observing the wall markers, the pose is not always particularly accurate. This should be kept in mind when assessing the deviations of our algorithm with respect to this “ground truth.”

In Fig. 4 we show the results of pose estimation on some of the test images we used. When our algorithm matches an input image to a database one, we visualize the associated pose by projecting a 3D face mask in that pose. In Fig. 5, we plot the corresponding distribution of pose estimation errors for various settings. Fig. 5(a) shows that the *post accumulation cross voting* scoring technique of Section 3.1.2 brings a moderate improvement over *best score first*, but that the best actually is *post accumulation simple voting*. These results were obtained using vocabulary trees of depth 4. Fig. 5(b) demonstrates that increasing the depth to 5 and 6 further improves the performance. Unfortunately, increasing further would require a training database larger than the one we have. In Fig. 5(c), we show that the pose refinement of Section 3.3 does indeed bring a substantial improvement. The mean error of 20° after image retrieval has decreased to 13.7° after pose refinement, which indicates a good precision since the database orientations are quantized in 15° increments and are not particularly accurate, as discussed above. Similarly, the number of frames where the error is smaller than 30° goes from 75% to 95%, meaning that the number of gross errors has been significantly reduced.

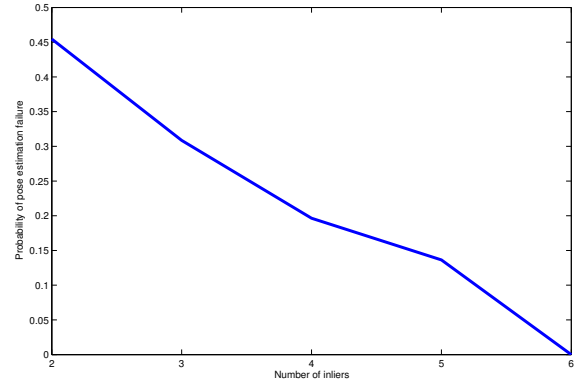


Figure 8. Probability of the pose estimation failure in respect to the number of inliers for the video of Fig. 6. The ground truth is obtained by visual inspection.

4.2. Qualitative Experiments

The database images were all acquired against uniform backgrounds and under the same lighting conditions. To show that our approach works under more realistic conditions, we acquired a video of a subject moving his head and changing his facial expression against a cluttered background with uncontrolled lighting. The subject is of course not one of the people who were photographed to build the database. In Fig. 6, we show some of the more extreme positions he reached. For each one, we show the video sequence frame we used as a query image, the database image that is most geometrically consistent with it, and a 3D wireframe mask projected in the recovered pose. Note that we do not exploit temporal consistency and treat each frame independently.

In the top row of Fig. 7, we show some algorithm failures. They tend to occur when the number of inliers after pose refinement is rather small, which means that the retrieval phase was already giving mainly wrong poses. We observed that this happens when the overall number of face features decreases, mainly because of lightening changes, shadows or motion blur causing bad pose retrieval. To verify this claim we computed the probability of the pose estimation failure in respect to the number of inliers as shown in Fig. 8. To remedy this problem we imposed a temporal consistency to the independent pose estimations. This results in smooth transitions between the detected poses as shown in the supplementary video.

5. Conclusion

We proposed a novel approach to 3D face pose estimation. Unlike many state-of-the-art approaches that rely on multi-view face detection and only provide qualitative 2D poses, our returns true 3D pose estimates. To this end, we



Figure 4. Results of pose estimation algorithm for the INRIA database where 930 images of 5 people filmed twice in 93 poses were used for testing and 1860 image of 10 people filmed twice, also in 93 poses, are used for training. On the left of each image is the query image. On the up-right is the selected person from a database which have the most similar pose as the one of query image. On the bottom-right the recovered 3D mesh model indicating the global face positions is depicted.

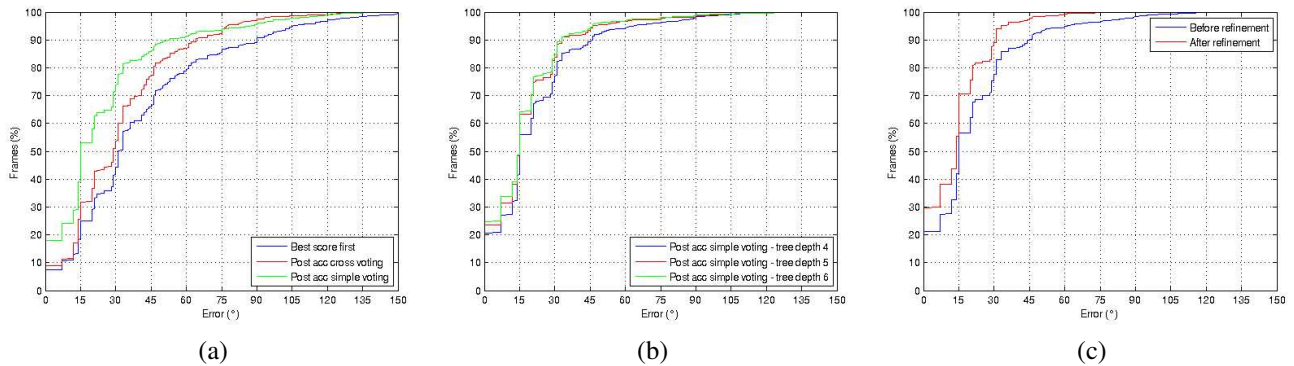


Figure 5. Accuracy of pose estimation for various implementation choices. In each graph, the x axis represents deviations between true and estimated poses in degrees and the y axis the percentage of poses for which the deviation is below a given value. (a) For initial pose retrieval, post accumulation simple voting gives better results than the two other scoring techniques. (b) Increasing the depth of the vocabulary tree from 4 to 6 improves the accuracy, but improving it even further would require a larger training database than the one we have. (c) Pose refinement brings a clear improvement in accuracy.

treat the image in which we are trying to compute the pose as a query image and search for the most similar one in a database of registered faces. This lets us take advantage of a powerful image retrieval technique and assign to the query

image the 3D pose of the image it matches.

In quantitative tests, we achieved an average angular deviation of 13.7° from the labels attached to the test images we used to evaluate our approach. This represents a good



Figure 6. Pose estimation against a cluttered background, arbitrary lighting and facial expressions. Each image includes, on the left a frame of a video sequence used as a query image and, on the right, the database image that matches it best and a 3D wireframe mask projected in the recovered pose.



Figure 7. Failures due to small number of inliers.

result given that these labels are not particularly accurate themselves and that a 15° angular error is barely noticeable by a human observer. Additional qualitative tests demonstrated robustness to extreme out-of-plane rotations, cluttered backgrounds, and facial expressions.

The current implementation treats each frame independently and can fail when the appearance of the face becomes such that the number of inliers is rather small to give good pose estimation. This could be remedied by imposing a temporal consistency as shown in supplementary video.

References

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.
- [2] V. Blanz and T. Vetter. A Morphable Model for The Synthesis of 3-D Faces. In *Computer Graphics, SIGGRAPH Proceedings*, pages 187–194, Los Angeles, CA, August 1999.
- [3] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [4] Y. L. Chang Huang, Haizhou Ai and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *International Conference on Computer Vision*, 2005.
- [5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.
- [6] F. Dornaika and J. Ahlberg. Fast and reliable active appearance model search for 3-d face tracking. *SMC-B*, 34(4):1838–1853, August 2004.
- [7] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications ACM*, 24(6):381–395, 1981.
- [8] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1408–1423, 2004.
- [9] A. H. Gee and R. Cipolla. Determining the gaze of face in images. Technical Report CUED/F-INFENG/TR 174, Trumpington Street, Cambridge CB2 1PZ, England, 1994.
- [10] L. Gu and T. Kanade. 3d alignment of face in a single image. In *CVPR (1)*, pages 1305–1312, 2006.

- [11] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–2108, 1979.
- [12] S. Z. Li and Z. Zhang. FloatBoost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9), 2004.
- [13] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.
- [14] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *British Machine Vision Conference*, pages 384–393, London, UK, September 2002.
- [15] S. McKenna and S. Gong. Real-time face pose estimation. *Real Time Imaging*, 4(5):333–347, October 1998.
- [16] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. In S. Nayar and T. Poggio, editors, *Early Visual Learning*, Oxford University Press, 1996, chapter 5, pages 99–130. Oxford University Press, 1996.
- [17] J. L. C. N. Gourier, D. Hall. Estimating face orientation from robust detection of salient facial features. In *Proceedings of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures*, Cambridge, UK, 2004.
- [18] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.
- [19] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Proceedings of the 1996 DARPA Image Understanding Workshop*, February 1996.
- [20] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Conference on Computer Vision and Pattern Recognition*, 1998.
- [21] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.
- [22] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 2002.
- [23] J. Sherrah and S. Gong. Fusion of 2d face alignment and 3d head pose estimation for robust and real-time performance. In *Proceedings of IEEE International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, September 1999.
- [24] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.
- [25] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.
- [26] G. E. T.F. Cootes and C. Taylor. Active Appearance Models. In *European Conference on Computer Vision*, pages 484–498, Freiburg, Germany, June 1998.
- [27] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.
- [28] P. Viola and M. Jones. Fast Multi-view Face Detection. In *Conference on Computer Vision and Pattern Recognition*, 2003.
- [29] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *International Conference on Computer Vision*, pages 734–741, 2003.