Head Pose Estimation in the Wild using Approximate View Manifolds

Kalaivani Sundararajan
University of Florida
Gainesville, FL, USA
kalaivani.s@ufl.edu

Damon L. Woodard
University of Florida
Gainesville, FL, USA
dwoodard@ufl.edu

Abstract

In this paper, we present a head pose estimation method for unconstrained images using feature-based manifold embedding. The main challenge of manifold embedding methods is to learn a similarity kernel that is reflective of variations only due to head pose and ignore other sources of variation. To address this challenge, we have used the feature correspondences of identity-invariant Geometric Blur features to learn a similarity kernel. To speed up the computation of the similarity kernel, we have used spatial pyramidal matching to approximate feature correspondences and random subsampling of training samples to approximate graph neighborhood. In addition to these approximations, we have used the Nyström approximation to embed out-of-sample test images in an efficient manner. Using these approximations, an approximate view manifold was learned for 14000 images in the Annotated Facial Landmarks in the Wild (AFLW) dataset. With the learned manifold, head pose estimation was performed on four in-the-wild face datasets - AFLW (remaining 7000 images), AFW, McGill and YouTube Faces. The Approximate View Manifold training achieves a 7X speedup compared to the non-approximated Learning-manifold-in-the-wild approach [15]. Further, pose estimation using the proposed approach shows significant improvement in accuracy and reduced Mean Angular Error(MAE) compared to other methods [36, 1, 29] on the challenging AFLW (7041 images), McGill (6833 images) and YouTube Faces (22534 images) datasets.

1. Introduction

Face processing algorithms constitute a significant portion of visual understanding in social media and deals with abundant unconstrained face images. Head pose estimation, being a fundamental face processing algorithm, is no exception and needs to generalize well for unconstrained scenarios. Head pose is considered an important social clue since it indicates a target object or location of interest. Head pose is used in various behavioral analytics like identifying social interactions [13, 31, 8, 22], focus of attention [2, 26, 10, 27], identifying social groups [20, 7] and identifying target of interest [21, 24, 4]. Automatic head pose estimation is also used for several other applications in biometrics, human computer interaction and robotics. Head pose estimation methods characterize head pose using three degrees of freedom - *yaw, pitch and roll*. Most automatic head pose estimation methods attempt to determine yaw, since it varies the most (-79.8°to 75.3°) [11] and frequently occurs in normal social scenarios.

Various approaches have been proposed in the literature for automatic head pose estimation [23]. One approach that has sustained research interest is the manifold embedding technique. Manifold embedding methods [25, 16, 3] attempt to project the high-dimensional face images onto a low-dimensional manifold since there are inherently only few dimensions in which head pose can vary. Manifold embedding techniques use a holistic representation of faces and are suitable for low-resolution images. One of the significant challenges in manifold embedding methods is to obtain a view manifold that models only changes due to pose and ignores other sources of image variations such as identity and lighting.

Manifold embedding techniques compute similarity of face images in high dimensions using a distance metric and attempt to preserve the distances in the low dimensional embedding as well. The similarity between two images is typically defined by the Euclidean distance between vectorized representations of aligned face images. For unconstrained scenarios, image alignment is cumbersome thereby making pose similarity estimation for faces-in-the-wild a challenging problem.

To address unconstrained scenarios in manifold embedding, we propose incorporating keypoint feature correspondences in manifold learning. It is imperative that the feature descriptors are identity invariant and robust to other noise factors. In our experiments, we found that the Geometric Blur feature descriptors [6] cater to these constraints as compared to the more discriminative SIFT descriptors.

Figure 1: Feature correspondences across various poses using SIFT and Geometric Blur correspondences. Note that Geometric Blur descriptors give meaningful correspondences for similar poses but for different people.

The feature correspondences obtained with Geometric Blur descriptors and SIFT descriptors across various poses is shown in Figure 1. It can be observed that the Geometric Blur descriptors provide meaningful correspondences across different poses and people.

Computing feature correspondences between all pairs of training images would make the similarity kernel computationally expensive even for small training sets. To address this limitation, we propose two approximations to the similarity kernel computation. First, the feature correspondences can be approximated using the Spatial pyramid match kernel [19]. Second, we sparsify the similarity kernel by using a small but different subset of randomly chosen samples for comparison with every training sample. While embedding a test sample, besides these two approximations, we use the Nyström approximation method to obtain a low-dimensional embedding. It should be noted that head pose estimation using the proposed method does not require any image alignment or localization of specific facial landmarks.

Our contributions can be summarized as follows: 1) To the best of our knowledge, this paper is the first to experimentally demonstrate the use of spatial pyramidal matching in manifold embedding framework, 2) We have shown that Geometric Blur feature descriptors can be useful for learning identity invariant similarity kernels, 3) We have used various approximation techniques to speed-up the similarity kernel computation both during manifold learning and during test sample embedding, 4) We have made the source code publicly available for research purposes. The source code for the method described in this paper can be found in $http://www.cise.ufl.edu/{\sim}kalaivan/manifold.html$.

2. Previous work

Various non-linear manifold embedding techniques like ISOMAP, LLE, Laplacian Eigenmaps or their linear equivalents have been used to learn face view manifolds [3, 16, 25, 12, 17]. Further enhancements have been proposed to make view manifolds person-independent and invariant to illumination [16, 35, 32]. Such techniques involve modeling a submanifold for every person and then unifying those submanifolds into a single global manifold that maximizes variances due to pose but minimizes variances due to other sources. However, this approach would not work for unconstrained images since it is difficult to obtain images of the same person under various pose and illumination conditions. Balasubramaniam *et al*. in [3] proposed biasing the kernel with the pose labels to learn a smooth manifold. However, they had used the perfectly aligned FacePix dataset images [3], taken under controlled settings, for their experiments. Further, all these methods use vectorized representations of images and hence are not suitable for unconstrained face imagery.

Few researchers have proposed using local keypoint feature correspondences to obtain similarity metrics for manifold embedding of unconstrained images. These techniques place emphasis on both the feature descriptor similarity and the spatial arrangement of features. Torki *et al*. in [28, 29] have proposed projecting keypoint features onto a feature embedding space such that similar feature descriptors as well as neighboring features in space are embedded closely while dissimilar feature descriptors and non-neighbors in space are embedded farther. Hausdorff distance between these embedded point sets of two images is used to compute the kernel matrix for Laplacian Eigenmaps to obtain a final image embedding. To embed new images, a regression function is learned based on the feature embedding to avoid dual out-of-sample predictions for the feature embedding space and the image embedding space. While this method proposed a novel framework for handling unconstrained imagery in manifold embedding, it is computationally intensive with dual embedding spaces and requires feature comparisons between all pairs of training images to obtain the kernel matrix. Hegde *et al*. in [15] proposed the Learning-manifolds-in-the-wild approach by using Earth Mover's distance with SIFT keypoint descriptors as a similarity metric for computing the kernel matrix. The Earth Mover's distance computes the cost required to move a certain distribution of points to another. Hence, the Learning-manifolds-in-the-wild approach considers both spatial distribution of features and feature de-

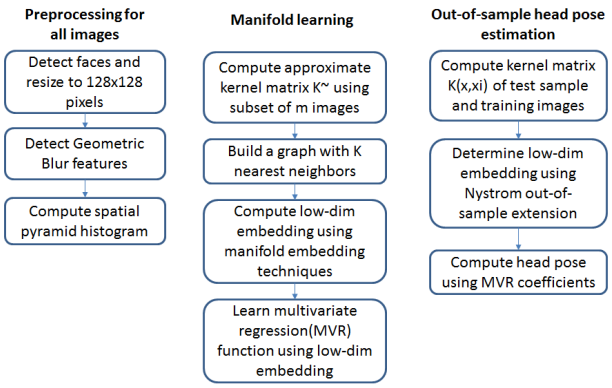


Figure 2: Overview of head pose estimation using view manifolds of unconstrained face images

scriptor similarity for manifold embedding. However, this method also suffers from the main bottleneck of computing feature correspondences for all pairs of images.

3. Proposed approach

An overview of the view manifold approach for unconstrained head pose estimation is shown in Figure 2.

3.1. Preprocessing

Face detection is performed on the training images and the detected faces are all resized to a certain size. Keypoint features are extracted from the resized images. Though any keypoint feature can be used, we found that SIFT keypoints with Geometric Blur descriptors were most effective in handling variance due to identity and geometric distortions of face images. Feature correspondences of these feature descriptors are used to obtain a similarity kernel.

3.2. Similarity kernel computation

In order to obtain feature correspondences, we use the Spatial Pyramid match kernel [19] which is an approximate correspondence technique typically used in object recognition. The Spatial pyramid match kernel takes both the feature descriptor similarity and the spatial distribution of features into account while computing feature correspondences. This suits our approach since we do not want corresponding features in two similar poses (scaled to same size) to be spaced far apart. The M cluster centers for the technique are obtained by clustering a few randomly chosen features from the training images using the K-means algorithm. This clustering simplifies the conventional $L2$-distance based feature correspondences to matching features only if they belong to the same class.

To compute the spatial pyramid histogram, an image is represented by a pyramid of $0, .., L$ levels with 2^l bins along each dimension at any level l. For each feature class m, a histogram representing the number of features that occupy each spatial pyramid bin is computed. The $(4^L * M)-$dimensional histogram of all feature classes is then normalized by the total number of features detected in an image to prevent biasing due to the number of features. The histogram representation of features thus enables the similarity measure computation to be independent of the number of features detected in each image.

Given two spatial pyramid histograms H_{X_m} and H_{Y_m} of feature class m for images X and Y, the number of feature correspondences at level l is given by the histogram intersection function [14]:

$$I_m^l = \sum_{i=1}^{4^l} min(H_{X_m}^l(i), H_{Y_m}^l(i)) \quad (1)$$

where, $H_{X_m}^l(i)$ and $H_{X_m}^l(i)$ indicate the number of features that fall into the i^{th} bin at level l for feature class m. The spatial pyramid match kernel for all L levels across all M feature classes is given by,

$$K_M^L(X,Y) = \sum_{m=1}^{M} \left[I_m^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} (I_m^l - I_m^{l+1}) \right] \quad (2)$$

where, the second term indicates the number of new matches obtained at subsequently coarser levels of the pyramid. Matching features of the same class m that fall into the same spatial bin are weighted based on the pyramidal level at which they match. The weight corresponding to each level is given by $\frac{1}{2^{L-l}}$, implying that matches at finer resolutions are of better quality.

Now, the spatial pyramid histograms of every training sample needs to be compared with those of every other training sample to obtain the similarity kernel for manifold learning. In order to speed up the kernel computation, we randomly choose a different subset of samples for comparison with every training sample. With a graph representation, this approach is equivalent to having every node connected to a random subset of nodes as compared to having a fully connected graph. The K nearest neighbors of each sample is then computed using the sparse similarity matrix which constitutes the first step for most manifold embedding techniques.

3.3. Learning Face Manifolds

We use Laplacian Eigenmaps [5] to learn the face manifold. Let $X = \{x_1, x_2, ..., x_N\}$ be N training samples such that $x_i \in R^D$. Laplacian Eigenmaps attempt to find a low-dimensional embedding, $Y = \{y_1, y_2, ..., y_N\}$, $y_i \in R^d, d \ll D$, that preserves the local distances observed in higher dimensions. Each training sample is represented by a graph node and the graph adjacency matrix

is represented using the similarity kernel. The first step for learning Laplacian Eigenmaps embedding involves building a graph where the K nearest neighbors of each node are determined using the similarity kernel.

1. Let W be the weight matrix such that W_{ij} represents a similarity measure if x_i and x_j are neighbors and W_{ij} = 0 otherwise. W is computed using the technique described in 3.2.

2. Let U represent the eigenvectors of a positive semidefinite matrix. Let D be a diagonal matrix such that $D_{ii} = \sum_j Wij$. The graph Laplacian is given by $L = D - W$. The objective function

$$\phi(Y) = \sum_i (y_i - \sum_j W_{ij} y_j)^2 \tag{3}$$

can be reformulated using graph Laplacian as

$$\underset{Y^T DY = I}{\operatorname{argmin}} \; tr(Y^T LY) \tag{4}$$

This objective function is solved using the generalized eigenvalue problem $Ly = \lambda Dy$. The low-dimensional embedding is given by eigenvectors corresponding to the d lowest eigenvalues with the exception of the trivial eigenvector corresponding to eigenvalue 0.

$$Y = U_d^T \tag{5}$$

Once the low-dimensional embedding of all training samples are obtained, a multivariate regression function (MATLAB mvregress) is used to map the low-dimensional embedding to corresponding pose labels.

3.4. Head pose estimation of test samples

When a new test sample needs to be embedded, the kernel matrix of the test sample is computed with respect to the training samples. In order to speed up the kernel computation, the test sample is compared only with a randomly chosen subset of training samples. This approximation seems to work fairly well in practice. The low-dimensional embedding is then computed using Nyström out-of-sample extension [33]. Let Λ and U be the eigenvalues and eigenvectors of the kernel matrix K computed with the training samples. The Laplacian eigenmaps embedding associated with a new point x is given by

$$Y(x) = \frac{1}{\Lambda} U^T K_x \tag{6}$$

where, $K_x = [K(x, x_1), ..., K(x, x_n)]$ and $\frac{1}{\Lambda} = diag\left(\frac{1}{\lambda_1}, ..., \frac{1}{\lambda_n}\right)$. The head pose is then estimated using the regression coefficients learned with the low-dimensional embedding of training samples.

4. Results

The proposed approach was validated using four in-the-wild face datasets - Annotated Facial Landmarks in the Wild (AFLW) [18], Annotated Faces in the Wild (AFW) [36], YouTube Faces dataset [34] and McGill dataset [9]. These challenging datasets were chosen since they contained unconstrained face images with poses ranging from -90°to +90°. From the AFLW dataset, 14000 face images were used to learn the manifold. For evaluating head pose estimation, we used the remaining 7041 face images from AFLW, 478 face images from AFW, 22534 randomly chosen face images from YouTube Faces dataset and 6833 images from McGill dataset. The AFLW and YouTube Faces datasets contain a substantial proportion of low-resolution and blurred images with the latter consisting of frames from YouTube videos.

4.1. Preprocessing

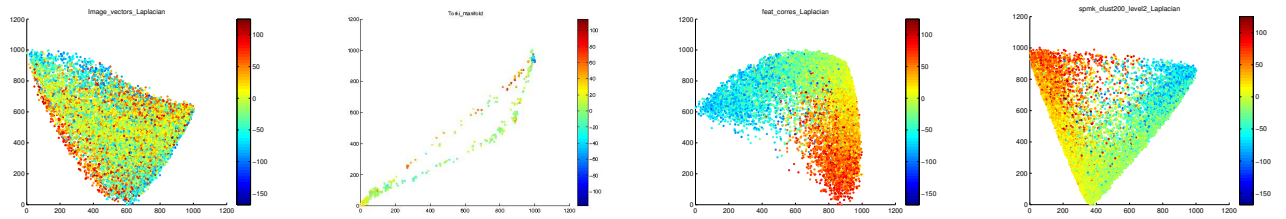
Viola-Jones face detection [30] was executed for the training images and the detected faces were resized to 128 $\times$ 128 pixels. All algorithms were compared using these detected faces. Up to 200 SIFT keypoints and their corresponding Geometric Blur descriptors were computed for each face image. The ground truth pose labels were obtained from the respective datasets.

4.2. Manifold learning

To illustrate the drawbacks of using vectorized images in unconstrained scenarios, we learned a manifold whose similarity metric was represented by the Euclidean distance of vectorized images. It can be observed from Figure 3(a) that the manifold learned using vectorized images does not represent a smoothly varying manifold where similar poses will be embedded close by in the low-dimensional space.

4.2.1 Comparison with other feature-based manifold learning approaches

We chose two in-the-wild feature-based manifold learning approaches for comparison - the Feature-embedding approach [28, 29] and the Learning-manifolds-in-the-wild approach [15]. For the Feature-embedding approach, we trained the system using 1000 randomly chosen images from the AFLW training images with 96 features per image. The number of training images and features per image were chosen similar to the experiments performed in [29]. The learned manifold using Feature-embedding approach is shown in Figure 3(b). It can be observed that the feature embedding corresponding to dissimilar poses are not spaced far apart. The Hausdorff distance between two embedded feature sets in this space might not always be representative of the pose similarity. Further, scaling this method to larger



(a) Vectorized images (b) Feature embedding (c) Learning-manifolds-in-the-wild (d) Approximate view manifolds

Figure 3: 2D manifolds learned using vectorized images and feature-based approaches with 14000 images in the AFLW dataset. Each dot represents a training sample and the different colors represent the ground truth pose labels ranging from -90°to +90°. It can be observed from these manifolds that the feature correspondences based approaches, Learning-manifolds-in-the-wild and Approximate View Manifolds, can better represent the pose similarity of unconstrained face images.

number of training samples seems computationally difficult along with memory constraints.

In the Learning-manifolds-in-the-wild approach, SIFT keypoints and decriptors are used in the computation of a summation kernel for representing image similarity. In our implementation of Learning-manifolds-in-the-wild approach, we used SIFT keypoints with Geometric Blur descriptors with a match kernel that enforces spatial similarity as well. Our implementation of the Learning-manifolds-in-the-wild approach is equivalent to performing feature correspondences without any approximations for similarity kernel computation. As can be seen from Figure 3(c), the Learning-manifolds-in-the-wild approach yields a smoothly varying manifold representative of pose similarity albeit being computationally expensive. The kernel computation for Learning-manifolds-in-the-wild approach using 14000 AFLW training images took about 2.44 hours on a 8x Intel Core i7-3770 CPU running at 3.4GHz with 16GB RAM.

4.2.2 Approximate View Manifold learning

In our Approximate View Manifold learning, the feature correspondences were approximated using the spatial pyramid match kernel. A spatial pyramid was learnt with 200 clusters and 2 levels using randomly chosen feature descriptors from the training images. The spatial pyramid histogram was computed for features in every training sample. The resulting 4000-dimensional spatial pyramid histogram was used to obtain image similarity using the histogram intersection function. The spatial pyramid match kernel thus allows the use of feature sets of unequal cardinality in a seamless manner for similarity kernel computation. Also, to further speed up the kernel computation, each training sample was only compared with approximately 5% of randomly chosen training samples. The kernel computation for the Approximate View Manifold learning using the same training images took only about 20 minutes on the same machine. Thus, it can be noted that the Approximate View Manifold learning provides 7X speedup compared to the Learning-manifold-in-the-wild approach.

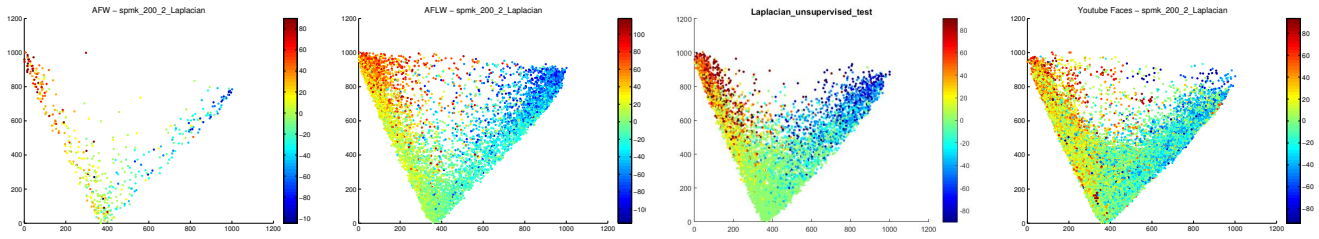
As shown in Figure 3(d), Laplacian Eigenmaps manifold was learned using the approximations described above. The Laplacian Eigenmaps provides a smoothly varying manifold with similar poses embedded close by. This shows that the spatial pyramid match kernel approximation still ensures that the feature correspondences are representative of pose similarity. Further, one has to keep in mind that the manifold learning was completely unsupervised with image similarity computed only using the feature descriptors and locations. This might cause the learned manifold to be noisy. However, this technique shows promise since the unsupervised manifold by itself is representative of pose similarity and can be made smooth and less noisy by biasing the similarity kernel with the pose labels.

4.3. Head pose estimation in the wild

The head pose estimation was evaluated using test images from the AFW (468 images), AFLW (remaining 7041 images), YouTube Faces (22534 images) and McGill Faces (6833 images) datasets. The poses were estimated using the Approximate View Manifold approach. We compared the pose estimation with that of the Feature-embedding approach and the Learning-manifolds-in-the-wild approach. For benchmarking with other categories of unconstrained head pose estimation techniques, we compare our approach with the Mixture of trees deformable model proposed in [36] and patch-based method proposed in [1]. Most other facial landmark-based methods handle only faces in pose range [-45,45] so that all facial landmarks are visible and hence were unsuitable for comparing extreme poses.

4.3.1 Embedding test samples

To obtain the low-dimensional embedding of the test samples, the kernel matrix of the test sample was computed



(a) AFW (b) AFLW (c) McGill Faces (d) YouTube Faces

Figure 4: Two-dimensional Approximate View Manifold embedding of unconstrained test images from AFW, AFLW, McGill and YouTube Faces datasets. Each dot represents a test sample and the different colors represent the ground truth pose labels of test data ranging from -90 to + 90. It can be observed that the test images of all datasets are embedded similar to the training images as shown in Figure 3(d).

with respect to the training samples as described in Section 3.4. The low-dimensional embedding of test images from the four datasets using the Approximate View Manifold approach is shown in Figure 4. It can be observed that the low-dimensional embedding of the test samples follow those of the training images shown in Figure 3(d) with the poses ranging from -90°to +90°embedded onto a roughly V-shaped manifold.

4.3.2 Pose estimation accuracy

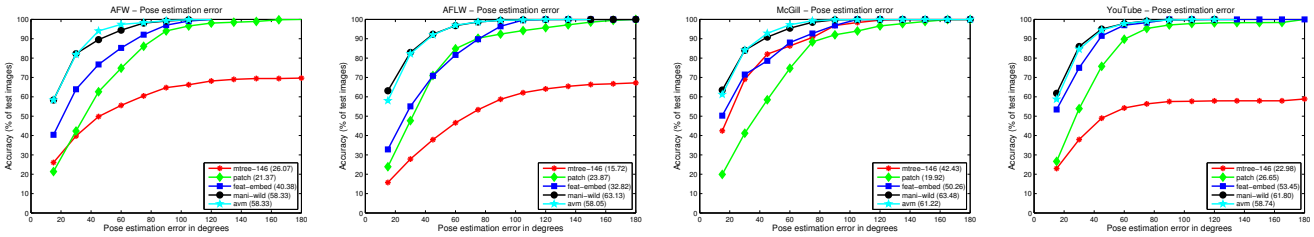
The pose estimation error was computed using the ground truth pose labels and the poses estimated using the regression coefficients of the learned manifold. The pose error was discretized in steps of 15°to allow us to compare with the Mixture of trees method which provides discrete pose labels. The accuracy plot representing the percentage of test images with pose estimation error within $\pm$ few degrees tolerance is shown in Figure 5. The pose estimation error was computed for the Feature Embedding approach, Learning-manifolds-in-the-wild approach, the Approximate View Manifold, the Mixture of trees method and the patch-based method of [1]. On the larger AFLW, McGill and YouTube Faces dataset, it can be seen that AVM performs better compared to the Mixture of trees method , patch-based method and the Feature-embedding approach with lesser MAE. Further, on all four datasets, it can be observed that the performance of both Learning-manifold-in-the-wild and the Approximate View Manifold are similar. The accuracy of AVM method may be 2-3% lesser than that of Learning-manifold-in-the-wild approach due to the trade-offs caused by the approximations in AVM. This suggests that the approximations due to the use of spatial pyramid match kernel or approximate neighborhood estimation does not hamper the pose estimation accuracy.

Table 1 shows the accuracy, i.e. percentage of images within $\pm$15°error, and the Mean Angular Error (MAE) of various methods on these four challenging datasets. It can be observed from this table that both Learning-manifold-in-the-wild approach and AVM approach perform better than the other methods in terms of both accuracy and MAE consistently across all four datasets. Further, the performance of AVM is comparable to that of Learning-manifold-in-the-wild approach with respect to both accuracy and MAE. This suggests that the various approximations used in AVM does not hamper performance. Also, learning and test sample embedding using AVM is considerably faster than that of Learning-manifold-in-the-wild approach. Pose estimation of a test sample takes 105ms on an average using the AVM approach.

Figure 6 shows the pose distribution of the training AFLW images and the average pose estimation accuracy for every pose category across all four datasets using AVM approach. The pose distribution of training images show that $\approx 75\%$ of images are within $\pm$45°. Pose labels were divided into 9 categories from 0° to $\pm$90° in steps of 15°. The right plot represents the percentage of images in each pose category that were estimated within few degrees error. As it can be observed from Figure 6, the pose estimation accuracy of 0° is highest and gradually drops till $\pm$60°. Beyond $\pm$75°, the pose estimation accuracy drops drastically with almost none of the images being estimated within $\pm$15° error. This can be attributed to the poor representation of extreme pose labels in the training images.

4.3.3 Effect of parameters on pose estimation accuracy

Three main parameters used by AVM are the number of clusters and number of tree levels in spatial pyramid matching and the number of randomly chosen samples for kernel approximation. In order to study the effect of various AVM parameters, we performed experiments by varying each of these parameters while keeping the other two parameters constant. Tables 2, 3 and 4 show the performance on AFLW test set for varying number of clusters, tree levels and randomly chosen samples respectively. These tables suggest

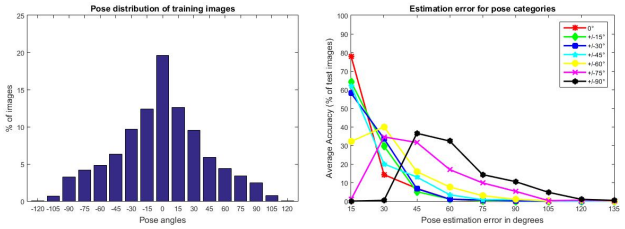


(a) AFW (468 images) (b) AFLW (7041 images) (c) McGill (6833 images) (d) YouTube Faces (22534 images)

Figure 5: The plots represent the pose estimation error for the Mixture of trees method [36], patch-based method [1], Feature-embedding method [29], Learning-manifold-in-the-wild method [15] and the Approximate View Manifold (avm). The numbers within parentheses in the legend represents the % of test images whose pose estimation error is within $\pm 15°$for each of these methods. These plots show that the AVM approach performs as well as the Learning-manifold-in-the-wild approach even after various approximations

Datasets	**AFW**		**AFLW**		**McGill**		**YouTube**	
Methods	Accuracy	MAE	Accuracy	MAE	Accuracy	MAE	Accuracy	MAE
Mixture of trees [36]	26.07	40.17	15.72	46.54	42.43	28.33	22.98	40.25
Patch-based [1]	21.36	41.67	23.87	38.39	19.92	43.44	26.65	45.95
Feature-embedding [29]	40.38	28.15	32.82	33.01	50.26	22.71	53.45	19.39
Learning-manifold-in-the-wild [15]	58.33	18.26	63.13	16.31	63.48	16.31	61.80	15.46
AVM (Our method)	58.33	17.20	58.05	17.48	61.22	16.29	58.74	16.47

Table 1: Accuracy (% of images with $\pm 15°$ error) and Mean Angular Error (MAE) for different methods on four challenging in-the-wild datasets



(a) Pose distribution of AFLW training images (b) Pose error per pose category

Figure 6: The left plot shows the pose distribution of training images in AFLW dataset. $\approx 75\%$ of images are within $\pm 45°$. The right plot shows the pose estimation accuracy for every pose category ($0°$, $\pm 15°$, ...) as % of images that belong to a specific pose category. The pose estimation accuracy seems to drop off slowly from $0°$ to $\pm 60°$. From pose category $\pm 75°$ onwards, the pose estimation accuracy drops drastically. This can be attributed to the poor representation of these categories while learning the manifold.

Num.clusters	Accuracy(%)	MAE(°)
100	57.05	18.07
200	58.05	17.48
300	57.63	17.52
400	60.47	16.75

Table 2: Performance results for different numbers of clusters and 2 levels in spatial pyramidal matching using AFLW test set

Num.levels	Accuracy(%)	MAE(°)
2	58.05	17.48
3	58.63	17.69

Table 3: Performance results for different numbers of levels and 200 clusters in spatial pyramidal matching using AFLW test set

that AVM is reasonably robust to changes in its parameters. However, increasing the number of clusters or levels of the spatial pyramid would result in an increase in the histogram length and hence will increase computation time of the similarity kernel. Similarly, increasing the number of randomly chosen samples would require comparison with more training samples thereby increasing computation time. From our experiments, 200 clusters and 2 levels of spatial pyramid works well along with 5% of training samples for similarity kernel computation.

% of random samples	Accuracy(%)	MAE(°)
5	58.05	17.48
10	57.93	17.44
20	57.62	17.38

Table 4: Performance results for different percentages of random samples with 200 clusters and 2 levels in spatial pyramidal matching using AFLW test set

5. Conclusion

In this paper, we have proposed an head pose estimation method for unconstrained face images using manifold embedding. To handle various nuisance factors in the unconstrained imagery, we have incorporated identity-invariant keypoint correspondences in the manifold embedding process. Our experiments revealed that the Geometric Blur descriptors were reasonably identity invariant thereby making the feature correspondences representative of the pose similarity. To speed up the feature-correspondence based manifold learning, we have used the spatial pyramid match kernel to approximate feature correspondences. The kernel computation was further approximated by comparing each training sample with only a random subset of other training samples. With these approximations, we obtained 7X speedup in manifold learning compared to the Learning-manifolds-in-the-wild manifold approach. Further, head pose estimation using the proposed method outperformed the other methods with improved accuracy and lesser MAE. Further, the performance of the AVM approach is at par with the Learning-manifolds-in-the-wild approach in spite of multiple approximations and is faster. Further, the performance of AVM is robust to variations of algorithm parameters. This suggests that the Approximate View Manifold approach shows great promise as a reliable and fast head pose estimation method for unconstrained low-resolution and blurred face images.

References

[1] J. Aghajanian and S. Prince. Face pose estimation in uncontrolled environments. In *BMVC*, volume 1, page 3, 2009. 1, 5, 6, 7

[2] S. O. Ba and J.-M. Odobez. Multiperson visual focus of attention from head pose and meeting contextual cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):101–116, 2011. 1

[3] V. N. Balasubramanian, J. Ye, and S. Panchanathan. Biased manifold embedding: A framework for person-independent head pose estimation. In *CVPR*, 2007. 1, 2

[4] R. H. Baxter, M. Leach, and N. M. Robertson. Tracking with intent. In *Sensor Signal Processing for Defence (SSPD), 2014*, pages 1–5. IEEE, 2014. 1

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 2003. 3

[6] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005. 1

[7] I. Chamveha, Y. Sugano, Y. Sato, and A. Sugimoto. Social group discovery from surveillance videos: A data-driven approach with attention-based cues. 2013. 1

[8] C.-W. Chen, R. C. Ugarte, C. Wu, and H. Aghajan. Discovering social interactions in real work environments. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 933–938. IEEE, 2011. 1

[9] M. Demirkus, D. Precup, J. J. Clark, and T. Arbel. Probabilistic temporal head pose estimation using a hierarchical graphical model. In *Computer Vision–ECCV 2014*, pages 328–344. Springer, 2014. 4

[10] M. W. Doniec, G. Sun, and B. Scassellati. Active learning of joint attention. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 34–39. IEEE, 2006. 1

[11] V. F. Ferrario, C. Sforza, G. Serrao, G. Grassi, and E. Mossi. Active range of motion of the head and cervical spine: a three-dimensional investigation in healthy young adults. *Journal of orthopaedic research*, 20(1), 2002. 1

[12] Y. Fu and T. S. Huang. Graph embedded analysis for head pose estimation. In *FGR*, 2006. 2

[13] A. Gaschler, S. Jentzsch, M. Giuliani, K. Huth, J. de Ruiter, and A. Knoll. Social behavior recognition using body posture and head pose for human-robot interaction. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2128–2133. IEEE, 2012. 1

[14] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005. 3

[15] C. Hegde, A. C. Sankaranarayanan, and R. G. Baraniuk. Learning manifolds in the wild. *Preprint, July*, 2012. 1, 2, 4, 7

[16] N. Hu, W. Huang, and S. Ranganath. Head pose estimation by non-linear embedding and mapping. In *ICIP*, 2005. 1, 2

[17] H. Ji, F. Su, and Y. Zhu. Robust head pose estimation via semi-supervised manifold learning with 1-graph regularization. In *IJCB*, 2011. 2

[18] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011. 4

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2, 3

[20] M. Leach, R. Baxter, N. Robertson, and E. Sparks. Detecting social groups in crowded surveillance videos using visual attention. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 467–473. IEEE, 2014. 1

[21] J. Leroy, F. Rocca, M. Mancas, and B. Gosselin. Second screen interaction: an approach to infer tv watcher's interest using 3d head pose estimation. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 465–468. International World Wide Web Conferences Steering Committee, 2013. 1

[22] M. J. Marín-Jiménez, A. Zisserman, and V. Ferrari. Heres looking at you, kid. *Detecting people looking at each other in videos. In BMVC*, 5, 2011. 1

[23] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE TPAMI*, 31(4), 2009. 1

[24] D. Parks, A. Borji, and L. Itti. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision research*, 2014. 1

[25] B. Raytchev, I. Yoda, and K. Sakaue. Head pose estimation by nonlinear manifold learning. In *ICPR*, 2004. 1, 2

[26] R. Stiefelhagen, L. Yang, and A. Waibel. Modeling focus of attention for meeting indexing based on multiple cues. *Neural Networks, IEEE Transactions on*, 13(4):928–938, 2002. 1

[27] R. Subramanian, Y. Yan, J. Staiano, O. Lanz, and N. Sebe. On the relationship between head pose, social attention and personality prediction for unstructured and dynamic group interactions. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 3–10. ACM, 2013. 1

[28] M. Torki and A. Elgammal. Putting local features on a manifold. In *CVPR*, 2010. 2, 4

[29] M. Torki and A. Elgammal. Regression from local features for viewpoint and pose estimation. In *ICCV*, 2011. 1, 2, 4, 7

[30] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 4

[31] M. Voit and R. Stiefelhagen. Deducing the visual focus of attention from head pose estimation in dynamic multi-view meeting scenarios. In *Proceedings of the 10th international conference on Multimodal interfaces*, pages 173–180. ACM, 2008. 1

[32] X. Wang, X. Huang, J. Gao, and R. Yang. Illumination and person-insensitive head pose estimation using distance metric learning. In *ECCV*. 2008. 2

[33] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, 2001. 4

[34] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011. 4

[35] S. Yan, H. Wang, Y. Fu, J. Yan, X. Tang, and T. S. Huang. Synchronized submanifold embedding for person-independent pose estimation and beyond. *IEEE Transactions on Image Processing*, 18(1), 2009. 2

[36] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 1, 4, 5, 7