A Single Classifier for View-Invariant Multiple Object Class Recognition

Gurman Gill and Martin Levine
Center for Intelligent Machines
McGill University, Montreal, Canada
gurman@cim.mcgill.ca

Abstract

Many state-of-the-art algorithms for object class recognition have recently appeared in the literature. These algorithms recognize one object at a time in that a dedicated classifier needs to be trained for each object class. However, no paper has yet reported a single classifier capable of recognizing the object class of any one of a number of classes. This paper sets out to recognize objects belonging to multiple classes, irrespective of the viewing and illumination conditions, using just a single classifier. This is achieved by means of the dimensionality reduction technique, Locally Linear Embedding. Object clusters corresponding to a specific range of views in the lower dimensional space are generated using supervised LLE. A test image is then projected into this space where it is labeled by a simple classifier. We have tested our technique on five object classes and attained recognition results comparable to those achieved in the literature using multiple, not single classifiers.

1 Introduction

Object class recognition has recently received attention from the vision research community. As against recognition of *specific* individual objects from images, object class recognition involves classification of objects belonging to a class such as faces, cars or bikes. The object class recognition problem is also termed generic object recognition or object categorization. This paper deals with a behavior that humans casually perform in everyday life: recognizing *multiple* object classes irrespective of the viewing and illumination conditions. Many recent state-of-the-art object class recognition algorithms [4][16][3][14][9] construct a classifier for classifying one object at a time. In contrast, we present a simple technique that allows us to build a *single* classifier for categorizing objects belonging to multiple classes. Our technique straightforwardly deals with different views of an object and is largely invariant to stark illumination changes.

Most recent work in object class recognition makes use of local feature detectors to represent an image in terms of distinctive parts, which after learning with respect to a set of training images are used to represent an object class. The learning procedure is carried out using boosting [16][9] or expectation-maximization algorithm [4][3]. These algorithms are trained for a particular object at a time and can efficiently learn class-specific parts of that object. However, if there is more than one object class, they can be no longer used to construct a single classifier to distinguish all object classes. Other than

BMVC 2006 doi:10.5244/C.20.27

these feature-based methods, an appearance-based technique that computes histograms of qualitative shape indices is presented in [14]. The object class is represented by a set of such histograms of training images. A test image is classified using a similarity measure based on an inner product of histogram representations. The results are shown to improve after the objects are cropped from the images. The algorithm presented in this paper uses a simple image representation along with a dimensionality reduction technique to transform images from different object classes into a single global coordinate system. Simple classification techniques are then applied on this low-dimensional subspace to classify multiple objects. The input to the system is a rectangular region that just contains the object of interest. This preliminary segmentation is not required in feature-based methods mentioned above. These locate salient interest points by maximizing a certain measure over a range of scales and choosing a characteristic scale. This permits scale invariance that in our case is achieved by normalizing all segmented rectangular regions to a fixed size.

A few earlier attempts have been made to achieve view-invariance for object class recognition. Schniederman and Kanade [12] present a view-based object detector which employs multiple detectors specialized to a specific orientation of the object. In the specific context of multi-view face detection, [17] builds a pyramid containing detectors operating on a coarse-to-fine view range, [15] applies a bootstrap method to train hierarchical SVMs and [7] constructs separate face detectors for separate view segments. All of these require multiple detectors to account for different object views. Murase and Nayar in [8] presented a technique that can be used to naturally account for view variation without constructing multiple detectors. They represent an object in different possible views and illumination as a single manifold in eigenspace. A query image is projected to the eigenspace and its location on the object's parameterized manifold then determines the pose. This technique used PCA to form the manifold. Recently, Locally Linear Embedding (LLE) [10] has been used to compute low-dimensional, neighborhood preserving embeddings of high-dimensional data. It has been shown that PCA fails to preserve the neighborhood structure of nearby images [10][5]. Therefore, LLE is better suited for representing true view variation in the embedded space. This was demonstrated for faces in [5], where the embeddings perfectly recovered the different poses.

LLE is an unsupervised technique that derives the embeddings solely from the geometric properties of nearest neighbors in the high-dimensional space. Therefore it does not exploit the class information of data points when known. [6] and [2] have proposed a supervised variation of LLE whose main purpose is to facilitate classification. Supervised LLE (sLLE) tends to reduce intra-class and increase inter-class distances resulting in good data partitioning into different classes. We have used sLLE to construct distinct view clusters for each object class in a lower-dimensional space. Therefore, each cluster represents an object class in a particular view segment. During the training phase, all training images of multiple objects at different views are mapped onto a lower-dimensional space using sLLE. During testing, a query image is projected to this space using a non-parametric method [10]. Finally, using a simple classification technique, the query image is classified to a particular object class at a particular view.

The paper is organized as follows: We discuss the supervised LLE algorithm in Section 2. Our complete algorithm is then described in step-wise fashion in Section 3. Section 4 presents the experiments and results. Finally, Section 5 contains the concluding remarks and scope for future work.

2 Supervised Locally Linear Embedding

LLE maps high-dimensional data to a single global coordinate system in a manner that preserves neighbourhood relationships. Suppose the data consist of N real-valued vectors $\vec{X_i}$, $i = 1, 2, .., ..., ..N$, each of dimensionality D ($\vec{X_i} \in R^D$) that have been sampled from some smooth underlying manifold. Each data point can be linearly reconstructed from its neighbours. The reconstruction error can be then measured by the cost function:

$$e(W) = \sum_{i=1}^{N} |\vec{X_i} - \sum_{j=1}^{K} W_j^i \vec{X_{N(j)}}|^2 \tag{1}$$

where K is the number of neighbours and W_j^i denotes the contribution of j^{th} neighbour to the i^{th} reconstruction.

We note that for one vector $\vec{X_i}$ and weights W_j^i, the contribution to the reconstruction error (Equation 1) can be rewritten as

$$e^i(W) = |\sum_{j=1}^{K} W_j^i (\vec{X_i} - \vec{X_{N(j)}})|^2 \tag{2}$$

$$= \sum_{j=1}^{K} \sum_{m=1}^{K} W_j^i W_m^i Q_{jm}^i \tag{3}$$

where Q^i is the $K \times K$ matrix:

$$Q_{jm}^i = (\vec{X_i} - \vec{X_{N(j)}})^T (\vec{X_i} - \vec{X_{N(m)}}) \tag{4}$$

To compute the weights, this cost function is minimised subject to the constraint that enforces the invariance of data points to translation. This is accomplished by setting $\sum_j W_j^i = 1$. The optimal weights subject to these constraints are found by solving a least squares problem.

Now, let us assume that the data actually lie on or near a smooth manifold of lower dimensionality $d << D$. The transformation from the high-dimensional coordinates of each neighbourhood to global internal coordinates on the manifold can be achieved as a linear mapping consisting of translation, rotation and rescaling. By construction, the weights are invariant exactly to these transformations since they reflect intrinsic geometric properties of the data. Therefore, it is expected that these same weights would reconstruct each data point from its neighbours in the low-dimensional space. LLE constructs a neighbourhood-preserving mapping based on this idea. The high-dimensional observation $\vec{X_i}$ is mapped to a low-dimensional vector $\vec{Y_i}$ that represents the global internal coordinates on the manifold such that the embedding cost function is minimized:

$$\phi(Y) = \sum_{i=1}^{N} |\vec{Y_i} - \sum_{j=1}^{K} W_j^i \vec{Y_{N(j)}}|^2 \tag{5}$$

The above cost function is optimized for the d-dimensional vectors $\vec{Y_i}$ using the optimized weights W_{ij} derived from equation 1. This cost function can be minimized by solving a sparse $N \times N$ eigenvalue problem whose bottom d non-zero eigenvectors provide the required orthogonal vectors $\vec{Y_i}$.

If class information $\omega_i \in \Omega(\|\Omega\| = c)$ corresponding to each data vector $\vec{X_i}$ is available, then it can be used to supervise the LLE algorithm such that the embeddings separate the within-class structure from the between-class structure. [2] proposed a method in which the degree of supervision can be controlled by a parameter α. Note that Q^i (Equation 4) can also be calculated based on just the squared Euclidean distance matrix D between all samples in the data set:

$$Q^i_{jm} = \frac{1}{2}\left(D_{i,N(j)} + D_{i,N(m)} - D_{N(j),N(m)}\right) \tag{6}$$

Now this distance is simply increased for data points in different classes:

$$D' = D + \alpha\ max(D)\Delta \tag{7}$$

where $\Delta_{jm} = 1$ if $\omega_j \neq \omega_m$, and 0 otherwise.

Given a specific set of data embeddings, the non-parametric method to compute embedding for an unseen example relies on the natural mapping between the low and high-dimensional spaces. For vector $\vec{X_{new}}$, the K nearest neighbours are identified and new weights W_j^{new} that best reconstruct $\vec{X_{new}}$ are calculated. The embedding for this new vector is simply the linear combination of embeddings of its neighbours weighted appropriately.

$$\vec{Y_{new}} = \sum_{j=1}^{K} W_j^{new} \vec{Y_{N(j)}} \tag{8}$$

This method is derived from nearest-neighbour interpolation and provides a simple way to generalize to new data when the assumption of local linearity is met.

Using a nearest mean classifier, the new data point is represented by the class mean of the embedded coordinates closest to it.

3 The Multiple Object Class Categorization Algorithm

The aim of the algorithm is to categorize an input image into one of many possible object classes. The input image is a rectangular sub-region that just contains an arbitrarily illuminated object of interest set against an arbitrary background. No assumptions are made regarding the spatial viewpoint of the object. Illumination normalization is achieved by employing the phase of the image. Phase images have proven to be quite robust to stark illumination changes [11]. With respect to viewpoint, we build view clusters which are sub-classes of each object class so that an input image can be classified according to the view of the object. We elaborate the details below.

The algorithm requires the object of interest to be roughly within the image window. This condition is imposed since LLE uses neighbourhood relationships between the training data points to compute embeddings in the lower-dimensional space. For objects from the same class to cluster in the embedding space, it is imperative that the objects be similar in the high-dimensional space. Since we use a processed raw image to represent the objects, all unnecessary background should be cropped so that objects from the same class will retain their similarity. It is not required that this cropping be done precisely and the algorithm achieves excellent results notwithstanding a significant amount of visible background. This is meant to mimic the results of a preprocessing step which performs a focus of attention analysis or a complete multi-scale scanning of the image.

All images from all of the object classes were pre-processed for scale and illumination normalization. This involved resizing the cropped image window to a fixed size. Subsequently, the phase of each image was computed by measuring the phase angle between the imaginary and real part at each frequency in the Fourier transform of the image, thereby removing the Fourier magnitude information. We then computed the inverse Fourier transform of the phase component and used this so-called phase image as the input $\vec{X}_i$. In addition to the class labels, the phase images are used to train the supervised LLE algorithm to produce embeddings in the lower-dimensional space. For a given object class, images are labeled according to separate sub-classes on the basis of the viewpoint angle. The embeddings were computed for various values of the algorithm parameters such as the number of nearest neighbors K, the dimensionality d of the embedding space and the amount of supervision α. The resulting embeddings clustered according to the appropriate sub-classes and classes. Therefore, the embedded space contains well-defined view-clusters corresponding to each object class.

Once the embeddings have been computed, the test images (scale and illumination normalized) are projected onto the lower-dimensional space, as explained in section 2. The test images are expected to fall into one of the view-clusters corresponding to a particular object class[1]. A simple classification technique, such as the one described in section 2, returns the predicted class for each test image.

Since the algorithm performs multiple object class recognition, the recognition results cannot be represented by true positive-false positive rate. This is because the false positives for each class can come from any of the remaining classes. Therefore, to compute the false-positive rate for a given class, we cannot divide by the total number of images in the remaining classes. Hence, in multiple object class recognition it is not possible to evaluate the false positive rate. Instead, the recognition results can be represented by a confusion matrix whose $(i,j)_{th}$ entry corresponds to the total number of times an object class i is classified as object class j. The true positives for class i will be given by the $(i,i)_{th}$ entry. The total number of false positives for class i will be sum of elements at column i except for the $(i,i)_{th}$ entry. The confusion matrix allows us to calculate the recall-precision rate [1] for each object class. To compare the performance of the algorithm for different values of parameters such as (d,K,α), we calculate the average recall and precision rate of all the classes. These can be computed by measuring the total number of true and false positives for each class except the background class. The best result is indicated by the maximum value of the F-measure, defined as 2*Recall*Precision/(Recall + Precision) [1].

4 Experiments and Results

We have used 5 publicly available databases to evaluate the algorithm: CMU PIE faces [13], UIUC cars [1], Caltech motorbikes, Caltech airplanes and Caltech background datasets [4]. From the CMU PIE database, we used 1700 images comprised of 68 individuals viewed in 5 different poses (camera ids: 22, 37, 27, 11, 34) subject to 5 different illuminations (flash ids: 03, 05, 08, 14, 16). These poses and illuminations roughly correspond to -90, -45, 0, 45, and 90 degrees rotation from the optical axis (frontal view). Each of the 5 different poses were assigned a separate class label. Each of the rest of the

[1] We did not test with objects for which the algorithm was not trained. However, this situation could be easily accounted for

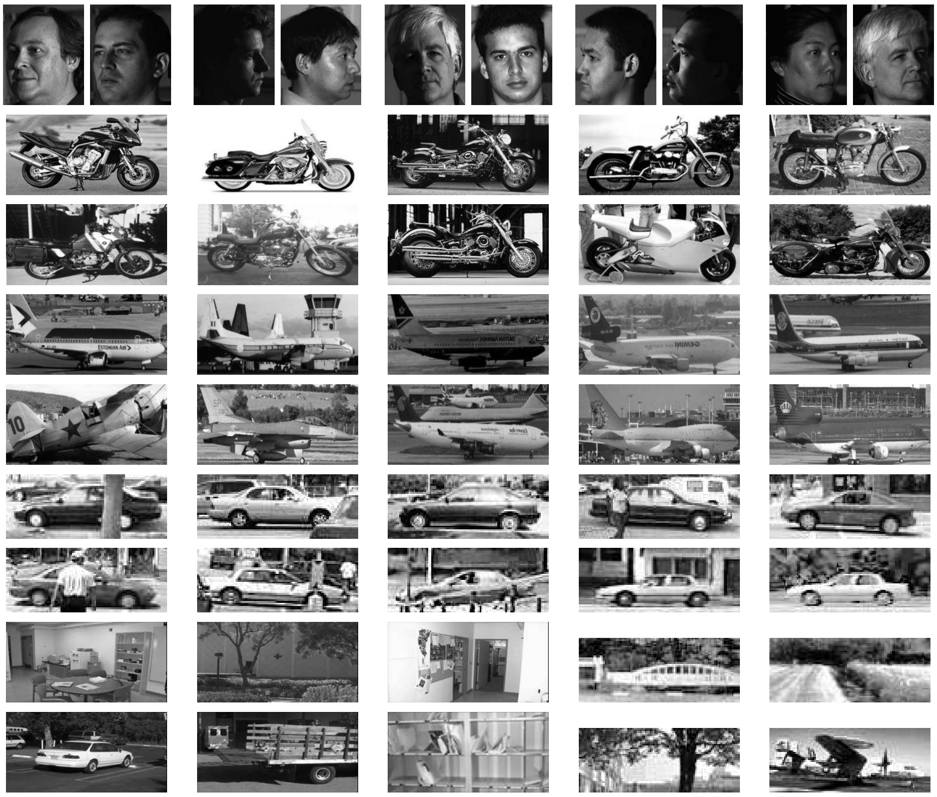


Figure 1: Examples of five object classes taken from test set. In the first row, only correctly classified images are shown[3]. The remaining rows occur in pairs. The first row of each pair illustrates an object class which was correctly classified while the second row was misclassified.

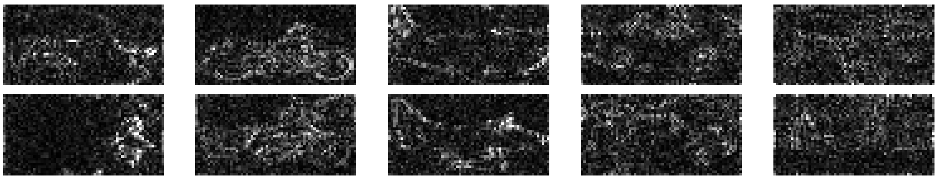
objects (cars, motorbikes and airplanes) is represented by a single view that forms a separate class. The non-car set in the UIUC database together with the Caltech background set form the background class in our experimental setup. In all there are a total of 9 classes.

Objects were manually cropped from the CMU PIE database, Caltech airplanes, and Caltech motorbikes. The UIUC car and Caltech background databases were used without modification. Due to asymmetric objects, the cropped rectangular regions usually contained considerable background, especially for the airplane class. After the cropping step, all of the images were resized to a fixed size of 32×64, yielding an image dimensionality of $D = 2048$. Figure 1 shows an example of the cropped images from all of the object classes considered in this paper. Figure 2 shows some of the corresponding phase images.

For all of the object classes indicated above, half of the set was used for training and the other half for testing. For the former, this translated into 170 images each for 5 face classes, 250 cars [4], 413 motorbikes, 537 airplanes and 725 background images, for a total

[3] The authors of [13] do not permit displaying misclassified images.

[4] Note that we have only used the training portion from the UIUC car database and divided it into training and test sets for our experiments.



Faces Motorbikes Airplanes Cars Background

Figure 2: Phase images of some of the examples in figure 1.

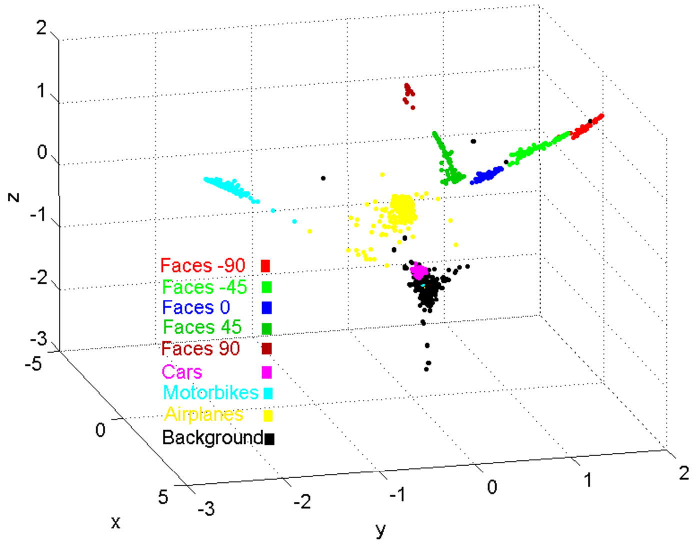


Figure 3: Example of object class embeddings of the training data in 3D space. Note the 9 object clusters.

of 2775 images. The images were chosen sequentially from the database, with the odd ones forming the training set and the even ones the test set.

The training set is used as an input to supervised LLE, which requires initializing parameters K and α to produce embeddings in d dimensional space. We computed embeddings for $(d, K, \alpha) = (3, 15, 1)$ for the purpose of visualization. Figure 3 shows this embedding. Note the nine distinct clusters in this three-dimensional space. The test data were projected onto the embedding space and classified.

Many experiments were carried out by varying K between 15 and 60 with a step size of 5, d between 6 and 16, with a step size of 2, and α between 0.4 and 1, with a step size of 0.3. Classification of the test data was carried out using the nearest mean classifier. We recorded the recall and precision rate for each object class and computed the average recall and precision rate and the associated F-measure (Section 3) for each of the experiments. Figure 4 shows the variation of F-measure as a function of the number of nearest neighbors K for different values of dimensionality d. The F-measure is computed by keeping $\alpha = 1$ since this value yielded the best results. The maximum F-measure is

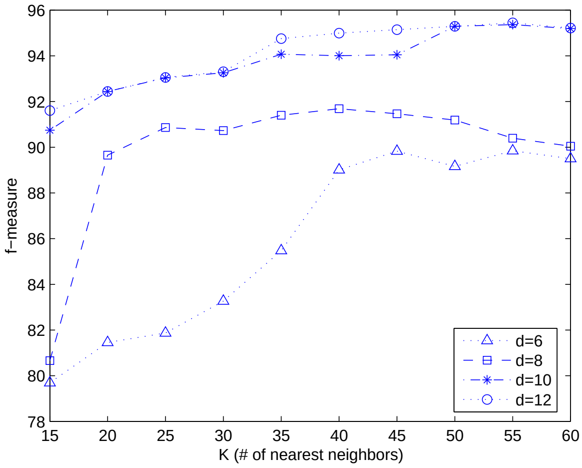


Figure 4: Variation of F-measure with respect to K for different values of d.

Classes										R	1-P
Faces (-90^o)	166	1	0	0	0	0	0	0	6	97.65	1.19
Faces (-45^o)	1	159	0	0	0	0	0	0	10	93.53	1.24
Faces (0^o)	0	1	159	0	0	0	0	0	10	93.53	0
Faces (45^o)	0	0	0	155	1	0	0	0	14	91.18	3.13
Faces (90^o)	0	0	0	4	164	0	0	0	2	96.47	0.61
Cars	0	0	0	0	0	258	0	0	17	93.82	1.15
Motorbikes	0	0	0	0	0	0	398	0	15	96.37	4.56
Airplanes	0	0	0	1	0	1	11	478	46	89.01	3.04
Background	1	0	0	0	0	2	8	15	674	96.29	14.79

Table 1: Confusion Matrix and Recall (R) and 1-Precision (1-P) Rate for each object class

reached at $(d,K,\alpha) = (12,55,1)$. Table 1 shows the confusion matrix for the number of classifications, along with the recall-precision rate for each object class using this set of parameters. According to [6], the dimensionality of the embedded space should be one less than the number of classes in order to completely specify all of them (this implies $d = 8$ in the above experiments). The additional dimensions required for our experiments could be needed in order to account for the background variations that were quite significant in a few of the object classes. Our experiments showed best recognition results with $\alpha = 1$. Since each object class contains enormous variation, complete supervision must be needed to obtain distinct object clusters. However, according to experiments done by Ridder et al. in [2], lower values of α yield the best generalization of sLLE.

The results show the effectiveness of our approach in dealing with different views of the same object class for the CMU PIE face database. Despite the presence of stark illumination changes within a class, the high object class recognition results indicate the efficacy of using phase information for illumination normalization. The individual recall-precision rates of other object classes are comparable to those in the literature, although

Databases	Our	[4]	[3]	[14]
UIUC cars	93.81	88.5		
Caltech airplanes	89.01	90.2	98.75	89.2
Caltech motorbikes	96.36	92.5	99.5	94.9

Table 2: Comparison of the results obtained using the algorithm discussed in this paper with some of the methods appearing in the literature.

these are the ROC or RPC equal error rates while our results are the recall-precision values at the maximum value of F-measure. In addition, the former were all obtained using local part features and by creating individual classifiers for each object class. Table 2 provides a comparison of our results with those in the literature[5]. In comparison to the other appearance-based technique[14], our results are superior for the motorbike class but slightly inferior for the airplane class. Overall, our algorithm has achieved multiple object class recognition with encouraging results. It should be noted that these results were obtained with the same set of parameters and a single classifier for all of the object classes, whereas in the literature optimal parameters are often determined for a particular object class.

5 Conclusions and Future Work

We have presented an algorithm for recognizing sub-images of objects which belong to multiple object classes. The algorithm can be straightforwardly used for multiple object detection in images by scanning a moving window over an image to provide input data to the recognition algorithm we have presented here.

The algorithm deals with viewpoint variations by building separate view clusters for each object sub-class and achieves illumination normalization by using phase image information. Thus, an input can be an object subject to any viewing or illumination conditions and the algorithm will return its class with high accuracy. Such a classification system that does not require separate classifiers for different objects closely mimics the human visual system. We plan to investigate further this approach to multiple object class recognition. The current literature which emphasizes an individual classifier for each object class is not practical for multiple object class recognition.

Our approach of building view clusters for each object class not only allows the input image to be classified irrespective of its view, but also yields an approximate pose of the input image. This pose estimation can be used as an input to applications that require pose-specific information for further processing.

Since the algorithm builds view clusters, it requires many images from a specific object sub-class in a given view-range. Other than faces, these are not yet publicly available for other object classes. As part of our future work, we intend to construct such a database and evaluate our algorithm on multiple objects at multiple views. We will also evaluate the number of view segments our algorithm can deal with. After a certain number, the view clusters may start to merge which would introduce inaccuracies in the determination of the sub-class but obviously not in the overall generic class. Lastly, we will test our algorithm to determine its efficacy when the objects are subject to partial occlusion.

[5] Only those results from the literature are compared that use the same distribution of training and test images

References

[1] S. Agarwal and A. Awan. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *IEEE Trans. PAMI*, pages 1475–1490, 2004.

[2] D. de Ridder and R. Duin. Locally Linear Embedding for Classification. Technical Report PH-2002-01, Pattern Recognition Group, Delft Univ. of Tech., Delft. 2002.

[3] G. Dorkó and C. Schmid. Object Class Recognition Using Discriminative Local Features. Rapport de recherche RR-5497, INRIA - Rhone-Alpes, February 2005.

[4] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *CVPR*, volume 2, pages 264–271, June 2003.

[5] A. Hadid, O. Kouropteva, and M. Pietikäinen. Unsupervised Learning Using Locally Linear Embedding: Experiments with Face Pose Analysis. In *ICPR (1)*, pages 111–114, 2002.

[6] O. Kouropteva, O. Okun, A. Hadid, M. Soriano, S. Marcos, and M. Pietikinen. Beyond Locally Linear Embedding Algorithm. Technical Report MVG-01-2002, University of Oulu, Machine Vision Group. 2002.

[7] Y. Li, S. Gong, J. Sherrah, and H. Liddell. Support Vector Machine based Multi-view Face Detection and Recognition. *Image and Vision Computing*, 22(5):413–427, May 2004.

[8] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3-d Objects from Appearance. *Int. J. Comput. Vision*, 14(1):5–24, 1995.

[9] A. Opelt, M. Fussenegger, and P. Auer. Generic Object Recognition with Boosting. *IEEE Trans. PAMI*, 28(3):416–431, 2006.

[10] L. K. Saul and S. T. Roweis. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. *J. Mach. Learn. Res.*, 4:119–155, 2003.

[11] M. Savvides, B.V.K.V. Kumar, and P.K. Khosla. Eigenphases vs Eigenfaces. In *ICPR*, volume 3, pages 810–813, 2004.

[12] H. Schneiderman and T. Kanade. A Statistical Model for 3d Object Detection Applied to Faces and Cars. In *CVPR*, June 2000.

[13] T. Sim, S. Baker, and M. Bsat. The Cmu Pose, Illumination, and Expression (PIE) Database. In *AFGR*, May 2002.

[14] J. Thureson and S. Carlsson. Appearance Based Qualitative Image Description for Object Class Recognition. In *ECCV (2)*, pages 518–529, 2004.

[15] P. W. and Q. Ji. Multi-View Face Detection under Complex Scene based on Combined SVMs. In *ICPR (4)*, pages 179–182, 2004.

[16] W. Zhang, B. Yu, G. Zelinsky, and D. Samaras. Object Class Recognition using Multiple Layer Boosting with Multiple Features. In *CVPR*, pages II:323–330, 2005.

[17] Z. Zhang, L. Zhu, S. Z. Li, and H. Zhang. Real-Time Multi-View Face Detection. In *AFGR*, pages 142–147, 2002.