Journal of Computational Information Systems 9: 15 (2013) 6089–6096
Available at http://www.Jofcis.com

An Efficient Data Dimensionality Reduction Scheme Based on SIFT for Face Recognition ⋆

Xianzhong LONG*, Hongtao LU, Yong PENG, Lei HUANG

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract

In order to accelerate data processing and improve classification accuracy, some classic dimension reduction techniques have been proposed in the past few decades, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Locally Preserving Projections (LPP), etc. In this paper, we put forward an efficient data dimensionality reduction scheme based on Scale Invariant Feature Transform (SIFT). Specifically, SIFT features of all images are first extracted, and then a dictionary is constructed by using k-means clustering algorithm, each image is finally represented according to their SIFT features and the obtained dictionary. A series of experimental results are carried out over two benchmark face databases to demonstrate the efficacy of our proposed scheme.

Keywords: Scale-invariant Feature Transform; K-means; Support Vector Machine; Face Recognition

1 Introduction

As one of the most challenging tasks in computer vision and pattern recognition fields, face recognition have recently attracted many researchers' attention. Some face recognition techniques have been proposed in the past few decades. We usually represent a face image of size $m \times n$ pixels by an $m \times n$ dimensional vector. However, these $m \times n$ dimensional vectors are too large to allow fast processing. In order to resolve this problem, many dimensionality reduction techniques have been proposed, such as Principal Component Analysis (PCA) [1], Linear Discriminant Analysis (LDA) [2], Locality Preserving Projections (LPP) [3], etc. Some corresponding projection matrices are generated after using these methods mentioned above. Each column of these projection matrices is a basis image, so the dimensionality reduction techniques are used to learn the representation of a face as linear combination of basis images. The basis images of PCA are orthogonal and have a statistical interpretation as the directions of the largest variance of data. LDA tries to find a linear transformation that can maximize the between-class scatter matrix and meanwhile minimize the within-class scatter matrix. However, traditional PCA and LDA dimensionality

⋆Project supported by the National Basic Research Program of China (973 program) (No. 2009CB320901), National Natural Science Foundation of China (No. 61272247).

*Corresponding author.
Email address: lxz85@sjtu.edu.cn (Xianzhong LONG).

1553–9105 / Copyright © 2013 Binary Information Press
DOI: 10.12733/jcisP0212
August 1, 2013

reduction techniques do not take into account the local geometric structure information of data, LPP seeks to preserve the intrinsic geometry of the data and local structure.

In the face recognition field, PCA, LDA and LPP are used to operate on the pixel values directly. For example, we can transform an image of 119×112 size into a 32×32 matrix by using down-sampling technique. Then the image can be represented by a 1024×1 column vector. Recently, some local descriptors have been used in image classification and object recognition, such as Scale-Invariant Feature Transform (SIFT) [4], Histograms of oriented Gradients (HoG) [5], Affine Scale-Invariant Feature Transform (ASIFT) [6], Oriented Fast and Rotated BRIEF (ORB) [7], etc. The existing experimental results denoted that these descriptors are useful because local information of images are helpful for some actual applications. In this paper, we reduce the dimensionality of data based on the SIFT descriptors of images and use SVM [8] to classify. Experiments show that our scheme achieves better recognition accuracy than those classical dimensionality reduction techniques.

The remainder of the paper is organized as follows: Section 2 introduces the basic idea of existing dimensionality reduction techniques. Our method is proposed in Section 3. In Section 4, the comparison results of face recognition on two widely used databases are reported. Finally, conclusions are made in Section 5.

2 Related Work

Let $\mathbf{X}$ be a data matrix of n m-dimensional samples $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$. Each column of $\mathbf{X}$ represents a face image with m dimensions. Usually, the value of m is very large and this may lead to slow recognition speed and low recognition accuracy. So dimensionality reduction is necessary before recognition, this section briefly reviews three classical dimensionality reduction techniques.

2.1 Principal component analysis (PCA)

Principal Component Analysis (PCA) [1] tries to find a subspace whose basis vectors correspond to the maximum-variance direction in the original image space. Without loss of generality, let $\mathbf{W} \in \mathbb{R}^{m \times k}$ represent the linear transformation that maps the original m-dimensional space onto a k dimensional feature subspace where $k \ll m$, the new feature vectors $\mathbf{y}_i \in \mathbb{R}^k (i = 1, 2, \cdots, n)$ are obtained via the linear transformation:

$$\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i \tag{1}$$

The columns of $\mathbf{W}$ are the first k eigenvectors $\mathbf{w}_j \in \mathbb{R}^m (j = 1, 2, \cdots, k)$, which can be achieved by solving the following problem:

$$\mathbf{C}\mathbf{w}_j = \lambda_j \mathbf{w}_j \tag{2}$$

where $\mathbf{C} = \mathbf{X}\mathbf{X}^T$, $\mathbf{C} \in \mathbb{R}^{m \times m}$ is the covariance matrix and λ_j is the eigenvalue associated with the eigenvector $\mathbf{w}_j$. It is noteworthy that we should accomplish two things before obtaining the eigenvectors of $\mathbf{C}$: 1) the column vectors in $\mathbf{X}$ are normalized such that $\|\mathbf{x}_i\| = 1$ and 2) the average vector of all images is subtracted from all column vectors of $\mathbf{X}$.

2.2 Linear discriminant analysis (LDA)

Linear Discriminant Analysis (LDA) [2] seeks those vectors in the low-dimensional space that best discriminate among classes. From all samples, two matrices are defined. The first is called between-class scatter matrix, given by

$$\mathbf{S}_b = \sum_{t=1}^{c} N_t(\mu^t - \mu)(\mu^t - \mu)^T \tag{3}$$

where $\mathbf{S}_b \in \mathbb{R}^{m\times m}$, c is the number of classes, N_t is the number of training samples in class t, $\mu^t \in \mathbb{R}^m$ is the mean vector of samples belonging to class t, and $\mu \in \mathbb{R}^m$ represents the mean vector of all samples. The second matrix is called within-class scatter matrix:

$$\mathbf{S}_w = \sum_{t=1}^{c}\sum_{i=1}^{N_t}(\mathbf{x}_i^t - \mu^t)(\mathbf{x}_i^t - \mu^t)^T \tag{4}$$

where $\mathbf{S}_w \in \mathbb{R}^{m\times m}$, and $\mathbf{x}_i^t \in \mathbb{R}^m$ is the i-th sample of class t. The goal of LDA is to maximize the between-class scatter matrix while minimizing the within-class scatter matrix.

2.3 Locality preserving projections (LPP)

Locality preserving projection (LPP) [9] searches for embedding that preserves the intrinsic geometry of the data. The objective function of LPP is as follows:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{i,j=1}^{n} \|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{x}_j\|^2\mathbf{M}_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} \mathbf{D}_{ii}\|\mathbf{W}^T\mathbf{x}_i\|^2 = 1 \end{aligned} \tag{5}$$

where $\|\cdot\|$ stands for the vector L_2 norm, $\mathbf{M}_{ij} = exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\}$ is a heat kernel function which is used to calculate the similarity matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ of data. $\mathbf{D}_{ii}$ is the row (or equivalently column, since $\mathbf{M}$ is symmetrical) sum of the similarity matrix $\mathbf{M}$, i.e., $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{M}_{ij}$.

3 Data Dimensionality Reduction Scheme Based on SIFT

Suppose that there are n images in one specific database. For an image, a matrix $\mathbf{s}_i \in \mathbb{R}^{128\times c_i}(i = 1, 2, \cdots, n)$ is used to represent the image by extracting SIFT. Each column of the matrix $\mathbf{s}_i$ is a descriptor corresponding to a key point in the original image. Then we can combine the SIFT matrices $\mathbf{s}_i$ of all the images in this database to form a large matrix $\mathbf{S}$. Specifically, we can use the following equation to explain this process.

$$\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n] = [\mathbf{s}_1^*, \mathbf{s}_2^*, \cdots, \mathbf{s}_N^*] \tag{6}$$

where $\mathbf{S} \in \mathbb{R}^{128\times N}$ and $N = c_1 + c_2 + \cdots + c_n$ is the total number of all SIFT descriptors extracted from the images. Each $\mathbf{s}_j^* \in \mathbb{R}^{128\times 1}(j = 1, 2, \cdots, N)$ is a SIFT descriptor. Then we impose the

k-means clustering algorithm on the matrix $\mathbf{S}$. The k-means can be formulated as the following optimization problem:

$$\min_{\mathbf{U}} \sum_{j=1}^{N} \min_{k} \|\mathbf{s}_j^* - \mathbf{u}_k\|^2 \tag{7}$$

where $k = 1, 2, \cdots, K$ and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K] \in \mathbb{R}^{128\times K}$ is the matrix consisting of K cluster center vectors. It is noteworthy that the number of cluster center K is much larger than the number of categories. $\|.\|$ stands for the vector L_2 norm.

After getting the cluster center matrix $\mathbf{U}$, we can use a $K \times 1$ column vector to represent each image by counting how many descriptors belong to each cluster center. We can use the following specific example to illustrate the process of new representation of image.

We set a full zero vector $\mathbf{r}_1 \in \mathbb{R}^{K\times 1}$ in advance. For the first image in the database, we suppose its SIFT matrix is $\mathbf{s}_1 \in \mathbb{R}^{128\times 200}$, i.e., c_1=200, which denotes there are 200 descriptors in this image. We calculate the Euclidean distance between each descriptor of $\mathbf{s}_1$ and each cluster center in the matrix $\mathbf{U}$ and find the minimum distance. If the $1th$ and $3th$ descriptor of $\mathbf{s}_1$ are closest to the $\mathbf{u}_{K-1}$, then the value in the location $(K-1, 1)$ of $\mathbf{r}_1$ will be equal to two. According to this way, we can construct a column vector $\mathbf{r}_1$ and the sum of all entries in the $\mathbf{r}_1$ equals 200. For the whole database, we can generate a new matrix:

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_n] \tag{8}$$

where $\mathbf{R} \in \mathbb{R}^{K\times n}$, each column $\mathbf{r}_i \in \mathbb{R}^{K\times 1}$ represents an image. So, we can consider our method as a kind of dimensionality reduction technique when the value of K is far less than the dimensionality of original image.

4 Experimental Results

In this section, we first illustrate our experiment settings and then compare our scheme with other classical methods on two databases, i.e., ORL face database and Georgia Tech face database.

4.1 Experiment settings

In our experiments, when we repeat the classical methods, such as PCA, LDA and LPP, all the face images are first manually resized to a resolution of 32×32, with 256 gray levels per pixel. The pixel values are then scaled to $[0, 1]$. Each face image is represented as a 1024-dimensionality vector. In order to get a comprehensive comparison, we also test the Baseline method. For the Baseline method, the recognition accuracy is simply performed in the original 1024-dimensional image space without any dimensionality reduction. However, for our methods, each column vector based on SIFT features is used to describe an image. We obtain the 128 dimensional SIFT descriptor which densely extracted from image patches on a grid with step size of 6 pixels with patch size 16×16. We resize the maximum size (length or width) of each image to 300 pixels. In our method, we set the number of cluster center $K = 300$. The linear SVM will be used in all methods for the final classification. We randomly select some images per class as training data and use the rest as testing data. For getting a more stable estimation of recognition accuracy, all

the results for each group of training data and testing data are repeated 50 times. The average accuracy and the standard deviation are reported. All experiments are conducted in MATLAB, which is executed on a server with an Intel X5650 CPU (2.66GHz and 12 cores) and 32GB RAM.

4.2 ORL face database

The ORL face database[1] consists 400 images of 40 different subjects in PGM format. Each subject has 10 images. Subjects were asked to face the camera and no restrictions were imposed on expression, only limited side movement and limited tilt were tolerated. For most subjects, the images were shot at different times and with different lighting conditions, but all the images were taken against a dark homogeneous background. Some subjects were captured with and without glass.

In our method, we first extract the SIFT descriptors from each image by using the codes provided by [4]. Each descriptor is represented by a 128-dimensional column vector. The total number of SIFT descriptors extracted from the ORL face database is 88400, i.e., $\mathbf{S} \in \mathbb{R}^{128\times 88400}$. Then we apply the k-means clustering algorithm to generate 300 clusters, i.e., $\mathbf{U} \in \mathbb{R}^{128\times 300}$. At last, for the whole ORL face database, we can generate a new matrix $\mathbf{R} \in \mathbb{R}^{300\times 400}$ and each column of $\mathbf{R}$ represents an image.

We report the mean accuracy and standard deviation of the 50 different runs for the ORL face database with different training numbers in Table 1. In our experiment, we randomly select some images from each class and use the remaining images of each class to test. From the Table 1, we can see that the recognition accuracy achieved by using our method is higher than the other methods and the standard deviation of our method is the lowest. In the Figure 1, left image is a schematic diagram corresponding to the Table 1 and it clearly shows that recognition accuracy of our method always outperform other methods, furthermore, the standard deviation of our method is also lower than others. Right image shows the relationship between recognition accuracy and iteration number of five methods when we randomly select five images from each class to train and use the remaining images of each class to test. The x-axis denotes 50 different iterations and the y-axis is the corresponding recognition accuracy with respect to 50 different iterations.

Table 1: Recognition accuracy on the ORL face database (mean±std-dev)%

Method	2 Train	3 Train	4 Train	5 Train	6 Train	7 Train
Baseline	83.61 ±2.4	89.78 ±2.4	93.76 ±2.1	95.19 ±1.8	96.13 ±1.3	97.23 ±1.5
PCA	83.66 ±3.0	89.49 ±2.3	93.29 ±2.2	94.89 ±1.6	96.40 ±1.6	97.37 ±1.4
LDA	82.69 ±2.3	88.55 ±2.2	91.00 ±1.8	93.02 ±1.8	93.06 ±1.5	94.48 ±1.7
LPP	76.77 ±2.5	82.80 ±2.7	86.30 ±2.7	88.21 ±1.8	89.76 ±2.3	90.85 ±2.4
Our method	**87.24 ±1.9**	**93.02 ±1.6**	**95.73 ±1.5**	**97.01 ±1.1**	**97.94 ±1.2**	**98.65 ±1.0**

4.3 Georgia tech face database

The Georgia Tech face database[2] contains 750 images of 50 different subjects and is stored in JPG format. For each individual, there are 15 color images. Most of the images were taken in two

[1]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[2]http://www.anefian.com/face_reco.htm

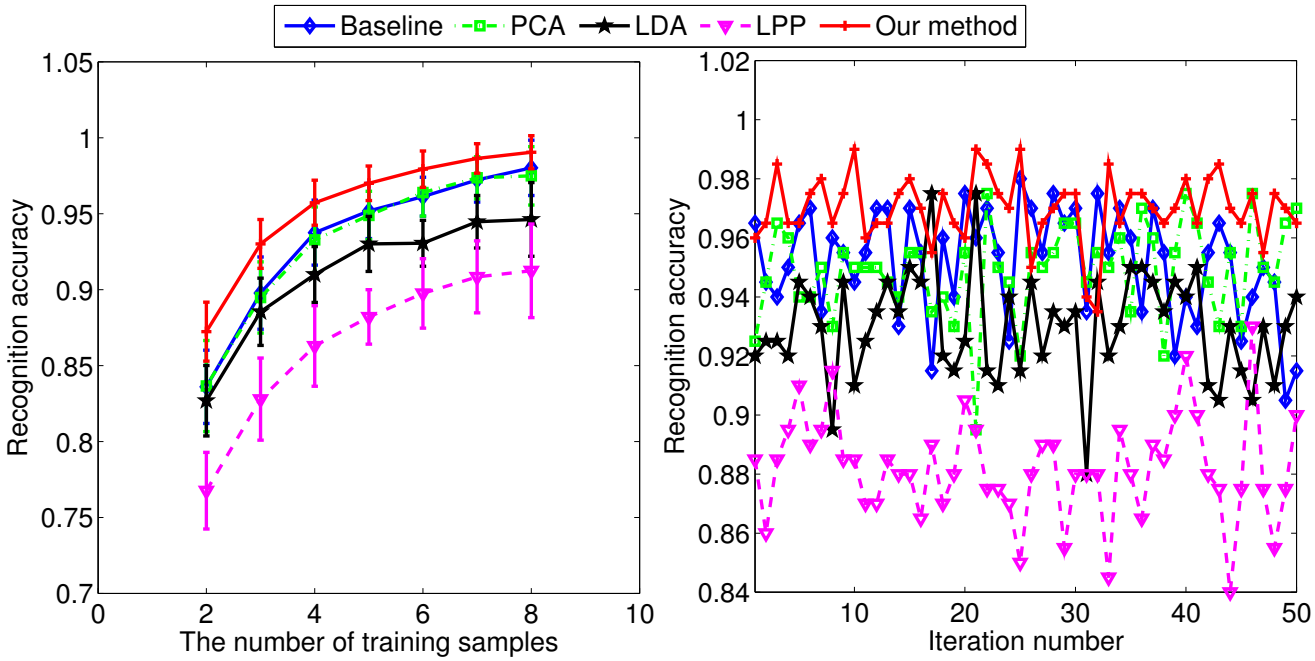

Fig. 1: Left image is recognition accuracy versus number of training samples on the ORL face database. Right image is recognition accuracy versus iteration number on the ORL face database.

different sessions to take into account the variations in illumination conditions, facial expression, and appearance. In addition to this, the faces were captured at different scales and orientations.

We do the same processing for each image like the operation on ORL face database. For the Georgia Tech face database, the total number of SIFT descriptors is 534461. The mean accuracy and standard deviation of the 50 different runs for the Georgia Tech face database with different training numbers are recorded in Table 2. From the Table 2, our method achieves more than 13% improvement in all cases over the best of the other methods. In the Figure 2, left image is a graphical representation corresponding to the Table 2. Right image shows the relationship between recognition accuracy and iteration number under different methods when we randomly select nine images from each class to train and use the remaining images of each class to test. It can reflect that our method achieves the best performance in each iteration.

Table 2: Recognition accuracy on the Georgia Tech face database (mean±std-dev)%

Method	6 Train	7 Train	8 Train	9 Train	10 Train	11 Train
Baseline	72.33 ±2.0	74.14 ±2.0	76.34 ±1.9	77.89 ±2.1	78.97 ±2.1	80.44 ±2.9
PCA	72.15 ±2.1	74.72 ±1.7	76.89 ±2.5	77.79 ±1.7	78.86 ±2.7	80.05 ±2.5
LDA	56.51 ±1.5	56.34 ±2.2	56.41 ±1.9	56.17 ±2.6	54.51 ±2.1	53.76 ±3.0
LPP	38.79 ±2.2	38.80 ±2.4	37.91 ±3.1	37.71 ±2.8	36.87 ±2.9	33.91 ±3.1
Our method	**88.42 ±1.5**	**89.99 ±1.2**	**91.49 ±1.4**	**92.43 ±1.5**	**93.01 ±1.6**	**94.15 ±1.6**

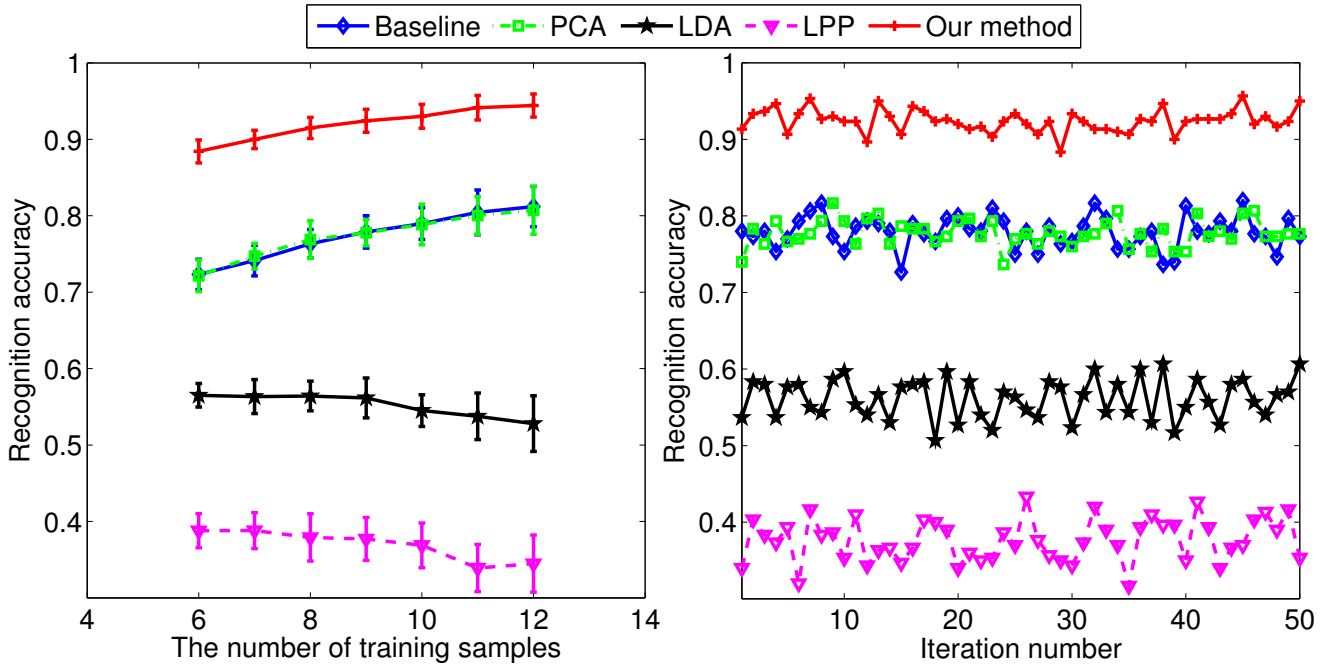

Fig. 2: Left image is recognition accuracy versus number of training samples on the Georgia Tech face database. Right image is recognition accuracy versus iteration number on the Georgia Tech face database.

5 Conclusions

In this paper, an efficient data dimensionality reduction scheme based on SIFT is proposed. We have compared our scheme for face recognition with four methods, including Baseline, PCA, LDA and LPP. In order to accelerate the speed of face recognition and improve the recognition accuracy, PCA, LDA and LPP are used as one dimensionality reduction technique to reduce the dimensionality of original image vector. In our method, we first use k-means to cluster all the SIFT descriptors of images and construct a cluster center matrix. Then, we represent each image as a column vector according to the distance between the each descriptor of image and cluster center matrix. Experiments on two databases demonstrate that recognition accuracy of our scheme is much better than the previous classical methods.

Acknowledgement

This work is supported in part by the National Basic Research Program of China (973 program) under Grant 2009CB320901, the National Natural Science Foundation of China under Grant 61272247, the National High Technology Research and Development Program of China (863 program) under Grant 2008AA02Z310, the European Union Seventh Framework Programme under Grant 247619, the Shanghai Committee of Science and Technology under Grant 08411951200, and the Innovation Ability Special Fund of Shanghai Jiao Tong University under Grant Z030026.

References

[1] Turk, M., Pentland, A.: Face recognition using eigenfaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (1991) 586–591.

[2] Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7) (1997) 711–720.

[3] Niyogi, X.: Locality preserving projections. In: Advances in neural information processing systems. Volume 16. (2004) 153.

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

[5] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2005) 886–893.

[6] Morel, J., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences **2**(2) (2009) 438–469.

[7] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: International Conference on Computer Vision. (2011).

[8] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3) (1995) 273–297.

[9] He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face recognition using laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(3) (2005) 328–340.