International Journal of Electronics and Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 4, Issue 4, Jun- Jul 2015, 29-38
© IASET

DIMENSIONALITY REDUCTION OF FEATURE DESCRIPTORS USING PCA AND SPARSE REPRESENTATION

ANITHA M MALLURKAR, THANMAYA H R, MANJULA AV & RAJESH N
Department of ECE, Nitte Meenakshi Institute of Technology, Bangalore, India

ABSTRACT

Feature matching is used in many applications such as environmental monitoring, aerial imaging, and surveillance etc. In this paper we discuss the methods for reducing dimensionality of feature descriptors. The extracted features are required to be invariant to different transformation of image like image rotation, scale change and illumination. The combination of different existing algorithms like SIFT, PCA and Sparse representation for feature detection and descriptor development has been discussed.

KEYWORDS: SIFT, PCA, SPARSE

INTRODUCTION

The market requirements encourage the development of high performance computer vision algorithms. A lot of different feature detectors and descriptors have successfully been applied in many computer vision tasks. However, these methods require high computation time and memory storage. Dimension reduction [2] is the process in which high-dimensional data [1] is transformed into a important representation of reduced dimensionality. Feature descriptors have wide variety of applications in face recognition, image matching, image retrieval, object recognition, image registration etc. Most of these Feature descriptors usually require high dimension to represent the objects of interest. Greater the dimension more will be the consumption of resources such as memory, space and computational time. Dimensionality reduction is important, as it permits visualization, classification, and compression of high dimensional data. The large number of dimensions generated by the descriptors can become a problem for certain types of tasks. To understand why we need this reduction, we can think about a video tracking application, in a film which consists of 30 frames per second, the descriptors here could detect hundreds of interest points in each frame. Even if we ignore some frames, the amount of information generated could be large, which would result in more consumption of space for memory storage and computational time.

Many dimensionality reduction techniques [2] were introduced in recent years. All these methods have the capacity to deal with complex nonlinear data. The commonly used techniques for dimensionality reduction are Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA).

Principal Components Analysis (PCA) [1] gives a low-dimensional representation of the data that describes most of variance in the data as possible. It is done by finding a linear basis of reduced dimensionality for the data, in which the amount of variance in the data is maximal. Linear Discriminant Analysis (LDA) maximizes the linear separability between data points belonging to different classes. It finds a linear mapping M that maximizes the linear class separability in the low-dimensional representation of the data. In this paper we chose to implement PCA because of its simplicity as it doesn’t require large computational time and also there is reduced complexity while using this technique with images.

PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) [1] is a standard technique for dimensionality reduction [2] and it has been used in computer vision problems such as feature selection, objects recognition and face recognition. PCA is a best algorithm used to represent key point patches[3] after they are transformed into a canonical scale, orientation and position, and this representation considerably increases SIFT's[4] matching performance. In general to reduce the dimension of large data sets, PCA uses vector space[1]. Using such mathematical projection, the original data set, which is usually composed of large variables, can often be interpreted in just a few variables that is the principal components.

In PCA amount of principal components is less than or equal to the amount of original variables. The transformation can be defined such that the main principal component that is the first principal component has the largest possible variance and hence accounts for maximum variability in the data and every component following in turn has the highest variance but under the constraint that it is orthogonal to the previous components. Principal components are orthogonal because they are the Eigen vectors of the covariance matrix, which is symmetric.

Basics Terminologies in PCA

Covariance

Covariance [1] is defined as a property which offers us the quantity of difference (variance) of two dimensions from its mean. It gives us a measure of distance performed on the pixel values of image and therefore provides the intensity measure of an image. Covariance can also be stated as a measure of how intensely correlated the two variables are. It is represented as follows

$$\mathrm{cov}(x_i x_j) = \frac{\mathrm{cov}(x_i x_j)}{\sigma_i \sigma_j} \tag{1}$$

Where σj is the standard deviations, it gives us the statistical Correlation of xi and xj

Covariance is found to be symmetric as given in the equation below

$$\mathrm{cov}(x, y) = \mathrm{cov}(y, x) \tag{2}$$

It is accomplished on the extracted matrix. It is calculated using the following formulae.

Suppose there are 'n' sets of variants given as follows {x1…………xn} then the 1st order "covariance matrix" is defined by

$$V_{ij} = \mathrm{cov}(x_i x_j) = \left(\left(x_i - \mu_i\right)\left(x_j - \mu_j\right)\right) \tag{3}$$

The term 'μ_i' is defined as the mean

The higher order matrices can be generally given as

$$V_{i_m j_n} = \left(\left(x_i - \mu_i\right)\left(x_j - \mu_j\right)\right) \tag{4}$$

Characteristics Equation

Characteristic equation is obtained from the Covariance matrix. It is a "cubic root equation". The roots which is said to be maximum are calculated using the "Cardans" technique. By this root we get the Eigen value which is principal component of the data and it exclusively recognizes an image.

The “Eigen Values” and the “Eigen Vectors:

Eigenvectors also called as characteristic vector of a matrix which is a vector. This vector points in a direction which is said to be invariant for transformations that are considered to be linear. For example let us consider a term “v” which is a vector and assume it to be non-zero, then it gives the Eigenvector of a square matrix “X”. Suppose Xv is a multiple scalar of the term “v”. let us now give this condition in the form of equation as follows :

$$A_v = \lambda_v \tag{5}$$

In the above equation the term “λ” is also considered as a scalar term, it is identified as the Eigen value or characteristic value associated with the Eigen vector “v”.

It does have a correspondence among (n × n) square matrices and hence it is a linear transformation from itself to “n” dimensional vector space. This is the purpose why we describe “Eigenvalues” and “Eigenvectors” as same, by either using the language of linear transformation or matrices. By using a symmetric matrix that is covariance matrix, we can determine orthogonal basis by determining the “Eigen values” and its resultant “Eigen vectors”. The Eigen vectors “ei” and the Eigen value “λ_i” gives us the solutions of the equation

$$c_x e_i = \lambda_i e_i \text{ , where i = 1,2, …… ,n} \tag{6}$$

For ease let us predict that the Eigen values "λi" are discrete. These standards can be set up in the following way that is by finding the solutions of the characteristic equation

$$|C_x - \lambda I| \tag{7}$$

“I” from above equation is known as the “identity matrix” which has a same order as Cx.

Also |.| denotes the determinant of matrix.

Here we face with a conflicting goal. That is we need to solve the problem by declining the dimension of the representation along with preserving the original information to a maximum extent. Hence PCA gives an appropriate approach to manage the compromise between reducing the information and making the problem simple.

Methodology of “Principal Component Analysis”

This method is formed on the hypothesis that high variance is present in higher data information. The input data is projected to a new co-ordinate space after the extraction of its features. And hence we get axis in new coordinate that signifies a principal constituent vector. The 1st principal component is the path that gives the maximum variance which means it is the direction along which the positive variance is extreme. The 2nd Principal Component is then given in a direction which is orthogonal to the 1st and for which the variance is extreme and so on.

Let us consider a data denoted in terms of a matrix “X” of order (m×n). The “n” columns are considered to be the samples or observations. The “m” rows are known as variables. Now we have to transform the matrix “X” linearly onto a matrix “Y” whose dimension is also (m×n), so that for a (m × m) matrix “P” we can write

$$Y = PX \tag{9}$$

The above equation (9) symbolizes “change of basis”.

Suppose if we consider “P” rows to be vectors (p1, p2, . . . , pm) and the columns of “X” to be the column vectors (x1, x2, . . . , xn) then it can be given in the form of equation as follows

$$P_x = \left(P_{x_1}\ P_{x_2} \dots \dots\ P_{x_n}\right) = \begin{bmatrix} p_1x_1 & p_1x_2 & \cdots & p_1x_n \\ p_2x_1 & p_2x_2 & \cdots & p_2x_n \\ \vdots & \vdots & \ddots & \ddots \\ p_mx_1 & p_mx_2 & \cdots & p_mx_n \end{bmatrix} = Y \tag{10}$$

It expresses the original data “X” which is projected onto a column of “P”. As a consequence we acquire the rows of “P” {p1, p2……….. pm} that gives us a fresh basis for demonstrating the columns of “X”. The rows of “P” will then be the principal component directions.

Properties of PCA

- It has Feature components which are said to be linear.
- Its main property is Dimensionality reduction.
- It has Gaussian distributed feature components.
- The feature components exhibit orthogonality.
- It is based on linear transformation

Steps to Implement PCA

The following are the steps to perform Principal Component analysis on a set of data

Get Data

Consider a data set with an observations of “p” variables, our goal lies in lessening the data in such a way that each and every observation could be described as only “A” variables, where (A < p).

The arrangement of data is done as a set of “n” vector data (x1…………xn), where “xi” symbolizes distinct gathered observation of “p” data variables and (x1…………. xn) are the row vector data, that has “p” columns.

Now place the row vectors into a single matrix X of order (n × p).

Mean Computation

Now we compute the mean for every dimension j=1,…., p and then dwell the computed mean into an mean vector “u” of order (p × 1).

$$u[j] = \frac{1}{n}\sum_{i=1}^{n} X[i, j] \tag{11}$$

Subtract the Mean with the Original Data

Subtract the mean from every data which gives us the average across each dimension. It is done to produce a data set whose mean is zero and gives us deviation from the mean. We do this for the purpose of centering the data. The observed mean “u” is then subtracted from every row of the data matrix “X” and thus we store the subtracted mean data in the (n × p) matrix

$$B = X - h_u\, T \tag{12}$$

The term “h” is a column vector of order (n×1).

Determine the Covariance Matrix

The (p×p) Covariance matrix “C” is now computed from the external product of matrix “B” with itself as given in the equation below

$$C = \frac{1}{(n-1)} B^{*} \cdot B \tag{13}$$

The operator “*” is the transpose conjugate operator.

Determining the “Eigen values” and “Eigen vectors” of the covariance matrix:

As we know that a Covariance matrix is also known as a symmetric matrix. The Eigen values and its vectors are computed for this matrix. This is a significant step, as they give us the worthwhile information about our data. Moreover, they offer with information about the data patterns.

Hence by this practice of taking the Eigen vectors, we extract the lines describing the data.

We then calculate matrix “V” of Eigen vectors that diagonalizes the covariance matrix C

$$V^{-1}\, CV = D \tag{14}$$

The term “D” is the diagonal matrix of C which gives us the Eigen values. Here we make the use of a built in algorithm for manipulating the “Eigenvectors” and “Eigenvalues”. The matrix “D” takes the order of (p × p) diagonal matrix.

D [k, l] = λk , for all values of k=1 which is the jth Eigenvalue of the covariance matrix

D [k,l] = 0 for all values of k ≠ l

Eigenvalues and Eigenvectors are then well-arranged and paired. That is jth Eigenvalue correspondence to the jth Eigenvector. We have to now arrange the columns of the Eigen vector “V” and Eigen value “D” in declining order of Eigen value. At this point w0e must be definite to keep accurate combinations between the columns in each matrix.

We then estimate the energy for all the Eigenvector as follows

$$g[j] = \sum_{k=1}^{j} D[k,k] \text{ , for } j = 1,\ldots\ldots\ldots, p \tag{15}$$

Selecting Components and Making a Feature Vector

Idea of reduced dimensionality and compression of data comes into picture in this step. That is the Eigen vector comprising of top Eigen value gives Principal constituent of data. The significant relationship is established among the data dimensions. After the determination of Eigen vectors, the succeeding action is sequencing the Eigen value from a range of high to low. We now get the constituents in sequence of their significance.

The reason to do this is to disregard the constituents of lesser significance. While doing this we drop certain information, but as the Eigen values are slight, we don’t miss considerable information. The resultant data will now have lesser dimensions than the input data.

Now choose a subclass of eigenvectors to be the basis vectors and then save 1st “A” columns of “V” as a matrix W with an order of (p ×A).

$$W[k,l] = V[k,l], \text{ for } k = 1,\ldots\ldots, p \quad (16)$$

$l = 1,\ldots\ldots, A$

Where $1 \leq A \leq p$

Here we introduce a term “g” denoted as a threshold in selecting a proper value assessment to “A”. We aim at selecting “A” in such a way that it is small along with attaining a sensibly greater value of “g” on the basis of percentage. To clarify this with an example, suppose if we select “A” such that the energy of “g” is exceeding certain threshold like 95%. Then we get the smallest value of A

$$\frac{g[A]}{g[p]} \geq 0.95 \quad (17)$$

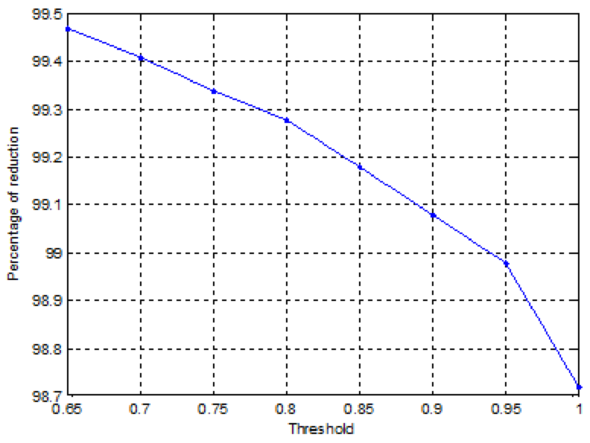

Figure 1: Graph Showing the Threshold Selection and its Corresponding Reduction

Changing the Input Data to Score “Z” and Calculate the (n × p) “Z” Matrix

$$Z = \frac{B}{h \cdot s'} \quad (18)$$

Projecting the Z Scores of Data onto the New Basis

This is done by using the equation below. This projected vector gives us the columns of the matrix

$$T = Z \cdot W \quad (19)$$

Advantages of Using PCA

PCA’s important advantages are low noise sensitivity, reduced requirements for memory and capacity and better efficiency. The advantages of PCA are discussed below:

- As it has got a orthogonal components there is lack of redundancy of the data
- There is reduced complexity while implementing with images.
- Due to a reduced database representation, we only need to store the trainee images in the form of the projections on a basis of reduction.

- It has got reduced noise because the maximum variation basis is selected and so the minor variations are ignored automatically in the background.

SIFT (Scale Invariant Feature Transform)

Important steps in SIFT are as follows:

- Scale space peak selection: This is first stage, where possible points are recognized by scanning the image in terms of location and scale.
- Localization of key points: In the second stage, key points are then localized to sub-pixel accuracy and discarded in case they are unstable.
- Orientation assignment: This stage detects orientations for every key points created on the patch.
- Key point descriptor: This section of the Sift forms a representation for every key point built on a pixels patch to compare local neighborhood.

PCA Based SIFT Descriptor

This algorithm for local descriptors [3] (termed PCA-SIFT) receives the same input as the standard SIFT descriptor, the sub-pixel location, scale, and dominant orientations of the key point. And then extract a 41×41 patch at the given scale, centered over the key point, and rotated to align its dominant orientation to a canonical direction.

Properties of a Patch

Each of the patches [3] should satisfy the following properties It should be centered on a local maximum in

PCA-SIFT descriptor:

PCA-SIFT descriptor [3] is implemented as given below

- Select a representative set of pictures: Detect all key in these pictures.
- Extraction of image patch: For each key point extract an image patch around it with size 41*41 image patch around the feature point, and then rotate the image patch to a direction using a transformation matrix.
- Computation of image gradient: Except every pixel on the edge around the patch. Calculate the gradient of each pixel in both horizontal and vertical directions. Therefore 39*39*2=3042 dimension vector is obtained.
- Projection of descriptor: Firstly we have to calculate the projection matrix, if there are “s” key-points detected, then the projection matrix has P= s*3042 vectors. The covariance matrix R is obtained of matrix P. The Eigen values [1] $\lambda_1,\lambda_2,\ldots\ldots\ldots\ldots\ldots,\lambda_{3042}$ and Eigen vectors $e_1,e_2,\ldots\ldots\ldots\ldots,e_{3042}$ of the matrix A are calculated.

The Eigen values are then sorted in descending order and the first 20 values are selected. Their corresponding Eigen vectors are chosen to be the principal components direction, thus we have 20*3042 projection matrix. Thus the 3042 element vector $(d_1,d_2,\ldots\ldots\ldots\ldots\ldots\ldots d_{3042})$ is projected onto the feature space by projection matrix P and thus reduced to a 20 dimensional PCA-SIFT descriptor[3].

Descriptor Matching

The best match [5] for each key-point is determined by identifying its nearest neighbor in the database of key-point. The nearest neighbor is defined as the key-point with minimum Euclidean distance for the invariant descriptor vector. Most of the features from an image will have not have correct match in the database because they origin from background clutter or are not detected in the training images. Hence it would be useful to eliminate features that do not have any good match to the database. A threshold is set on distance to the closest feature and it does not perform well, as few descriptors are more discriminative than others. A better measure can be the comparing of distance of the closest neighbor to that of the second-closest neighbor. If there are multiple training images of the same object, then we define the second-closest neighbor as being the closest neighbor.

The figure 2 below describes the graph for percentage of matching of two images at different angles for both SIFT and PCA-SIFT and figure 3 shows the matching operation with respect to scaling.

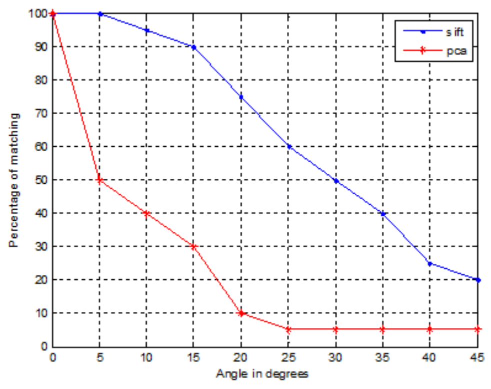

Figure 2: SIFT vs. PCA-SIFT on a Matching Task where the Images are Rotated to Different Angles

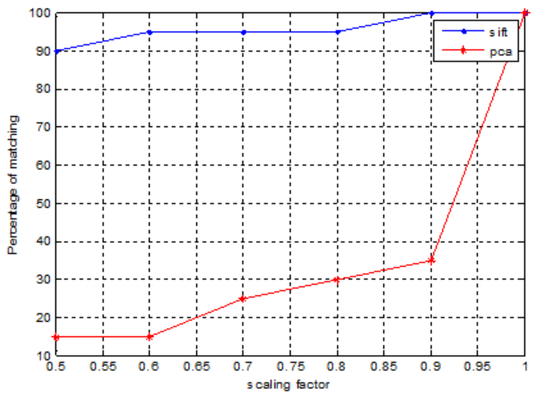

Figure 3: SIFT vs PCA-SIFT on a Matching Task where Images are Scaled

SPARSE APPROXIMATION

The "Sparse Approximation"[6] describes the low-dimensional data structure. The model uses a dictionary. The model assumption is that every data vector from the source could, in principle, be described as a linear combination of few atoms from the dictionary. Sparse vectors require less space when being stored on a computer as only the position and value of the entries need to be recorded.

In image feature extraction [8], the feature points of an image are to be detected, followed by depicting every feature point using a descriptor. Then, sparse representation for all the descriptors of an image has been proposed. The advantage is that a feature descriptor is sparsely represented in terms of a dictionary, in order to obtain simple but efficient feature representation. The fundamental unit achieved by arranging a set training data can be termed as atom. The sparse representation of the data is linear combination of the few atoms as that of the dictionary.

Sparse Representations in Mathematical Terms

Let y be a vector of dimension N and D a matrix of dimension N × M with M << N. The columns dk of D can be seen as basis functions or atoms of a dictionary that will be used to represent the vector y. Note that there is an infinite number of ways to choose the M dimensional vector X such that y = Dx. The aim of sparse representations is to search among all these solutions of y = Dx, that are sparse, Indeed one quite generally does not seek an exact reconstruction but rather seeks a sparse representation that satisfies:

$\|y - Dx\|_2^2 < e$ (20)

Where ‘e’ characterizes an admissible reconstruction error

Greedy Algorithms for Approximation of Sparse Vector

These algorithms attempt to build the support of y, one non-zero element at a time. The most basic of algorithm is the Matching Pursuit (MP)[7]. This is an iterative algorithm that starts by finding the one atom that best describes the input signal, i.e., finding the index

$\hat{\imath} = \text{argmin}_i \, \text{min}_c \, \|y - c \cdot d_i\|_2^2$ (21)

Once found, compute the signal residual as y − c di with the optimally found constant c and the atom di. Then, at each iteration, one atom that best describes the residual is added to the support, with the appropriate coefficient contributing to reducing the residual the most. This process is repeated until the norm of the threshold drops below the given threshold. There are several variants of Matching pursuit algorithms such as: Orthogonal matching pursuit(OMP), Stage-wise matching pursuit(StOMP), Regularized matching pursuit(ROMP) and so on. Every time the atom is added OMP algorithm re-computes set of coefficients. StOMP is similar to OMP, but the difference is that, in StOMP several atoms are added to the vector at a time. StOMP [9] compares the values of the dot product of y with the columns of D and the same is repeated with a residue vector. An advantage of this method is that it can produce a good approximation with a small number of iterations and the disadvantage is determining an appropriate value for the threshold. ROMP does not use a pre-set threshold value instead it uses the vectors which have a similar dot product with the required vector. The advantage of ROMP is that vectors are used which would a similar contribution to the required vector. This process will be repeated by updating residue vector at each iteration.

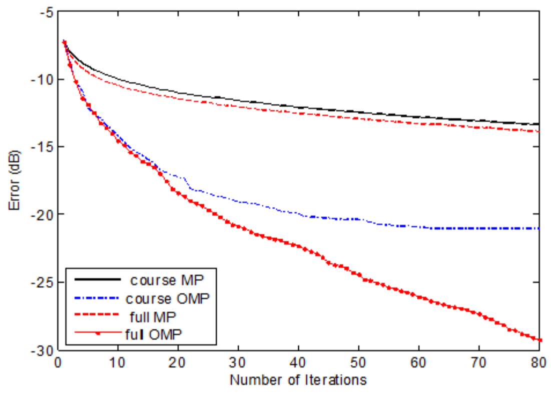

Figure 4: Comparison of MP and OMP Algorithms for Reconstruction Error

Figure 4 gives the Comparison iteration between algorithm MP and OMP with fixed coarse sub-dictionary and full dictionary. Input signal is of dimension N= 128. Input signal is N/2 vectors chosen from D(dictionary) with normal weights. Here fixed sub-dictionary cases, i.e. Coarse MP/OMP, have saturated pattern since the sub-dictionary. Even if working on sub-dictionaries four times smaller, performing pursuits on the full dictionaries gives the better outcome. It can be observed from the, Figure 4, that Potential gains are more in the case of OMP algorithm with full dictionary. The probabilistic parsing of the full dictionary has better error decaying rate than when the fixed sub-dictionary (Coarse OMP) is used.

CONCLUSIONS

In this paper, feature dimension reduction techniques based on PCA and Sparse techniques have been discussed. PCA reduces the feature dimension for the tune of at least 15 times as compared to SIFT. However PCA-SIFT method is not as robust as SIFT. Five different Sparse Approximation Algorithms for Sparse Vector creation has been discussed.

REFERENCES

1. Jonathon Shlens, “A Tutorial on Principal Component Analysis”
2. Padraig Cunningham University College Dublin Technical Report On Dimensionality Techniques Ucd-Csi-20.
3. Yan Ke And R. Sukthankar, “Pca-Sift: A More Distinctive Representation For Local Image Descriptors,” Ieee Computer Society Conference On Computer Vision And Pattern Recognition, 2004, Pp. 506–513.
4. D. G. Lowe, “Object Recognition from Local Scale-Invariant Features,” In Ieee International Conference On Computer Vision. Ieee Computer Society, 1999, Pp. 1150–1157.
5. Ricardo Eugenio, William Robson Schwartz, Helio Pedrini “Dimensionality Reductions Through Pca Over Sift And Surf Descriptors”.
6. Philip Breen “Algorithm for Sparse Approximation” School of Mathematics University of Edinburgh 2009
7. Chao-Yung Hsu, Hung-Wei Chen, Chun-Shuen Lu, “Feature Based Sparse Representation For Image Similarity Assessment”, Ieee Trancation On Multimedia, Vol.13, No.5, October 2011
8. Mamuel Moussallam, L. Daudet, G Richard, “Matching Pursuit with Random Sequential Sub-Dictionaries”. Signal Processing Elsevier, May 2012
9. Yaakov Tsaig David L Donoho, “Sparse Solution of Underdetermined Linear”, March 2006