2010 International Conference on Pattern Recognition

Dimensionality Reduction for Distributed Vision Systems Using Random Projection

Vildana Sulić, Janez Perš, Matej Kristan, and Stanislav Kovačič
Faculty of Electrical Engineering
University of Ljubljana
Trzaska 25, SI-1000 Ljubljana, Slovenia
vildana.sulic@fe.uni-lj.si

***Abstract*—Dimensionality reduction is an important issue in the context of distributed vision systems. Processing of dimensionality reduced data requires far less network resources (e.g., storage space, network bandwidth) than processing of original data. In this paper we explore the performance of the random projection method for distributed smart cameras. In our tests, random projection is compared to principal component analysis in terms of recognition efficiency (i.e., object recognition). The results obtained on the COIL-20 image data set show good performance of the random projection in comparison to the principal component analysis, which requires distribution of a subspace and therefore consumes more resources of the network. This indicates that random projection method can elegantly solve the problem of subspace distribution in embedded and distributed vision systems. Moreover, even without explicit orthogonalization or normalization of random projection transformation subspace, the method achieves good object recognition efficiency.**

***Keywords*-dimensionality reduction; random projection; distributed vision systems;**

I. Introduction

Dimensionality reduction techniques are especially important in the context of embedded vision systems, such as smart cameras. The reasons for that are the specific characteristics and limitations of smart cameras (i.e., low processing power, limited storage space and limited network bandwidth) [1]. Processing of dimensionality reduced data usually requires far less scarce resources that are available in such systems. One of the widely used methods for dimensionality reduction is Principal Component Analysis (PCA). PCA (also known as the Karhunen-Loéve transform) is a vector-space transform that reduces multidimensional data sets to lower dimensions in a way that low-dimensional subspace of the data describes as much of the variance in the data as possible. When PCA is used for object recognition in the distributed sensor network, each sensor needs an access to the low-dimensional subspace. However, access to or distribution of a subspace is a problem by itself. In practice this subspace would have to be transmitted to each and every node in the network, which would cause significant network traffic. Alternatively, cameras could use fixed subspace, which is built into the camera at the time of manufacture or installation. Therefore, the question that we pose is: *How to make available common subspace to all cameras in a visual-sensor network (VSN, network of embedded cameras) without transmitting large amount of data?*

Using Random Projection (RP), where original high-dimensional data is projected onto a low-dimensional subspace using a random matrix, there is a possibility that all cameras in the network recreate exactly the same subspace with minimum of transmitted information. This is possible, if pseudorandom generator is used to generate the random matrix. The only information that has to be known to each camera in the network is the pseudorandom *state*. In practice, it is sufficient that each camera knows only one parameter, called *seed* of the pseudorandom generator. Since same seed results in exactly the same pseudorandom sequence, recreation of the same random matrix in each camera is possible. This means that each camera in the network projects the input data (images) through the same random matrix into the same subspace.

To find out if RP can solve the problem of subspace distribution, we examine the recognition efficiency of RP and compare it to the well-known PCA. Other researchers have explored RP in the context of various applications. The most closely related works are work by Goel et al. [2] and work by Fradkin et al. [3]. In the first work authors investigated the feasibility of the RP for the face recognition. They performed a large number of experiments involving PCA. They used Gram-Schmidt algorithm for orthogonalization of random matrix, which is computationally demanding, and therefore, less appropriate for embedded systems. In the second work authors performed extensive comparison between PCA and RP using different data set and classification methods, however, they did not use visual data, which is used in our case.
RP can be implemented in many ways, and not all of them are appropriate for the resource-constrained embedded cameras. Thus, we decided to explore different variants of RP and their effect on recognition performance. For instant, in standard RP approach, the matrix is normalized and for this reason we are interested also in recognition performance of non-normalized vectors. Moreover, we are also interested in performance of RP, if its transformation vectors are sorted according to the highest variance in the data, as it is done in the PCA.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.101

II. Random projection

RP has been found computationally efficient and a sufficiently accurate method for dimensionality reduction of highly dimensional data sets (e.g.,[1], [4], [5], [6]). The concept of RP is as follows: Given a data matrix $\mathbf{X}$, the dimensionality of data can be reduced by projecting it onto a lower-dimensional subspace formed by a set of random vectors [4],

$$\mathbf{A}_{[k\times N]} = \mathbf{R}_{[k\times d]} \cdot \mathbf{X}_{[d\times N]}, \tag{1}$$

where N is the total number of points, d is the original dimension, and k is the desired lower dimension. The central idea of RP is based on the Johnson-Lindenstrauss (JL) lemma [7], which states that if we perform projection of n points in some high-dimensional space onto a random $O(\log n)$-dimensional plane, then the distances between points are preserved (i.e., not distorted more than a factor of $1 \pm \varepsilon$), for any $0 < \varepsilon < 1$.

Selecting the random matrix An efficient method for dimensionality reduction based on the JL lemma employs a random matrix $\mathbf{R}$, whose elements are drawn independently and identically distributed (i.i.d.) from a zero mean, bounded variance distribution [1]. There are many choices for the random matrix. A random matrix with elements generated by a normal distribution $r_{i,j} \sim N(0,1)$ is one of the simplest in terms of analysis [8]. The problem of this type of RP is its computational complexity due to the dense nature of the projection matrix. Achlioptas [6] suggested the use of two simpler distributions that generate sparse projection matrices with elements drawn i.i.d. as:

$$r_{i,j} = \begin{cases} +1 & \text{with probability } \frac{1}{2} \\ -1 & \text{with probability } \frac{1}{2} \end{cases} \tag{2}$$

or

$$r_{i,j} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ -1 & \text{with probability } \frac{1}{6} \end{cases} \tag{3}$$

Distributions shown in Eq.(2) and (3) reduce computational time for the calculation of $\mathbf{R} \cdot \mathbf{X}$. For the second distribution, the speedup is threefold because only one-third of the operations are needed. In this work, we use the sparse projection matrix, presented in Eq.(3).

Orthogonalization If the random vectors were orthogonal, then the similarities between the original vectors would be preserved under RP [4]. Ideally random matrix should be orthogonal but, unfortunately, orthogonalization is very costly (e.g., a complexity of Gram–Schmidt method is $O(nm^2)$ if m principal base vectors of dimension n have to be determined [9]). The cost of the orthogonalization defeats the purpose of using RP on resource-constrained embedded cameras. However, in practice, it seems that in high-dimensional space, there are far more nearly orthogonal than truly orthogonal vectors, which is also shown in [10]. Hence it follows that random vectors might be sufficiently close enough to orthogonal to offer a reasonable approximation of the original vectors [11] and for this reason we decided to use non-orthogonal random matrix.

III. Experimental analysis

We performed a series of experiments, including both PCA and RP. The recognition performance of both methods in terms of percentage of correctly classified and misclassified objects on a standard database was examined.

Data We have used the standard COIL-20 database, which consists of images of 20 different objects; each one is rotated with 5 degree angle interval, corresponding to 72 images per object. That sums up to 1,440 images for the whole database [12].

Preprocessing Image dimensions were the same for both methods, i.e., 128×128 pixels. First, the database was split into two parts of approximately the same size (training set and testing set). For every object, every second image was selected for recognition (testing), and the remaining images were used to build PCA subspace and for training. Images selected for recognition were transformed into the PCA and RP subspaces. This way feature vectors for both methods were obtained.

Protocol All vector projections from the testing set were compared to the projections of every second image (training set) and using Euclidean distance in the RP subspace, the nearest neighbor was found. If the nearest neighbor was from the same class (one class corresponds to all images of the same object), the comparison resulted in a match between the image from the training set and the tested image. If the match was found between samples of two different classes, the number of misclassified was increased. This protocol was conducted separately for PCA and RP. Additionally, to rule out the influence of seed selection, experiments for RP were repeated ten times with different seeds of a pseudorandom generator. All the experiments were conducted several times with different degrees of dimensionality reduction (i.e., for subspace dimensions of 512, 256, 128, 64, 32, 16, 8, 4, 2, and 1).

Benchmarking First, performance of the PCA was tested to obtain reference values for percentage of correctly classified and misclassified objects.

Random projection Major portion of experiments were dedicated to the RP method. We explored the following: how results vary with different seeds, whether is transformation vector normalization really necessary, and whether does the method benefit from selection of the transformation vectors based on the preserved variance in data (as it is in the PCA).

Random projection matrix $\mathbf{R}$ was generated following the suggestion by Achlioptas [6] as defined in Eq.(3). The maximum dimensions of the matrix were 512×16384 elements. The Mersenne twister [13] was used as pseudorandom generator. First, we checked that equal seeds result in equal pseudorandom sequences and then precalculated RP matrices for ten randomly generated seeds. In practical applications the matrices would be calculated on the fly on each of the cameras. Similar as with the PCA, every second image in the database was transformed to the random subspace. Procedure was repeated for all ten different RP matrices in combination of normalized/non-normalized vectors (important for preserving similarities in the low-dimensional space) and sorted/unsorted vectors according to the highest preserved variance in data (as in the PCA). For this reason, a larger random projection matrix was generated and rows which modelled the most of the data variance were selected and therefore, rows were sorted accordingly.

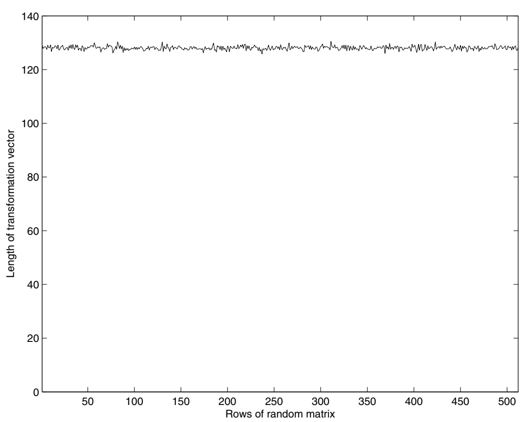

Figure 1. Length of the rows of the random matrix.

Results Results are shown in Table I. We can see that even if only 16 feature vectors are used, RP correctly classifies objects with almost 97% efficiency. Although PCA correctly classifies objects with 100% it is still necessary to distribute its (data dependent) subspace. We can conclude that our results support the use of RP instead of PCA in embedded systems. First, RP can elegantly solve the problem of shared subspace distribution. Additionally, the results show that there is no significant difference if RP vectors are normalized or not. Indeed, if we observe the length of RP projection vectors, shown in Figure 1 it is obvious that they already have similar lengths. The mean length of RP vectors is $128,062 \pm 0,7161$. This

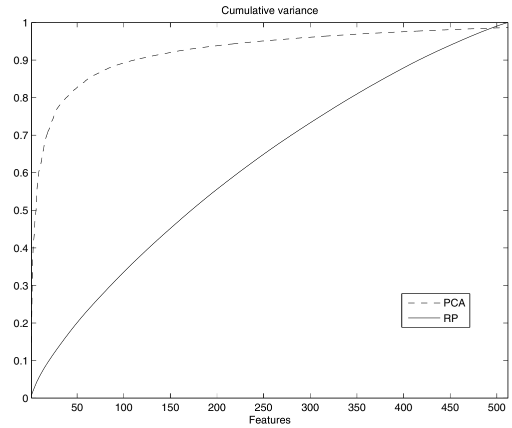

Figure 2. Cumulative variance for the PCA and RP calculated from the projection through the PCA subspace and the random matrix $\mathbf{R}$ for the RP.

is not surprising due to the dimensionality of input data – each vector has 16384 elements which can take only three possible values (0 and ± 1). Therefore, by discarding RP vector normalization, only scaling factor is introduced into the transformation. The results also show that there is no significant difference if RP vectors are sorted according to preserved variance or not. Figure 2 shows the cumulative variance across individual RP dimensions. The differences are too small to have significant impact on method performance. In contrast, same figure shows that PCA indeed compresses information in small number of features. We can conclude that the most appropriate RP subspace for the resource-constrained embedded systems is non-normalized and unsorted.

IV. Discussion and conclusion

The focus of our work was random projection and its potential use in embedded vision systems. RP projects original high-dimensional data through the random matrix onto the low-dimensional space. Data in low-dimensional space consume less resources, which are usually scarce in embedded systems. RP is data independent, preserves the structure of the data without significant distortion, is computationally simple and it does not require distribution of shared subspace. On the other hand, while PCA is known to give good results and has a lot of useful properties, it is also computationally expensive and requires distribution of shared subspace, which consumes more resources in the network.

Performance of the RP was compared to the performance of the PCA in terms of recognition efficiency on the COIL-20 image data set. Results show good performance of the RP even without explicit orthogonalization (computationally demanding) or normalization of transformation subspace. Moreover, sorting of feature

Table I
EXPERIMENTAL RESULTS FOR DIFFERENT SUBSPACE TYPES AND DEGREES OF DIMENSIONALITY REDUCTION

Subspace type	Number of feature vectors									
	512		256		128		64		32	
	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]
PCA	100.00	0	100.00	0	100.00	0	100.00	0	100.00	0
RP, normalized	99.97	0.059	99.93	0.073	99.88	0.102	99.58	0.254	98.92	0.485
RP, non-normalized	99.97	0.059	99.93	0.073	99.87	0.102	99.58	0.254	98.93	0.494
RP, non-normalized, sorted	99.97	0.059	99.94	0.072	99.88	0.110	99.74	0.231	98.96	0.431
RP, normalized, sorted	99.97	0.058	99.94	0.072	99.87	0.122	99.74	0.231	98.97	0.415
Subspace type	Number of feature vectors									
	16		8		4		2		1	
	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]	$\bar{x}^*$ [%]	σ[%]
PCA	100.00	0	99.44	0	94.44	0	71.80	0	31.80	0
RP, normalized	96.89	0.924	87.99	2.212	61.11	3.657	28.94	3.838	12.36	2.199
RP, non-normalized	96.92	0.921	87.97	2.212	61.18	3.703	28.96	3.830	12.38	2.276
RP, non-normalized, sorted	96.44	0.541	88.29	1.699	62.29	4.584	29.07	4.388	13.90	2.012
RP, normalized, sorted	96.42	0.539	88.30	1.665	62.29	4.598	29.11	4.457	13.86	1.916

* $\bar{x}$ represent percentage of correctly classified objects, and is in the case of RP computed as the average of results across 10 different seeds of a pseudorandom generator.

vectors in accordance to the preserved variance (as it is in the PCA) did not pay off either. Our results indicate that RP can be used in embedded systems and distributed visual-sensor networks, where RP can elegantly solve the problem of shared subspace distribution.

Our future efforts will also include testing of Fast Johnson-Lindenstrauss Transform [14], which is known as method that can generate orthogonal projections with very low computational complexity.

ACKNOWLEDGMENT

The work was supported by research program P2-0095 and research grant 1000-07-310221, both by Slovenian research agency.

REFERENCES

[1] G. Tsagkatakis and A. Savakis, "A random projections model for object tracking under variable pose and multi-camera views," in *Proc. of ICDSC*, 2009, pp. 1–7.

[2] N. Goel, G. Bebis, and A. Nefian, "Face recognition experiments with random projection," vol. 5779, no. 1. SPIE, 2005, pp. 426–437.

[3] D. Fradkin and D. Madigan, "Experiments with random projections for machine learning," in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 517–522.

[4] S. Kaski, "Dimensionality reduction by random mapping: fast similarity computation for clustering," in *Proc. of IJCNN*, 1998, pp. 413–418.

[5] S. Dasgupta, "Experiments with random projection," in *Proc. of UAI*, 2000, pp. 143–151.

[6] D. Achlioptas, "Database-friendly random projections," *J. Comput. Syst. Sci.*, vol. 66, no. 4, pp. 671–687, 2003.

[7] W. B. Johnson and J. Lindenstrauss, "Extensions of lipshitz mapping into hilbert space," in *Contemporary Mathematics, Conference in modern analysis and probability*, vol. 26, 1984, pp. 189–206.

[8] S. S. Vempala, *The Random Projection Method*. American Mathematical Society, 2004, vol. 65.

[9] R. Möller, "First-order approximation of gram-schmidt orthonormalization beats deflation in coupled pca learning rules," *Neurocomput.*, vol. 69, no. 13-15, pp. 1582–1590, 2006.

[10] R. Hecht-Nielsen, "Context vectors: General purpose approximate meaning representations self-organized from raw data," in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II, and C. J. Robinson, Eds. IEEE Press, 1994.

[11] J. Lin and D. Gunopulos, "Dimensionality reduction by random projection and latent semantic indexing," in *Proc. of SDM*, 2003.

[12] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20), cucs-005-96," Department of Computer Science, Columbia University, Tech. Rep., 1996.

[13] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[14] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast johnson-lindenstrauss transform," in *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM, 2006, pp. 557–563.