Computer Vision Winter Workshop 2010, Libor Špaček, Vojtěch Franc (eds.)
Nové Hrady, Czech Republic, February 3–5
Czech Pattern Recognition Society

Efficient Dimensionality Reduction Using Random Projection

Vildana Sulić, Janez Perš, Matej Kristan, and Stanislav Kovačič

Faculty of Electrical Engineering, University of Ljubljana, Slovenia
vildana.sulic@fe.uni-lj.si

Abstract *Dimensionality reduction techniques are especially important in the context of embedded vision systems. A promising dimensionality reduction method for a use in such systems is the random projection. In this paper we explore the performance of the random projection method, which can be easily used in embedded cameras. Random projection is compared to Principal Component Analysis in the terms of recognition efficiency on the COIL-20 image data set. Results show surprisingly good performance of the random projection in comparison to the principal component analysis even without explicit orthogonalization or normalization of transformation subspace. These results support the use of random projection in our hierarchical feature-distribution scheme in visual-sensor networks, where random projection elegantly solves the problem of shared subspace distribution.*

1 Introduction

In visual systems we usually deal with large amounts of digital image data. Data has to be archived or exchanged between numerous users and systems [3], consuming expensive resources, such as storage space or transmission bandwidth.

In digital imaging the basic unit of image is pixel. Therefore, an image can be represented as a feature vector, where each pixel corresponds to one feature value. Even standard resolution images (e.g., VGA) contain large number of pixels and therefore the resulting feature vector representation has usually high dimensionality. In order to handle real-world data adequately, the dimensionality needs to be reduced [27]. Using dimensionality reduction techniques is especially important when data is distributed across networks with limited bandwidth.

A dimensionality reduction technique that is capable to reduce the data into a lower-dimensional model, while preserving the reconstructive or discriminative properties of the original data can be marked as ideal. However, in practice information is lost as the dimensionality is reduced. Therefore, a method which efficiently reduces dimensionality, while preserving as much as possible information from the original data is needed. One solution is to reduce the dimensionality of data by projecting it onto a lower-dimensional subspace [18].

Dimensionality reduction techniques using linear transformations have been very popular in determining the intrinsic dimensionality of the manifold as well as extracting its principal directions (i.e., basis vectors). The most famous method in this category is the Principal Component Analysis (PCA) [11]. PCA (also known as the Karhunen-Loéve transform) is a vector-space transform that reduces multidimensional data sets to lower dimensions while minimizing the loss of information. A low-dimensional representation of the data is constructed in such a way that it describes as much of the variance in the data as possible. This is achieved by finding a linear basis of reduced dimensionality for the data (a set of eigenvectors) in which the variance in the data is maximal [27].

Besides PCA, many other dimensionality reduction techniques exist. Recently, Random Projection (RP) [28, 16, 5] has emerged as a powerful method for reducing dimensionality. The most important property of the RP method is that it is a general data reduction method. Unlike PCA, it does not depend on a particular training data set. Unlike Discrete Cosine Transform (DCT) or Discrete Fourier Transform (DFT) its basis vectors do not exhibit particular frequency or phase properties.

In RP, the original high-dimensional data is projected onto a low-dimensional subspace using a random matrix, whose columns have unit length. If compared to other methods, for instance PCA, which compute a low-dimensional subspace by optimizing certain criteria (e.g., PCA finds a subspace that maximizes the variance in the data), RP does not use such criteria, therefore, it is data independent. Furthermore, it represents a computationally simple and efficient method that preserves the structure of the data without significant distortion [11]. There exist theoretical results supporting that RP preserves for example volumes and affine distances [19] or the structure of data (e.g., clustering) [6].

1.1 Motivation

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) [26]. Processing of dimensionality reduced data usually requires far less resources than processing of unmodified data. In our previous work [24, 25] we proposed a framework of hierarchical feature-distribution (HFD) for object recognition in a network of visual sensors, which utilizes network in a more balanced way than trivial network flooding. HFD for visual-sensor networks (VSNs) is based on hierarchical distribution of the information, where each individual node retains only a small amount of information about the objects seen by the network. Nevertheless, this amount is sufficient to efficiently route queries through the network without any degradation in the recognition performance. The amount of data transmitted through the network can be significantly reduced using our hierarchical distribution.

One of the methods for image feature extraction used in our work was PCA. We used the variant of the PCA, which assumes that the eigenvectors (the subspace) were obtained in advance. We assumed that each sensor has an access to the global subspace. However, distribution of subspace is problem by itself. In practical implementation 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 problem that we are aiming to solve is as follows. *How to distribute common subspace to all cameras in a network without transmitting large amount of data?*

Using RP, 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 thing 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 pseudorandom 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 same subspace.

Therefore, since RP method enables reduction of data dimensionality, is computationally simple, preserves the structure of the data and is increasingly used in distributed visual-sensor networks, we decided to examine the recognition efficiency of RP and compare it to the well-known PCA. Other researchers have explored RP in the context of many applications, but to best of our knowledge, a detailed comparison to PCA has not been done in the terms of recognition efficiency. Moreover, RP can be implemented in many ways, and many of them are not appropriate for the resource-constrained embedded cameras. Therefore, our aim is to determine the performance of RP implementation, which can be directly used in embedded cameras. Furthermore, we decided to explore several variant of RP and their effect on recognition performance. Therefore, our main contribution is extensive comparison between RP and PCA (our baseline) in terms of recognition accuracy, in a way that is directly relevant to use of RP in embedded camera systems and VSNs.

The remainder of this paper is organized as follows. In the Section 2 we provide theoretical background of the RP, including selection of the random matrix and its orthogonalization, and we present different applications of the RP. Experiments and results of tests are reported and discussed in the Section 3. Section 4 provides discussion and conclusion.

2 Random projection

Random projection is a powerful dimension reduction technique that uses random projection matrices to project data from high-dimensional subspace to a low-dimensional subspace [11]. The RP technique has been used for designing algorithms for problems from a variety of areas, such as combinatorial optimization, information retrieval and machine learning [28]. Some theoretical background of RP and a brief review of its applications is presented below.

2.1 Background

The main idea of RP is that using a random matrix whose columns have unit lengths, the original high-dimensional data is projected onto a lower-dimensional subspace [3]. RP has been found computationally efficient and a sufficiently accurate method for dimensionality reduction of highly dimensional data sets (e.g., [14, 16, 4, 6, 1] just to name a few).

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 [16],

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

where N is the total number of points, d is the original dimension, and m is the desired lower dimension. The central idea of RP is based on the Johnson-Lindenstrauss lemma (JL lemma) [15]:

Johnson-Lindenstrauss lemma

For any $0 < \varepsilon < 1$ and any integer n, let k be a positive integer such that k

$$k \geq 4\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \ln n. \tag{2}$$

Then for any set V of n points in $\mathbb{R}^d$, there is a map $f : \mathbb{R}^d \mapsto \mathbb{R}^k$ such that for all $u, v \in V$ [7],

$$(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2, \tag{3}$$

where $f(u)$ and $f(v)$ are the projections of u and v.

Using the above lemma, [7] shows that if we perform an orthogonal projection of n points in a vector space ($\mathbb{R}^d$) onto a selected lower-dimensional subspace, then distances between points are preserved (i.e., not distorted more than a factor of $1 \pm \varepsilon$), for any $0 < \varepsilon < 1$. For complete proofs on the lemma refer to [10, 7]. The JL lemma can be proven for sparse transformation matrices $\mathbb{R}$ as well, for details see [1, 17, 2].

Selecting the random matrix. The choice of random matrix $\mathbf{R}$ is one of the crucial points of interest.

An efficient method for dimensionality reduction using 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 [26]. 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 [28]. The problem of this type of RP is its computational complexity due to the dense nature of the projection matrix. Achlioptas [1] suggested the use of two simpler distributions that generate sparse projection matrices with elements drawn i.i.d. as:

$$r_{i,j} = \begin{cases} +1 & \text{with probability } \frac{1}{2} \\ -1 & \text{with probability } \frac{1}{2} \end{cases} \tag{4}$$

or

$$r_{i,j} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ -1 & \text{with probability } \frac{1}{6} \end{cases} \tag{5}$$

Distributions shown in Eq.(4) and (5) 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. Authors in [17] further explored the idea of sparse RP and suggested a method for achieving a $\sqrt{n}$ - fold increase in speedup with a small penalty in the preservation of the pairwise distances. In this work, we use the sparse projection matrix, presented in Eq.(5). Usage of sparse RP in a distributed environment is presented in [29].

Orthogonalization If the random vectors were orthogonal, then the similarities between the original vectors would be preserved under RP [16]. Ideally we would want the random matrix to be orthogonal but, unfortunately, orthogonalization is very costly (e.g., Gram–Schmidt method has a complexity in the order of nm^2 if m principal eigenvectors of dimension n have to be determined [21]). The cost of the orthogonalization defeats the purpose of using RP on resource constrained embedded cameras. However, Hecht-Nielsen et al. [16, 13] have noted that in a high-dimensional space, there exist a much larger number of nearly orthogonal than truly orthogonal vectors. Therefore, in a high-dimensional space even random vectors might be sufficiently close enough to orthogonal to offer a reasonable approximation of the original vectors [18].

To summarize, RP combines very interesting characteristics, making it ideal for computer vision. First, it tackles the “curse of dimensionality” by projecting the data to a much lower dimensional space. In addition, problems that deal with large amounts of data can be tackled more efficiently by operating on fewer data [26]. In contrast to similar dimensionality reduction techniques, such as PCA, RP is data independent, while still preserving the structure of the input data. RP can be applied on various types of data such as text, image, audio, etc.

2.2 Applications

There are several successful applications of RPs to computer vision problems. Tsagkatakis et al. [26] use RP for object tracking under variable pose and multi-camera views. Han et al. [12] used RP and robust linear discriminant analysis for face recognition. Wright et al. [30] extended their use by using RP and a novel l_1-norm minimization for face recognition. An insightful observation shared by many researchers that utilize RP for computer vision applications, as noted in [30], is that the choice of features is not as important as the number of features. This observation could prove to be of great significance for resource constrained environments such as embedded vision systems, where elaborate feature extraction may not be feasible due to the imposed limitations [26]. Kaski [16] presented experimental results using RP in the context of a system for organizing textual documents using Self-Organizing Map (i.e., WEBSOM). In this case, the results were as good as those obtained using PCA, and almost as good as those obtained using the original vectors. Lin and Gunopulos [18] compared RP and Latent Semantic Indexing (LSI) in the area of information retrieval. Bingham and Mannila [5] compared several dimensionality reduction techniques, such as PCA (based on data covariance matrix), RP and Discrete Cosine Transform (DCT) on image and text data. Their results indicate again that RP preserves distances and has performance comparable to that of PCA while being faster. Work, which is related to [5] and is focused on using RP for lossy image compression is presented in [3]. Dasgupta [6] described experiments on learning mixtures of Gaussians in high dimensions using RP and PCA. Li et al. [17] used RP with EM to learn geometric object appearance for object recognition. Motivated by the results of [6] Fern and Brodley [8] investigated the application of RP for clustering high-dimensional data. More recently, Fradkin and Madigan [9] evaluated RP in the context of supervised learning. In particular, RP was compared with PCA on a number of different problems using different machine learning algorithms. They concluded that although RP was slightly worse than PCA, its computational advantages might make it attractive in certain applications.

RP has demonstrated good performance in a number of applications, yielding results comparable to conventional dimensionality reduction techniques, such as PCA, while having much lower computational requirements. Feasibility of RP for face recognition is also investigated in [11], where authors compared RPs performance to the performance of PCA. However, even if theirs results suggest that RP is comparable to PCA, they used Gram-Schmidt algorithm for orthogonalization of random matrix, which is computationally wasteful. In this paper we do not orthogonalize random matrix since according to Hecht-Nielsen et al. there exist a much larger number of nearly orthogonal than truly orthogonal vectors [16, 13].

3 Experimental analysis

We performed a series of experiments, including both PCA and RP. The recognition performance of both methods in terms of percentage of false positives and false negatives on a standard database was examined.

Experiments were divided in two parts. In the first part general properties of the RP were tested and compared to the PCA. In the second part we used RP in the VSN simulator in conjunction with our hierarchical feature-distribution (HFD) scheme. In both parts of experiments, PCA was used as a benchmark. The experimental protocol was as follows.

3.1 Experimental procedure

Database 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 [22].

Preprocessing Image dimensions were the same for both methods, i.e., 128×128 pixels. First, the database was split into two parts of approximately same size. For every object, every second image was selected for recognition, and the remaining images were used to built PCA subspace. Images selected for recognition were transformed into the PCA and RP subspaces. This way feature vectors for both methods were obtained. Images that corresponded to zero orientation of each object and the corresponding features were used as a reference. This is consistent with the testing protocol, used in our VSN simulator [25].

Protocol All vector projections have been compared to the projections of zero orientation images and Euclidean distance was calculated. If the distance was below the predefined threshold T, the comparison resulted in a match between the reference image from the training set and the tested image. If the match was between images of two different classes (one class corresponds to all images of the same object), the number of false positives (FPs) was increased. If there was no match between the two images from the same class, number of false negatives (FNs) was increased. This way, results for a particular threshold T were obtained. This protocol was conducted separately for PCA and RP.

The procedure was repeated for a range of thresholds, which yielded FPs and FNs percentages.

Additionally, to rule out the influence of seed 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 proportion of FPs and FNs. PCA transforms the data to a new coordinate system in which basis vectors follow modes of greatest variance in training data [23]. Therefore, PCA is data dependent method. Every second image from the COIL-20 database was used to build a PCA subspace in advance. Properly constructed PCA feature vectors contain feature values that are ordered by decreasing importance in terms of the reconstruction quality of the original data.

The results are shown in Figure 1, which depict number of FPs and FNs in relation to the threshold T. We show only the results for subspace dimensions of 512, 16 and 1.

3.2 RP

Major portion of experiments were dedicated to the RP method. We explored the following questions.

- How do results vary with different seeds of used pseudorandom generator?
- Is transformation vector normalization really necessary?
- Does the method benefit from sorting of the transformation vectors based on the preserved variance in data?

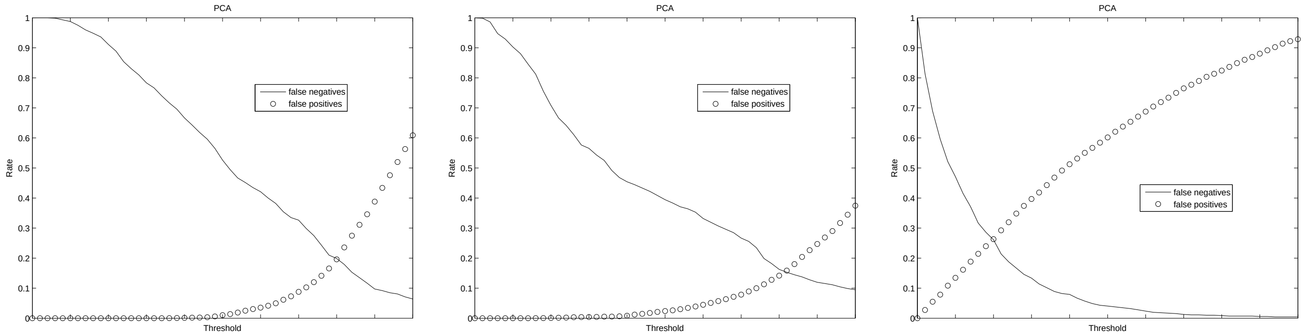
Although sorting of the transformation vectors according to the preserved variance in the data is not common practice in RP (as it is in the PCA), we are interested if this would improve recognition performance of RP.

Random projection matrix $\mathbf{R}$ was generated following the suggestion by Achlioptas [1] as defined in Eq.(5). The maximum dimensions of the matrix were 512×16384 elements. The Mersenne twister [20] was used as pseudorandom generator. We verified that same seeds result in same 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 and Figures 2 – 5 were generated by running tests with different thresholds T. Results for RP with different random seeds were compiled to single graphs using boxplots, as seen in Figures 2 – 5. Finally, the whole procedure was repeated for the following scenarios.

- RP using normalized vectors (Figure 2).
- RP using non-normalized vectors (Figure 3).
- RP using normalized vectors, sorted according to the highest preserved variance in data (Figure 4).
- RP using non-normalized vectors, sorted according to the highest preserved variance in data (Figure 5).

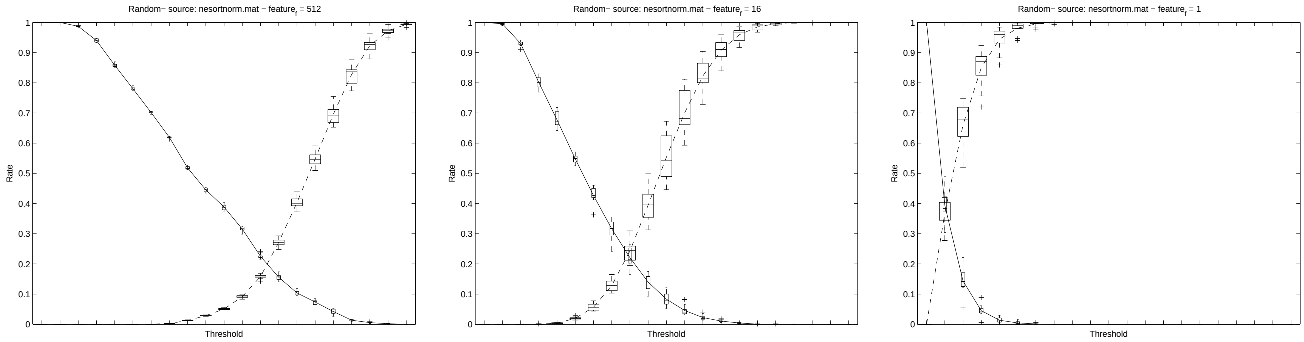
Observing Figures 2 – 5 we can conclude the following:

With 512 feature vectors used there is practically no difference between the PCA and RP and the scatter around the median value in the RP results is negligible. When 16 feature vectors are used PCA outperforms RP regardless of normalization or RP feature vector sorting. In this case, PCA yields FP/FN rates of 20%, while RP yields FP/FN rates in $25\% - 30\%$ range. Scatter in RP results is slightly larger but still small at the point of intersection of FPs and



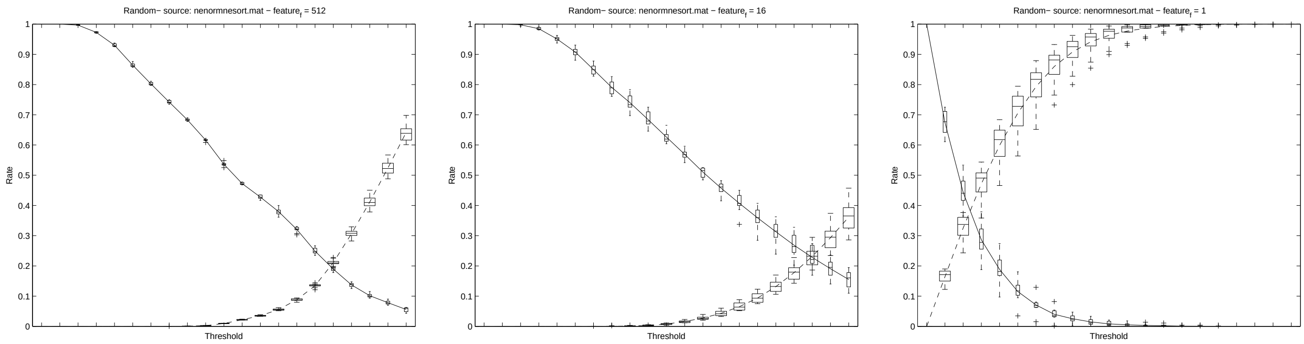
(a) 512 feature vectors (b) 16 feature vectors (c) 1 feature vector

Figure 1: Recognition rates (FPs and FNs) for PCA depending on the number of features used.



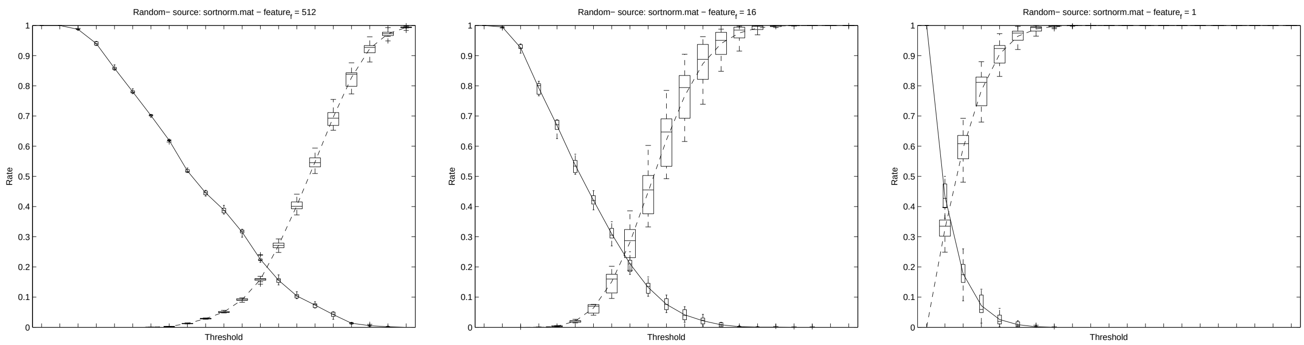
(a) 512 feature vectors (b) 16 feature vectors (c) 1 feature vector

Figure 2: Recognition rates (FPs and FNs) for RP (normalized feature vectors) depending on the number of features used.



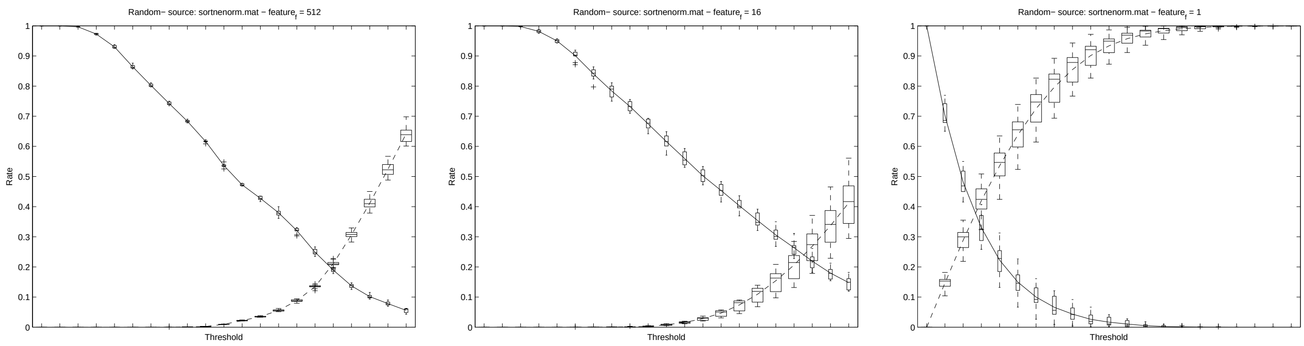
(a) 512 feature vectors (b) 16 feature vectors (c) 1 feature vector

Figure 3: Recognition rates (FPs and FNs) for RP (non-normalized feature vectors) depending on the number of features used.



(a) 512 feature vectors (b) 16 feature vectors (c) 1 feature vector

Figure 4: Recognition rates (FPs and FNs) for RP (normalized and sorted feature vectors) depending on the number of features used.



(a) 512 feature vectors (b) 16 feature vectors (c) 1 feature vector

Figure 5: Recognition rates (FPs and FNs) for RP (non-normalized and sorted feature vectors) depending on the number of features used.

FNs curves. With only one feature vector used performance of RP deteriorates significantly and scatter is even larger.

It can also be seen 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 6 it is obvious that they already have similar lengths. This is not surprising due to the dimensionality of input data – each vector has 16384 elements which can take only three possible values. Therefore, by discarding RP vector normalization, only scaling factor is introduced into the transformation.

Figure 8 shows that there are differences in reconstructive power of individual RP vectors. However, those differences are too small to have major impact on method performance. Therefore, this implies that it did not pay off to sort 512 random vectors according to the preserved variance.

In contrast, Figure 7 shows that PCA indeed condenses the information in small number of features.

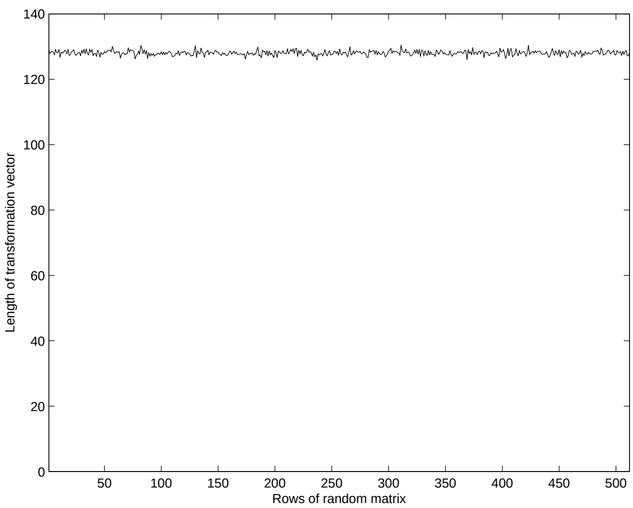


Figure 6: Length of the rows of the random matrix.

3.3 Application to simulator for VSNs

To test the performance of RP in our previously proposed hierarchical feature-distribution scheme [24, 25], we used a distributed network simulator. It runs on a standard desktop

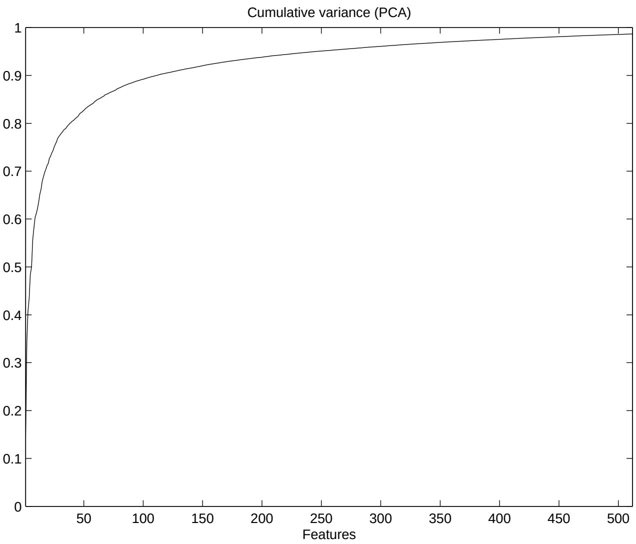


Figure 7: Cumulative variance (PCA) calculated from the projection through the subspace (every second image).

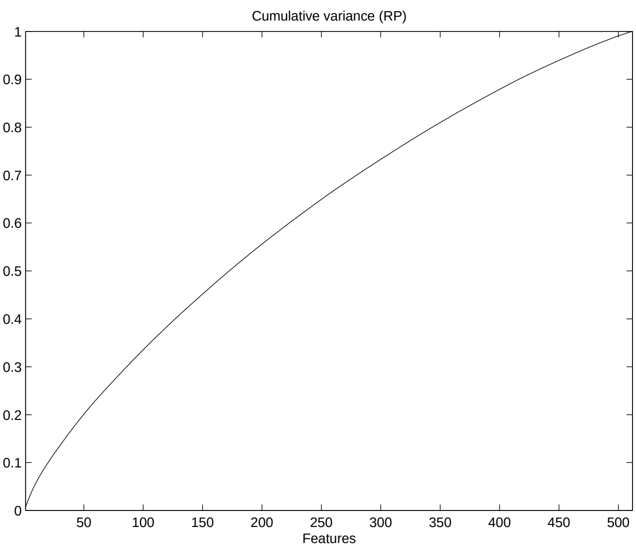


Figure 8: Cumulative variance (RP) calculated from the projection of every second image through the random matrix $\mathbf{R}$.

Table 1: Experimental results for hierarchical feature-distribution method

Recognition method	Training		Recognition		Recognition rate	
	Hops $[\frac{1}{sample}]$	Traffic $[\frac{kbytes}{sample}]$	Hops $[\frac{1}{sample}]$	Traffic $[\frac{kbytes}{sample}]$	FPs	FNs
PCA	614	44	108	342	15%	21%
RP	614	44	160	552	15%	22%

computer and is written in Matlab. The simulator measures both the amount of traffic transmitted between the nodes and the number of nodes (hops) over which the traffic is transmitted.

For the experiments, we used a network consisting of 99 nodes, arranged in a 11×9 rectangular, 4-connected grid.

The simulator for VSNs is used for testing the hierarchical feature-distribution (HFD) across the network of visual sensors (cameras). HFD has an important property. It significantly reduces the amount of transmitted data in the task of distributed object recognition (e.g., recognizing the objects that have been seen by other cameras in the network). The efficiency of our method in terms of data transmission is directly related to the recognition efficiency of the used object recognition method, as recognition errors significantly increase amount of transmitted data. In our previous work, we have established that PCA is appropriate for use in our HFD scheme, however, distribution of subspace across the network would be extremely wasteful in terms of network traffic. If RP is used instead, the only thing that we have to distribute is the seed of the pseudorandom generator (maximum eight bytes if Matlab implementation of Mersenne twister pseudorandom generator is used as a reference). The HFD method also ensures that the final decision on the identity of the recognized object is made using all available features. Therefore, in terms of recognition accuracy we are not concerned with the lower performance of RP when small number of features is used. However, this affects the amount of transmitted data.

Experiment was divided in two phases. The first (training) phase measured the performance of the network during training. Twenty nodes, evenly distributed through the network, were injected with images of the twenty different objects from the database. Those images corresponded to the zero orientation in the COIL-20 database. Next, the simulation cycle was started, and, after the network traffic stopped, the statistics on the network load (number of hops and the total network traffic per sample) was examined.

The second (recognition) phase measured the performance of the network during recognition. A pseudo-random sequence (same for all tests) was used to choose any image from the database and any node from the network. The image was injected into the chosen node, and the simulation cycle was started. After the network activity stopped, the result of the recognition was read from the same node, and the statistics on the FPs and the FNs was updated. The process of injecting the random image to a random node was repeated 5,000 times, and the statistics on the network load (number of hops and the total network traffic per sample) was recorded. The results for training and recognition for both PCA and RP are shown in Table 1.

It can be seen that substitution of PCA with RP degrades performance of the network. Recognition rate is almost the same, however, both traffic and number of hops in the recognition phase are increased due to the poor performance of RP with small number of features. However, our method still outperforms naive feature-distribution schemes, such as flooding by almost 1:2 [25] in terms of network traffic. Additionally, use of RP does not change performance of the network in training phase, since the format of data is practical identical for both PCA and RP. Nevertheless, considering the huge advantage of RP in distributing the subspace to all cameras, the RP method seems perfectly fit for the use with our hierarchical feature-distribution (HFD) scheme [25].

- It is a good substitute for PCA when dimensionality reduction is needed.
- The combination of HFD and RP still significantly reduces the amount of traffic across the network, even though this reduction is not as high as when HFD and PCA are used.
- A lower recognition rate of RP in comparison to PCA at same dimensionality does not influence the overall recognition rate of our simulated network, and the reduction in network efficiency is relatively small.

4 Discussion and conclusion

Focus of our work was random projection in a context of possible 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 is usually constrained 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 sharing subspace. On the other hand, while principal component analysis is known to give good results and has a lot of useful properties it is also computationally expensive and require distribution of shared subspace, which results in consuming more resources in the network. Therefore, our aim was to determine the performance of the RP implementation, which could be directly used in embedded cameras. Some minor modifications of RP (the absence of row normalization of the random matrix, and additional sorting of the rows of the random matrix considering highest variance in the data – as it is in the PCA) and their effect on the recognition accuracy were systematically explored. Performance of the RP was compared to the performance of the PCA in conjunction with recognition efficiency on the COIL-20 image data set. Results show surprisingly good performance of the RP in comparison to the PCA even without explicit orthogonalization (computationally wasteful) or normalization (important for preserving similarities in the low-dimensional space) of transformation subspace. Moreover, in our case even sorting of feature vectors in accordance to the preserved variance did not pay off. Our results indicate that RP can be used with our hierarchical feature-distribution scheme in visual-sensor networks, where RP can elegantly solve the problem of shared subspace distribution.

Acknowledgement

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

References

[1] D. Achlioptas. Database-friendly random projections. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[2] 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*, pages 557–563, New York, NY, USA, 2006. ACM.

[3] J. J. Amador. Random projection and orthonormality for lossy image compression. *Image Vision Comput.*, 25(5):754–766, 2007.

[4] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, page 616, Washington, DC, USA, 1999. IEEE Computer Society.

[5] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, New York, NY, USA, 2001. ACM.

[6] S. Dasgupta. Experiments with random projection. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 143–151, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[7] S. Dasgupta and A. Gupta. An elementary proof of the johnson-lindenstrauss lemma. Technical report, 1999.

[8] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *In Proc. of ICML*, pages 186–193, Washington, DC,, 2003.

[9] 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*, pages 517–522, New York, NY, USA, 2003. ACM.

[10] P. Frankl and H. Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory Ser. A*, 44(3):355–362, 1987.

[11] N. Goel, G. Bebis, and A. Nefian. Face recognition experiments with random projection. volume 5779, pages 426–437. SPIE, 2005.

[12] Pang Ying Han and A. B. J. Teoh. Random projection with robust linear discriminant analysis model in face recognition. In *CGIV '07: Proceedings of the Computer Graphics, Imaging and Visualisation*, pages 11–15, Washington, DC, USA, 2007. IEEE Computer Society.

[13] R. Hecht-Nielsen. *Context Vectors: General Purpose Approximate Meaning Representations Self-Organized from Raw Data*. IEEE Press, 1994.

[14] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, New York, NY, USA, 1998. ACM.

[15] W. B. Johnson and J. Lindenstrauss. Extensions of lipshitz mapping into hilbert space. In *In Conference in modern analysis and probability*, volume 26 of Contemporary Mathematics, pages 189–206. Amer. Math. Soc., 1984.

[16] S. Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering. In *Proceedings of International Joint Conference on Neural Networks*, pages 413–418, 1998.

[17] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296, New York, NY, USA, 2006. ACM.

[18] J. Lin and D. Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *In Proc. of SDM*, San Francisco, CA, 2003.

[19] A. Magen. Dimensionality reductions that preserve volumes and distance to ane spaces, 2002.

[20] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.

[21] R. Möller. First-order approximation of gram-schmidt orthonormalization beats deflation in coupled pca learning rules. *Neurocomput.*, 69(13-15):1582–1590, 2006.

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

[23] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson Learning, third edition, 2008.

[24] V. Sulić, J. Perš, M. Kristan, and S. Kovačič. Hierarchical feature encoding for object recognition in visual sensor networks. In *Proceedings of the 17th International Electrotehnical and Computer Science Conference, ERK 2008*, pages 221–224, Portoro}, Slovenia, September 2008.

[25] V. Sulić, J. Perš, M. Kristan, and S. Kovačič. Efficient feature distribution in visual sensor networks. Computer Vision Winter Workshop – electronic proceedings, 2009.

[26] G. Tsagkatakis and A. Savakis. A random projections model for object tracking under variable pose and multi-camera views. In *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, pages 1–7, 30 2009-Sept. 2 2009.

[27] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. 2007.

[28] S. S. Vempala. *The Random Projection Method*, volume 65. American mathematical Society, 2004.

[29] W. Wang, M. Garofalakis, and K. Ramchandran. Distributed sparse random projections for refinable approximation. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 331–339, New York, NY, USA, 2007. ACM.

[30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.