Comparing the algorithms of computing PCA

Marcin Cegielski

***Abstract* —Three methods of computing the Principal Component Analysis (PCA) are compared in these paper. The autocovariance matrix is computed using Discrete Wavelet Transformation (DWT), Discrete Cosine Transformation (DCT), Discrete Sine Transformation (DST) and Discrete Hadamard Transformation (DHT). The aim of this comparison is to find the speed and accuracy of these methods, and discussing the arrived results.**

I. Introduction

THE Principal Component Analysis (PCA) also known as the Hotelling Transformation or Karhunen-Loeve Transformation is built of eigenvectors of autocovariance matrix

$$\mathrm{cov}(X) = E[(X - E(X))(X - E(X)^T)] \quad (1)$$

The eigenvectors correspond to eigenvalues, which have been set in descending order. The KL transformation gives the orthogonal basis functions as the eigenvectors of the covariance matrix. This transformation is optimal in that it is a canonical transform minimizing the mean square error between a truncated representation and the actual data.

The cost of computation KL transformation creates a limitation of practical use of it.

Process of computing KL matrix is divided into two blocks.

In the first step covariance matrix is computed from input signal. The computation cost of the covariance matrix is $O(M^4)$ when the size of the reference image is $M \times M$. Coefficients of the covariance matrix are real and the matrix is symmetrical.

In the second step eigenvectors and eigenvalues are computed form the covariance matrix. We obtain the optimal approximation of the input data by selecting eigenvectors in decreasing order of magnitude of the eigenvalues. [5].

II. Explanation of used methods

The method of estimation PCA using Wavelets [3] was compared with other algorithms using Cosine and Sine transformation. All used transformations are orthogonal and move energy of the signal into the first coefficients.

A. Computing PCA with using Wawelets

The covariance matrix $\mathbf{C_x}$ of vectors $\mathbf{x_i}$ is

$$C_x = \frac{1}{M} XX^T \quad (1)$$

where **X** is a matrix of vectors $\vec{x}'_i$.

Wavelet equation of variable **X** may be written in matrix form.

$$X = A\Phi \quad (2)$$

By substituting (1) to (2) covariance matrix is obtained in form (3).

$$C_x = \frac{1}{M} A\Phi(A\Phi)^T \quad (3)$$

Since set of wavelets functions is orthogonal formula (3) is reduced into form (4)

$$C_x = \frac{1}{M} A\Phi(A\Phi)^T = \frac{1}{M} A\Phi\Phi^T A^T = \frac{1}{M} AA^T \quad (4)$$

It means that covariance matrix may be computed as a multiplication of matrix and its transposition. Matrix consists of wavelets coefficients.

B. Computing PCA with using DCT

Forward Discrete Cosine Transformation may be written by matrix formula (5)

$$Y = XA \quad (5)$$

and Inverse Discrete Cosine Transformation is in formula (6)

$$X = YA^T \quad (7)$$

where A is a matrix of cosine coefficients.

By substituting (1) to (7) covariance matrix is obtained in form (8)

$$C_x = \frac{1}{M} YA^T (YA^T)^T = \frac{1}{M} YA^T AY^T = \frac{1}{M} YY^T \quad (8)$$

The covariance matrix **C** may be computed as a multiplication of transformed matrix **X** and its transposition.

C. Computing PCA with using DST

The forward and inverse discrete sine transformation is

$$Y = XB \qquad X = YB \quad (9)$$

where **B** is a matrix of sine coefficients. The sine transformation is real, symmetric and orthogonal, that covariance matrix may by computed by formula (10)

$$C_x = \frac{1}{M} YB(YB)^T = \frac{1}{M} YBB^T Y^T = \frac{1}{M} YY^T . \quad (10)$$

Algorithms of computing covariance matrix using DST and DST are that same. Using of block version of transformation may reduce time of computing DCT or DST. Every vector $\vec{x}'_i$ is divided into 8 elements blocks. Cosine or sine transformation is computed for every block . Number of multiplications may be reduced in all line or in every block.

Author is with the Institute of Computer Science, Technical University of Lodz, ul. Wólczańska 215, 93-005 Łódź, Poland, e-mail: *mceg@ics.p.lodz.pl*

D. Computing PCA with using Hadamard transformation

Hadamard matrix is real and orthogonal. Method of computing covariance matrix is identical to DCT method and it is described by formula (8).

The algorithm of estimation PCA using DWT, DCT, DST or DHT is performed in third steps.

In the first step average vector must be computed and it must be subtracted from input matrix ***X***. DWT, DCT, DST or DHT transforms each column of this matrix. In the second step, transformed matrix ***X*** is multiplied by its transposition. Eigenvectors and eigenvalues are computed from this covariance matrix in the last step.

It is possible to use property of DWT, DCT, DST and DWT to concentrate signal energy in the firsts coefficients. It permits to reduce number of multiplications of coefficients of matrix ***X***. It influences number of mathematical operations in computing covariance matrix. The results of experiments are presented in the next section.

III. Presentation of experimental results

The covariance matrix was generated from the input picture of size 256 x 256 pixels with 8 bit gray level. Number of used multiplications for computing of one element of covariance matrix is in the first columns of tables. .Second and third columns describe time of computing transformation and covariance matrix in milliseconds. The mean square error and signal to noise ratio were computed for all covariance matrices. The results were compared with classical generated covariance matrix. Diagrams present distortion covariance matrices and computed PCA matrices.

TABLE I. METHOD USING HAAR WAVELETS

Number of multiplications	Time of transformation [ms]	Time of computing Cov. [ms]	pmse	SNR [dB]
256	10	441	0,0000	301,39
128	10	240	0,0000	37,50
64	10	141	0,0000	32,50
32	10	80	0,0004	21,63
16	10	60	0,0010	17,44
8	10	40	0,0024	13,70
4	10	30	0,0117	6,86

TABLE II. METHOD USING D4 WAVELETS

Number of multiplications	Time of transformation [ms]	Time of computing Cov. [ms]	pmse	SNR [dB]
256	10	440	0,0000	303,46
128	10	240	0,0000	38,73
64	10	141	0,0000	31,63
32	10	80	0,0003	23,30
16	10	60	0,0009	18,13
8	10	40	0,0147	5,89
4	10	30	0,0470	0,84

TABLE III. METHOD USING DCT

Number of multiplications	Time of transformation [ms]	Time of computing Cov. [ms]	pmse	SNR [dB]
256	3900	441	0,0000	300,96
128	3900	240	0,0000	40,18
64	3900	141	0,0000	32,80
32	3900	80	0,0002	24,28
16	3900	60	0,0008	18,52
8	3900	40	0,0012	16,68
4	3900	30	0,0061	9,71

TABLE IV. METHOD USING BLOCK DCT

Number of multiplications	Time of transformation [ms]	Time of computing Cov. [ms]	pmse	SNR [dB]
256	20	441	0,0000	310,12
128	20	240	0,0126	6,54
64	20	141	0,0468	0,85
32	20	80	0,0512	0,47
16	20	60	0,0537	0,26
8	20	40	0,0553	0,13
4	20	30	0,0553	0,13

TABLE V. METHOD USING BLOCK DCT WITH REDUCTION OF COEFICIENTS

Number of multiplications	Time of transformation [ms]	Time of computing Cov. [ms]	pmse	SNR [dB]
8	20	531	0,0000	310,12
7	20	311	0,0000	49,33
6	20	461	0,0000	45,34
5	20	240	0,0000	42,34
4	20	400	0,0000	39,96
3	20	211	0,0000	36,40
2	20	180	0,0000	33,20

TABLE VI. METHOD USING DST

Number of multipli cations	Time of transfor mation [ms]	Time of computi ng Cov. [ms]	pmse	SNR [dB]
256	3900	441	0,0000	300,72
128	3900	240	0,0000	39,66
64	3900	141	0,0000	31,86
32	3900	80	0,0003	23,44
16	3900	60	0,0010	17,38
8	3900	40	0,0018	14,99
4	3900	30	0,0041	11,39

TABLE VII. METHOD USING BLOCK DST

Number of multipli cations	Time of transfor mation [ms]	Time of computi ng Cov. [ms]	pmse	SNR [dB]
256	20	441	0,0000	306,87
128	20	240	0,0126	6,54
64	20	141	0,0468	0,85
32	20	80	0,0512	0,47
16	20	60	0,0537	0,26
8	20	40	0,0553	0,13
4	20	30	0,0553	0,13

TABLE VIII. METHOD USING BLOCK DST WITH REDUCTION OF COEFICIENTS

Number of multipli cations	Time of transfor mation [ms]	Time of computi ng Cov. [ms]	pmse	SNR [dB]
8	20	531	0,0000	306,87
7	20	311	0,0000	49,06
6	20	461	0,0000	41,88
5	20	240	0,0000	38,18
4	20	400	0,0000	30,81
3	20	211	0,0001	29,70
2	20	180	0,0008	18,62

TABLE IX. METHOD USING HADAMARD TRANSFORMATION

Number of multipli cations	Time of transfor mation [ms]	Time of computi ng Cov. [ms]	pmse	SNR [dB]
256	3900	441	0,0000	INF
128	3900	240	0,0381	1,750
64	3900	141	0,0539	0,24
32	3900	80	0,0556	0,10

Number of multipli cations	Time of transfor mation [ms]	Time of computi ng Cov. [ms]	pmse	SNR [dB]
16	3900	60	0,0565	0,04
8	3900	40	0,0569	0,01
4	3900	30	0,0569	0

TABLE V. MSE ERRORS OF COVARIANCE MATRIX

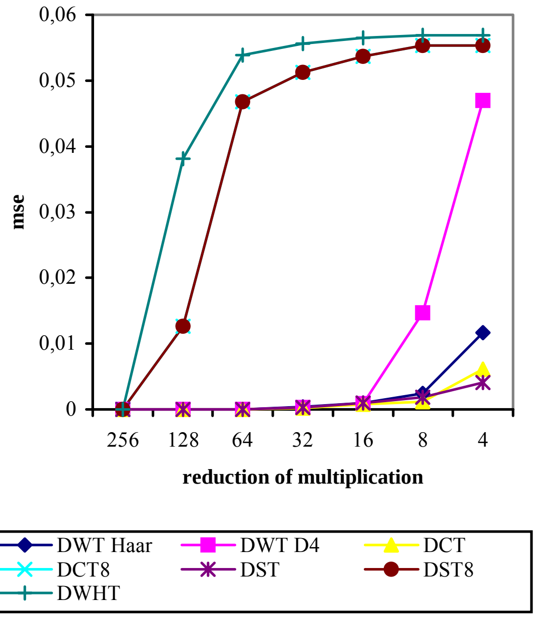
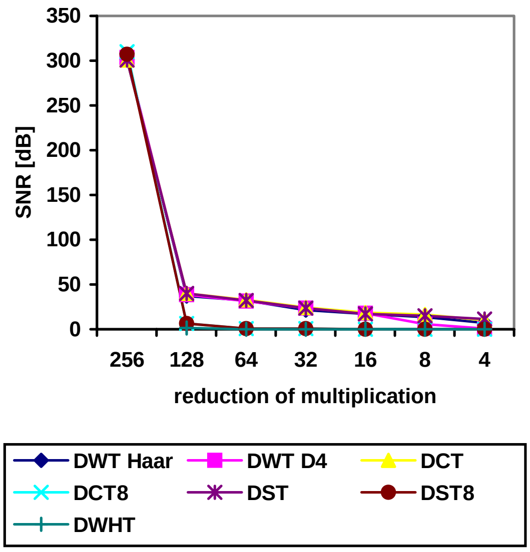


TABLE VI. SNR OF COVARIANCE MATRIX



Peak Mean Square Error was computed from equation (11)

$$PMSE = \frac{\frac{1}{M*N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[c_{x,y} - \hat{c}_{x,y}]^2}{[\max\{c_{x,y}\}]} \qquad (11)$$

Where $c_{x,y}$ is classic covariance matrix and $\hat{c}_{x,y}$ is covariance matrix estimated using DWT, DCT,DST or Hadamard matrix.

Signal to Noise Ratio was computed from equation (12)

$$SNR = 10\log_{10}\left(\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[c_{x,y}]^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[c_{x,y} - \hat{c}_{x,y}]^2}\right) \qquad (12)$$

where $c_{x,y}$ are coefficients of normal covariance matrix and $\hat{c}_{x,y}$ are coefficients of estimated covariance matrix C [6].

IV. CONCLUSION

The best results estimation of covariance matrix gets algorithm using wavelet transformation. Distortion of covariance matrix has insignificant impact on computing eigenvalues and eigenvectors. Cost of computation wavelet transformation is little and may be omitted. Good results of estimation PCA using DCT and DST are reduced by high computation cost. Computation cost of DCT,DST algorithm may be reduced, when line will be divided into 8 elements blocks. But results will be worse than other algorithms. Algorithm using Hadamard transformation is the worst. It gets maximum mean square error. Estimation error depends on correlation of input data.

Next investigations will be concentrated on Daubechies Discrete Wavelet Transformation of rank 4,6,8,12 and 20 and problem of error minimizing. This algorithm will be used for many kinds of images.

REFERENCES

[1] K. Sayood. “Kompresja danych wprowadzenie”, Wydawnictwo RM, 2002

[2] W. Press, S. Teukolsky, W. Vetterling, B. Flannery. “Numerical Recipes in C", Cambridge University Press

[3] M. Cegielski, “The Method of Improving Estimation Discrete Karhunen-Loeve Transformation Using Wavelets”, Proceedings of the International Conference TCSET’2004, Lviv-Slavsko 24-28 February, 2004

[4] J. Castrillon-Candas, K. Amarantuga, “Fast Estimation of Continuous Karhunen-Loeve Eigenfunctions Using Wavelets”, IEEE Transactions on Signal Processing, Vol. 50, No. 1, January 2002

[5] L. Smith , “A tutorial on Principal Components Analysis”, 2002, www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[6] W. Skarbek, “Multimedia Algorytmy i Standardy Kompresji”, Akademicka oficyna wydawnicza PLJ, Warszawa 1998