

Multiresolution Approach in Computing NTF

Arto Kaarna^{1,2}, Alexey Andriyashin³, Shigeki Nakauchi², and Jussi Parkkinen³

¹ Lappeenranta University of Technology, Department of Information Technology

P.O. Box 20, FIN-53851 Lappeenranta, Finland

² Toyohashi University of Technology

Department of Information and Computer Sciences

1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, 441-8580, Japan

³ University of Joensuu, Laboratory of Computer Science

P.O. Box 111, FI-80101 Joensuu, Finland

Abstract. The computation of non-negative tensor factorization may become very time-consuming when large datasets are used. This study shows how to accelerate NTF using multiresolution approach. The large dataset is preprocessed with an integer wavelet transform and NTF results from the low resolution dataset are utilized in the higher resolution dataset. The experiments show that the multiresolution based speed-up for NTF computation varies in general from 2 to 10 depending on the dataset size and on the number of required basis functions.

1 Introduction

Non-negative basis for data description is useful for two reasons. First, the approach is natural since many measuring devices output only non-negative values. Secondly, non-negative filters can be physically implemented. Thus, many application possibilities exist for non-negative bases. They include feature extraction in image databases, band selection in spectral imaging, and even image compression.

In general, the basis is a low-dimensional mapping of a high-dimensional data. The traditional approaches are the principal component analysis, the vector quantization, and the singular value composition [1]. Their outputs are either representatives for groups of samples or eigenimages, which are then mixed to get the reconstruction. The mixing coefficients may be both positive, negative, or they may have zero values.

Two approaches to find a non-negative factorization of a data set V have arisen recently. The non-negative matrix factorization (NMF) generates the basis functions W and their multipliers H for a composition $V_r = WH$ [1]. The number of columns in W and the number of rows in H is the rank k . If the data set contains images, then these images must be vectorized to apply NMF. Thus, the approach is not able to utilize the structural features of images. In the non-negative tensor factorization (NTF) the original shape of the data is maintained [2]. The factorization outputs the reconstruction as $V_r = \sum u \otimes v \otimes w$, where u , v , and w are the factors for each domain, and the sum is over the rank k .

In NTF, the factorization in practical applications results in a unique solution, while in NMF that may vary depending on the initial values for the solution process.

Both in NMF and NTF the factorization is obtained through an iterative process. The target is to minimize the distance between the original data and the reconstructed data. Depending on the application, the distance may be measured as the energy norm, the entropy, or as the Kullback-Liebler divergence.

The iteration is a time-consuming process. Typically, hours of computation time is needed to find the factorization for the data set. In most cases the basis functions are only needed and in the literature, these results are mostly reported, compared, and evaluated [1,2,3,4,5,6,7].

In this study, our hypothesis is that the number of iterations can be limited with a selection of relevant initial values. Instead of random initial values, a starting point generated from the dataset can speed up the iteration. In the proposed approach, the multiresolution of the data set is used to enhance the computation of the non-negative tensor factorization. The multiresolution is obtained through the integer wavelet transform. For a low resolution image the NTF is computed and then the components u , v , and w for that resolution are interpolated for the next higher resolution level.

The structure of the report is as follows. In Chapter 2 we introduce the NTF process. In Chapter 3 we give the background for the multiresolution approach based on the integer wavelet transform. In Chapter 4 we consider the computational complexity of the multiresolution approach for NTF computation. In Chapter 5 we show the results from the experiments. The discussion and the conclusions are in Chapter 6.

2 Non-negative Tensor Factorization

Recently, new approaches have emerged to define non-negative bases for datasets. Two of the methods are the non-negative matrix factorization (NMF) [1], and the non-negative tensor factorization (NTF) [2]. The basic approach for both of these is to find a solution for the problem

$$\min_{V_o, V_r \geq 0} \|V_o - V_r\| \quad (1)$$

where V_o is the original data and V_r is the reconstructed data. In composing V_r , all the components or substructures required in composition are non-negative.

In NTF [2] the reconstruction V_r is obtained as a sum of tensor products

$$V_r = \sum_{j=1}^k u^j \otimes v^j \otimes w^j \quad (2)$$

where u^j are bases for the first domain, v^j are bases for the second domain and w^j are bases for the third domain. k is the rank, a normal requirement is $(r + s + t)k < rst$, where r , s , and t are the number of samples in each domain.

Every element i in u^j , v^j , and w^j is non-negative. The number of domains naturally depends on the dataset. Three domains, u , v , and w are needed for example for the analysis of grayscale facial image databases, where the first two domains are the spatial domains of the images and the third domain comes from the stack of several facial images [2], [3] [4], [5].

We have used the well-known iterative process [2], that minimizes the reconstruction error in energy sense. Now the iteration steps for u_i^j , v_i^j , and w_i^j are defined, respectively, as

$$u_i^j \leftarrow \frac{u_i^j \sum_{s,t} G_{i,s,t} v_s^j w_t^j}{\sum_{m=1}^k u_i^m \langle v^m, v^j \rangle \langle w^m, w^j \rangle} \tag{3}$$

$$v_i^j \leftarrow \frac{v_i^j \sum_{r,t} G_{r,i,t} u_r^j w_t^j}{\sum_{m=1}^k v_i^m \langle u^m, u^j \rangle \langle w^m, w^j \rangle} \tag{4}$$

$$w_i^j \leftarrow \frac{w_i^j \sum_{r,s} G_{r,s,i} u_r^j v_s^j}{\sum_{m=1}^k w_i^m \langle u^m, u^j \rangle \langle v^m, v^j \rangle} \tag{5}$$

where $\langle ., . \rangle$ refers to the inner product, matrix G contains the values from V_0 .

3 Integer Wavelet Transform

For the multiresolution approach an approximation of the original data is required. This can be performed in many ways. A simple approach would be to subsample the data, i.e. select every second value from each domain. This approach is not suitable for basis function generation, since the bases are typically required to represent the low-frequency properties of the data. At least with synthetic data this approach would lead to problems since some features may be only one pixel wide [2].

The wavelet transform performs the appropriate approximation of the data. The original data is transformed to the approximative component and to the detail component [9]. In the inverse wavelet transform these two components are used to reconstruct the data. The wavelet transform carries the perfect reconstruction property. In Fig. 1, a), b), the principle of multiresolution is illustrated. The lower level approximation is received as values a_{j+1} etc. from the original values a_j . In practice the transform is performed using convolution with low-pass filter h and high-pass filter g . In definition of the filters different requirements can be set [9].

The wavelet transform is one-dimensional in nature. In the two-dimensional case, the one-dimensional transform is applied to the rows and columns of the image. In the three-dimensional case, the one-dimensional transform is applied to the spatial and spectral domains separately. In Fig. 1, c), the principle of the three-dimensional, separable transform is shown.

The datasets in imaging normally contain integer data. Thus, an integer version of the wavelet transform suits well to our case. Similarly to the floating case, there exists different integer wavelet transforms [10,11,13]. The integer wavelet

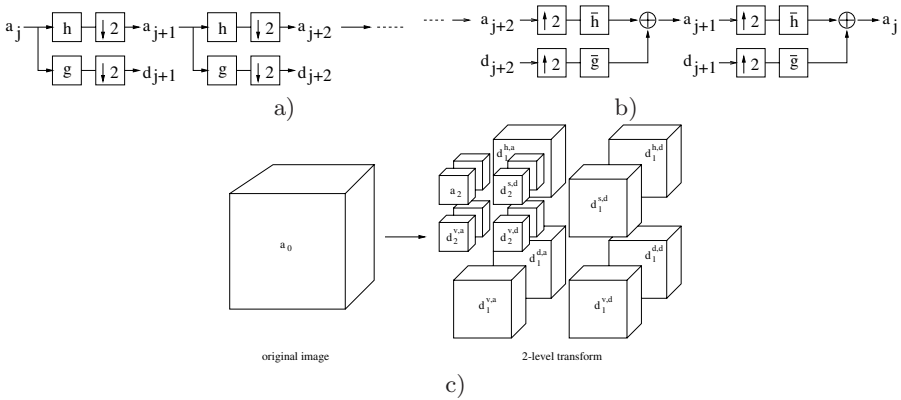


Fig. 1. Wavelet transform. a) Forward transform. b) Inverse transform, c) Separable three-dimensional wavelet transform applied twice.

transform is based on the lifting scheme: different filters are derived by combining the prediction step with the update step [11].

The basic form S of the integer wavelet transform subtracts the even samples from the odd samples to get the difference d_1 and the new approximation a_1 as

$$d_{1,l} = a_{0,2l+1} - a_{0,2l}, \quad a_{1,l} = a_{0,2l} + \lfloor d_{1,l}/2 \rfloor \tag{6}$$

where the original data is stored in $a_{0,..}$. The second subscript refers to the index in the sample vector. The exact reconstruction comes from calculating the values in reverse order as

$$a_{0,2l} = a_{1,l} - \lfloor d_{1,l}/2 \rfloor, \quad a_{0,2l+1} = a_{0,2l} + d_{1,l} \tag{7}$$

In general, the integer wavelet transform consists of prediction and of update based on the lifting where the number of vanishing moments is increased. In [11], [12], [13], [10] several integer wavelet transforms are defined. We have implemented the following transforms: TS -transform, $S+P$ -transform, $(2+2, 2)$ -transform, and $5/3$ -transform. In Eqs. 8, 9, 10, and 11 the forward transforms are given, respectively.

$$TS \quad \begin{cases} d_{1,l} = a_{0,2l+1} - a_{0,2l}, & a_{1,l} = a_{0,2l} + \lfloor d_{1,l}/2 \rfloor \\ d_{1,l} = d_{1,l} + \lfloor 1/4(a_{1,l-1} - a_{1,l}) + 1/4(a_{1,l} - a_{1,l+1}) \rfloor \end{cases} \tag{8}$$

$$S + P \quad \begin{cases} d_{1,l} = a_{0,2l+1} - a_{0,2l}, & a_{1,l} = a_{0,2l} + \lfloor d_{1,l}/2 \rfloor \\ d_{1,l} = d_{1,l} + \lfloor 2/8(a_{1,l-1} - a_{1,l}) + 3/8(a_{1,l} - a_{1,l+1}) + 2/8d_{1,l+1} \rfloor \end{cases} \tag{9}$$

$$(2 + 2, 2) \quad \begin{cases} d_{1,l} = a_{0,2l+1} - \lfloor 1/2(a_{0,2l} + a_{0,2l+2}) + 1/2 \rfloor \\ a_{1,l} = a_{0,2l} + \lfloor 1/4(d_{1,l-1} + d_{1,l}) + 1/2 \rfloor \\ d_{1,l} = d_{1,l} - \lfloor 1/8(-1/2a_{1,l-1} + a_{1,l} - 1/2a_{1,l+1}) + 1/8(-1/2a_{1,l} + a_{1,l+1} - 1/2a_{1,l+2}) + 1/2 \rfloor \end{cases} \tag{10}$$

$$5/3 \quad \begin{cases} d_{1,l} = a_{0,2l+1} - \lfloor 1/2(a_{0,2l+2} + a_{0,2l}) \rfloor \\ a_{1,l} = a_{0,2l} + \lfloor 1/4(d_{1,l} + d_{1,l-1} + 1/2) \rfloor \end{cases} \quad (11)$$

The integer wavelet transform outputs non-negative values for the approximation coefficients a if the original data is non-negative. When details d are added in the inverse transform, the output is still non-negative. Thus, the transform does not violate the requirement of non-negativeness of the data for NTF.

4 Definition and Computational Complexity of the Proposed Algorithm

Non-negative matrix factorization outputs k vectors for each domain. The number of samples for each domain are r , s , and t , if a three-dimensional dataset is used. This is the normal case, when NTF is used with a grayscale facial image dataset or with a spectral image. For the spectral dataset, one domain, like v , can be neglected, since the data is only two-dimensional, the first domain is the spectral domain and the second domain consists of the large number of spectra.

4.1 Definition of the Algorithm

The algorithm for the multiresolution approach for computing the NTF consists of the integer wavelet transform (IWT) and of NTF computation. The details are given in Algorithm 1.

Algorithm 1

1. Compute the lowest resolution transform using IWT for the original data set.
2. Compute u , v , and w for this lowest level in multiresolution.
3. Interpolate u , v , and w for the next higher level in multiresolution.
4. Use inverse IWT to compute the next higher level in multiresolution.
5. Compute u , v , and w for the current multiresolution level.
6. If u , v , and w are computed for the highest level in multiresolution, then Stop. Otherwise, goto Step 3.

4.2 Computational Complexity

The one-dimensional wavelet transform is of order $O(n)$, where n is the number of samples. In three-dimensional case the number of samples n is $n = rst$. At each step to lower level in resolution, the number of samples is divided by eight, so one half of the samples in each domain is transferred to the next level. Each IWT described in Eqs. 8, 9, 10, and 11, require from 4 (S -transform, Eq. 6) to 24 ((2 + 2, 2)-transform, Eq. 10) operations to get the two new values in the lower resolution level. Normally, a low number of levels are needed in the transform, like from 3 to 5 levels.

In NTF, u , v , and w are obtained through an iterative process. In each iteration step j , see Eqs. 3, 4, 5, the computational load is proportional to $O(rst)$.

The number of iterations depends on the data set used in the application. Typically, hundreds or even thousands of iterations are needed for the process to converge [8].

In the implementation, we selected various number of iterations for each resolution level. At low resolution, it was possible to select a large number of iterations with only a nominal effect to the whole computational time. The final target was to minimize the required number of iterations at the highest resolution, since at the highest resolution the iteration was most time-consuming in finding the final components u , v and w for the data set.

5 Experiments

Three phases were performed in the experiments. The first phase was to select a suitable integer wavelet filter for the multiresolution analysis. The second phase considered the number of levels in the multiresolution. In the last phase NTF within the multiresolution approach was performed.

5.1 Experimental Data

Three data set were used in the experiments. The first set is a two-dimensional set containing $t = 1269$ color spectra. Each spectrum had $r = 384$ samples. Typical representatives of the spectra set are shown in Fig. 2 [14].

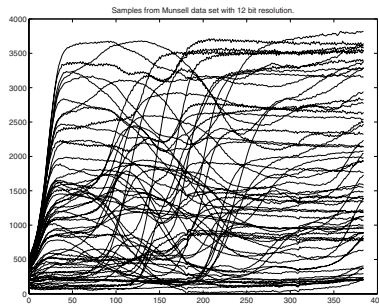


Fig. 2. Test set 1: Samples from Munsell color spectra set

The second data set was constructed from facial images from CBCL data set [15]. The number of images was $t = 192$, and each image was of size $rs = 96 \times 96$ pixels. Samples of the data set are shown in Fig. 3.

The third data set is a spectral image of size $rst = 256 \times 256 \times 224$. The image is part of Moffet Field remote sensing image captured using AVIRIS equipment [16].



Fig. 3. Test set 2: Facial image dataset, samples from the set



Fig. 4. Test set 3: Moffet Field spectral image, bands 11, 21, 41, 61, and 91

5.2 Selection of the Integer Filter and the Number of Levels in Multiresolution

In [10], the best lossless compression results for gray-scale images were obtained with the 5/3-transform. This means that the filter can capture essential features from the data set to a low number of coefficients. Our target is different. The purpose is to describe the data set such that features from a lower resolution to a higher resolution remain similar. Thus, the first task is to select a suitable IWT filter.

We computed the algorithm with $k = 3$ for the third data set with various IWT filters. The results are collected to Table 1. The quality is computed as the signal-to-noise ratio and it is expressed in decibels (dB). For filter S the results for the whole multiresolution solution are shown. For other filters, only the final result is shown. The starting point for the iteration was the same for all filters.

The conclusion from this experiment is that all filters act similarly, except filter 5/3, which outputs slightly worse results than the others. The final selection

Table 1. IWT filter selection, $k = 3$

Filter name	level	# of iterations	Quality (dB)	Relative time
S	4	2000	18.453	1.080
	3	600	15.975	2.190
	2	300	14.333	27.910
	1	50	13.119	74.540
	0	1	12.408	24.730
TS	0	1	12.408	31.780
SP	0	1	12.408	42.830
$(2 + 2, 2)$	0	1	12.405	41.070
5/3	0	1	12.393	31.730

criterion is the computational time which is lowest for the filter S . Thus, all the experiments were performed with filter S .

Next, we wanted to find out, how many steps in the multiresolution ladder are needed. The experiment was performed with filter S , with the spectral image data set. The results are shown in Table 2.

Table 2. Selection of the number of levels in the multiresolution. IWT filter S , $k = 3$.

level	# of iterations	Quality (dB)	Relative time
3	600	15.974	2.020
2	300	14.332	24.430
1	50	13.119	70.050
0	1	12.408	24.300
2	300	14.320	32.230
1	50	13.112	77.640
0	1	12.402	24.400
1	50	12.627	67.740
0	1	12.020	22.630

From Tables 1 and 2 we can conclude that at least three levels of multiresolution are required to achieve the good quality in reconstruction. In the next experiments, four levels in multiresolution were applied, since the computational cost in the lowest level is very small compared to the whole process.

5.3 Computational Results for NTF

The last experiment considers the whole process described in Alg. 1. The three datasets were used, in IWT the filter was filter S , and four levels in the multiresolution ladder were used. The results from the experiment are shown in Figs. 5 a), b), and c). On the horizontal axis there is the relative computational time with a logarithmic scale, and on the vertical axis there is the reconstruction quality in dB.

Each subfigure in Fig. 5 contains ranks $k = 1, 2, 3, 4, 6, 8, 16, 32$. For each k without multiresolution, the number of iterations were 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. The qualities for the latest iterations are shown for each k . For the multiresolution case, only the final result is shown as a single box (\square) for each k .

6 Conclusions

In this study, we have enhanced the computation of u , v , and w for the non-negative tensor factorization with a multi-resolution approach. The multiresolution was computed using the integer wavelet transform.

The following conclusions can be made from the experiments. In general, the proposed approach is from 2 to 10 times faster than the original computation. Especially, for a data set with large values for r and s , the approach is very good. When the rank k is large, the original solution requires time that is fifty-fold

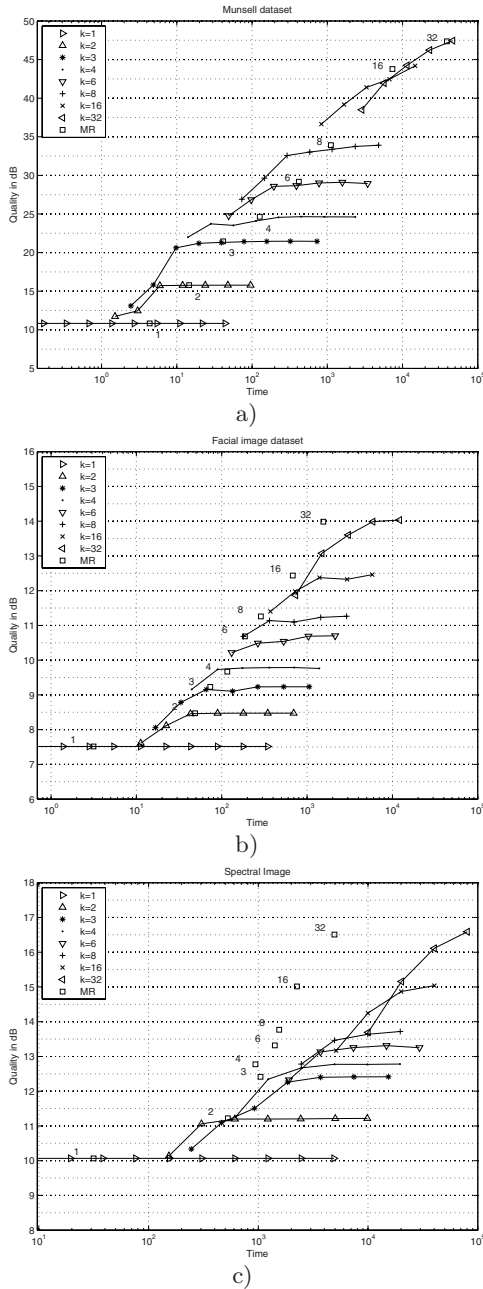


Fig. 5. Computational time vs. reconstruction quality, a) Munsell data set. b) Facial image data set. c) Moffet Field spectral image. For each run 2,4,8,16,32,64,128,256, and 1024 iteration steps were used. For larger rank k , only last steps are marked. For each k , the computational time for the highest quality of the proposed method is marked by \square .

compared to the proposed approach, see Fig. 5, c). The gain from the proposed approach is from 2 to 10 fold. For facial image data set, the same conclusions can be drawn, see Fig 5, b).

For the Munsell data, the approach provides clear gain when k is larger, like $k = 4, 6, 8, 16, 32$. With $k = 1$ the proposed approach is not usable, there is the extra load of the IWT compared to the original solution, see Figs. 5 a), b), c). In practice, a small number of iterations (even 2 iterations) results in the converged solution.

References

1. Lee, D.D., Seung, N.S.: Learning the Parts of Objects by Non-negative Matrix Factorization. *Nature* 401, 788–791 (1999)
2. Hazan, T., Polak, S., Shashua, A.: Sparse Image Coding using a 3D Non-negative Tensor Factorization. *IEEE International Conference on Computer Vision (ICCV'05)* (2005)
3. Hoyer, P.O.: Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research* 5, 1457–1469 (2004)
4. Li, S., Hou, X.W., Zhang, H.J., Cheng, Q.S: Learning Spatially Localized, Part-Based Representation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Hawaii, USA 1, 207–212 (2001)
5. Wild, S., Curry, J., Dougherty, A.: Improving Non-negative Matrix Factorizations through Structured Initialization. *Pattern Recognition* 37, 2217–2232 (2004)
6. Shashua, A., Hazan, T.: Non-negative Tensor Factorization with Applications to Statistics and Computer Vision. In: *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, pp. 792–799 (2005)
7. Heiler, M., Schnörr, C.: Controlling Sparseness in Non-negative Tensor Factorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 56–67. Springer, Heidelberg (2006)
8. Yuan, Z., Oja, E.: Projective Nonnegative Matrix Factorization for Image Compression and Feature Extraction. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) *SCIA 2005*. LNCS, vol. 3540, pp. 333–342. Springer, Heidelberg (2005)
9. Daubechies, I.: *Ten Lectures on Wavelets*, CBMS-NSF. Regional Conference Series in Applied Mathematics, 61, SIAM, USA (1992)
10. Adams, M.D., Kossentini, F.: Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis. *IEEE Transactions on Image Processing* 9(6), 1010–1024 (2000)
11. Calderbank, A.R., Daubechies, I., Sweldens, W., Yeo, B-L.: Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis* 5(3), 332–369 (1998)
12. Calderbank, A.R., Daubechies, I., Sweldens, W., Yeo, B-L.: Lossless Image Compression using Integer to Integer Wavelet Transforms. *IEEE International Conference on Image Processing (ICIP'97)* 1, 596–599 (1997)
13. Daubechies, I.: Recent results in wavelet applications. *Journal of Electronic Imaging* 7(4), 719–724 (1998)
14. Spectral Database. University of Joensuu Color Group, Accessed: October 26, 2006 Available: <http://spectral.joensuu.fi/>
15. Face Recognition Database. MIT-CBCL, accessed: November 10, 2006, available: <http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>
16. Homepage, AVIRIS: accessed: November 10, 2006 available: <http://aviris.jpl.nasa.gov/>