Proc. Int. Joint Conf. on Neural Networks IJCNN 2004, Budapest, pp 2535-2539

Transformation-invariant representation and NMF

Julian Eggert, Heiko Wersing and Edgar Körner
HONDA Research Institute Europe GmbH
Carl-Legien-Straße 30
63073 Offenbach/Main, Germany
E-mail: {Julian.Eggert,Heiko.Wersing,Edgar.Koerner}@honda-ri.de

***Abstract*— Non-negative matrix factorization (NMF) is a method for the decomposition of multivariate data into strictly positive activations and basis vectors. Here, instead of using unstructured data vectors, we assume that something is known in advance about the type of transformations that either the input data or the basis vectors may undergo. This would be the case e.g. if we assume input vectors that are translationally shifted versions of each other, but it applies to any other transformations as well. The key idea is that we factorize the data into activations and basis vectors modulo the transformations. We show that this can be done by extending NMF in a natural way. The gained factorization thus provides a transformation-invariant and compact encoding that is optimal for the given transformation constraints.**

I. Introduction

NMF has been introduced by Lee and Seung [3], [4] as an effective factorization method for decomposing multivariate data under the constraints of non-negative components. These constraints result in a representation that is parts-based, because the framework allows only additive, and not substractive, combinations of components. It has been shown [3] that the non-negativity property leads to very different representations than other factorization techniques such as principal components analysis and vector quantization, which learn more holistic representations.

Whereas the general NMF method makes no assumptions on the data with exception of the non-negativity of all its components, it is often the case that NMF is used for data with particular symmetry properties. This is e.g. the case for input images which could be presented at different positions (translational symmetry), scales, angles of rotation, etc. Consider the case of NMF being used as a method for modeling receptive fields of the mammal early visual pathway, e.g. the primary visual cortex. The resulting basis vectors usually look like elongated bar patches, similar to gabor filters, anchored at different positions of the receptive field. This form of representation embodies a large amount of redundancy, since the different basis vectors can be considered as transformed versions of each other.

If the symmetry properties of the input data are known in advance (e.g., if the input consists of image patches drawn at random positions of a larger image, then several input vectors are translated versions of each other), we can take advantage of them and include them explicitly into the learning and encoding framework. By this way, a much less redundant representation is gained, with a reduced set of basis vectors that can be interpreted as the extended, original set modulo the transformations. Here we set up a generative model that encodes data in an effective way under consideration of symmetry constraints and the constraint of non-negativity in all its components, taking advantage of the NMF framework to manage this task. In the next section (section II) we include transformation-invariant representations into the NMF formulation (calling our approach "Overlapping NMF"). Simulation results gained using the modified algorithm that show how the algorithm works are presented afterwards in section III.

II. Overlapping NMF: Inclusion of transformation-invariant representations into the NMF framework

A. Standard Euclidean NMF algorithm

In the NMF formulation, we start with a non-negative matrix V containing the original input data and the task is to find non-negative matrices W and H such that their linear combination (the reconstruction of V) R is close to V,

$$V \approx R(W,H) = WH \ . \tag{1}$$

Eq. (1) can be rewritten on a column by column basis as

$$\mathbf{V}_i \approx \mathbf{R}_i(W,H) = \sum_j H_i^j \mathbf{W}_j \ , \tag{2}$$

with $\mathbf{V}_i = V_i^l$, $\mathbf{R}_i = R_i^l$ and $\mathbf{W}_j = W_j^l$ beeing the columns of V, R and W, respectively, and H_i^j the activities used for the encoding of the data. Therefore, each data vector $\mathbf{V}_i$ is approximated by its reconstruction $\mathbf{R}_i$ which is a linear combination of the basis vectors $\mathbf{W}_j$ weighted by their activities H_i^j. An Euclidean cost function that quantifies the degree to which this approximative reconstruction has been achieved,

$$F(W,H) = \frac{1}{2}||V - WH||^2 \ , \tag{3}$$

or, equivalently,

$$F(W,H) = \frac{1}{2}\sum_i \left|\left| \mathbf{V}_i - \sum_j H_i^j \mathbf{W}_j \right|\right|^2 \ . \tag{4}$$

It is shown in [4] that the following NMF update rule(s) minimize (4) with respect to W and H, while preserving non-negativity in all its components. First, the matrices W and H are initialized with components that are non-negative and larger than zero. Then, we proceed according to:

1) Calculate the reconstruction according to

$$\mathbf{R}_i = \sum_j H_i^j \mathbf{W}_j \ . \tag{5}$$

2) Update the activities according to

$$H_i^j \leftarrow H_i^j \odot \frac{\mathbf{V}_i^T \mathbf{W}_j}{\mathbf{R}_i^T \mathbf{W}_j} \ . \tag{6}$$

3) Calculate the reconstruction with the new activities according to

$$\mathbf{R}_i = \sum_j H_i^j \mathbf{W}_j \ . \tag{7}$$

4) Update the basis vectors according to

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i H_i^j \mathbf{V}_i}{\sum_i H_i^j \mathbf{R}_i} \ . \tag{8}$$

5) Return to point 1 until convergence.

(The multiplications and divisions in (6) and (8) are applied componentwise; for the multiplication this is indicated by the $\odot$.)

The update rules can be understood in a gradient descent framework (meaning an additive update rule which modifies each component in the direction of the negative gradient of the cost function (4), proportional to some stepsize variable), with diagonally rescaled stepsize variables, which causes the stepsize parameters to disappear (see eqs. 6,7 from [4] for a hint on this issue) and which results in the purely multiplicative update rules (6) and (8).

B. Overlapping reconstruction and cost function

The idea of incorporating transformation symmetries into the NMF framework can be thought of in two different ways. We can either assume symmetry properties of the input vectors and demand the factorization matrices to reflect these properties (e.g. meaning that the same input image is presented at different positions, and that such a shift in the positions is reflected in the activity as well; in short, that a shifted input gives rise to a correspondingly shifted activation pattern), or we can incorporate the transformations into the reconstruction procedure (2) and the cost function (4). The two ways of thinking lead to the very same result, here, we are going to extend on the second way of thinking because its formulation is more similar to the original NMF approach.

As in the original NMF formulation, the task is to find non-negative matrices W and H such that their linear combination (the reconstruction of V) R is close to V. The difference now is that instead of the simple reconstruction (2), which made direct use of the basis vectors $\mathbf{W}_j$, we have that

$$\mathbf{V}_i \approx \mathbf{R}_i(W,H) = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} T^{\mathbf{m}} \mathbf{W}_j \ , \tag{9}$$

so that the reconstruction is gained by linearly *overlapping* the contributions of $\mathbf{W}_j$ transformed by matrices $T^{\mathbf{m}}$ [1]. In a sense, we take one out of a pool of original basis vectors, transform it in every possible way allowed by the transformation parameter $\mathbf{m}$, and then use all these transformed basis vectors to reconstruct the input. In the translational symmetry case, this means that we translate the original basis vector to all positions, and use all translated basis vectors with their respective activities to gain an optimal reconstruction. The tranformation $T^{\mathbf{m}}$ is a matrix that converts from the (eventually smaller) space of basis vectors to the space of the input and reconstruction images [2], and that transforms the original basis vector according to the parameter $\mathbf{m}$.

Using the overlapping reconstruction (9), the Euclidean cost function is written as

$$F(W,H) = \frac{1}{2} \sum_i \left|\left| \mathbf{V}_i - \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} T^{\mathbf{m}} \mathbf{W}_j \right|\right|^2 \ . \tag{10}$$

The task is now to find the original basis vectors $\mathbf{W}_j$ (i.e., the basis vectors modulo transformation) that allow an optimal reconstruction of the input. Remark that the activities retain their index $\mathbf{m}$ since they have to be stored *for every possible transformation*.

C. Overlapping NMF algorithm

Intuitively, the standard NMF update rules can be "understood" as being gained from the original cost function (4), using gradient descent, separating positive and negative terms, and observing that the fixed point of the update rules is reached when the gradient approaches zero, respectively the quotient approaches 1. Similarly to the quotients of the standard NMF, we can reformulate the fixed point conditions for the cost function (10) into update rules. The complete algorithm for overlapping NMF is then given by the following set of rules:

1) Calculate the reconstruction according to

$$\mathbf{R}_i = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} T^{\mathbf{m}} \mathbf{W}_j \ . \tag{11}$$

2) Update the activities according to

$$H_i^{j,\mathbf{m}} \leftarrow H_i^{j,\mathbf{m}} \odot \frac{\mathbf{V}_i^T T^{\mathbf{m}} \mathbf{W}_j}{\mathbf{R}_i^T T^{\mathbf{m}} \mathbf{W}_j} \ . \tag{12}$$

3) Calculate the reconstruction with the new activities according to

$$\mathbf{R}_i = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} T^{\mathbf{m}} \mathbf{W}_j \ . \tag{13}$$

4) Update the original (non-transformed) basis vectors according to

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{V}_i^T T^{\mathbf{m}}}{\sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{R}_i^T T^{\mathbf{m}}} \ . \tag{14}$$

5) Return to point 1 until convergence.

[1] This is the reason for the name "Overlapping NMF".

[2] Thinking of $T^{\mathbf{m}}$ as a transformation rather than an arbitrary matrix would seemingly imply boundary condition problems when the result of the transformation gets out of range. This can be dealt with in arbitrary ways, as long as $T^{\mathbf{m}}$ remains a matrix. In our translation example from section III, we used cyclic boundary conditions.

Convergence of the gained algorithm can be proven in a way analogous to the original NMF, see [4]. Comparing the algorithm for overlapping NMF with that of standard (non-overlapping) NMF, we see that much of the original form has been maintained. The greatest modification is the incorporation of the transformation matrices $T^{\mathbf{m}}$ in the equations. Nevertheless, two problems arise. First, the calculation of the transformed vectors makes the computations more expensive, and second, we have a larger dimensionality of the activities since the $H_i^{j,\mathbf{m}}$'s are now additionally indexed with $\mathbf{m}$ (i.e., we have activities for all input images i, for all basis vectors j and each transformation $\mathbf{m}$), which for large $\mathbf{m}$'s results in an overcomplete representation of the data. However, the standard NMF is not able to deal with overcomplete representations, resp. settles at trivial or uninteresting solutions for them. This can be dealt with in two ways: Either we do not use every possible transformation parameter $\mathbf{m}$, i.e., we *subsample* accordingly (e.g., for translational symmetry, the activity would only be calculated at selected anchor points which do not cover all possible pixel shifts of an image, but only a subsampled set of them), or we include sparsity constraints on the activity into the cost function (10), which can be done for NMF as proposed in [1].

For the special case of translational symmetry being a part of the total transformation $T^{\mathbf{m}}$, which is common when processing e.g. images as input vectors, we get very simple expressions for the NMF dynamics. Then, the reconstruction $\mathbf{R}_i$ is calculated using a convolution of the basis vectors with the activities. Similarly, the update rules for the activities and the basis vectors contain correlations between the input vectors / the reconstruction vectors and the basis vectors resp. the activities, so that the calculation of the overlapping NMF algorithm amounts to simple, straightforward calculations.

III. Results

A. Overlapping NMF simulations

Here we show simulation results of the overlapping NMF algorithm for the translational symmetry transformation case. For simplicity, we use 4×4 pixel images with cyclic boundary conditions as data vectors. Figure 1 shows a few of the input images used for the task, generated by superposition of horizontal and vertical lines, thresholding and normalization. The task is to find the set of original (i.e., non-transformed) basis vectors that together with the transformations properly encode the input images. In our case, we search for those basis vectors that can be superposed linearly to appropriately reconstruct the input images while allowing arbitrary shifts in position.

For 8 allowed basis vectors, the standard (i.e., non-overlapping) NMF algorithm finds a solution which consists of all horizontal and vertical lines at the $(4+4=8)$ different positions. This is shown in fig. 2. Since we are interested in the basis vectors modulo translational transformation, we expect the overlapping NMF representation to be much more compact, since the 4 horizontal and the 4 vertical lines are translated versions of each other. The corresponding overlapping NMF solution would therefore consist of only 2 basis vectors, one horizontal and one vertical.

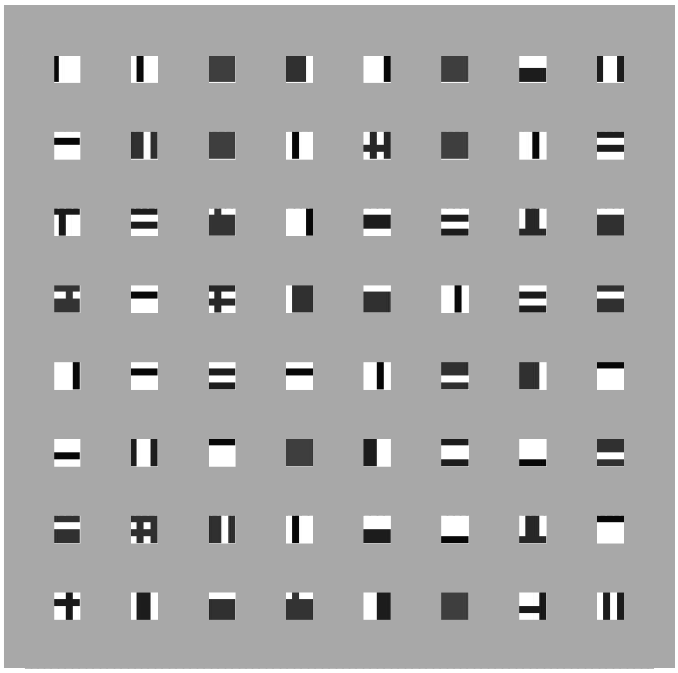


Fig. 1. An extraction of 64 of the 4x4-pixel-sized input images used as input for learning the basis vectors. They are generated by linear superposition of 1 to 4 randomly selected horizontal or vertical lines at arbitrary positions. An additional threshold reduces all pixels to values 0 and 1, afterwards the images have been normalized using an Euclidean norm. Here, dark colors indicate high values (black=1, white=0).

Fig. 2. Solution of the standard NMF algorithm for 8 basis vectors. The basis vectors comprise all 4 horizontal and all 4 vertical lines. The transformation properties of the input are encoded implicitly in the basis vectors; they are translated versions of each other. To the contrary, the overlapping NMF algorithm would consider the translations explicitly, and only 2 basis vectors would be needed: One horizontal and one vertical. The input is then reconstructed by shifting these basis vectors to all of their 4 possible positions.

Figure 3 shows the solution of the overlapping NMF algorithm for 2 basis vectors without sparsity constraints. In this case, we get a one-pixel basis vector, because with it, any image can be encoded with 100% fidelity. (Since we allowed translations, we can use the single pixel basis vector, translate it to any pixel of the image and add the contribution to the total reconstruction, reproducing the input images on a pixel-per-pixel basis.) The second basis vector is not needed at all for the reconstruction, it just remains unstructured. This is the trivial solution of overlapping NMF.

To prevent the system to run into the trivial, but uninteresting solution, we constrain the activity with a sparsity

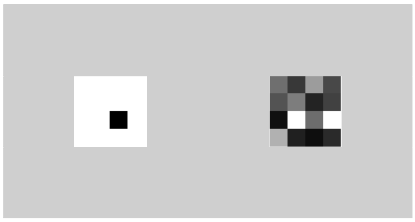


Fig. 3. Solution of the overlapping NMF algorithm for 2 basis vectors without sparsity constraints. A single one-pixel basis vector emerges, since with overlapping reconstruction this suffices to reconstruct any input pattern. The second basis vector remains unstructured, it does not contribute to the reconstruction.

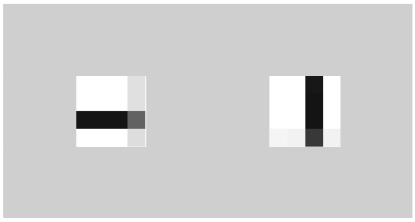


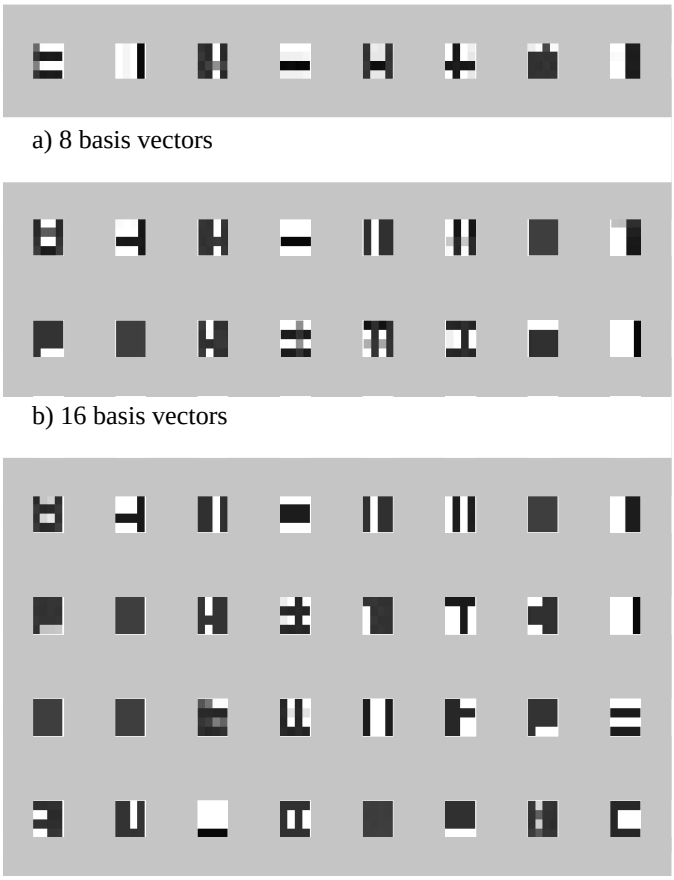
Fig. 4. Solution of the overlapping NMF algorithm for 2 basis vectors, now including sparsity constraints. We get a horizontal and a vertical bar, which are all single bar configurations modulo the translation transformation. With these two basis vectors, the overlapping NMF reconstruction of the input vector set is already better than for the standard NMF solution with the 8 bars from fig. 2.

term in the cost function. Figure 4 shows the solution of the overlapping NMF algorithm for 2 basis vectors and sparsity constraints. The solution settles into the horizontal and the vertical basis vector as expected [3]. The reconstruction for overlapping NMF with the 2 bar basis vectors is quantitatively as good as for the standard NMF with the 8 bar basis vectors.

For an increasing number of basis vectors, we get basis vectors that cover the more complex input patterns, such as those composed of 2 or more bars (parallel bar patterns, crosses, etc.). This is shown in fig. 5, with results for 8 (a), 16 (b) and 32 (c) basis vectors. Although in the input images each pattern can appear at any position, only one version of each distinguishable (by bar configuration) pattern is stored as a basis vector. This can be seen in fig. 5 (a), where e.g. the two-bar cross only appears once. In combination with the allowed shift transformations (and the cyclic boundary conditions), the single cross basis vector thus can be used to encode all possible crosses appearing at any of the 16 pixel positions of the input image. The basis vectors appear in an order according to this compositionality principle; first the single bar patterns, then more complex ones, etc. The overlapping NMF algorithm indeed extracts such complex patterns in a reliable way; solutions for an increased number of basis vectors are shown in fig. 5 (b) and (c).

Since the overlapping NMF framework extracts the basis

[3]Nevertheless there are slight deviations from the bar solution, this is because of the nonlinearties in the generation process. For a linear superposition scheme, we get the "clean" bar solution.



a) 8 basis vectors

b) 16 basis vectors

c) 32 basis vectors

Fig. 5. Solution of the sparse overlapping NMF algorithm for 8, 16 and 32 basis vectors. For increasing number of basis vectors more and more complex patterns are encoded explicitly. The maximal number of different pattern configurations is 20, therefore, for 32 basis vectors, some of the basis vectors remain unstructured or appear repetitively.

vectors modulo transformations, the resulting codebook is more compact than that of the standard factorization frameworks. Comparing fig. 2 with fig. 5 (a), we see that the standard factorization dedicates much effort to encode all the shifted varieties of single bars, etc., whereas the transformation invariant factorization encodes all single bars with only 2 basis vectors (horizontal and vertical orientation) and can then use the remaining basis vectors to encode the more subtle pattern variations, such as double and triple bar patterns in all their different configurations. The simulations were calculated with 250 input images and run for 1000 steps. The matrices W and H were initialized with equidistributed random values between 0.5 and 1.0.

B. Discussion

Factorizing algorithms encode data regardless of any invariance and transformation properties. The result is expressed in basis vectors and activations that contain these invariances implicitly, such as it is the case for the translational transformation example, where a standard NMF algorithm learns

the "same" basis vector at a multitude of different positions, leading to an undesirable redundancy in the coding scheme. Moreover, the learned basis vectors cannot generalize over the transformations, i.e., the set of input vectors eventually has to be rich enough to contain all transformational variations for the system to be able to extract all transformed basis vectors.

In this paper we included transformations explicitly into the learning process. This is sensible for sensory applications such as visual processing, where a number of transformation properties are desirable and known in advance. To some extent, overlapping NMF can be understood as beeing equivalent to an extensive method that presents every input vector for all its allowed transformations, and afterwards tries to compact and group together basis vectors according to the symmetry properties determined by the transformations. After learning, this results in the fact that transformed input vectors are reconstructed equally well as the original ones. This means e.g., that transformed input vectors can be presented to the system and the activities (for fixed basis vectors W) will converge to a distribution that reflects the transformation but still allows a reliable reconstruction. (E.g., in the translational case, a shift of the input vector will simply lead to an according shift of the activities.)

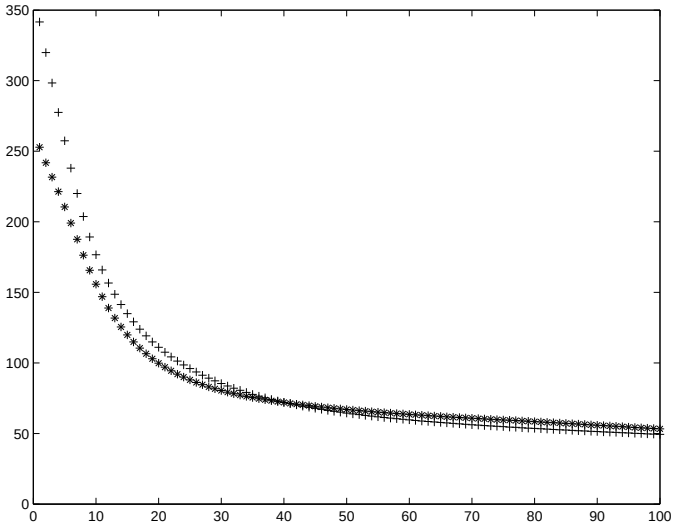


Fig. 6. Comparison of the standard NMF with overlapping NMF to show the equivalence. Plotted is the reconstruction error vs. iteration step for standard NMF with 8 basis vectors (stars) and overlapping NMF with 2 basis vectors (crosses) for the bar example. The error settles at an asymptotic baseline because of the limited number of basis vectors and the nonlinear threshold which cuts input activities at 1.

Still, if we use NMF with sufficient basis vectors, so that the system has the opportunity to generate basis vectors for each choice of transformation parameters (e.g., in the translational case, the "same" bar pattern at every possible position), then it would be equivalent to the corresponding overlapping NMF as shown in figure 6.

A further interesting point is that, in case of a translational invariance such as it is the case for retinotopic inputs and receptive field modeling, the method allows a reconstruction scheme with overlapping patches, differing from most standard reconstruction schemes, which usually reconstruct only local patches considered in isolation from the neighbouring patches.

In a first application to the learning of visual cortex receptive fields in a two-stage network similar to the setup of Hoyer and Hyvärinen [2], we found feature sets that contain more structural variation, thus resulting in a less redundant representation with respect to translations than in the standard case, see [5] for details.

NMF has been claimed to result in a "parts-based" encoding scheme, such as shown by Lee and Seung [3] in their 1999 paper for the example of human faces. There indeed, the resulting basis vectors were parts of faces such as eyes or mouth parts, which the algorithm puts together additively to reconstruct the entire face. Although these parts seemingly appeared to be spatially localized, the terms parts-based, localized and sparse should not be taken as describing the same property. Since nowwhere in the algorithm formulation (neither in the original NMF nor in the presented extension) there is included a concept of spatial neighborhood, the localization of the basis vectors in the case of faces is a result of statistical properties of the input. On the other hand, NMF leads to sparser basis vectors than PCA, and the localization together with the sparsity make up the "parts-based" character seen in the results from [3].

In this paper, we have chosen stimuli with bars, and the resulting set of basis vectors seems to be holistic rather than parts-based. This is again a consecuence of the input statistics together with the sparsity constraints on the activity and the number of allowed basis vectors and is not in contradiction to the original NMF formulation, as can be seen from the results of the standard NMF in figure 2. The ideal (but trivial) parts-based set of basis vectors would be the single pixel basis vectors, as shown in figure 3. We found this case rather uninteresting, but if it is desired, the degree of parts-basedness can be controlled by the sparsity parameter and the number of basis vectors. Less sparsity in the activity and a larger number of basis vectors leads to more sparsity in the basis vectors. The contribution of overlapping NMF may therefore be seen less in the parts-based character of the solutions but in the generation of an increasingly complex set of basis vectors under consideration of transformation symmetries.

REFERENCES

[1] J. Eggert and E. Koerner. Sparse coding and NMF. Submitted to IJCNN 2004.

[2] P. O. Hoyer and A. Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593–1605, 2002.

[3] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[4] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[5] H. Wersing, J. Eggert, and E. Körner. Sparse coding with invariance constraints. In *ICANN/ICONIP 2003 conference proceedings*, 2003. Conference proceedings paper.