TRANSFORMATION INVARIANT SPARSE CODING

Morten Mørup and Mikkel N. Schmidt

Section for Cognitive Systems, DTU Informatics, Technical University of Denmark
Richard Petersens Plads bld 321, 2800 Kgs. Lyngby, Denmark
e-mail: {mm,mns}@imm.dtu.dk

ABSTRACT

Sparse coding is a well established principle for unsupervised learning. Traditionally, features are extracted in sparse coding in specific locations, however, often we would prefer invariant representation. This paper introduces a general transformation invariant sparse coding (TISC) model. The model decomposes images into features invariant to location and general transformation by a set of specified operators as well as a sparse coding matrix indicating where and to what degree in the original image these features are present. The TISC model is in general overcomplete and we therefore invoke sparse coding to estimate its parameters. We demonstrate how the model can correctly identify components of non-trivial artificial as well as real image data. Thus, the model is capable of reducing feature redundancies in terms of pre-specified transformations improving the component identification.

1. INTRODUCTION

Sparse coding and the closely related independent component analysis (ICA) are well established principles for feature extraction in multi-media data [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The principle of sparse coding is to account for as much information as possible while transmitting as little information as necessary. Mathematically, this corresponds to attaining as few non-zero elements as possible in the code (i.e., an NP hard l_0 norm minimization problem), and sparse coding is thus closely related to redundancy reduction [11, 12]. Olshausen and Field [1] argue that the brain might employ sparse coding since it allows for increased storage capacity in associative memories; it makes the structure in natural signals explicit; it represents complex data in a way that is easier to read out at subsequent level of processing; and it is energy efficient. Thus, sparseness is a natural constraint for unsupervised learning, and often yields parsimonious features.

When we experience our surroundings, it is well known that our perception does not alter when we move the head or change gaze. Thus, our brain manages to interpret the world, despite the location, scale and orientation of the objects we navigate among. The visual area 1 (V1) in the visual cortex of the human brain is retinotopically organized, such that neighboring regions of the retina are also neighboring regions in V1 [13]; however, the visual processing in the human brain is also organized into orientation selective columns [14, 15]. Here, a typical simple cell responds best to some optimum stimulus orientation (illustrated in Figure 1), and the response measured by the number of impulses, as the receptive field is passed through, falls off over 10–20 degrees to either side of the optimum, declining steeply to zero outside this region. If an electrode is pushed through the cortex in a direction parallel to the surface, an amazingly regular sequence of changes in orientation occurs. Every time the electrode advances 0.05 millimeter, the preferred orientation shifts on average about 10 degrees clockwise or counterclockwise [15]. Thus, it seems neurons in the visual cortex are tuned to respond to given orientation and location of feature objects. Furthermore, as pointed out by Tanaka [16], neurons in the inferotemporal cortex respond to moderately complex features, icon alphabets, which are invariant to the position of the visual stimulus. Hence, these features are complex patterns rather than the Gabor-like features often obtained by sparse coding or ICA decomposition.

Inspired by these properties of feature extraction in the brain we find ample motivation for sparse coding incorporating invariance such as shift and rotation for analyzing image data. Furthermore, features invariant to shift could potentially constitute icon alphabets as observed in inferotemporal cortex.

2. TRANSFORMATION INVARIANT SPARSE CODING

It is demonstrated in [2] how sparse coding of image patches results in Gabor like features, based on the following model [2, 5]

$$I^{(k)}(x,y) \approx \mathbf{R}^{(k)} = \sum_{d=1}^{D} \alpha_{k,d} \Psi_d(x,y). \tag{1}$$

where, $I^{(k)}(x,y)$ denotes the kth image patch of the same size as the desired feature images, $\Psi_d(x,y)$, and α_d is the sparse code. Hence, each image patch is approximated by a sparse linear combination of the feature images.

In [17, 18] it is demonstrated how the sparse coding model can be extended to general transformation invariances of the feature images. The features are invariant to a pre-specified set of operators, T_r

$$I^k(x,y) \approx \mathbf{R}^{(k)} = \sum_{d=1}^{D} \sum_{r=1}^{R} \alpha_{k,d,r} T_r(\Psi_d)(x,y). \tag{2}$$

These operators, T_r, account for any desired transformation within each patch, such as scaling and rotation. The model is based on subdividing the image into image patches; thus, a drawback of the above approach is that the extracted features depend on how the image is subdivided, and the model cannot account for simple transformations such as shifts without introducing redundant features.

The models are estimated by

$$\arg\min_{\alpha,\Psi} \sum_{k=1}^{K} (\mathcal{D}(\mathbf{I}^{(k)}, \mathbf{R}^{(k)}) + \lambda \sum_{d=1}^{D} \log sp(\alpha_{k,d})) \tag{3}$$

where λ is a parameter, that defines the tradeoff between reconstruction error and sparseness of the code. $\mathcal{D}(\cdot,\cdot)$ is a distance measure of

This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

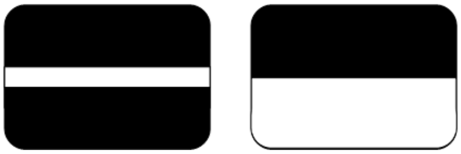

Fig. 1. Typical receptive-field maps for V1 simple cells [15]. The off-regions and on-regions of the cells are illustrated by the black and white colors.

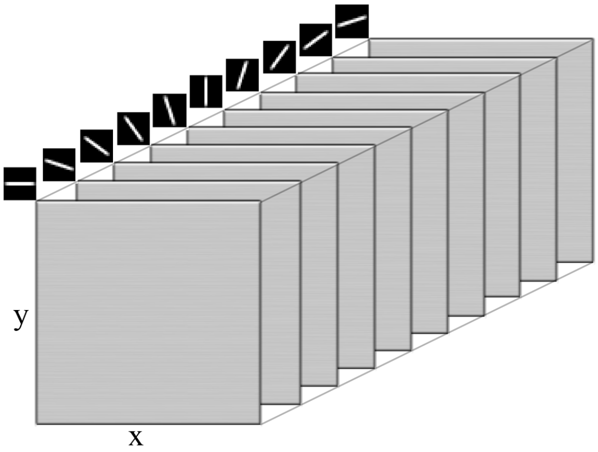

Fig. 2. Illustration of the sparse coding array for a given component, d, i.e. $\boldsymbol{\alpha}_{d,r}$ for $r \in \{1, 2, \ldots, 10\}$ of a total of R=10 rotations corresponding to the feature image of a bar being represented in the interval $[0°; 180°[$. The sparse code representation is similar to the organization of V1 of the human visual cortex, where the organization of the cells maintain the organization of the receptive field of the eye, i.e. the x and y coordinates, while each receptive field in V1 is organized into orientation selective columns, corresponding here to the indexing r over rotations.

the reconstruction error, for example the least squares error, and sp is the sparse prior distribution of α_d such as the Laplace distribution $sp(\alpha_d) \propto e^{-|\alpha_d|}$.

We presently propose the following model, that does not rely on subdividing the image into patches, and allows the features to be invariant to a given set of pre-specified transformations, T_r. Let $\mathbf{I} \in \mathbb{R}^{X \times Y}$ be the full image (without subdividing), then

$$\mathbf{I} \approx \sum_{d=1}^{D} \sum_{r=1}^{R} \boldsymbol{\alpha}_{d,r} * T_r(\boldsymbol{\Psi}_d). \tag{4}$$

where $*$ denotes 2-dimensional convolution, $\boldsymbol{\alpha}_{d,r} \in \mathbb{R}^{X+U \times Y+V}$ and $\Psi_d \in \mathbb{R}^{U \times V}$. The above model is related to shift invariant sparse coding [19, 20, 21, 22] with the extension of invariance to general transformations. The proposed model directly implements shift invariance through 2-D matrix convolution, which can be efficiently implemented in the Fourier domain. In the following, in addition to shift invariance, we consider invariance to rotation. Thus, T_r denotes a rotation operator, such that $T_r(\Psi_d)$ rotates the feature image, Ψ_d, $2\pi(r-1)/R$ radians clockwise. From this formulation of shift and rotation invariant sparse coding, a strong resemblance can be found between each component of the sparse code and the retinotopic organization in the human brain subdivided into orientation selective columns (see Figure 2).

To incorporate both shift and rotation invariance with respect to R different rotations, the sparse code, has a huge number of parameters, $(X+U) \cdot (Y+V) \cdot D \cdot R$, compared to previous shift and rotation invariant image decompositions, in which the analyzed images where subdivided into image patches prior to the analysis. Thus, it is not feasible to solve for the sparse code using traditional sparse coding algorithms based on computing the Hessian matrix. In order to estimate the parameters of the model, we use an efficient algorithm that relies only on gradient information.

The paper is structured as follows: In section 3, we derive an efficient sparse coding algorithm based only on gradient information, and compare this type of update to state-of-the-art algorithms for sparse coding. Based on this, we derive an efficient algorithm for the rotation and shift invariant sparse coding model. In section 4, we compare the features found by the rotational and shift invariant sparse coding model to the features obtained by the traditional sparse coding method, when analyzing the set of natural images described in [2].

3. METHOD

3.1. Solving efficiently for the sparse code

Consider again the sparse coding objective given in equation 3. The objective has two terms: a penalty for reconstruction error, and a penalty for non-sparseness. Using least squares as measure of reconstruction error and the Laplace prior as sparse distribution, corresponding to an l_1-norm penalty, the above problem becomes the well known LASSO [23] or basis pursuit denoising (BPD) [24] problem for a fixed value of Ψ

$$\arg\min_{\mathbf{S}} L(\mathbf{S}), \tag{5}$$

$$L(\mathbf{S}) = \frac{1}{2}\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2 + \lambda|\mathbf{S}|_1, \tag{6}$$

where $S_{d,k} = \alpha_{k,d}$, $X_{k,j(x,y)} = I^{(k)}(x,y)$, $A_{j(x,y),d} = \Psi_d(x,y)$, and $j(x,y)$ is a re-indexing of (x,y) that corresponds to vectorizing the feature images. Although the above objective is a convex optimization problem for fixed $\mathbf{A}$, no closed form solution exists, thus an iterative procedure must be employed in order to solve the problem. Several methods have been proposed: In [2], an algorithm based on conjugate gradient (Conj.Grad.) was used. In sparselab[1], the discontinuity of the derivative at zero is avoided, by turning the problem into a non-negative quadratic programming problem (this approach we presently denote BPD). In [19] the SignSearch algorithm was introduced, which is an active set procedure that estimates the sign of $\mathbf{S}$, such that a closed form solution can be obtained as $(\mathbf{A}^\top\mathbf{A})^\dagger(\mathbf{A}^\top\mathbf{X} - \lambda sgn(\mathbf{S}))$. In [20], the l_1-penalty is approximated by the quadratic penalty $|S|_1 = \sum_{d,j}|S_{d,j}| = \sum_{d,j}\frac{S_{d,j}^2}{Q_{d,j}}$, where $Q_{d,j} = \sqrt{S_{d,j}^2}$, and $\mathbf{Q}$ is kept fixed when computing the gradient and Hessian with respect to $\mathbf{S}$ despite it's dependence on $\mathbf{S}$. This procedure we will presently denote (BD-SC). [25, 26] introduce the least angle regression and selection (LARS) algorithm, that solves for $\mathbf{S}$ by computing the entire regularization path, i.e., the solution for all values of λ, at the computational cost of an ordinary least squares solution. All the above methods, except the conjugate gradient based approach, rely on computing the Hessian, and they are thus very memory intensive for large problems.

[1] `http://sparselab.stanford.edu/`

Algorithm 1 Gradient Based Sparse Coding (GB-SC)

```
repeat
    Update $\mathbf{S}$ according to reconstruction penalty,
    $\mathbf{S}^{new} = \mathbf{S} - \mu(\mathbf{A}^\top(\mathbf{AS} - \mathbf{X})$
    Update $\mathbf{S}^{new}$ according to the sparsity penalty, such that elements crossing zero are set to zero,
    $\mathbf{S}^{new}_{d,j} = \begin{cases} 0 & \text{if } |\mathbf{S}^{new}_{d,j}| < \mu\lambda \\ \mathbf{S}^{new}_{d,j} - \mu\lambda sgn(\mathbf{S}^{new}_{d,j}) & \text{otherwise} \end{cases}$
    if $L(\mathbf{S}^{new}) < L(\mathbf{S})$ then
        $\mu = 1.2\mu$
        $\mathbf{S} = \mathbf{S}^{new}$
    else
        $\mu = \mu/2$
    end if
until convergence
```

Unfortunately, simple gradient based methods normally fail in finding the optimal solution, since they tend to get stuck in very small step sizes, due to oscillations around zero. To see this, consider the gradient of the objective in Equation (5) given by

$$\mathbf{G} = \mathbf{A}^\top(\mathbf{AS} - \mathbf{X}) + \lambda sgn(\mathbf{S}). \tag{7}$$

A gradient based update would be given by

$$\mathbf{S}^{new} = \mathbf{S} - \mu\left(\mathbf{A}^\top(\mathbf{AS} - \mathbf{X}) + \lambda sgn(\mathbf{S})\right); \tag{8}$$

however, if $|\lambda sgn(\mathbf{S}^{old})|_{d,j} >> |\mathbf{A}^\top(\mathbf{AS}^{old} - \mathbf{X})|_{d,j}$, the regularization will dominate the update, and rather than be forced to zero, $\mathbf{S}_{d,j}$ will cross zero, and in subsequent updates, oscillate around zero, until the step size, μ, becomes infinitesimal small, even though the regularization is minimized when elements in $\mathbf{S}$ becomes zero. (see Figure 3). At first glance, this might appear to be a minor concern; however, when many elements of $\mathbf{S}$ are close to zero, the joint effect of all these oscillations will completely dominate the update.

To avoid the oscillations, we propose to split the gradient based update into the following simple two step procedure: update the solution, first, based on the gradient of the reconstruction error term, and second, based on the regularization term, as described in Algorithm 1. This simple algorithm avoids the oscillatory behavior encountered in regular gradient descent. Notice that $\mathbf{S}$ is only updated, if the update decreases the objective. Furthermore, when the step-size, μ, becomes very small, elements do not change sign, and the proposed update is equivalent to regular gradient descent, and hence, the proposed algorithm has the same fixed points as the regular gradient descent procedure. Notice also, that although the procedure in Algorithm 1 is given for least squares minimization with sparsity penalty based on the Laplace prior, (i.e., the L_1 norm,) the approach of splitting the gradient into an update for the reconstruction error and an update for the sparsity penalty generalizes directly to other types of reconstruction metrics and sparsity penalty measures (i.e., $\log sp(\alpha_{k,d})$). A simple but useful convergence criterion is to stop iterating when the relative change in the objective is below some small value, ϵ.

3.2. Transformation Invariant Sparse Coding Algorithm

We now return to the Transformation Invariant Sparse Coding (TISC) model

$$\mathbf{I} \approx \mathbf{R} = \sum_{d=1}^{D}\sum_{r=1}^{R} \boldsymbol{\alpha}_{d,r} * T_r(\boldsymbol{\Psi}_d). \tag{9}$$

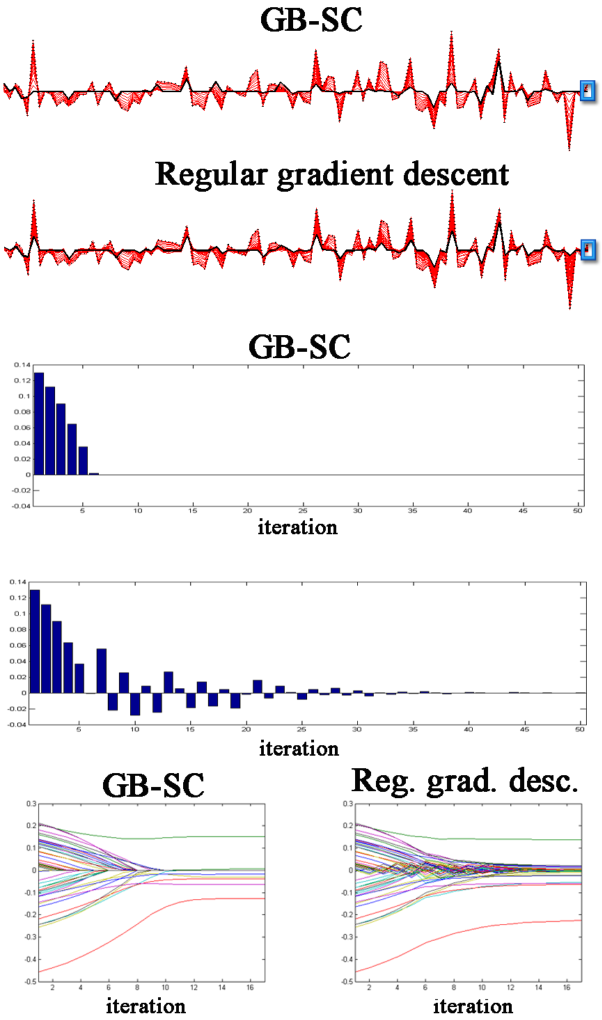

Fig. 3. Illustration of how the proposed gradient based sparse coding alleviates poor convergence due to oscillations around zero. **Top panel:** the progression through 50 iterations of the GB-SC algorithm as well as regular gradient descent on a problem with 100 variables. The dotted black line is the initial solution, the solid black line is the solution obtained after 50 iterations, and the red lines are intermediate results. For the GB-SC, the final solution is the global minimum of the problem. The regular gradient descent algorithm does not converge fast to the optimum, but oscillates around zero. **Middle panel:** Inspection of the progression of one variable (marked by blue boxes in the top panel). No oscillations are found for the GB-SC based method, whereas the regular gradient descent method oscillates around zero causing the algorithm to suffer from slow convergence. **Bottom panel:** The progression of the coefficients through 17 iterations for a problem with 50 variables. Even for this relatively small problem, regular gradient descent is stuck in suboptimal solution due to oscillations around zero, whereas the GB-SC efficiently finds the optimal solutions.

As we would like the model to extract features that are similar across various different images, we extend the model to N images of arbi-

	256×100	256×256	256×1000	256×2500
BD-SC	0.3641 ± 0.3044	11.6250 ± 4.4922	—	—
SignSearch	0.0750 ± 0.0359	0.1984 ± 0.1342	$\mathbf{0.3734 \pm 0.1759}$	$\mathbf{1.6969 \pm 0.6441}$
Conjugate gradient	0.4172 ± 0.0651	1.1219 ± 0.2560	9.0297 ± 1.8055	45.6297 ± 12.0142
LARS	0.0453 ± 0.0226	$\mathbf{0.1313 \pm 0.0787}$	0.4313 ± 0.1477	1.9813 ± 0.6342
BPD	0.5703 ± 0.0696	0.9313 ± 0.0748	2.8719 ± 0.1389	15.5047 ± 0.7882
GB-SC	$\mathbf{0.0125 \pm 0.0066}$	0.3172 ± 0.2121	2.0688 ± 1.0760	22.8828 ± 12.2846

Table 1. Comparison of the CPU time for various sparse coding algorithms on different problem sizes.

trary size

$$\mathbf{I}^{(n)} \approx \mathbf{R}^{(n)} = \sum_{d=1}^{D} \sum_{r=1}^{R} \boldsymbol{\alpha}_{d,r}^{(n)} * T_r(\boldsymbol{\Psi}_d). \qquad (10)$$

Hence, the nth image is modeled by a sparse code, $\boldsymbol{\alpha}_{d,r}^{(n)}$, convolved with the pre-specified transformations of set of feature images, $T_r(\boldsymbol{\Psi}_d)$, that is shared by all the N images, and summed over all rotations and features.

Using the least squares error for the reconstruction penalty (corresponding to a Gaussian noise model) and imposing a Laplace prior to promote sparsity we obtain the following objective

$$\sum_{n=1}^{N} \frac{1}{2} \|\mathbf{I}^{(n)} - \mathbf{R}^{(n)}\|_F^2 + \lambda \sum_{d,r} |\boldsymbol{\alpha}_{d,r}^{(n)}|_1. \qquad (11)$$

Presently, we consider rotation invariant features, thus, r indexes a set of predefined rotation operators. For an illustration of this, see Figure 2 and 4.

The derivative of the objective function (11) with respect to $\boldsymbol{\alpha}_{d,r}^{(n)}$ and Ψ_d is

$$\nabla \boldsymbol{\alpha}_{d,r}^{(n)} = \left(\mathbf{I}^{(n)} - \mathbf{R}^{(n)}\right) * \left(T_\pi(T_r(\boldsymbol{\Psi}_d))\right) + \lambda sgn\left(\boldsymbol{\alpha}_{d,r}^{(n)}\right), \quad (12)$$

$$\nabla \Psi_d = \sum_{n=1}^{N} \sum_{r=1}^{R} T_r^{-1}\left(\mathbf{I}^{(n)} - \mathbf{R}^{(n)}\right) * T_r^{-1}\left(T_\pi\left(\boldsymbol{\alpha}_{d,r}^{(n)}\right)\right), \quad (13)$$

where T_r^{-1} denotes the inverse rotation operator, and T_π denotes rotation of 180 degrees. We implemented the rotation operator T_r using linear interpolation between the image pixels. In image regions, where T_r^{-1} and T_r are not valid, we zero padded the data. Ψ_d was updated such that $\|\Psi_d\|_F = 1$ as proposed for sparse coding in [2, 20] based on the normalization invariant projected gradient approach proposed in [4].

4. RESULTS

In Table 1, the performance of the BD-SC, SignSearch, conjugate gradient, LARS, and BPD algorithms with the proposed GB-SC method are compared for a range of different problem sizes. The problem solved is $\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{s}|_F^2 + \lambda|\mathbf{s}|_1$, for $\lambda = 0.05$. $J \times D$ denotes the size of $\mathbf{A}$, (J image pixels and D basis vectors.) The mean and standard deviation is given for 10 randomly generated problems, each given by setting $\mathbf{A}$ to D randomly chosen columns from the natural images data set [2] and $\mathbf{x}$ to a randomly selected image patch, not already used in the dictionary, $\mathbf{A}$. Notice, SC-BD, SignSearch and LARS all find the global optimum. The remaining algorithms were stopped, when their deviation from the true optimum was less than 10^{-4}. For $D \leq J$, the proposed GB-SC is the fastest of all the algorithms, but for over-complete problems, i.e., $D \gg J$, the GB-SC algorithm is not in general as effective as the other algorithms, which use Hessian information; however, it is still faster than the conjugate gradient based method. Hence, the proposed algorithm is not only simple, but also efficient, and even outperforms state of the art algorithms for $D \leq J$. SC-BD for 256×1000 and 256×2500 was not included, as it was more than 100 times slower than the conjugate gradient algorithm. The conjugate gradient algorithm was obtained from `www.l1-magic.org`, whereas the BPD and LARS was obtained from `www.sparselab.stanford.edu`. The SignSearch algorithm was kindly provided by H. Lee [19].

Figure 4 shows the result of a rotation and shift invariant sparse coding of a synthetically generated dataset. The data consists of a number of bar and C-shapes, randomly rotated between 20 uniformly distributed orientations over the interval $[0; 360°]$. From the figure it can be seen that when the regularization strength λ is week, most of the information is coded in the sparse code, while for the "correct" degree of sparsity, the information of the bar and C-shape is coded in the features. When the regularization is too strong, only the most prominent regions are coded, which results in features that are highly localized.

Figure 5 shows the result of a regular sparse coding analysis of the natural scenes image data given in [2]. The data was preprocessed as described in [2]. Figure 6 and 7 show the corresponding results based on shift-invariance and rotation-and-shift-invariance. In the shift invariant model, we used 10 features, and in the rotation and shift invariant model, we used 2 features and 10 rotational representations covering the interval $[0; 180°[$. The data set consists of 10 natural images of size 512×512 from [2]. Thus, the resulting size of the sparse code was $512 \times 512 \times 10 \times 10 \times 2 = 52,428,800$ variables.

5. DISCUSSION

The transformation invariant sparse coding (TISC) model presently derived, codes images in a representation that resembles the organization of the visual processing in the visual area 1 in the human brain. Both the TISC model, as well as the visual processing system of the brain, code images such that the retinotopic mapping is preserved, while features are coded in orientation selective columns, as demonstrated in Figure 2.

In Table 1 it was seen, that the proposed gradient based sparse coding (GB-SC) algorithm, despite relying solely on gradient information, was comparable in performance to state of the art algorithms. It is even faster than the other algorithms when $J > D$. Thus, the proposed GB-SC forms a simple yet efficient algorithm for sparse coding. As the TISC model was solved by alternatingly solving for the features $\boldsymbol{\Psi}_d$ and the sparse code $\boldsymbol{\alpha}_{d,r}^{(n)}$ a benefit of the proposed gradient based approach is that rather than solving exactly

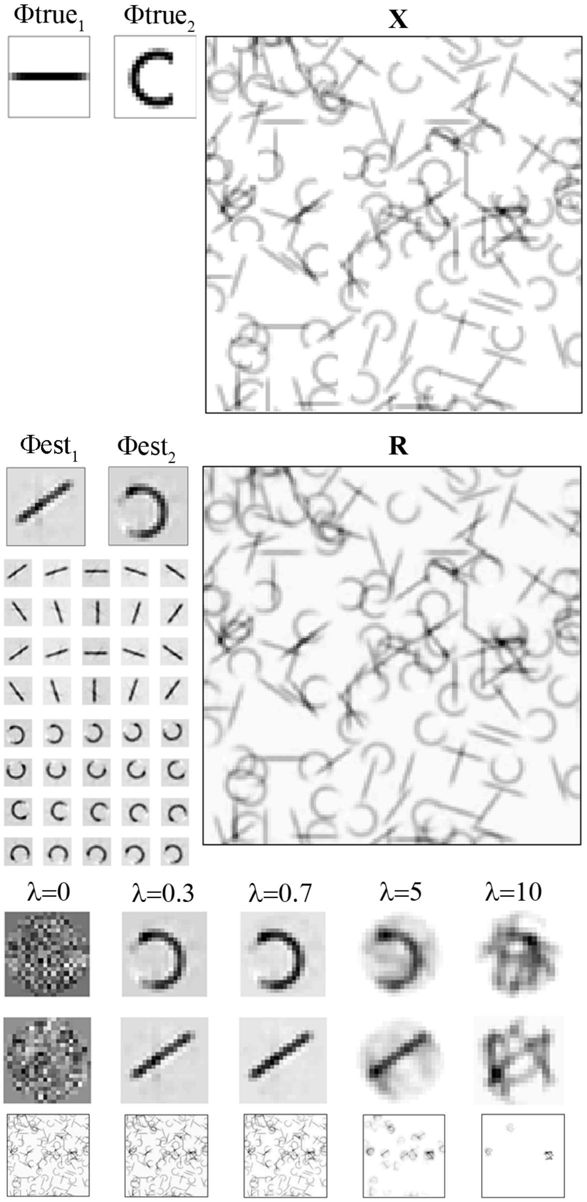

Fig. 4. A rotation and shift invariant sparse coding analysis of a synthetically generated dataset. **Top panel:** The feature images consist of a bar and a C shape, at random locations and orientations forming the synthetic image data $\mathbf{X}$. **Middle panel:** Estimated features and data using the rotation and shift invariant sparse coding algorithm. **Bottom panel:** Inspection of the results obtained for different values of the regularization parameter λ, given are the two estimated features as well as the reconstructed data. Note that the gray background of the estimated features are due to a different color axis used to show small regions of the estimated feature images with negative values.

for $\mathbf{\Psi}_d$ for fixed $\boldsymbol{\alpha}_{d,r}^{(n)}$ and vice-versa at each iteration at a high computational cost $\mathbf{\Psi}_d$ and $\boldsymbol{\alpha}_{d,r}^{(n)}$ were instead refined at a relatively low computational cost such that changes were propagated during each gradient step between $\mathbf{\Psi}_d$ and $\boldsymbol{\alpha}_{d,r}^{(n)}$.

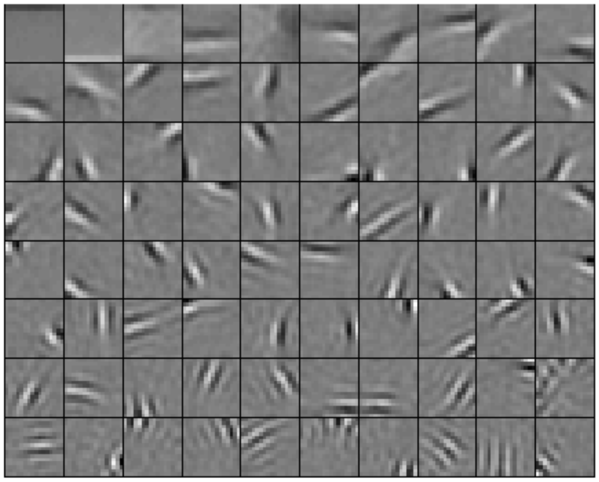

Fig. 5. Feature images, $\mathbf{\Psi}$, of size 16×16 obtained by analyzing the natural image data subdivided patches of size 16×16, according to the model given in Equation (1). The result is Gabor-like features, as reported in [2]; however, we note that the features appear redundant with respect to shift and rotation (many of the features are, more or less, shifted and/or rotated versions of other features). Hence, a representation that does not depend on the specific choice of subdivision of the images, while taking into account the shift and rotation redundancies, is desirable.

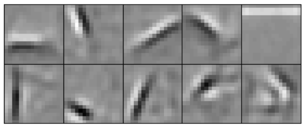

Fig. 6. Shift invariant feature images, $\mathbf{\Psi}$, of size 16×16 obtained when analyzing the natural image data using shift invariant sparse coding. Similar to sparse coding, Gabor-like features are obtained; however, the features are not redundant with respect to shift, since the model can use each feature at any position. The features appear, however, are redundant with respect to rotation.

In the analysis of the synthetically generated dataset of a bar and C-shape in random shifted and rotated positions, the TISC algorithm was able to correctly identify the correct features (see Figure 4.) The degree of sparsity, controlled by the parameter λ, was important for the success of the algorithm in extracting the components. A too low degree of sparsity made the model code most of the information in the code matrix, $\boldsymbol{\alpha}$, while the features were more or less given by random patterns. For a suitable degree of sparsity, the two features were correctly identified, and the sparse code consisted only of peaks corresponding to the location and orientation of the features in the data. Imposing to much sparsity on the other hand resulted in coding of only regions containing most prominent activity, such that the feature images over-fitted to these specific regions. This property was also found when analyzing the natural scene images.

In the classical application of sparse coding to natural images, using the data described in [2], we illustrated how the traditional sparse coding, where the images are subdivided into image patches, yielded features, that were highly redundant in terms of shift and rotation. By imposing rotation and shift invariance, this redundance is directly included in the model. In our analysis, where we used

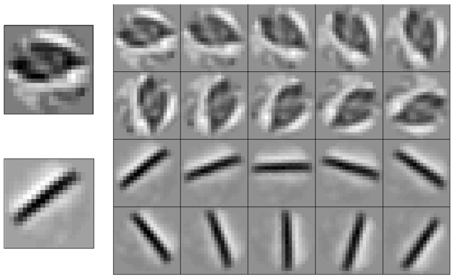

Fig. 7. Rotation and shift invariant feature images $\mathbf{\Psi}$ of size 20×20 obtained when analyzing the natural image data using the rotation and shift invariant sparse coding algorithm. Notice, due to the rotation invariance, only the central areas of the features are non-zero. The first feature obtained seem to mimic on-center off-surround behavior, while the second feature resembles an edge detector, corresponding to the simple cell behavior given in Figure 1. To the right, the 10 rotated representations of the features are shown.

only two features, one feature corresponded to low frequency on-center off-surround behavior, while the other corresponded to an edge, hence resembled the typical simple cell characteristic illustrated in Figure 1. Thus, the proposed TISC model extracts features, that more closely resemble simple cell behavior compared to traditional sparse coding, and the rotation and shift invariance is able to greatly reduce the redundancy of the extracted features. While we presently considered rotation invariance we note that the proposed TISC readily generalize to other types of invariances such as invariance to scale.

6. REFERENCES

[1] B A. Olshausen and D J. Field, “Sparse coding of sensory inputs,” *Current Opinion in Neurobiology*, vol. 14, pp. 481–487, 2004.

[2] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, pp. 607–609, 1996.

[3] P.O. Hoyer, “Non-negative sparse coding,” *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pp. 557–565, 2002.

[4] J. Eggert and E. Körner, “Sparse coding and nmf,” in *Neural Networks*, 2004, vol. 4, pp. 2529–2533.

[5] B A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1,” *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[6] A Hyvärinen and P. O. Hoyer, “A two-layer coding model learns simple and complex cell receptive fields and topography from natural images,” *Vision Research*, vol. 21, no. 18, pp. 2413–2423, 2001.

[7] T.-W. Lee and M. S. Lewicki, “Unsupervised image classification, segmentation, and enhancement using ica mixture models,” *IEEE tansactions on Image Processing*, vol. 11, no. 3, pp. 270–279, 2002.

[8] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley and Sons., 2001.

[9] P. O. Hoyer and A. Hyvärinen, “Independent component analysis applied to feature extraction colour and stereo images,” *Network: Computation in Neural Systems*, vol. 11, no. 3, pp. 191–210, 2000.

[10] T. Kolenda, L. K. Hansen, J. Larsen, and O. Winther, “Independent component analysis for understanding multimedia content,” in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII*, H. Bourlard, T. Adali, S. Bengio, J. Larsen, and S. Douglas, Eds., Piscataway, New Jersey, 2002, pp. 757–766, IEEE Press, Martigny, Valais, Switzerland, Sept. 4-6, 2002.

[11] H.B. Barlow, “Possible principles underlying the transformations of sensory messages,” *Sensory Communication, MIT Press*, pp. 217–234, 1961.

[12] D.J. Field, “What is the goal of sensory coding?,” *Neural Computation*, vol. 6, pp. 559–601, 1994.

[13] R.B. Tootell, M.S. Silverman, E. Switkes, and R.L. De Valois, “Deoxyglucose analysis of retinotopic organization in primate striate cortex,” *Science*, vol. 218, no. 4575, pp. 902–904, 1982.

[14] T. D. Albright, “Direction and orientation selectivity of neurons in visual area mt of the macaque,” *Journal of Neurophysiology*, vol. 52, no. 6, pp. 1106–1130, 1984.

[15] D. Hubel, *Eye, Brain and Vision*, http://hubel.med.harvard.edu/, 1995.

[16] K. Tanaka, “Representation of visual features of objects in the inferotemporal cortex,” *Neural Networks*, vol. 9, no. 8, pp. 1459–1475, 1996.

[17] J. Eggert, H. Wersing, and E. Körner, “Transformation-invariant representation and nmf,” in *Neural Networks*, 2004, vol. 4, pp. 2535– 2539.

[18] H. Wersing, J. Eggert, and E. Körner, “Sparse coding with invariance constraints,” *Proc. Int. Conf. Artificial Neural Networks ICANN*, pp. 385–392, 2003.

[19] H. Lee, A. Battle, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” *In Proceedings of the Neural Information Processing Systems (NIPS)*, vol. 19, 2007.

[20] T. Blumensath and M. Davies, “On shift-invariant sparse coding,” *International Conference on Independent Component Analysis and Blind Source Separation*, vol. 26, pp. 1205–1212, 2004.

[21] M. Mørup, M. N. Schmidt, and L. K. Hansen, “Shift invariant sparse coding of image and music data,” 2008.

[22] P. Smaragdis, B. Raj, and M. Shashanka, “Sparse and shift-invariant feature extraction from non-negative data,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2069–2072, 2008.

[23] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[24] S. C. Shaobing and D. Donoho, “Basis pursuit,” *28th Asilomar conf. Signals, Systems Computers*, 1994.

[25] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[26] M.R. Osborne, B. Presnell, and B.A. Turlach, “A new approach to variable selection in least squares problems,” *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.