(Murray:2004) IEEE Workshop on Machine Learning for Signal Processing, Sao Luis, Brazil, Sep 2004

SPARSE IMAGE CODING USING LEARNED OVERCOMPLETE DICTIONARIES

Joseph F. Murray[1] and Kenneth Kreutz-Delgado
University of California, San Diego
Electrical and Computer Engineering
9500 Gilman Dr Dept 0407
La Jolla Ca 92093-0407
Email: jfmurray@ucsd.edu, kreutz@ece.ucsd.edu

Abstract. Images can be coded accurately using a sparse set of vectors from an overcomplete dictionary, with potential applications in image compression and feature selection for pattern recognition. We discuss algorithms that perform sparse coding and make three contributions. First, we compare our overcomplete dictionary learning algorithm (FOCUSS-CNDL) with overcomplete Independent Component Analysis (ICA). Second, noting that once a dictionary has been learned in a given domain the problem becomes one of choosing the vectors to form an accurate, sparse representation, we compare a recently developed algorithm (Sparse Bayesian Learning with Adjustable Variance Gaussians) to well known methods of subset selection: Matching Pursuit and FOCUSS. Third, noting that in some cases it may be necessary to find a non-negative sparse coding, we present a modified version of the FOCUSS algorithm that can find such non-negative codings.

INTRODUCTION

We discuss the problem of representing images with a highly sparse set of vectors drawn from a learned overcomplete dictionary. The problem has received considerable attention since the work of Olshausen and Field [8], who suggest that this is the strategy used by the visual cortex for representing images. The implication is that a sparse, overcomplete representation is especially suitable for visual tasks such as object detection and recognition that occur in higher regions of the cortex. A key result of this line of work is that images (and other data) can be coded more efficiently using a learned basis than with a non-adapted basis (e.g. wavelet and Gabor dictionaries) [5]. Our earlier work has shown that overcomplete codes can be more efficient

[1]J.F. Murray was supported by the Sloan Foundation and the Arcs Foundation.

than complete codes in terms of entropy (bits/pixel), even though there are many more coefficients than image pixels in an overcomplete code [4].

Non-learned dictionaries (often composed of Gabor functions) are used to generate the features in many pattern recognition systems [12], and we believe that their performance could be improved using learned dictionaries that are adapted to the image statistics of the inputs.

Another natural application of sparse image coding is image compression. Standard compression methods such as JPEG use a fixed, complete basis (e.g. discrete cosines). Compression systems (based on methods closely related to those presented here) have shown that using learned overcomplete dictionaries can provide improved compression over such standard techniques [2]. Other applications of sparse coding include high-resolution spectral estimation, direction-of-arrival estimation, speech coding, biomedical imaging and function approximation [10].

In some problems, we may desire (or the physics of the problem may dictate) non-negative sparse codings. A multiplicative algorithm for non-negative coding was developed and applied to images [3]. A non-negative Independent Component Analysis (ICA) algorithm was presented in [9] (which also discusses other applications). In [3, 9] only the complete case was considered. Here, we present an algorithm that can learn non-negative sources from an overcomplete dictionary, which leads naturally to a learning method that adapts the dictionary for such sources.

SPARSE CODING AND VECTOR SELECTION

The problem of sparse coding is that of representing some data $y \in \mathbb{R}^m$ (e.g. a patch of an image) using a small number of non-zero components in a source vector $x \in \mathbb{R}^n$ under the linear model

$$y = Ax + \nu \,, \tag{1}$$

where the dictionary $A \in \mathbb{R}^{m \times n}$ may be overcomplete ($n \geq m$), and the additive noise ν is assumed to be Gaussian, $p_\nu = \mathcal{N}(0, \sigma_\nu^2)$. By assuming a prior $p_X(x)$ on the sources, we can formulate the problem in a Bayesian framework and find the maximum *a posteriori* solution for x,

$$\begin{aligned} \widehat{x} &= \arg\max_x p(x|A, y) \\ &= \arg\max_x [\log p(y|A, x) + \log p_X(x)] \,. \end{aligned} \tag{2}$$

By making an appropriate choice for the prior $p_X(x)$, we can find solutions with high sparsity (i.e. few non-zero components). We define *sparsity* as the number of elements of x that are zero, and the related quantity *diversity* as the number of non-zero elements, so that diversity = ($n-$sparsity). Assuming the prior distribution of the sources x is a generalized exponential of the form,

$$p_X(x) = c_x e^{-\gamma_p \, d_p(x)} \,, \tag{3}$$

where the parameter p determines the shape of distribution and c_x is a normalizing constant to ensure $p_X(x)$ is a density function. A common choice for the prior on x is for the function $d_p(x)$ to be the p-norm-like measure,

$$d_p(x) = \|x\|_p^p = \sum_{i=1}^{n} |x[i]|^p \,, \quad 0 \leq p \leq 1 \,, \tag{4}$$

where $x[i]$ are the elements of the vector x. When $p = 0$, $d_p(x)$ is a count of the number of non-zero elements of x (diversity), and so $d_p(x)$ is referred to as a *diversity measure* [4].

With these choices for $d_p(x)$ and p_ν, we find that,

$$\begin{aligned} \widehat{x} &= \arg\max_x [\log p(y|A,x) + \log p_X(x)] \\ &= \arg\min_x \|y - Ax\|^2 + \lambda \|x\|_p^p \,. \end{aligned} \tag{5}$$

When $p \to 0$ we obtain an optimization problem that directly minimizes the reconstruction error and the diversity of x. When $p = 1$ the problem no longer directly minimizes diversity, but the right-hand-side of (5) has the desirable property of being globally convex and so has no local minima. The $p = 1$ cost function is used by the Basis Pursuit algorithm [13].

FOCUSS and Non-negative FOCUSS

For a given, known dictionary A, the Focal Underdetermined System Solver (FOCUSS) was developed to solve (5) for $p \leq 1$ [10]. The algorithm is an iterative re-weighted factored-gradient approach, and has consistently shown better performance than greedy vector-selection algorithms such as Basis Pursuit and Matching Pursuit, although at a cost of increased computation [10]. Previous versions of FOCUSS have assumed that x is unrestricted on $\mathbb{R}^n$. In some cases however, we may require that the sources be non-negative, $x[i] \geq 0$. This amounts to a change of prior on x from symmetric to one-sided, but this results in nearly the same optimization problem as (5). To create a non-negative FOCUSS algorithm, we need to ensure that the $x[i]$ are initialized to non-negative values, and that each iteration keeps the sources in the feasible region. To do so, we propose the *non-negative* FOCUSS algorithm,

$$\begin{aligned} \Pi^{-1}(\widehat{x}_k) &= \mathrm{diag}(|\widehat{x}_k[i]|^{2-p}) \\ \lambda_k &= \lambda_{\max} \left(1 - \frac{\|y_k - A\widehat{x}\|}{\|y_k\|} \right) \,, \quad \lambda_k > 0 \\ \widehat{x}_k &\leftarrow \Pi^{-1}(\widehat{x}_k) A^T \left(\lambda_k I + A \Pi^{-1}(\widehat{x}_k) A^T \right)^{-1} y_k \\ \widehat{x}_k[i] &\leftarrow \begin{cases} 0 & \widehat{x}_k[i] < 0 \\ \widehat{x}_k[i] & \widehat{x}_k[i] \geq 0 \end{cases} \,, \end{aligned} \tag{6}$$

where λ_k is a heuristically-adapted regularization term, limited by $\lambda_{\max}$ which controls the tradeoff between sparsity and reconstruction error (higher values

of λ lead to more sparse solutions, at the cost of increased error). We denote this algorithm FOCUSS+, to distinguish from the FOCUSS algorithm [4] which omits the last line of (6). The estimate of x is refined over iterations of (6) and usually 10 to 50 iterations are needed for convergence (defined as the change in x being smaller than some threshold from one iteration to the next).

Sparse Bayesian Learning with Adjustable Variance Gaussian Priors (SBL-AVG)

Recently, a new class of Bayesian model characterized by Gaussian prior sources with adjustable variances has been developed [11]. These models use the linear generating model (1) for the data y but instead of using a non-Gaussian sparsity inducing prior on the sources x (as FOCUSS does), they use a flexibly-parameterized Gaussian prior,

$$p_X(x) = p(x|\alpha) = \prod_{i=0}^{n} \mathcal{N}(x[i]|0, \alpha_i^{-1}) \,, \tag{7}$$

where the variance hyperparameter α_i^{-1} can be adjusted for each component $x[i]$. When α_i^{-1} approaches zero, the density of $x[i]$ becomes sharply peaked making it very likely that the source will be zero, increasing the sparsity of the code. The algorithm for estimating the sources has been termed Sparse Bayesian Learning (SBL), but we find this term to be too general, as other algorithms (including the older FOCUSS algorithm) also estimate sparse components in a Bayesian framework. We use the term SBL-AVG (Adjustable Variance Gaussian) to be more specific.

To insure that the prior probability $p(x|\alpha)$ is sparsity-inducing, an appropriate prior on the hyperparameter α must be chosen. In general, a Gamma$(\alpha_i|a, b)$ distribution can be used for the prior of α_i, and in particular with $a = b = 0$, the prior on α_i becomes uniform. This leads to $p(x[i])$ having a Student's t-distribution which qualitatively resembles the ℓ_p-like distributions (with $0 < p \leq 1$) used to enforce sparsity in FOCUSS and other algorithms.

SBL-AVG has been used successfully for pattern recognition, with performance comparable to Support Vector Machines (SVMs) [11]. In these applications the known dictionary A is a kernel matrix created from the training examples in the pattern recognition problem just as with SVMs. The performance of SBL-AVG was similar to SVM in terms of error rates, while using far fewer support vectors (non-zero x_i) resulting in simpler models. Theoretical properties of SBL-AVG for subset selection have been elucidated [13], and simulations on synthetic data show superior performance over FOCUSS and other basis selection methods. To our knowledge, results have not been previously reported for SBL-AVG on image coding.

Modified Matching Pursuit (MMP): Greedy vector selection

Many variations on the idea of matching pursuit, or greedy subset selection, have been developed. Here, we use Modified Matching Pursuit (MMP) [1] which selects each vector (in series) to minimize the residual representation error. The simpler Matching Pursuit (MP) algorithm is more computationally efficient, but provides less accurate reconstruction. More details and comparisons can be found in [1]. For the case of non-negative sources, matching pursuit can be suitably adapted, and we call this algorithm MP+.

DICTIONARY LEARNING ALGORITHMS

In the previous section we discussed algorithms that accurately and sparsely represent a signal using a known, predefined dictionary A. Intuitively, we would expect that if A were adapted to the statistics of a particular problem that better and sparser representations could be found. This is the motivation that led to the development of the FOCUSS-CNDL dictionary learning algorithm. Dictionary learning is closely related to the problem of ICA which usually deals with complete A but can be extended to overcomplete A [6].

FOCUSS-CNDL

The FOCUSS-CNDL algorithm solves the problem (1) when both the sources x and the dictionary A are assumed to be unknown random variables [4]. The algorithm contains two major parts, a sparse vector selection step and a dictionary learning step which are derived in a jointly Bayesian framework. The sparse vector selection is done by FOCUSS (or FOCUSS+ if non-negative x_i are needed), and the dictionary learning A-update step uses gradient descent.

With a set of training data $Y = (y_1, \ldots, y_N)$ we find the maximum *a posteriori* estimates $\widehat{A}$ and $\widehat{X} = (\widehat{x}_1, \ldots, \widehat{x}_N)$ such that

$$(\widehat{A}, \widehat{X}) = \arg\min_{A,X} \sum_{k=1}^{N} \|y_k - Ax_k\|^2 + \lambda d_p(x_k)\,, \tag{8}$$

where $d_p(x) = \|x_k\|_p^p$ is the diversity measure (4) that measures the number of non-zero elements of a source vector x_k (see above).

The optimization problem (8) attempts to minimize the squared error of the reconstruction of y_k while minimizing d_p and hence the number of non-zero elements in $\widehat{x}_k$. The problem formulation is similar to ICA in that both model the input Y as being linearly generated by unknowns A and X, but ICA attempts to learn a new matrix W which by $Wy_k = \widehat{x}_k$ linearly produces estimates $\widehat{x}_k$ in which the components $\widehat{x}_{i,k}$ are as statistically independent as possible. ICA in general does not result in as sparse solutions as FOCUSS-CNDL which specifically uses a *sparsity-inducing* non-linear iterative FOCUSS algorithm to find $\widehat{x}_k$.

We now summarize the FOCUSS-CNDL algorithm which was fully derived in [4]. For each of the N data vectors y_k in Y, we update the sparse source vectors $\widehat{x}_k$ using one iteration of the FOCUSS or FOCUSS+ algorithm (6). After updating $\widehat{x}_k$ for $k = 1...N$ the dictionary $\widehat{A}$ is re-estimated,

$$\begin{aligned} \Sigma_{y\hat{x}} &= \frac{1}{N}\sum_{k=1}^{N} y_k \widehat{x}_k^T\,, \quad \Sigma_{\hat{x}\hat{x}} = \frac{1}{N}\sum_{k=1}^{N} \widehat{x}_k \widehat{x}_k^T\,, \quad \delta\widehat{A} = \widehat{A}\,\Sigma_{\hat{x}\hat{x}} - \Sigma_{y\hat{x}} \\ \widehat{A} &\leftarrow \widehat{A} - \gamma\left(\delta\widehat{A} - \mathrm{tr}\,(\widehat{A}^T\delta\widehat{A})\widehat{A}\right)\,, \quad \gamma > 0\,, \end{aligned} \tag{9}$$

where γ is the learning rate parameter. Each iteration of FOCUSS-CNDL consists of updating all $x_k, k = 1...N$ with one FOCUSS iteration (6), followed by a dictionary update (9) (which uses Σ calculated from the updated $\widehat{x}_l$ estimates). After each update of $\widehat{A}$, the columns are adjusted to have equal norm $\|a_i\| = \|a_j\|$, in such a way that $\widehat{A}$ has unit Frobenius norm, $\|\widehat{A}\|_F = 1$.

Overcomplete Independent Component Analysis (ICA)

Another method for learning an overcomplete dictionary based on ICA was developed by Lewicki and Sejnowski [5, 6]. In the overcomplete case, the sources must be estimated as opposed to in standard ICA (complete A), where the sources are found by multiplying by the learned matrix W, $\widehat{x} = Wy$. In [5] the sources are estimated using a modified conjugate gradient optimization of a cost function closely related to (5) that uses the 1-norm (derived using a Laplacian prior on x). The dictionary is updated by gradient ascent on the likelihood using a Gaussian approximations (cf. [5] eq. 20).

MEASURING PERFORMANCE

To compare the performance of image coding algorithms we need to measure two quantities: distortion and compression. As a measure of distortion we use a normalized root-mean-square-error (RMSE) calculated over all N patches in the image,

$$\mathrm{RMSE} = \frac{1}{\sigma}\left[\frac{1}{K}\sum_{k=1}^{N}(y_k - A\widehat{x}_k)^2\right]^{\frac{1}{2}}\,, \tag{10}$$

where σ is the variance of the elements in all the y_k. Note that this is calculated over the image patches, leading to a slightly different calculation than the mean-square error over the entire image.

To measure how much a given transform algorithm compresses an image, we need a coding algorithm that maps which coefficients were used and their amplitudes into an efficient binary code. The design of such encoders is generally a complex undertaking, and is outside the scope of our work here. However, information theory states that we can estimate a lower bound on the coding efficiency if we know the entropy of the input signal. Following

the method of Lewicki and Sejnowski (cf. [6] eq. 13) we estimate the entropy of the coding using histograms of the quantized coefficients. Each coefficient $\widehat{x}_k$ is quantized to 8 bits (or 256 histogram bins). The number of coefficients in each bin is c_i. The limit on the number of bits needed to encode each input vector is,

$$\#\text{bits} \geq \text{bits}_{\text{lim}} \equiv -\sum_{i=1}^{256} \frac{c_i}{N} \log_2 f[i]\,, \tag{11}$$

where $f[i]$ is the estimated probability distribution at each bin. We use $f[i] = c_i/(Nn)$, while in [6] a Laplacian kernel is used to estimate the density. The entropy estimate in bits/pixel is given by,

$$\text{entropy} = \frac{\text{bits}_{\text{lim}}}{m}\,, \tag{12}$$

where m is the size of each image patch (the vector y_k). It is important to note that this estimate of entropy takes into account the extra bits needed to encode an overcomplete ($n > m$) dictionary, i.e. we are considering the bits used to encode each *image pixel*, not each coefficient.

EXPERIMENTS

Previous work has shown that learned complete bases can provide more efficient image coding (fewer bits/pixel at the same error rate) when compared with unadapted bases such as Gabor, Fourier, Haar and Daubechies wavelets [5]. In our earlier work [4] we showed that overcomplete dictionaries A can give more efficient codes than complete bases. Here, our goal is to compare methods for learning overcomplete A (FOCUSS-CNDL and overcomplete ICA), and methods for coding images once A has been learned, including the case when the sources must be non-negative.

Comparison of dictionary learning methods

To provide a comparison between FOCUSS-CNDL and overcomplete ICA [6], both algorithms were used to train a 64×128 dictionary A on a set of 8×8 pixel patches drawn from images of man-made objects. For FOCUSS-CNDL, training of A proceeded as described in [4]. Once A was learned, FOCUSS was used to compare image coding performance, with parameters $p = 0.5$, iterations $= 50$, and the regularization parameter λ_{max} was adjusted over the range $[0.005, 0.5]$ to achieve different levels of compression. A separate test set was composed of 15 images of objects from the COIL database [7].

Figure 1 shows the image coding performance of dictionaries learned using FOCUSS-CNDL (which gave better performance) and overcomplete ICA. FOCUSS was used to code the test images, which may give an advantage to the FOCUSS-CNDL dictionary as it was able to adapt its dictionary to sources generated with FOCUSS (while overcomplete ICA uses a conjugate gradient method to find sources).

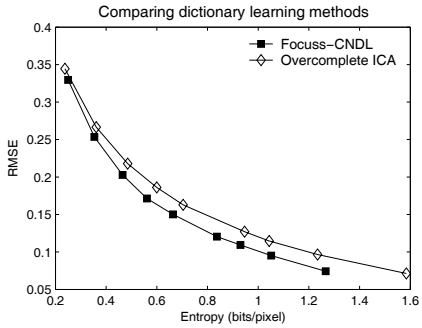


Figure 1: Image coding with 64x128 overcomplete dictionaries learned with FOCUSS-CNDL and overcomplete ICA.

Comparing image coding with MMP, SBL-AVG and FOCUSS

In this experiment we compare the coding performance of the MMP, SBL-AVG and FOCUSS vector selection algorithms using an overcomplete dictionary on a set of man-made images. The dictionary learned with FOCUSS-CNDL from the previous experiment was used, along with the same 15 test images. For FOCUSS, parameters were set as follows: $p = 0.5$, $\lambda_{\max} \in [0.005, 0.5]$. For SBL-AVG, parameters were: iterations = 1000 and the fixed noise parameter σ^2 was varied over $[0.005, 2.0]$. For MMP, the number of vectors selected varied from 1 to 13.

Figure 2b-f shows examples of an image code with the algorithms. FOCUSS was used in Figure 2b for low compression and Figure 2c for high compression. SBL-AVG was similarly used in Figure 2d and 2e. In both cases, SBL-AVG was more accurate and provided higher compression, e.g. MSE of 0.0021 vs. 0.0026 at entropy 0.54 vs 0.78 bits/pixel. In terms of sparsity, Figure 2e requires only 154 nonzero coefficients (of 8192, or about 2%) to represent the image.

Figure 3a shows the tradeoff between accurate reconstruction (low RMSE) and compression (bits/pixel) as approximated by the entropy estimate (12). The lower right of the curves represents the higher accuracy/lower compression regime, and in this range the SBL performs best, with lower RMSE error at the same level of compression. At the most sparse representation (upper left of the curves) where only 1 or 2 dictionary vectors are used to represent each image patch, the MMP algorithm performed best. This is expected in the case of 1 vector per patch, where the MMP finds the optimal single vector to match the input. Coding times per image on a 1.7 GHz AMD processor are: FOCUSS 15.64 sec, SBL-AVG 17.96 sec, MMP 0.21 sec.

Image coding with non-negative sources

Next, we investigate the performance tradeoff associated with using non-negative sources x. Using the same set of images as in the previous section,

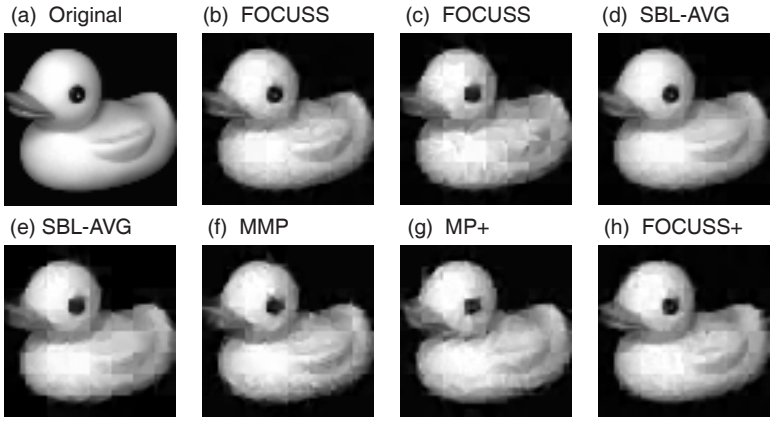
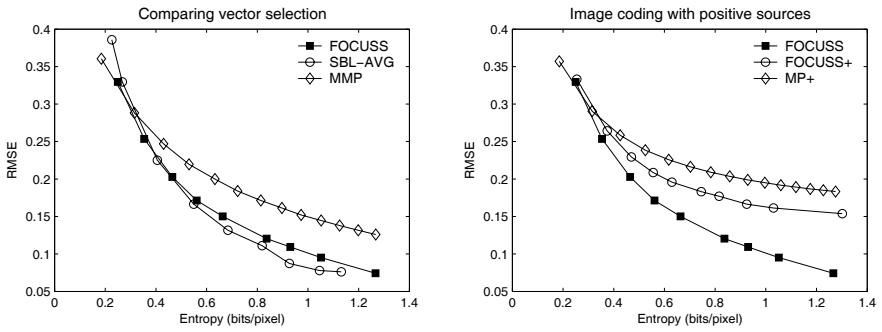


Figure 2: Images coded using an overcomplete dictionary. (a) Original image (b) FOCUSS 0.78 bpp (bits/pixel) (c) FOCUSS 0.56 bpp (d) SBL-AVG 0.68 bpp, 214 nonzero sources (out of 8192) (e) SBL-AVG 0.54 bpp, 154 nonzero sources (f) MMP 0.65 bpp (g) MP+ 0.76 bpp (h) FOCUSS+ 0.77 bpp, 236 nonzero sources. In (b)-(f) the dictionary was learned using FOCUSS-CNDL. In (g)-(h), non-negative codes were generated and the dictionary was learned with FOCUSS-CNDL+.

we learn a new $A \in \mathbb{R}^{64\times 128}$ using the non-negative FOCUSS+ algorithm (6) in the FOCUSS-CNDL dictionary learning algorithm (9). The image gray-scale pixel values are scaled to $y_i \in [0, 1]$ and the sources are also restricted to $x_i \geq 0$ but elements of the dictionary are not further restricted and may be negative. Once the dictionary has been learned, the same set of 15 images as above were coded using FOCUSS+. Figure 2g and 2h show an image coded using MP+ and FOCUSS+. FOCUSS+ is visually superior and provides higher quality reconstruction (MSE 0.0016 vs. 0.0027) at comparable compression rates (0.77 vs. 0.76 bits/pixel). Figure 3b shows the compression/error tradeoff when using non-negative sources to code the same set of test images as above. As expected, there is a reduction in performance when compared with methods that use positive and negative sources especially at lower compression levels.

CONCLUSION

We have discussed methods for learning sparse representations of images using overcomplete dictionaries, and methods for adapting those dictionaries to the problem domain. Images can be represented accurately with a very sparse code, with on the order of 2% of the coefficients being nonzero. When the sources are unrestricted, $x \in \mathbb{R}^n$, the SBL-AVG algorithm provides the best performance, encoding images with fewer bits/pixel at the same error when compared FOCUSS and Matching Pursuit. When the sources are required to be non-negative, $x[i] \geq 0$, the FOCUSS+ and associated dictionary learning algorithm presented here provide the best performance.



(a) (b)

Figure 3: (a) Comparison of sparse image coding. (b) Image coding using non-negative sources x, with the FOCUSS curve from (a) included for reference. Both experiments use a 64x128 overcomplete dictionary.

REFERENCES

[1] S. F. Cotter, J. Adler, B. D. Rao and K. Kreutz-Delgado, "Forward sequential algorithms for best basis selection," **IEE Proc. Vis. Image Sig. Proc.**, vol. 146, no. 5, pp. 235–244, October 1999.

[2] K. Engan, J. H. Husoy and S. O. Aase, "Frame Based Representation and Compression of Still Images," in **Proc. ICIP 2001**, 2001, pp. 1–4.

[3] P. O. Hoyer, "Non-negative sparse coding," in **Proc. of the 12th IEEE Workshop on Neural Networks for Sig. Proc.**, 2002, pp. 557– 565.

[4] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee and T. J. Sejnowski, "Dictionary Learning Algorithms for Sparse Representation," **Neural Computation**, vol. 15, no. 2, pp. 349–396, February 2003.

[5] M. S. Lewicki and B. A. Olshausen, "A Probabilistic Framework for the Adaptation and Comparison of Image Codes," **J. Opt. Soc. Am. A**, vol. 16, no. 7, pp. 1587–1601, July 1999.

[6] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," **Neural Computation**, vol. 12, no. 2, pp. 337–365, February 2000.

[7] S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-100)," Techn. Report CUCS-006-96, **Columbia University**, 1996.

[8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" **Vis. Res.**, vol. 37, pp. 3311–3325, 1997.

[9] M. D. Plumbley, "Algorithms for nonnegative independent component analysis," **IEEE Trans. Neural Net.**, vol. 14, no. 3, pp. 534–543, May 2003.

[10] B. D. Rao and K. Kreutz-Delgado, "An Affine Scaling Methodology for Best Basis Selection," **IEEE Trans. Sig. Proc.**, vol. 47, pp. 187–200, 1999.

[11] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," **Journal of Machine Learning Research**, vol. 1, pp. 211–244, 2001.

[12] D. M. Weber and D. Casasent, "Quadratic Gabor filters for object detection," **IEEE Trans. Image Processing**, vol. 10, no. 2, pp. 218–230, February 2001.

[13] D. P. Wipf and B. D. Rao, "Sparse Bayesian Learning for Basis Selection," **to appear IEEE Trans. Sig. Proc.**, 2004.