Submitted to ICCV'99. Please do not divulgate.

Independent Component Analysis of Textures

Roberto Manduchi

Autonomy and Control Group
Jet Propulsion Laboratory
Pasadena, CA 91030
manduchi@jpl.nasa.gov

Javier Portilla

Instituto de Optica
Consejo Super. de Investig. Cientificas
Serrano 121, Madrid, Spain
iodpm79@pinar2.csic.es

Abstract

A popular method for texture representation is based on the marginal statistical description of filtered channels. We propose a technique, based on Independent Component Analysis theory, to choose the filters in a given space that yield the most informative marginals. We implement our algorithm using a steerable filter space, and present experiments of classification and synthesis. The results show that Independent Component Analysis is superior to Principal Component Analysis for modeling the joint statistical description of filtered channels for both natural and synthetic textures.

1 Introduction

One of the main goals of computer vision is the compact representation of visual entities. Textures, the visual objects considered in this paper, are best represented by their statistical properties. We introduce an information-theoretic framework that provides a better understanding of filter-based approaches to texture analysis and suggests a technique for improving the quality of texture representations based on marginal statistics.

Loosely speaking, textures are characterized by two basic properties: *homogeneity* and *locality of representation*. In other words, the "visual flavour" of a texture can be captured by its statistical behavior within a limited size window. Suitable statistical models for textures are stationary Markov Random Fields (MRF), which are completely characterized by their joint probability density function (pdf) within a suitable neighborhood. Estimating and representing even low-dimensional joint pdf's, however, can be an overwhelming task. The size of the representation grows exponentially with the model dimension, as does the minimum number of samples required to achieve a given estimation accuracy [7]. In fact, techniques that attempt to completely characterize joint distributions either consider very small neighborhoods [21], measure only pairwise dependency [12], or use specific MRF models [9], [5].

A different approach is based on the analysis of feature vectors formed by the output of a filter bank [15],[6],[16],[18]. With a suitable choice of the analysis filters, we can assume that the feature vectors capture the local visually significant texture character. In other words, the filters map the image values in each neighborhood onto a "perceptually relevant" subspace, thereby reducing the representation size while preserving structural information.

Barring a few exceptions [21],[10],[23], typical filter-based algorithms do not estimate the complete statistical description of the texture features. Instead, they build models based on the *marginal statistics* of the feature components, usually represented by the channel variances or their histograms. For example, by analyzing the directional characteristics of a texture along a dense (ideally continuous) set of orientations and scales, one obtains its "scale/orientation signature", which can be used for classification [18], [22].

Texture representation by marginal statistics is a simple and attractive approach. To make efficient use of it, though, we should understand how well a given set of marginals represents the joint statistical description of a texture feature. Zhu et al. pointed out that, in general, one needs the set of *all* the marginal pdf's to completely characterize the joint feature vector pdf. Thus, one approach to improving the quality of representation is to use a large number of filters, perhaps exploiting steerable schemes [11], [20] for computational efficiency.

In this work we follow a different route, and devise a technique for finding the basis of a given filter space which generates the most informative marginals for a given texture. We show by experimental tests of classification and synthesis that simply by selecting such

"optimal" filter basis the quality of the representation increases significantly.

Our basis selection algorithm is based on Independent Component Analysis (ICA) theory. ICA has been an area of intense research during the last decade, stimulated by problems of blind source separation and deconvolution. ICA can be regarded as a technique for statistical modeling that proves superior to Principal Component Analysis in the case of non-gaussian random vectors. It is an established notion that natural images do not behave like gaussian processes [17] [3]; our experiments prove that ICA provides better models for natural and synthetic textures.

This paper is organized as follows. Section 2 describes the filter spaces that we use for texture analysis, and introduces the filter basis selection problem. Section 3 provides some necessary background of ICA theory, and shows its application to texture modeling. Section 4 describes experiments of texture classification and synthesis, which allow us to assess the performance of ICA modeling. Section 5 has the conclusions.

2 Steerable spaces for texture analysis

Given an image $l(x)$ and a set of N filters $\{h_i(x)\}$, we will say that the vector $f(x) = [f_1(x) = l * h_1(x), \ldots, f_N(x) = l * h_N(x)]$ is the *feature representation* of the image at point x. We will assume that a texture is completely represented by the joint pdf of its feature vector $f(x)$ (which, by stationariety, is independent of x.) However, in this paper we will deal with reduced representations formed by the marginal statistics of the feature vectors. Let us recall that the marginal density of the i-th component of a random vector z, $p_i(z_i)$, is the projection of the joint pdf $p(z)$ onto the i-th axis. The set of marginal statistics is univocally represented by the outer product of the marginal pdf's: $\bar{p}(z) = \prod_i p_i(z_i)$, which is equal to $p(z)$ if and only if the feature vector has independent component.

Which filters produce meaningful feature representations of textures? Several design criteria have been proposed. Basically, they all share the following characteristics: 1) zero-DC response, to enforce invariance to slow-varying illumination bias; 2) good spatial/spectral concentration, to ensure energy separation while preserving locality of description.

Another useful property of analysis filter banks is *steerability* [11], defined as follows. Let H be the linear space spanned by the N kernels $\{h_i(x)\}$, and let $\mathcal{G}$ be a suitable group of domain transformations (e.g., rotations or isotropic scaling.) We will say that H is steerable over $\mathcal{G}$ if for every kernel $h(x)$ in H, and for every transformation $A(\cdot)$ in $\mathcal{G}$, the "transformed" kernel $h_A(x) = h(A(x))$ belongs to H. For example, assume the filter space is steerable over the group of rotations. Let $f(x)$ and $f^R(x)$ be the feature vectors of a texture, and of the rotated version of the texture respectively (R is the rotation matrix). Then, $f(x)$ and $f^R(R^{-1}x)$ are related by a linear transformation (a matrix), which is a function of the rotation angle only. Steerability, thus, ensures uniformity of representation for transformed versions of the same texture. Methods exist to design exact and approximate finite dimensional steerable filter spaces over rotations and isotropic scaling in the scale range of interest [20].

Joint spatial/spectral concentration is achieved, for example, using Gabor kernels or directional derivatives of gaussian kernels [16],[18]. Such filters effectively capture directional texture attributes. If a N-dimensional steerable analysis space is used, the texture signatures are completely characterized by the joint statistics of the output of N basis filters. Note that one can always find a set of basis filters which are scaled/rotated versions of any prototype kernel in the steerable space.

In this work we are concerned with the selection of the "optimal" basis in a steerable filter space. Note that the choice of the basis is irrelevant if the joint pdf of the feature vector is considered. Indeed, if f_1 and f_2 are the feature vectors at point x, computed using two different filter basis, then $f_2 = Af_1$, where A is a full-rank matrix. Thus, the joint pdf's of the feature vectors, $p_1(z)$ and $p_2(z)$, are related to one another as $p_2(z) = p_1(A^{-1}z)/|\det(A)|$. This one-to-one relation, however, holds for joint pdf's only. In general, the marginal pdf's of f_1 *cannot* be computed from the marginal pdf's of f_2. This somewhat paradoxical observation stems from the fact that marginal pdf's are projections of a joint pdf along different directions. A finite set of marginals does not carry enough information to allow reconstructing all the other projections.

Assume, however, that a particular choice of basis filters gives statistically independent feature components. Then, as noted above, the marginals completely characterize the joint pdf of the feature vector: from this set of marginals we can reconstruct all the other marginals. The converse is not true: other filter basis will produce, in general, features with "less informative" marginal statistics.

Unfortunately, only very few textures can be represented by independent component features [3]. Still, it is almost always the case that some filter bases produce "more informative" marginals than others. These intuitive notion is given a rigorous formal char-

acterization by the Independent Component Analysis theory, which also provides us with algorithms for selecting "optimal" filter basis.

3 Independent Component Analysis

In this section we report some results of Independent Component Analysis theory that are instrumental to the development of the texture classification and synthesis algorithms discussed in later sections. We refer the reader to the excellent tutorial of Cardoso [4] for a general overview of the theory.

We first introduce the ICA problem as a statistical approximation tool, and then briefly outline Comon's ICA algorithm, which we have adopted in our experiments. We also present a simple case study that the shows the effectiveness of ICA modeling for textures analysis in a steerable space.

3.1 Problem formulation

The theory of Independent Component Analysis is traditionally associated with the Blind Source Separation (BBS) problem. In its simplest formulation, the BBS problem assumes that N independent causes (random variables) have been linearly combined by a full rank matrix A to produce N observed variables. The goal is to identify the mixing matrix from the observations, possibly using prior information about the statistics of the causes if available.

ICA, however, may also be regarded as a general estimation method [8], without explicit reference to the BSS model. If z is a random vector with joint pdf $p(z)$ and full-rank covariance matrix[1], ICA seeks a full rank matrix A such that the pdf of the vector $y = Az$ is "best represented" by the outer product of its marginal pdf's. The quality of the separable approximation is measured by the *contrast*, which is a deterministic function of a joint pdf. Let $e(z)$ be the "model error", defined as the difference between the logarithm of the density $p(z)$ and the log-likelihood of the separable model:

$$e(z) = \log p(z) - \log \prod_i p_i(z_i) \qquad (1)$$

where $p_i(z_i)$ are the marginal pdf's of the components of z. We define *contrast* of $p(z)$ [8] the expectation of the model error $e(z)$, which is equal to the Kullback-Leibler (KL) divergence between $p(z)$ and the separable model pdf:

$$E\left[e(z)\right] = \int_{-\infty}^{\infty} p(z) \log \frac{p(z)}{\prod_i p_i(z_i)} \, \mathrm{d}z \qquad (2)$$

$$= \sum_i H_i(z_i) - H(z)$$

where $H(\cdot)$ represents the differential entropy of its argument: $H(z) = -\int_{-\infty}^{\infty} p(z) \log p(z)$. This contrast takes also the name of *mutual information* of z, and being a KL divergence it is always positive (it vanishes if and only if z has independent components). Intuitively, the mutual information tells us how much we loose in terms of average information if we neglect to consider the statistical dependence among the components of z.

Other contrast functions have been proposed for ICA. For example, if the distributions $g_i(z_i) = \int_{-\infty}^{z_i} p_i(\hat{z}) \mathrm{d}\hat{z}$ of the original independent components in a BSS model are known, we may define the "infomax" contrast [1] of the vector z as $-H(g(z))$, where $g(z) = [g_1(z_1), \ldots, g_N(z_N)]$. However, in the case of texture descriptors we don't know the distributions $g_i(z_i)$, and mutual information is a more appropriate criterion. Thus, we define the ICA problem as follows:

> **ICA problem:** find a full-rank matrix A such that $y = Az$ has minimal mutual information.

If z is jointly gaussian, its mutual information is null if and only if its covariance matrix is diagonal. Principal Component Analysis (PCA, also known as Karhunen-Loève transform) is a technique to diagonalize a vector covariance by an orthonormal matrix, and therefore solves the ICA problem for gaussian vectors. In the general non-gaussian case, however, diagonal covariance is not sufficient to ensure minimal mutual information, and more work is needed.

It can be shown [8] that, if z has been "prewhitened" to unit covariance (by PCA followed by axis rescaling), the ICA problem is solved by an orthonormal linear transformation that minimizes the sum of the marginal entropies of the transformed vector. This fact has a nice counterpart in the context of BSS. Since the gaussian distribution has the highest entropy for a given variance, we may argue that the demixing matrix is the one that produces marginals as "far from gaussian" as possible. This observation agrees with intuition: indeed, as suggested by the central limit theorem, linear mixing of independent causes produces gaussian-like distributions.

3.2 Comon's ICA algorithm

An efficient ICA algorithm, proposed by Comon [8], is based on higher-order statistics. In fact, this technique minimizes a different contrast than mutual information; the relation between the two contrast functions can be highlighted using Edgeworth functional

[1] If the covariance matrix V is not full rank, we may consider the projection of z onto the range space of V [8].

expansions [14]. An intuitive justification of higher-order methods is based on the fact that all the cross-cumulants of an independent component vector are null [14]. Thus, one may try to minimize mutual information by actually minimizing the n-th order dependence among the vector components. For example, Comon's technique finds an orthonormal matrix A that minimizes the sum of the squared fourth-order cross-cumulants of $y = Az$, provided that z has been pre-whitened. It can be shown that this is equivalent to maximizing the sum of the squared marginal kurtosis (fourth-order cumulants) of y.

It is a well known fact that gaussian distributions have zero kurtosis [14]: Comon's algorithm does push the vector components away from gaussianity. Leptokurtic, heavy tailed exponential-like distributions are characteristic of "sparse codes" [19], and are often used to model the output of wavelet filter banks for image coding [17]. Platykurtic distributions are also often observed in texture analysis.

Comon's design algorithm is based on Jacobi iterations. Its complexity is $\mathcal{O}(N^4)$ (including pre-whitening,) and it always converges to a (possibly local) minimum of the contrast. Due to these desirable properties, we have selected Comon's technique for our experiments.

3.3 A case study

In this section we present a simple example that shows the effectiveness of ICA-based modeling for texture analysis. The two textures of Figure 1 (a),(b) are analyzed in the steerable space spanned by the two $(0° - 90°)$ filters of Figure 1 (c),(d) (upper row). Texture A is a mosaic of patches filled at random with either a constant signal, horizontal or vertical sinusoidal gratings, or the sum thereof. Texture B is a mosaic of patches filled with either horizontal or vertical sinusoidal gratings. The $0°$ $(90°)$ filter removes the horizontal (vertical) grating component completely.

Our goal is to find two filters in the steerable space that best describe each texture by means of the marginal statistics of their outputs. Figure 1 (c),(d) (lower row) show the filtered versions of the two textures using the $0° - 90°$ filters. Figure 1 (e),(f) show the joint pdf's of the filter outputs, measured by a Parzen window estimator. In both cases the two channels are uncorrelated with identical variance (as proved by the axial symmetries of their pdf's.) However, only the channels of Texture A are statistically independent. Thus, the outer product of their marginal densities reproduces the joint pdf of Texture A exactly, as shown in Figure 1 (g). In the case of Texture B, though, the separable model yields a very poor approximation of the joint pdf (compare Figure 1 (h) and (f)). As a matter of fact, the marginal pdf's of the $0° - 90°$ channels of Texture B and Texture A are identical: based on such marginal descriptions, the two textures look the same! Note also that PCA cannot help us finding better bases, since the channels of both textures are already uncorrelated.

If we run the ICA algorithm separately on each texture (by analyzing the output of the two filters) we find that the $0° - 90°$ kernels are indeed the correct choice for Texture A, while the $45° - 135°$ kernels of Figure 1(i) (upper row), obtained by multiplying the outputs of the $0° - 90°$ filters by a $45°$ rotation matrix, are more appropriate for Texture B. Figure 1 (j) shows the outer product of the corresponding marginal pdf's, after rotation back to the original $0° - 90°$ space. By comparing Figure 1 (f), (h) and (j) it is clear that ICA has dramatically increased the quality of the separable approximation .

4 Experimental tests

4.1 Supervised classification

In this section we use ICA pdf modeling in a test of supervised texture discrimination. For the sake of simplicity, we perform training and classification on the same data set, formed by the $K = 8$ textures of Figure 2 (chosen from the MIT VisTex database). Each texture is modeled by the pdf's of the selected channels in a fixed steerable space, together with the corresponding ICA matrix. The marginal densities are estimated and represented by 50-bins histograms.

An image formed by the mosaic of all textures is classified pixel-wise using a Bayesian technique [2]. A simple heuristic iterative procedure, inspired by the "Perceptually Organized Expectation Maximization" algorithm of Weiss and Adelson [25], estimates posterior class probabilities from the likelihoods, enforcing local spatial coherence. The experiment proceeds as follows:

- **Training:** for each texture model,
  1. filter the training image with a fixed filter bank;
  2. compute the ICA matrix from the output of the filters;
  3. multiply the output vectors by the ICA matrix and compute the channel histograms.
- **Classification:** apply the fixed filter bank to the test image. For each texture model,
  1. multiply the filter output vectors by the model ICA matrix, and use the corresponding channel histograms to compute the marginal likelihoods;

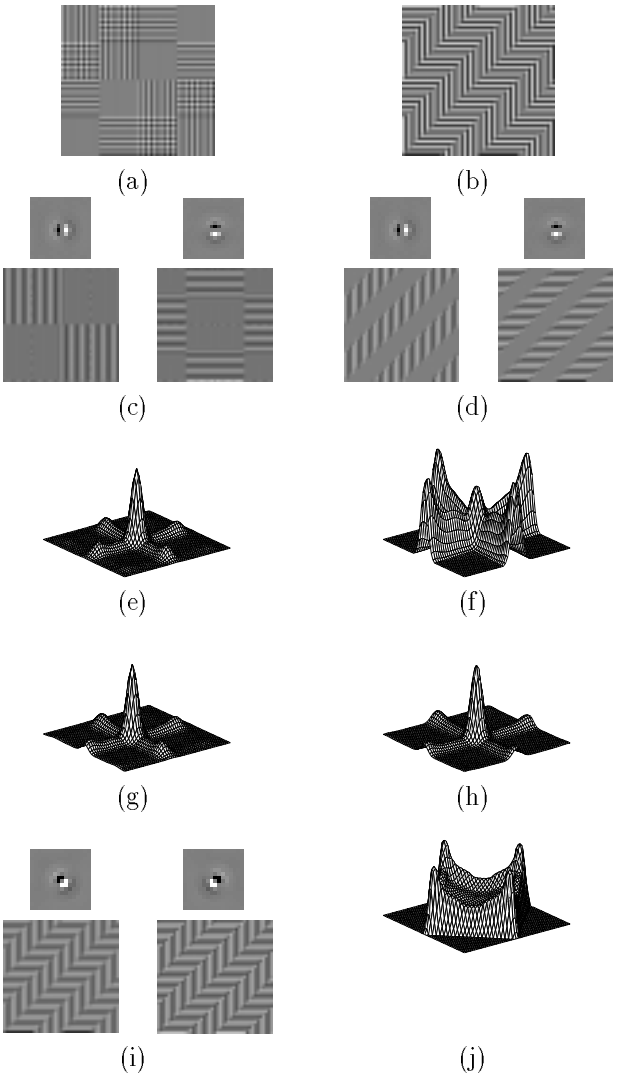

Figure 1: A case study (see text.) (a),(b): Textures A and B. (c),(d): The $0° - 90°$ kernels (upper row), and the filtered versions of the two textures (lower row.) (e),(f): The joint channel pdf's of the two textures. (g),(h): The separable approximations of the joint pdf's using the $0° - 90°$ filters. (i): The $45° - 135°$ kernels (upper row), and the filtered version of texture B (lower row.) (j) The separable approximation of the joint pdf of texture B using the $45° - 135°$ filters.

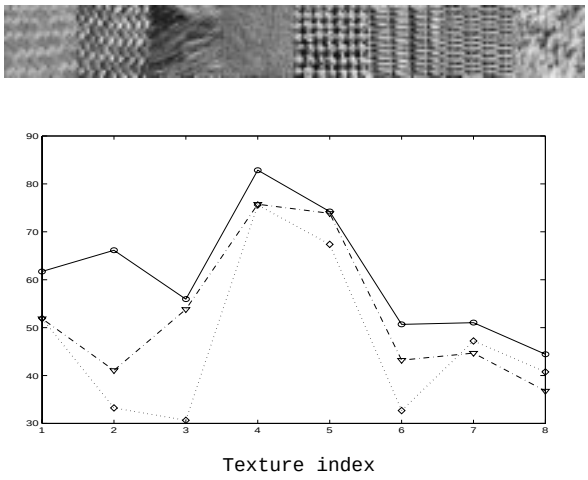

Figure 2: Supervised classification test (see text). The system is trained and tested on the eight textures shown above. The plots represent the percentage of correct classification for each texture. Solid line: using ICA transformation. Dot-dashed line: using PCA transformation. Dotted line: without transformation.

2. the likelihood of each pixel given the model is equal to the product of the marginal likelihoods, times the absolute value of the determinant of the ICA matrix.

Compute the posterior class probabilities and perform pixel-wise Bayesian classification.

For our experiment we have chosen a steerable pyramid filter bank [24] with just two orientations and three scales ($N = 6$ filters overall). For each training texture (size 256×256 pixels), Comon's Matlab implementation of ICA required 27 seconds of computation time on a Power Macintosh G3 266 MHz. The results, in terms of percentage of correctly classified pixels for each texture, are shown by the solid line plot of figure 2. If PCA is used instead of ICA, we obtain the correct classification rates shown by dot-dashed line; the classification performances of the system without transformation (i.e., using the original filters) are shown by the dotted line. ICA pdf modeling consistently yields the highest correct classification rates.

We have observed that the performance gain provided by ICA and PCA modeling decreases if the number of orientations in the steerable filter bank is increased. This should not come as a surprise: many marginal statistics can provide sufficient information even if they are not statistically independent, at the price of increased representation size.

4.2 Texture synthesis

Synthesis-by-analysis algorithms [13],[26],[23] create new textures which "look like" a given prototype one. This is equivalent to sampling a random process whose statistical description has been estimated from a given sample.

A simple and effective synthesis technique has been proposed by Heeger and Bergen [13]. First, the prototype texture is analyzed by a filter bank which includes the "identity" filter, and the marginal channel histograms are recorded. Then, a random image is generated, and its channel histograms (computed by the same filter bank) are iteratively adjusted to match those of the prototype texture. The algorithm usually converges to a texture that has the same channel histograms as the prototype: the two textures are therefore identical on the ground of marginal statistical description.

We have adopted such an algorithm as a testbed for our experiments because it represents the ideal "Turing test" to validate texture models based on marginal channel statistics. Our only addition to Heeger and Bergen's model is the multiplication of the channel vectors at the output of the filter bank by the ICA matrix computed for the prototype texture, as in the scheme of section 4.1. After the channel histograms have been adjusted to match those of the prototype texture, the channel vectors are multiplied by the inverse of the ICA matrix. Note that the presence of the ICA matrix requires that all the channels be kept to the same sampling rate.

Heeger and Bergen use steerable pyramid analysis filter banks [24], and synthesize samples that successfully reproduce the fine grain characteristics of the prototype texture. Such simple marginal statistics modeling, though, fails to capture more complex spatial structures, such as elongated patterns. A simple way to boost the performance of Heeger and Bergen's scheme is to enrich the filter space with the *shifted* versions of the filters. In other words, for each filter $h(x)$ we add the filters $h^n(x) = h(x+n)$ with n belonging to a suitable neighborhood of the origin. The feature representation thus "looks around" over a larger neighborhood of each point. This new information comes at no cost: the output of filter $h^n(x)$ at point x_0 is equal to the output of filter $h(n)$ at point $x_0 + n$. Shifted filters, though, are useless if the marginals statistical description is built directly from their output: by stationariety, the marginal pdf of a filter output does not change if the filter is shifted. The contribution of the shifted filters becomes relevant if a different basis of the steerable space is chosen, that is, if the output of the filters is multiplied by the ICA matrix.

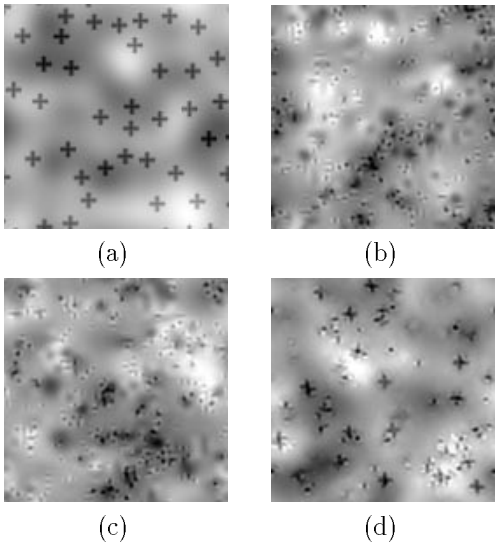

Figure 3: Examples of texture synthesis. The prototype texture "Crosses on clouds" (a) is synthesized using Heeger and Bergen's technique with no channel transformation (b), with PCA transformation (c) and with ICA transformation (d).

We show in figures 3 and 4 examples of synthesis from the prototype textures "Crosses on clouds" and "Squares". For the "Crosses on clouds" texture we have used a steerable pyramid filter bank which comprised both odd-symmetric and even-symmetric kernels at two orientations and four scales (overall $N = 16$ filters). For the "Squares" texture we have used odd-symmetric kernels at two orientations and three scales, shifted over five different positions (overall $N = 30$ filters). These prototype textures are highly non-gaussian, which makes PCA modeling inadequate (see figure 3(c) and 4(c)). ICA modeling proves superior in both cases. In particular, for the "Crosses on clouds" texture, ICA does a good job at separating the low-pass, isotropic components (the "clouds") from the line-shaped, oriented ones (the "crosses".) In the case of the "Squares" texture, ICA is able to extract the local structure by correctly combining the shifted filters outputs.

5 Conclusions

We have presented an algorithm that chooses the basis filters in a steerable space yielding the most informative marginals for texture representation. This method is based on the minimization of the mutual channel information via Independent Component

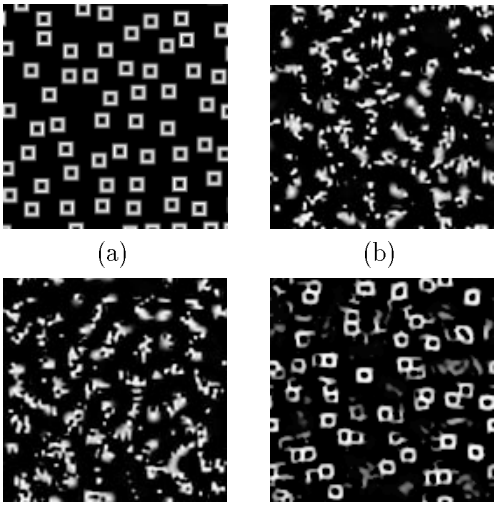

Figure 4: Examples of texture synthesis for the prototype texture "Squares" (see caption of figure (3)).

Analysis. The experimental results show the superiority of ICA modeling with respect to PCA for natural and synthetic textures and small dimensional filter spaces.

An interesting open problem, which we are currently investigating, is subspace selection based on information–theoretic criteria. We expect that by choosing a small number of highly informative channels, one may obtain reduced size representations that maintain good discrimination properties.

References

[1] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129-1159, 1995.

[2] C.M. Bishop. *Neural networks for pattern recognition.* Clarendon Press, Oxford, U.K., 1995.

[3] R.W. Buccigrossi and E. Simoncelli. Progressive wavelet image coding based on a conditional probability model. *Proc. IEEE ICASSP*, 4:2957–2960, Munich, 1997.

[4] J.-F. Cardoso. Blind signal separation: Statistical principles *Proceedings of the IEEE*, 86(10:2009–2025,October 1998.

[5] R. Chellappa and S. Chatterjee. Classification of textures using Gaussian Markov Random Fields models. *IEEE Trans. Acoust., Speech, Signal Proc.*, 33(4):959–963, August 1985.

[6] J.M. Coggins and A.K. Jain. A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, 3:195–203, May 1985.

[7] P. Comon. Supervised classification: A probabilistic approach. *Proc. ESANN*, 111–128, Verleysen, April 1995.

[8] P. Comon. Independent Component Analysis: A new concept? *Signal Processing*, 36(3):287–314, April 1994.

[9] G.C. Cross and A.K. Jain. Markov Random Field texture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5:25–39,1983.

[10] J.S. De Bonet and P. Viola. Texture recognition using a non-parametric multi-scale statistical model. *Proc. IEEE CVPR*, 641–647, Santa Barbara, June 1998.

[11] W. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891-906, 1991.

[12] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804. May 1979.

[13] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. *Proc. ACM SIGGRAPH*, August 1995.

[14] M. Kendall and A. Stuart. *The advanced theory of statistics.* MacMillan Publishing, New York, 1977.

[15] K.I. Laws. Texture energy measures. *Proc. Image Underst. Workshop*, 47–51, November 1979.

[16] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journ. Optical Society of America*, 7(5):923–932, May 1990.

[17] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:674–693, July 1989.

[18] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, August 1996.

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

[20] P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):488-499, May 1995.

[21] K. Popat and R. Picard. Cluster-based probability model and its application to image and texture processing. *IEEE Trans. Image Proc.*, 6(2):268–284, February 1997.

[22] Y. Rubner and C. Tomasi. A metric for distributions with applications to image databases. *Proc. 6th ICCV*, 59–66, Bombay, 1998.

[23] E.P. Simoncelli and J. Portilla. Texture characterization via second-order statistics of wavelet coefficient amplitudes. *Proc. IEEE ICIP'98*, 1998.

[24] E. Simoncelli and W.T. Freeeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. *Proc, of 2nd IEEE ICIP*, 444-447, Washington DC, October 1995.

[25] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. *Proc. IEEE CVPR'96*, 321–326, San Francisco, 1996.

[26] S.C. Zhu, Y. Wu and D. Mumford. Filters, random fields and maximum entropy (FRAME.) *Int. Journal of Computer Vision*, 27(2):1–20, March/April 1998.