Machine Vision and Applications (1997) 9: 262–271

Machine Vision and
Applications

A multidimensional scaling approach to explore the behavior of a texture perception algorithm

Maria Vanrell[⋆], **Jordi Vitrià, Xavier Roca**

Computer Vision Center, Universitat Autònoma de Barcelona, E-08193 Bellaterra, Barcelona, Spain

Abstract. This paper presents a methodology for behavior characterization of an algorithm in terms of the parametric description of input images. To develop the work we have selected an algorithm which implements a model of texture perception and provides a texture representation. The approach is based on the definition of an input parametric texture space, where parameters are related to texton attributes. Multidimensional scaling provides a dimensional reduction of space of representation. It allows interpretation of the behavior of the algorithm in a low-dimensional space where points represent textures and distances represent dissimilarities between textures, preserving the metric of the algorithm representation in a monotonic sense. The resulting behavior space establishes the basis to construct a quantitative causal model of an algorithm.

Key words: Multidimensional scaling – Algorithm behavior – Texture perception – Texture representation

1 Introduction

Recent work in artificial intelligence [3] focuses on empirical methods for characterization of the features of programs, environments, tasks and behaviors. The study of specific programs using empirical methods can help us find their general features. An empirical generalization strategy has been defined in five steps: (1) build a program that exhibits an interesting behavior; (2) identify specific features of the program, the tasks and the environments that influence the target behavior; (3) develop and test a causal model on how these features influence the target behavior; (4) generalize the features so that other programs, tasks and environments are encompassed by the causal model; (5) test whether the general model predicts accurately the behavior of this larger set of programs, tasks and environments.

In this paper we focus on the second step of this strategy, on a classical problem of computer vision. We have chosen an algorithm that implements a general computational model of preattentive texture segmentation [12]. The paper has been organized in the following steps.

Firstly we introduce the texture perception problem and the algorithm we have studied in this work. Secondly we give a brief introduction to the multidimensional scaling method. Then we propose a methodology to construct a behavior space in terms of the input parameters. Finally we test the results on some natural images.

2 Texture

Texture is an important visual cue due to the repetition of image patterns. It is widely used in several applications, such as classification of materials, scene segmentation and attentional mechanisms.

Much work has focused on the texture perception problem. Interesting reviews have been presented [5, 7, 21, 27]. Psychophysical experiments and neurobiological evidence have provided the basis for the definition of texture perception models [1, 6, 9, 14]. These results have led to computational multichannel approaches that are based on the responses of linear mechanisms. This type of approach implies a large amount of image data distributed in a high number of channels. This means it is difficult to understand how the model will behave on specific images. In this work we experiment on this type of approach to develop a methodology to explore the behavior of algorithms.

2.1 An improved algorithm

The work presented by Malik and Perona [12] has been considered one of the most important computational models of human preattentive texture perception. In order to explore the model's behavior, we firstly propose an efficient algorithm to compute it. We have divided the model into four stages.

First stage. An image-filtering step with a set of linear filters. F_i represents the filter of the i channel, whose response is given by

$$R_i(x, y) = F_i(x, y) * I(x, y) \qquad (1)$$

[⋆] e-mail: maria@cvc.uab.es

Correspondence to: M. Vanrell

The selected filters are the DOOG family, which are formed by differences of offset gaussian functions as:

$$G(x_0, y_0, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-((\frac{x-x_0}{\sigma_x})^2+(\frac{y-y_0}{\sigma_y})^2)} \tag{2}$$

The expressions for the filters are given in Table 1. DOG1 and DOG2 are radially symmetric filters modelling nonoriented simple cells, and DOOG2 is a directional even-symmetric filter to model bar-sensitive simple cells. The filter depends on a size parameter σ. Taking 12 possible sizes for each filter and considering six different orientations on filters DOOG2, 96 responses are obtained.

In this first step the filtering is followed by a half-wave rectification that duplicates the number of responses

$$\begin{aligned} R^{+}(x,y) &= max\{R(x,y),0\} \\ R^{-}(x,y) &= max\{-R(x,y),0\} \end{aligned} \tag{3}$$

Thus, the result of this first stage is given by the following set of 192 responses:

$$\begin{aligned} R_{2k} &= (I * F_k)^{+}(x,y) \\ R_{2k+1} &= (I * F_k)^{-}(x,y) \quad \forall k \in \{0,...,95\} \end{aligned} \tag{4}$$

Second stage. A non-linear inhibition in two steps. We have ameliorated these two steps by introducing fast morphological operations [15] to compute them. Local maximum operations have been implemented by a morphological dilation with an isotropic structuring element [23, 24].

The first inhibition removes spurious responses. That is, responses in non-optimally tuned channels are suppressed. This is done by subtracting the following function from each channel i

$$T_i(x_0, y_0) = \max_j\{(\alpha_{ji}R_j(x,y)) \oplus B_{I_{ji}}\} \tag{5}$$

Values for I_{ji} and α_{ji} depend on the inhibition that is selected. In our experiments we have used model A (see Tables 1 and 2 in [12]). I_{ji} represents the radius of the isotropic structuring element B.

The second inhibition expands strong responses to their own neighborhood. The postinhibition response is given by

$$PIR_i(x_0, y_0) = (\frac{1}{1-\alpha_{ii}}[R_i(x,y) - T_i(x,y)]^{+}) \oplus B_{S_i} \tag{6}$$

where $k = 1/(1-\alpha_{ii})$ also depends on model A and S_i is the radius of the neighborhood.

Third stage. A gradient detection on each channel is computed

$$Grad_i(x,y) = \nabla(PIR_i * G_{\sigma'})(x,y) \tag{7}$$

Fourth stage. A combination stage where the distributed representation in channels is lost and all gradients are joined in a unique response. In this model this step is computed by a maximum operation over all gradient of the previous step, obtaining the textural gradient of the image.

3 Multidimensional scaling

Multidimensional scaling (MDS) methods have been developed to analyze and visualize data in such a way that configurations of points in a space of dimension m can be seen in a space of dimension k ($k < m$), preserving the interpoint distances in a monotonic sense [8, 13, 28]. Although, we will work on euclidean distances we have based our approach on nonmetric MDS, since dissimilarity between representations could be defined with any other measure.

Starting with an $n \times n$ matrix, $\Delta = (\delta_{ij})$, where δ_{ij} represents a dissimilarity measure between points p_i and p_j in a m-dimensional space, the objective is to find a configuration of $x_1, ..., x_n$ points in a k-dimensional space having d_{ij} as interpoint distances between x_i and x_j. The values of d_{ij} must be monotonically related to δ_{ij}, that is

$$\delta_{ij} \le \delta_{qp} \rightarrow d_{ij} < d_{qp} \;\; \forall\, (i,j) \in \{1,..,n\} \times \{1,..,n\} \tag{8}$$

Given an initial arbitrary configuration of points in $\mathbb{R}^k$ with interpoint distances $\hat{d}_{ij}$, the problem consists of applying an iterative method that minimizes the following expression:

$$S = \sqrt{\frac{\sum_{i,j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i,j}\hat{d}_{ij}^2}} \tag{9}$$

The minimum is taken over values of d_{ij}'s that verify the monotonic relation with δ_{ij}. Thus, S gives information about how good the computed configuration is; it is called "stress" and was first defined by Kruskal [10]. In this sense, we will consider the minimum stress value as a measure of badness-of-fit, hence, $S = 0$ represents an exact fit on the rank order of the recovered configuration. Finding the most adequate value for k has been defined as the dimensionality problem. There is an interesting approach to this problem in [11].

First approaches to the construction of configurations based on dissimilarities were presented by Shepard [16, 17]. In [18], he proposes to use MDS as a way to discover the structure of what he calls the psychological space. It is formed by the consequential regions which behaves as natural kinds or classes. In this way a new object is easily recognized or classified by being introduced in this structured psychological space. The psychological space is constructed from a set of dissimilarity data obtained from psychophysical experiments. Our work has been motivated by this type of approach since we are trying to explore the underlying structure of high-dimensional representation spaces provided by certain algorithms.

4 Defining the behavior space

In a previous section we introduced a texture perception algorithm from which we extract a high-dimensional texture representation

$$r(t) = (v_1, ..., v_{192}) \tag{10}$$

where

$$v_i = \sum_{(x,y)\in\{0,...,N\}^2} \frac{PIR_i(x,y)}{N^2} \tag{11}$$

Table 1. Expressions corresponding to the filters $DOG1$, $DOG2$ and $DOOG2$.

Filter			σ	Factor	Offset
$DOG1(\sigma)$	=	$aG(0,0,\sigma_i,\sigma_i)$	$\sigma_i : \sigma : \sigma_0$	$a : b$	
	+	$bG(0,0,\sigma_0,\sigma_0)$	0.71 : 1 : 1.14	1 : −1	
$DOG2(\sigma)$	=	$aG(0,0,\sigma_i,\sigma_i)$	$\sigma_i : \sigma : \sigma_0$	$a : b : c$	
	+	$bG(0,0,\sigma,\sigma)$	0.62 : 1 : 1.6	−1 : 2 : −1	
	+	$cG(0,0,\sigma_0,\sigma_0)$			
$DOOG2(\sigma,r)$	=	$aG(0,y_a,\sigma_x,\sigma_y)$	$\sigma_y = \sigma$	$a : b : c$	$y_a = -y_c = \sigma$
	+	$bG(0,y_b,\sigma_x,\sigma_y)$	$\sigma_x = r \cdot \sigma$	−1 : 2 : −1	$y_b = 0$
	+	$cG(0,y_c,\sigma_x,\sigma_y)$	$(r = 3)$		
				$a_{DOG1} : a_{DOG2} : a_{DOOG2}$ 3 : 4.15 : 2	

where t is an image of size $N \times N$ presenting one texture. Dimension of vector r depends on the number of channels in the algorithm[1].

The defined representation permits consideration of a texture as a point in a representation space of 192 dimensions, in which discriminable textures must be separated by a distance in the space. From this point of view, the study of the discriminability of the algorithm can be seen as the study of the spatial proximities between textures in this representation space. With this purpose, we define a dissimilarity measure, δ, such that

$$\delta_{ij} = \sqrt{\sum_{k=0}^{192} (r(t_i)_k - r(t_j)_k)^2} \tag{12}$$

where t_i and t_j are two different texture images and $r(t_i)$, $r(t_j)$ are their corresponding points in the representation space and have been computed by the algorithm.

As we have argued before, the purpose of this work is to identify specific features that influence the algorithm behavior and how they do it. In order to do this, we will define a parametric texture space. It will allow us to work on sets of images whose features can be isolated and controlled, in order to be able to prepare valid experiments to interpret the program behavior.

We have assumed that a texture image is a repetition of blobs and bars with similar properties. Actually, this assumption has also been made in the work of Vorhees and Poggio [25], where bars and blobs are first detected and then their attributes are measured. Contrast, size and orientation are the most frequently used attributes. In fact, we are treating blobs and bars as textons [9].

Hence, to carry out the study we will work on synthetic images formed by aligned blobs or bars[2]. We have introduced a constant density on all images that agrees with the inhibition radius of the program. This does not affect the program behavior since it does not depend on texton location but on textons density.

The foregoing considerations permit definition of a texture as a point in a parametric space of four dimensions. Each axis is associated to a texton attribute as size, length, orientation and contrast, where

$$t = i(p_1, p_2, p_3, p_4) = i(s, l, \theta, c) \tag{13}$$

represents an image texture, which is formed by the repetition of one type of texton, and whose attributes are given by s, l, θ and c. These attributes represent the position of image t in the parametric space[3] (Examples of images in this parameteric space can be seen in Figs. 5a, 6a and 7a).

Variation on a given axis represents a variation of the corresponding attribute on the image texton. As we have stated before, the images that we are generating present a constant density of textons and a fixed background gray level.

Heretofore, we have defined a parametric texture space which will provide interesting sets of images. The algorithm permits representation of the input image in a representation space. To explore this representation space we will carry out a set of experiments according to the following steps:

1. Select a set of n images in the texture parametric space. Selection must be made to characterize any given aspect of the representation space.
2. Apply the algorithm to obtain the representation, r, of each image.
3. Calculate the set of all interpoint dissimilarities between texture representations and construct the $n \times n$ matrix of dissimilarities.
4. Select the k dimension for the recovered space.
5. Apply a nonmetric MDS algorithm to obtain the configuration of the n points in the k dimensional space, we will show it by a k-dimensional plot. This process provides a stress measure as the badness-of-fit of the new configuration.

This MDS-based approach will allow construction of the behavior space of the algorithm, which preserves the same underlying structure of the representation space. In Fig. 1 we can see a scheme of this methodology.

4.1 Variations on one parameter

We will first apply this approach to explore, separately, how the program behaves with variations on each parameter at the input space. Variations on size and length axis on the parametric texture space have been recovered by monotonic variations on a one-dimensional space (badness-of-fit=0.001,

[1] The average of the inhibition response has been computed to avoid the effects of noisy spurious responses.

[2] Generated from the expression of a two-dimensional Gaussian function, size of blobs and bars is given by the standard deviations of the function.

[3] In our experiments parameters have been taken from the following space: $[1, 10] \times [1, 10] \times [0^\circ, 180^\circ] \times [-128, 128]$. In fact we are taking a subspace of it, since we only consider points accomplishing $s \leq l$, otherwise we could have two different points representing the same image texture.

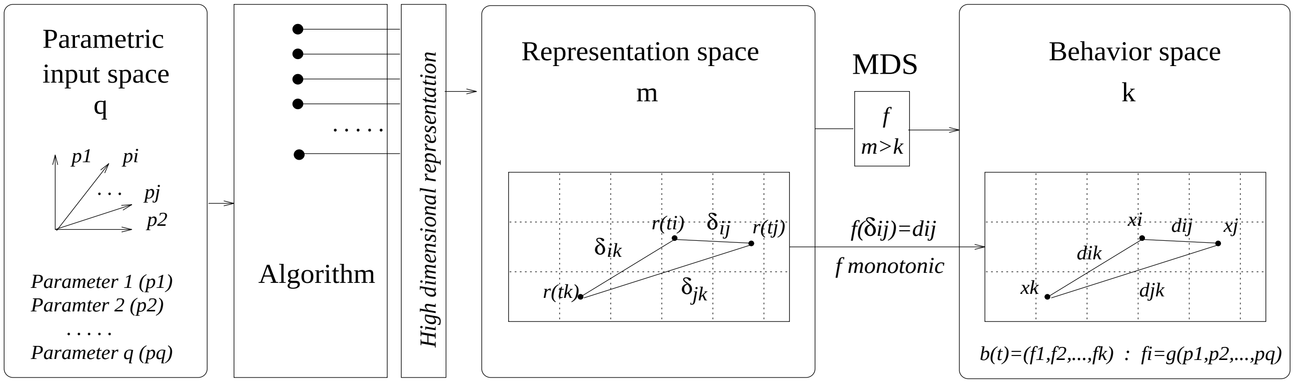


Fig. 1. Diagram of the steps in the proposed methodology. In our approach the parametric input space corresponds to a texture space (q = 4) and parameters have been defined as size, length, orientation and contrast of the image texton. The algorithm implements a computational model of texture perception, which implies a representation space with m = 192. The reduction of the MDS method provides a behavior space defined in terms of the input parameters, where dimensions verify $k < m$

0.0008, respectively), as is shown in Fig. 2. At this moment, it implies two dimensions in the behavior space

$$(f_1(s), f_2(l)) \tag{14}$$

where f_1 and f_2 represent monotonic functions.

On the other hand, variations along the orientation axis imply a non-monotonic one-dimensional plot with an important stress measure (badness-of-fit=0.26). But it decreases significantly for a two-dimensional space, presenting the configuration of Fig. 3 (badness-of-fit=0.0002). This circular configuration appears indistinctly for whatever size, length and contrast.

From this result we can state that the algorithm presents maximum distances whenever orientation differs by 90° and that there is no coordinate origin for the orientation axis. The algorithm does not represent directions but axes. That is, bars' orientation is represented by their inertia axis. These observations lead us to hypothesize that the algorithm representation behaves in a two-dimensional space like this:

$$(f_3(\cos 2\theta), f_4(\sin 2\theta)) \tag{15}$$

for variations in orientation, where f_3 and f_4 determine the radius of the circular configuration. At present, we are not able to hypothesize about the shape of these functions.

Likewise, contrast variations through a zero cross implies a non-monotonic variation of the algorithm representation on a one-dimensional space (badness-of-fit=0.2), whereas for a two-dimensional space the badness-of-fit is 0.00004. We can see this in Fig. 4. The important reduction of the stress value from one to two dimensions shows that there is a significant fact that is not represented in a one-dimensional space. Hence, the inflexion introduced by the zero-cross in the parametric space has to be considered in the representation space.

From these results, we can observe that there exists a symmetric behavior between dark and bright textons, and an increasing difference between them as the absolute value of contrast increases. Therefore, we can hypothesize that the algorithm represents textures with different contrast sign according to the following two-dimensional space:

$$(sgn(c) \cdot f_5(|c|), f_6(|c|)) \tag{16}$$

where sgn function represents the sign of the argument and f_5, f_6 seems to be monotonic functions.

At this point we can give a first approximation of the structure of the behavior space, in terms of s, l, θ and c parameters. In this case we go from a 19two-dimensional representation to an approximation on a six-dimensional space, b, such as

$$b(t) = (f_1, ..., f_k) \simeq (f_1, f_2, f_3, f_4, f_5, f_6) \tag{17}$$

where k would represent the correct dimensionality[4] and f_1, f_2, f_5, f_6 are monotonic functions of the following expressions

$$b(t) \simeq (f_1(s), f_2(l), f_3(\cos 2\theta), f_4(\sin 2\theta), \\ , sgn(c) \cdot f_5(c), f_6(|c|)) \tag{18}$$

4.2 Interactions between parameters

In order to refine the previous hypothesis, we will try to explore how dimensions interact between them. We can do it by constructing sets of images where variation occurs in more than one parameter.

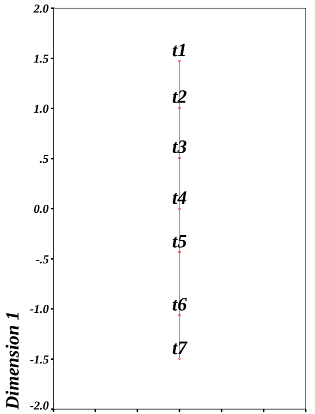
Parameters one and two vary in a monotonic sense. We can see this effect in Fig. 5, where the triangle configuration in the parametric space is preserved by the algorithm representation in a monotonic sense. There is a deviation due to image t_c, but we can not derive any clear interaction between these two axes. Other triangle configurations, that has been experimented in this subspace, present certain deviation for some images, as in this case, but do not allow one to infer any specific relation.

A contrast variation with respect to size can be seen in Fig. 6a. From the recovered configuration in Fig. 6b, we can deduce that distances between textures of the same contrast increase when contrast increases, that is

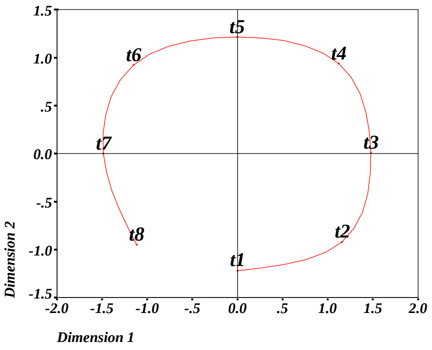
$$f_1 = g_1(s, c) \tag{19}$$

where g_1 is a monotonic function of s and c. While distances between textures of different contrast depend only on contrast difference, whatever the other parameters are

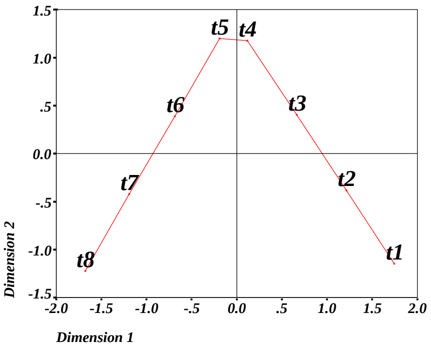
[4] The hypothetical true dimensionality which underlies the data



2



3



4

Fig. 2. MDS plot configuration of a set of images, $t_1, \ldots, t_7$, where each one presents size variations

Fig. 3. MDS plot configuration of a set of images, $t_1, \ldots, t_8$, where each one present orientation variations from 0° to 180°

Fig. 4. MDS plot configuration of a set of images, $t_1, \ldots, t_8$, where each one presents contrast variations with a zero cross between t_4 and t_5

$$f_6 = g_6(|c|) \tag{20}$$

such that, g_6 is a monotonic function depending only on $|c|$.

Equally, the same conclusions can be reached from square configuration with images with negative contrast. The squares are equally deviated from the input configuration, confirming expressions for 19 and 20.

This interpretation has also been done from experiments where variations on size have been substituted by length variations [22], obtaining

$$f_2 = g_2(l, c) \tag{21}$$

where g_2 is a monotonic function.

Interaction between orientation and the rest of the parameters has also been checked. In Fig. 7a we have selected a set of images varying in contrast and orientation, and in Fig. 7b we see the consequential configuration for the algorithm representation, where we can approximately say that on the recovered space dimension 1 corresponds to contrast variation and dimensions 2 and 3 correspond to orientation in the sense that we have expressed with f_3 and f_4 functions. We can see that images varying in orientation and with the same contrast are positioned in a circle configuration over the same plane. We can also observe that for a given orientation the algorithm representations are on the same plane and maintaining the configuration that we have seen in Fig. 4 for a contrast sign change. Finally we can state that the radius of the circle configuration depends on contrast values.

$$f_3 = g(|c|)\cos(2\theta) \tag{22}$$

$$f_4 = g(|c|)\sin(2\theta) \tag{23}$$

Following this last inference we have tried to prove if the radius depends only on contrast, or if it also depends on the rest of parameters. Graphics in Fig. 8 give support to this hypothesis, since, as we can observe, the radius varies for different lengths of bars (Fig. 8a), and for different size (Fig. 8b). From this observation we can deduce the following expressions

$$f_3 = g(s, l, |c|)\cos(2\theta) \tag{24}$$

$$f_4 = g(s, l, |c|)\sin(2\theta) \tag{25}$$

which means that the radius depends on monotonic variations of size, length and absolute value of contrast. It has been expressed with the same g function, since the shape of the circular configuration presents no distorsion in either of its two dimensions.

Other experiments can be found in [22] which confirm the expressions of the behavior space that we have constructed.

Thus far, we have obtained the following expression for the six-dimensional space, in which we have expressed the behavior of the algorithm, that is, the behavior space in this way:

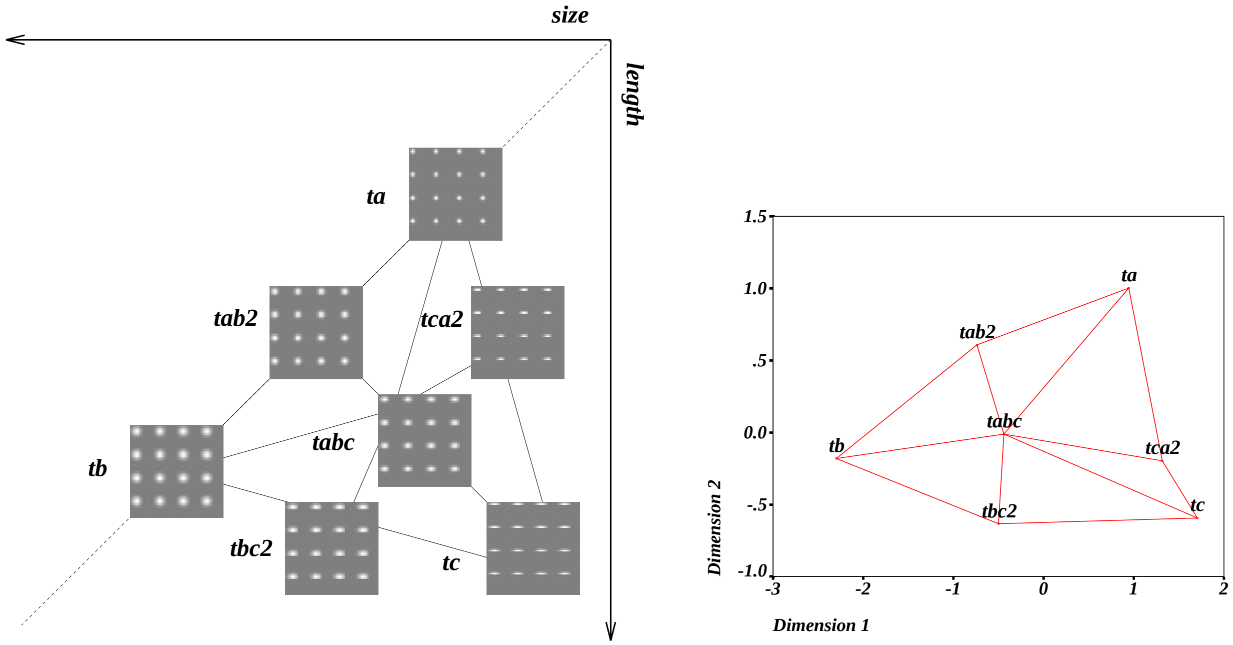
$$\begin{aligned} b(t) \simeq (&g_1(s, c), g_2(l, c), g(s, l, c)\cos(2\theta), \\ &, g(s, l, |c|)\sin(2\theta), sign(c) \cdot g_5(|c|), g_6(|c|)) \end{aligned} \tag{26}$$

4.3 Underlying dimension

The last question that we will attempt to respond is about the true dimensionality for the constructed space. Classical techniques to solve this problem have been based on a Monte Carlo approach [19, 20]. A computer simulation using random data with known configurations was used as the basis to construct sets of dissimilarities, which were scaled by a nonmetric MDS algorithm. Some errors were introduced in these sets of data. The stress values of these simulated data can be used as a guide to recover the unknown dimensionality.

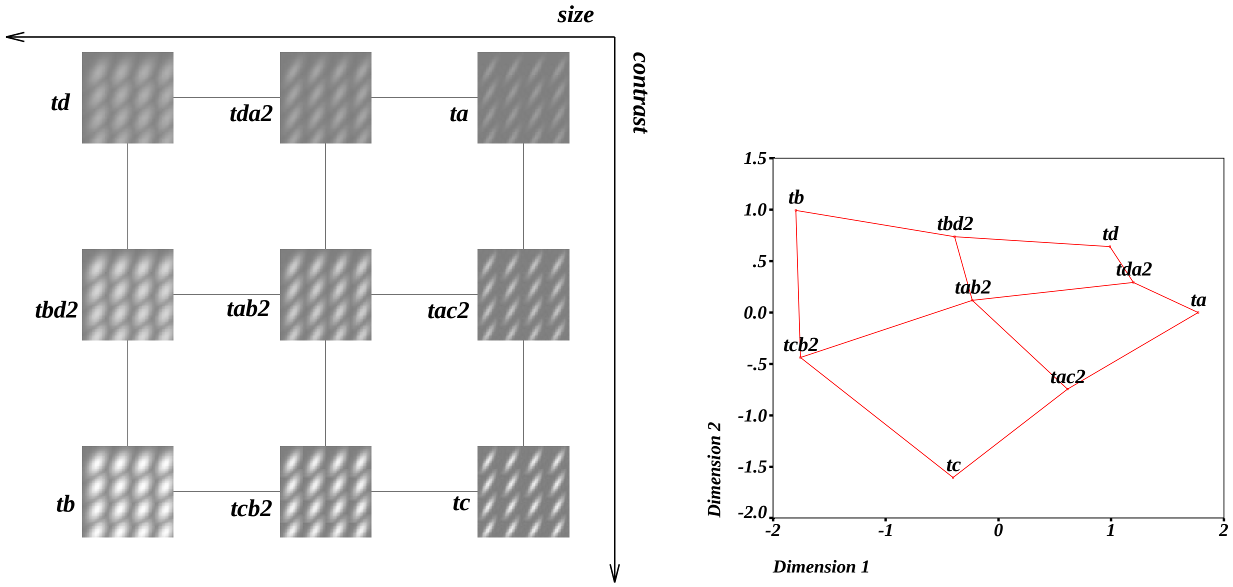
We have computed the stress values for a set of 36 images from our experiments. In Fig. 9 we show the obtained stress values versus the stress values of the simulated data in the Monte Carlo approach, given in [19]. From this we can deduce that our data present an underlying dimension of 4.

Taking this into account, we will give a last hypothesis about the behavior space of the algorithm. We can eliminate previous redundant dimensions and give the following final approach:



5a

5b



6a

6b

Fig. 5. **a** Points in the texture parametric space in a triangle configuration. Images t_a, t_b and t_c are at the vertex of an equilateral triangle, t_{ij2} is the midpoint between t_i and t_j and t_{abc} is the triangle centroid. Variations are taken on size and length parameters (assuming size≤length). **b** 2D MDS solution of the dissimilarities computed from the algorithm representation of points in **a** (badness-of-fit=0.003)

Fig. 6. **a** Square configuration of points in the parametric space. Variations of size and contrast. **b** 2D MDS solution of the algorithm dissimilarities on points of **a** (badness-of-fit=0.00061)

$$b(t) \simeq (g(s,l,c)\cos(2\theta), g(s,l,|c|)\sin(2\theta),$$
$$, sign(c) \cdot g_5(|c|), g_6(|c|)) \qquad (27)$$

since parameters in the two first dimensions were also considered in the others. This new space maintains all the constraints that we have been introducing during the study. At any rate, the relevant conclusion is that we have an important reduction and we know how the input parameters interact, independently of knowing the true dimensionality.

5 Experiments on natural images

To give support to the behavior space defined in the previous section, we have tested it on some natural images. We have taken a set of images from the Brodatz album [2] that can be perceived as points in the defined parametric texture model.

We have applied the process on the set of images in Fig. 10a, presenting a predominant direction on bars and with different bar contrast. The 2D MDS configuration of the resulting representations present the form that we have predicted in the behavior space (see Fig. 10b). Dissimilarities

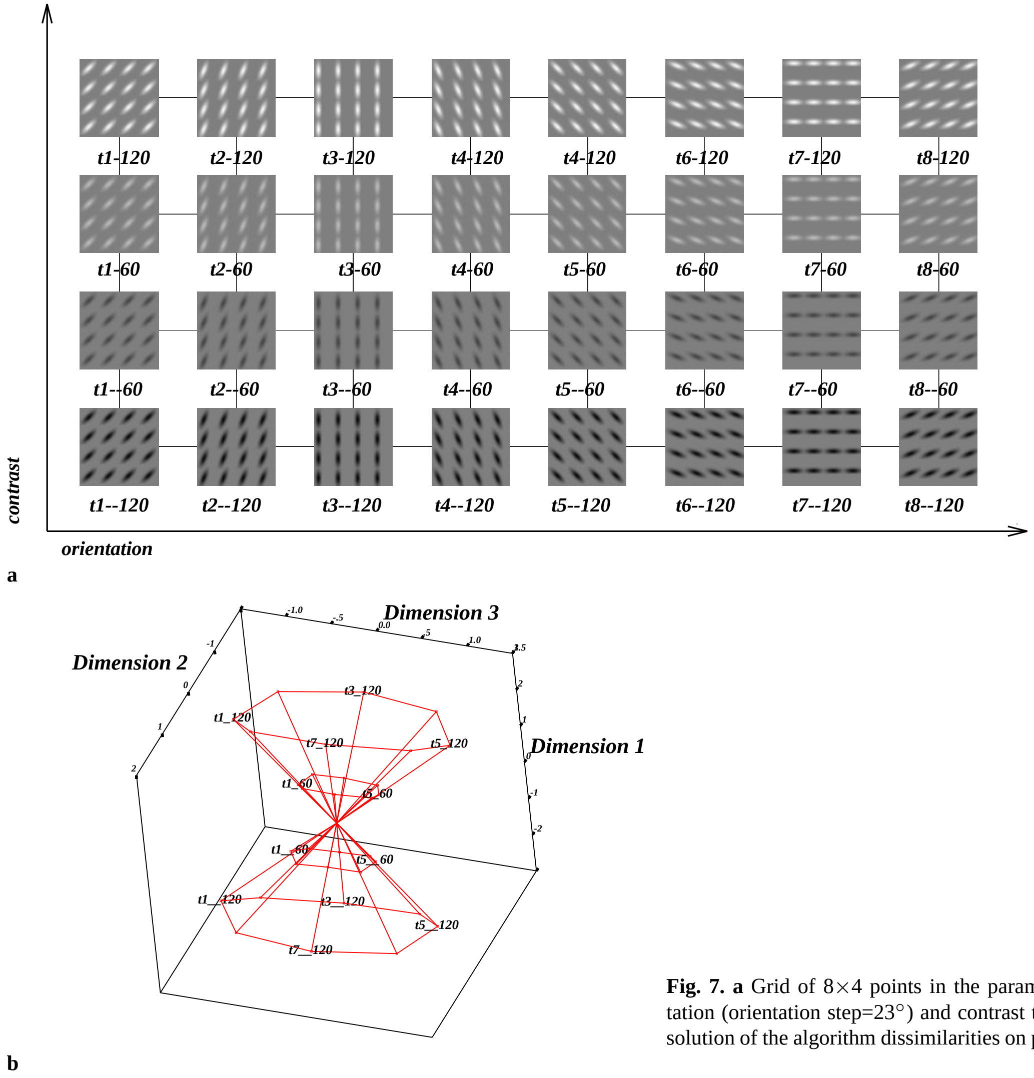


Fig. 7. a Grid of 8×4 points in the parametric space. Variations of orientation (orientation step=23°) and contrast through a zero cross. **b** 3D MDS solution of the algorithm dissimilarities on points of **a** (badness-of-fit=0.038)

Table 2. Stress values for scaled data to different dimensions expressed as k. Dissimilarity for our data has been computed over a set of 36 points in the representation space. Selected points corresponds to a subset of the images used in this work. The simulated data from a Monte Carlo approach have been obtained from the work of Spence and Graef [19]

	Stress	Monte-Carlo simulation	
k	(our data)	e=0.0000	e=0.0625
1	0.37402	0.387	0.447
2	0.19609	0.204	0.233
3	0.10843	0.097	0.125
4	0.05063	0.000	0.060
5	0.02685	0.000	0.055
6	0.02000	-	-

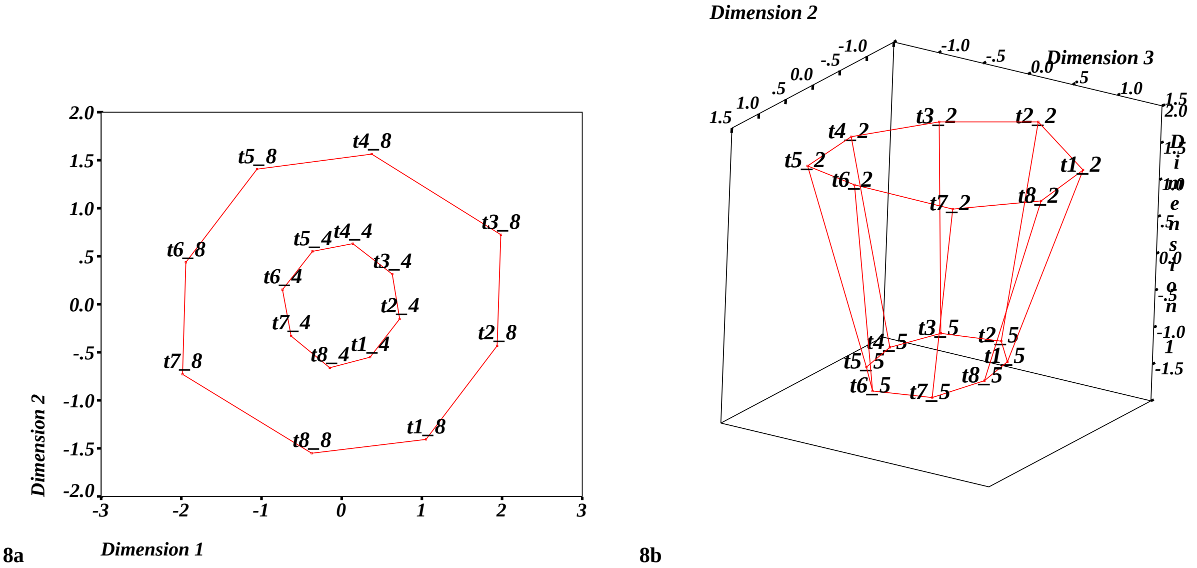
due to orientation are configured on a circle, whose radius depends on the bar contrast value.

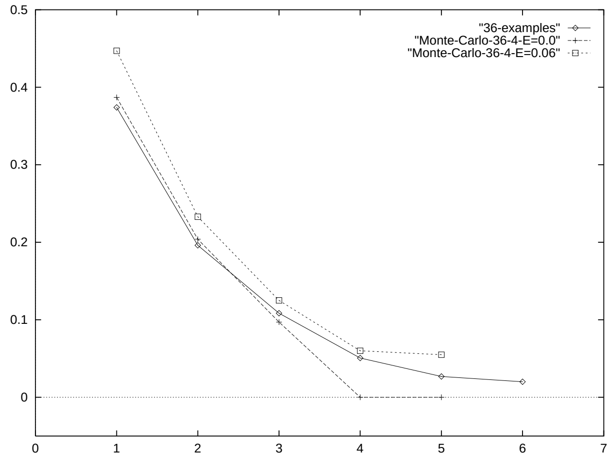
In a similar way we have tested the size parameter by reducing image resolution. We have constructed a square configuration of textures varying in contrast and resolution, simulating variations on contrast and size of texture blobs (see Fig. 11a). The 2D MDS configuration of the resulting representations preserve the square configuration in the same way as we had predicted for textures of the parametric space (see Fig. 11b).

6 Conclusions

In this work we present a methodology to explore the behavior of algorithms. We have based our approach on new tendencies in artificial intelligence which use empirical methods to establish causal models of programs. The approach relies on the extraction of general features of environments, tasks and behaviors.

We have particularly worked on the texture perception problem. Defining a simplified parametric texture space we have obtained a way to synthesize images by controlling certain interesting parameters. Any image has been formed by the repetition of a blob with specific parameters of size, length, orientation and contrast. We are not concerned about the specific location of textons, we only consider attribute densities as in Julesz's texton theory. This parametric space also agrees with Marr's image model based on bars and blobs and their attributes, which has been computationally defined in the work of Voorhees and Poggio [26].





9

Fig. 8. a 2D MDS solution of the algorithm dissimilarities on two sets of points $\{t1_4, \ldots, t8_4\}$ and $\{t1_8, \ldots, t8_8\}$, where tk_i represents the texture with orientation θ_k and length i (badness-of-fit=0.016) **b** 3D MDS solution of the algorithm dissimilarites on two sets of points $\{t1_2, \ldots, t8_2\}$ and $\{t1_5, \ldots, t8_5\}$, where tk_j represents the texture with orientation θ_k and length j (badness-of-fit=0.002). (In both cases consecutive orientations are separated by 23°)

Fig. 9. Graphical representation of the stress values in Table 2 versus values from the Monte Carlo approach on four-dimensional data with two different error levels

The selected image model defines the algorithm environment. We propose to test the algorithm in its environment by isolating specific features of it, in the same way that psychologists do in psychophysical experiments.

Assuming that the algorithm returns a high-dimensional representation of the input image, we have based our approach on the use of the multidimensional scaling method. It has been applied from a dissimilarity matrix constructed by distances between representations of different images. An exhaustive study on variations of specific parameters of the input images has permitted formulation of a four-dimensional behavior space for this algorithm on this simplied image model. Considering that the bars and blobs model has been defined as a general image model, the results on this parametric space should be able to be extended to any image.

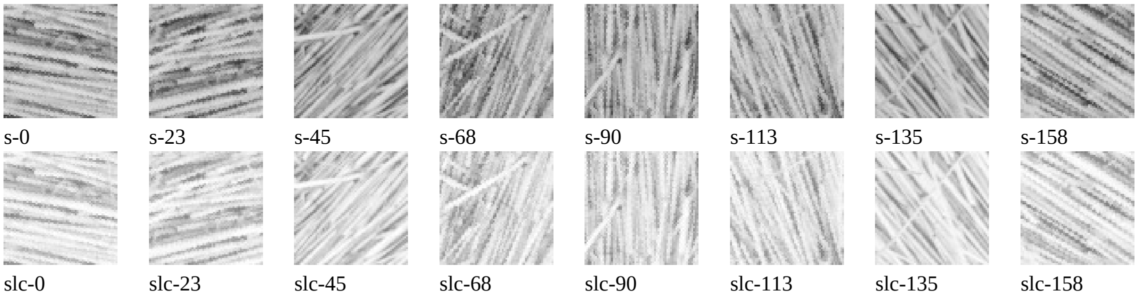
This behavior space can be understood as a qualitative causal model, which is the basis for development of the quantitative causal model of the algorithm. It also provides a representation of the algorithm in such a way that it facilitates the comparison between different algorithms.

Finally, we conclude that the construction of the behavior space of any given texture representation is a relevant step in solving automatic classification problems. In this sense we can see this behavior space as a representation space with a known underlying structure which will allow us to construct systems based on this representation, i.e. systems able to describe any new texture by their proximity to some given set of prototypes in this space, as it has been done with the shape recognition problem in [4].

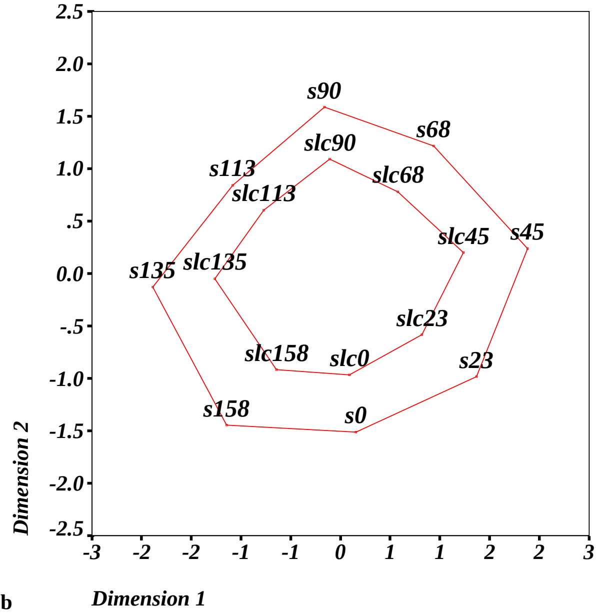
An immediate continuation to this approach is to test it on images presenting more than one kind of textons. In this case, the approach would permit representation of any texture in terms of texture prototypes of the defined model.

It would be very interesting to test the approach on different neural networks in which the high dimensionality problem of certain intermediate steps sometimes makes prediction of their behavior difficult. In this case, we should extract the information of an intermediate stage in the network and use it as a representation space.

The construction of the behavior space of a computational model based on the human visual system opens a

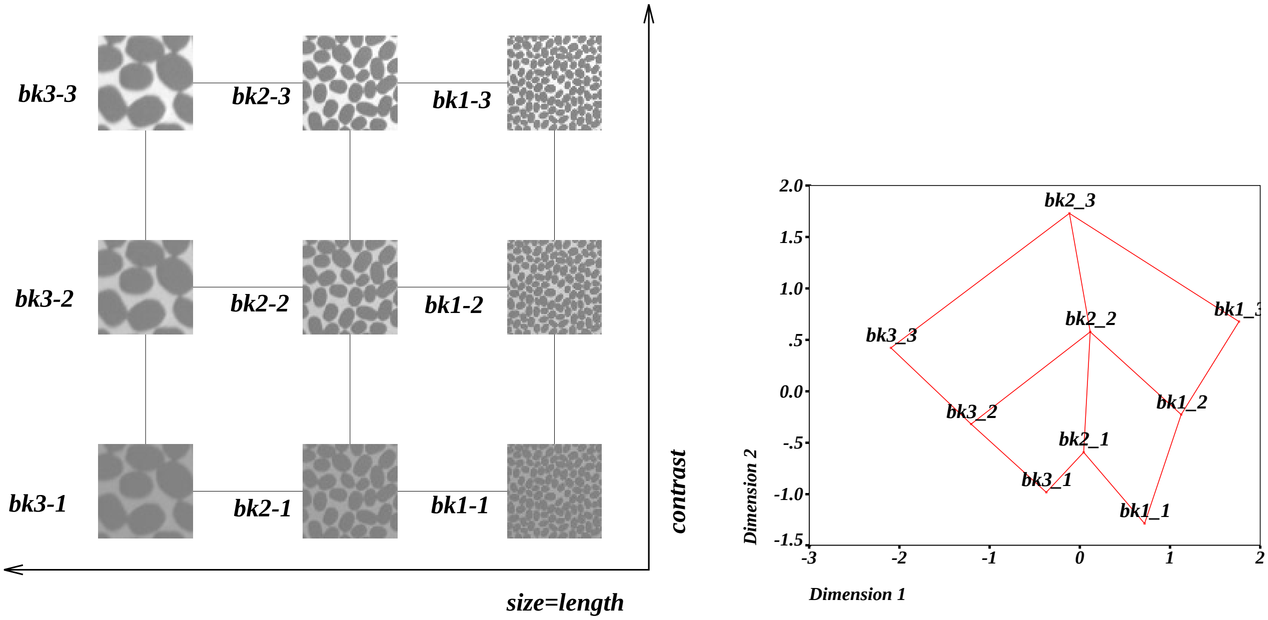


10a



10b

Fig. 10a,b. A set of textures with a predominant bar orientation. Textures {*s-0,...,s-158*} present a photometric transformation on the range of 0 to 255 gray levels. Textures {*slc-0,...,slc-158*} present a photometric transformation on the range of 128 to 255. At the bottom we present the 2D MDS configuration of these two sets of images (stress= 0.05)



11a

11b

Fig. 11. a Image transformations on resolution and background contrast. The resolution variation is positioned along the axis where size=length. Contrast variations on background is on the contrast axis. **b** 2D MDS configuration of the algorithm representations of images in **a** (stress=0.004) new line to systematically test whether it behaves as humans do. This point of view implies construction of the perceptual space of human vision from a set of psychophysic experiments on stimuli constructed from the same parametric model. These experiments will allow construction of a dissimilarity matrix. A computational model of vision will be valid if the behavior space presents the same structure of the perceptual space constructed from the same set of images.

References

1. J.R. Bergen, M.S. Landy (1991) Computational Models of Visual Processing, chapter Computational Modeling of Visual Texture Segregation, pp 253–271. MIT Press, Cambridge, Mass.
2. P. Brodatz (1966) Textures. New York: Dover
3. P. R. Cohen (1995) Empirical methods for Artificial Intelligence. MIT Press, Cambridge, Mass.
4. S. Duvdevani-Bar, S. Edelman (1995) On similarity to prototypes in 3d object representation. Technical report, Weizmann Institute
5. L.van Gool, P. Dewaele, A. Oosterlinck (1985) Survey: Texture analysis anno 1983. Computer Vision, Graphics and Image Processing 29:336–357
6. N. Graham, A. Sutter (1996) Effect of spatial scale and background luminance on the intensive and spatial nonlinearities in texture segregation. Vision Research 36:1371–1390
7. R.M. Haralick (1979) Statistical and structural approaches to texture. In Proceedings of the IEEE, pp 304–322, May
8. M.J. Norusis/ SPSS Inc. (ed) (1994) SPSS Professional Statistics 6.1
9. B. Julesz, J.R. Bergen (1983) Textons, the fundamental elements in preattentive vision and perception of textures. Bell Systems Technological Journal 62:1619–1645
10. J.B. Kruskal (1964) Nonmetric multidimensional scaling: A numerical method. Psychometrika 29:115–129
11. J.B. Kruskal, M. Wish (1978) Multidimensional Scaling. Sage University Paper
12. J. Malik, P. Perona (1990) Preattentive texture discrimination with early vision mechanisms. Journal of the Optical Society of America 7:923–932
13. K.V. Mardia, J.T. Kent, J.M. Bibby (1995) Multivariate Analysis, chapter Multidimensional Scaling, pp 394–423. Academic Press, New York
14. D. Sagi (1991) Early Vision and Beyond, chapter The Psychophysics of Texture Segmentation, pp 69–78. MIT Press, Cambridge, Mass.
15. J. Serra (1982) Image analysis and mathematical morphology. Academic Press, New York
16. R.N. Shepard (1962) The analysis of proximities: multidimensional scaling with an unknown distance function. i and ii. Psychometrika 27:125–140,219–246
17. R.N. Shepard (1980) Multidimensional scaling, tree-fitting, and clustering. Science 210:390–210
18. R.N. Shepard (1987) Toward a universal law of generalization for psychological science. Science 237:1317–1323
19. I. Spence, J. Graef (1974) The determination of the underlying dimensionality of an empirically obtained matrix of proximities. Multivariate Behavioral Research 9:331–342
20. I. Spence, J.C. Oglivie (1973) A table of expected stress values for random rankings in nonmetric multidimensional scaling. Multivariate Behavioral Research 8:511–517
21. M. Tuceryan, A.K. Jain (1993) Handbook of Pattern Recognition and Computer Vision, chapter Texture Analysis, pp 235–276. World Scientific, Teaneck, NJ
22. M. Vanrell (1996) Exploring the behavior of a texture perception algorithm. Technical Report 12, CVC
23. M. Vanrell, J. Vitrià (1995) Optimal decomposable disks for morphological transformations. Image and Vision Computing. Second revision
24. M. Vanrell, J. Vitrià (1996) 3×3 decomposition of circular structuring elements. In: International Conference on Image Processing, volume 3, pp 5–8. IEEE
25. H. Voorhees, T. Poggio (1987) Detecting textons and texture boundaries in natural images. In First International Conference on Computer Vision, pp 250–258, 1987.
26. H. Voorhees, T. Poggio (1988) Computing texture boundaries from images. Nature 333:364–367
27. H. Wechsler (1980) Texture analysis – a survey. Signal Processing 2:271–282
28. F.W. Young, R.A. Faldowski, M.M. McFarlane (1993) Handbook on Statistics 9 – Computational Statistics, chapter Multivariate statistical visualization, pp 959–998. North Holland, Amsterdam

Maria Vanrell is an Associate Professor of Artificial Intelligence in the Computer Science Department at the Universitat Autònoma de Barcelona. She received an MSc in Graphics, Computer Vision and Artificial Intelligence from the UAB. In 1996 she received a PhD at the UAB for work in texture analysis and representation. Her research interest includes texture representation and artificial intelligence.

Jordi Vitrià is an Associate Professor in the Computer Science Department of the Universitat Autónoma de Barcelona. In 1990 he received a PhD at the UAB for work in mathematical morphology and image analysis. He is currently in charge of the research area at the Computer Vision Center. He has worked in several funded projects in the areas of visual inspection and medical imaging. He has published several papers on image analysis, computer vision and its applications to visual inspection, and medical images.

Xavier Roca is an Associate Professor of Software Engineering in the Computer Science Department at the Universitat Autònoma de Barcelona. He received an MSc in Graphics, Computer Vision and Artificial Intelligence from the UAB. He is currently pursuing a PhD in Computer Science and is involved in development of Active Vision systems. His research interest includes focus and active vision.