USING MARKOV RANDOM FIELDS IN CERAMIC DESIGN

E. de Ves[1], A. Simó[2], O. Belmonte[1]

[1]Instituto de Robótica, Universidad de Valencia
C/ Hugo de Moncada, 4 6010 Valencia
[2]Dept. of Mathematics, Universitat Jaume I

1. INTRODUCTION

In recent years computer assisted design has been widely used in branches of industry such as ceramics, textiles, footwear, etc.

Specifically in the ceramic industry, the production of floor and wall tile designs resembling natural textures like granite, marble, wood, etc. as closely as possible, would appear to be a subject of considerable interest.

It would be very useful to have a user-friendly tool that allowed the professional designer to generate different configurations of the same tile, while keeping the same visual appearance (i.e. having a common characteristic that would permit differentiating all the realisations belonging to the same texture). However, on observing these in detail, they would be as different as two pieces of the same rock.

This led to setting two closely related objectives:

- In the first place, designing different textures by specifying just a few parameters.
- Secondly, obtaining different realisations of the same texture, in order to approach nature's random behaviour.

2. GENERAL DESCRIPTION OF THE METHOD

The mathematical methods chosen for generating the ceramic textures are called

Markov Random Fields (MRF)[1]. These techniques were selected as a result of the very nature of the problem itself. Multiple random realisations are sought of the same design, and MRF are precisely based on the consideration that multiple variables behave in a random fashion while following a probability distribution. In our case, the random variables are the pixels in the image, which can take on different levels of grey.

The present paper is structured as follows; Section 2 provides a detailed explanation of Markov random fields and describes two of the energy functions used for our purposes. Section 3 gives an overview of the applied simulation methods and concludes with a presentation of some of the most significant results obtained with the method.

3. MARKOV RANDOM FIELDS

A Markov random field will be a probability distribution on the set of all possible images that satisfy certain properties. The distribution will have a characteristic shape, so that certain configurations are more likely to be found than others.

In the images that follow a MRF there is a certain relation between the colour of neighbouring points. This relation can be expressed mathematically as a potential difference between these points.

An image can be interpreted as a function in plane XY, which assumes discrete values between [0,255][2]. Thus a representation of the image would be the function shown in Fig. 1.

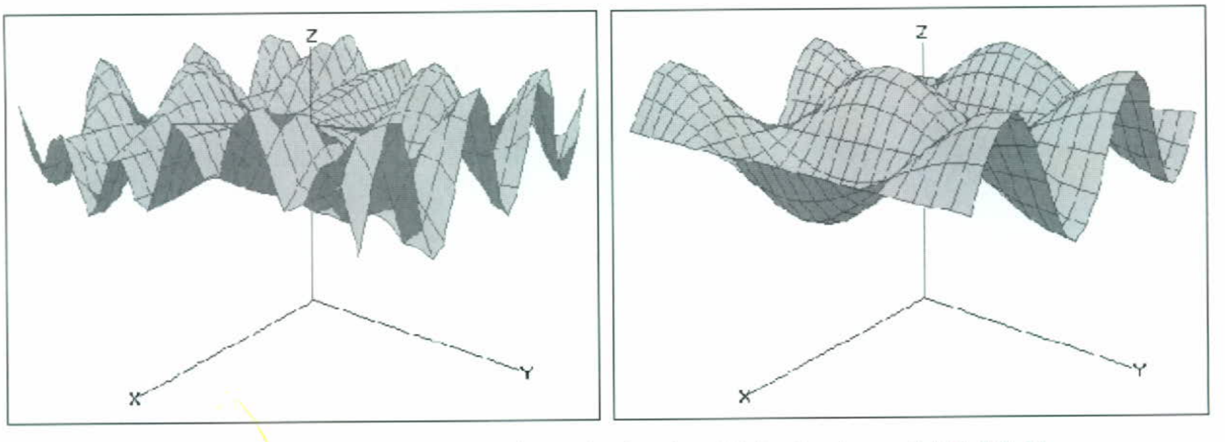

Fig 1. Representation of an image as a 3 dimensional surface. (a) Starting image. (b) Modelled image.

Let us assume a starting image as shown in Fig. 1, in which however the function only assumes values at certain points of plane XY (discrete function), given by a grid of positions s=(x , y), which we will call pixels. A level of grey (or colour) is associated with each pixel in the range [0,255]. This defines a surface and each configuration will have a different surface. Let us now imagine that the surface can be modelled at will, by introducing for this purpose certain forces in each position (local forces), which will make the surface smoother or rougher depending on the type of force defined in each position.

[1]. Rama Chellapa and Paul Jain. *Markov Random Fields, Theory and Applications.* Academic Press, Inc, 1991.
[2]. Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing.* Addison-Wesley Publishing Company, 1987.

The objective is to define certain local forces and allow the image to evolve, obtaining an end image that will be the targeted texture in which the colour of a position s=(x, y) only depends on its neighbours.

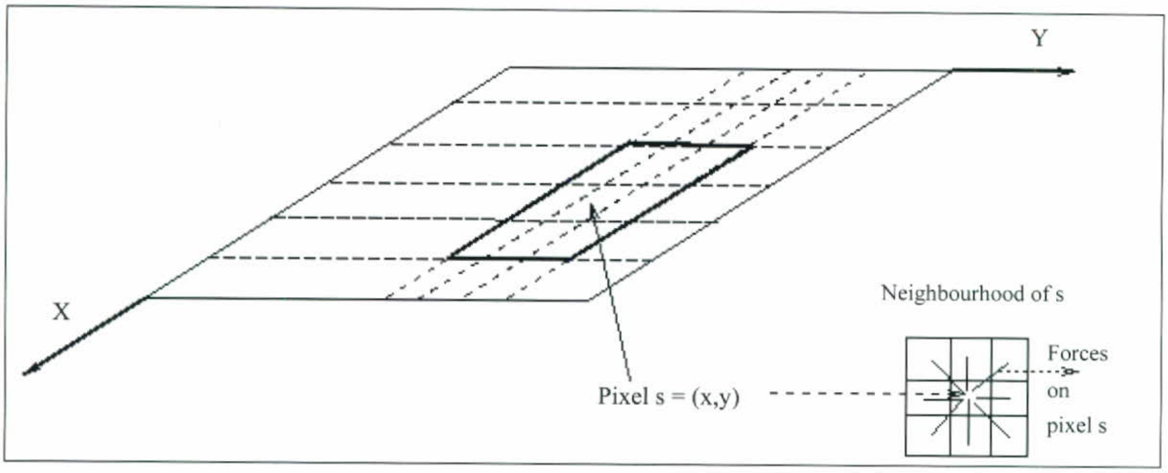

Fig. 2. Representation of the image plane and neighbourhood used.

Having set out this idea in an intuitive form, we can now go on to formalise it. Let an energy function given by E(x) represent the forces on the different pixels in the previously described image. This energy defines a probability distribution given by:

$$P(x) = \frac{1}{Z} exp(-E(x))$$

where P(x) is the probability that a given realisation x is obtained of the image, and $Z=\sum exp(-E(x))$ is a standardisation constant whose value is obtained by calculating the summation of the probabilities on the set of all the possible images. This distribution is called the Gibbs Field[3]. The meaning of the foregoing equation is that the most probable images in this distribution are precisely those with the least energy.

For the above distribution to be a Markov distribution, the following two requirements need to be satisfied for energy:

- That a neighbourhood relation δ(s) be defined on the pixels in the image.
- That function E(x) be locally defined, according to defined neighbourhood δ(s), as a summation of local potentials $V(x_S)$ that will depend for each given pixel s on the neighbouring pixels.

Another feature that has not yet been mentioned, but which appears very interesting from a texture design point of view, is that of achieving continuity at the image boundaries, that is, when the same image is set various times, the texture is not broken at the edges. This has been done by defining the neighbourhood δ(s) at the image boundaries in a special way. See fig. 3.

[3]. Anand Rangarajan and Rama Chellapa. Markov Random Field Models in image processing. *The handbook of Brain Theory and Neural Networks*, (pp. 564-567),1995.

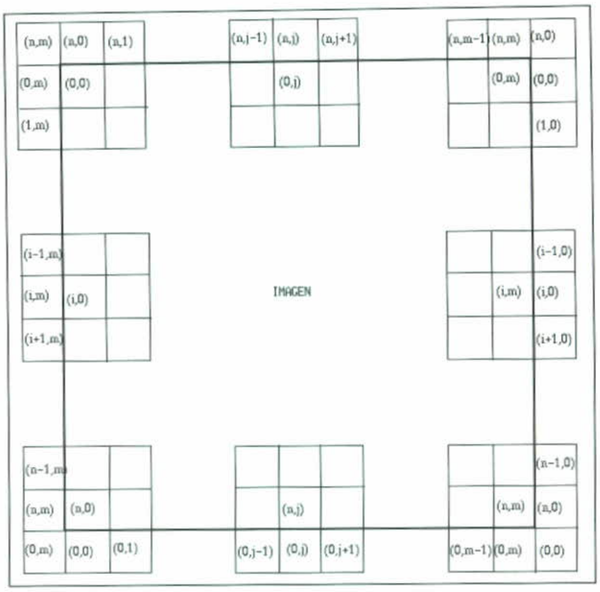

Fig. 3. Boundary definition of neighbourhood.

The work to be done will involve defining the energy function (equivalent to the forces acting locally) and proposing algorithms that allow any image (that will act as a seed) to evolve, in order to obtain other, different, most probable configurations, according to a locally specified probability distribution.

Fig. 3 shows a general schematic of the whole system used, in which the part corresponding to the definition of local Energy remains to be explained. This is set out below in Section 3, followed by the simulation algorithms.

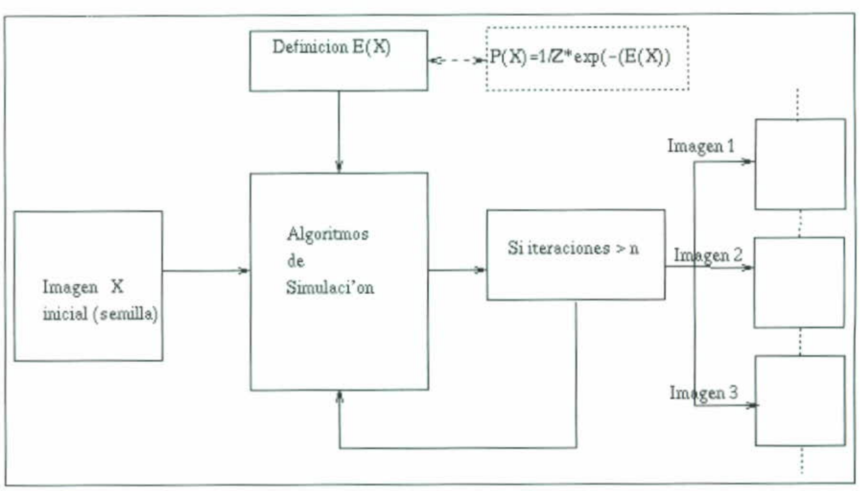

Fig. 4. Schematic of the system used in generating random textures.

3.1. TEXTURE MODELS

We have already stated that defining a random Markov field only involves inputting an energy function expressed in a potential summation on image neighbourhoods.

At this point we will describe some Energy functions used for texture synthesis by Markov Random Fields.

The models provided depend on a set of parameters θ_i such that different types of texture can be obtained, depending on the value of these parameters.

3.1.1. MODEL ϕ

Now we will look at an example of Energy that meets all the necessary requirements for the resulting distribution to be a Markov distribution.

Let the energy function be given by[4],

$$E(x) = \sum_{i}^{4} \sum_{<s,t>_i} \theta_i \phi(x_s - x_t)$$

The symbol $<s,t>_i$ indicates that positions s and t form one of 4 different types of click (direction in which t lies relative to s as shown in Fig. 5.

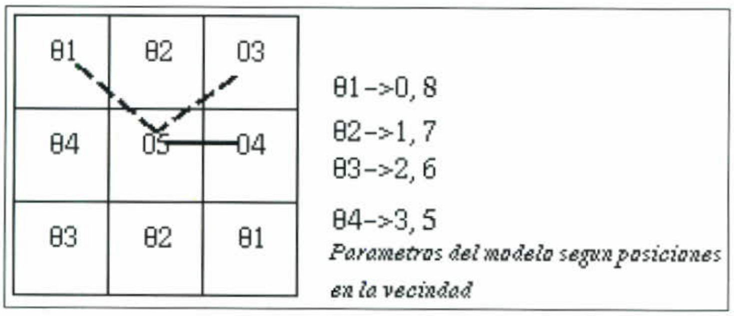

Fig. 5. Neighbours of a given s pixel of order 2 and associated parameters.

Function ϕ called the disparity function can be, for example:

$$\phi\ (\Delta) = \frac{-1}{1 + \frac{\Delta^2}{\delta}}$$

It is to be noted that positive values of parameters θ_i favour images with similar levels of grey, while parameters of $\theta_i < 0$ favour images exhibiting great discrepancies between grey levels. Moreover, this model allows generating isotropic or anisotropic textures depending on whether identical or different θ_i are used.

3.1.2. AUTOBINOMIAL MODEL

This model is characterised by its given energy function for a pure texture.

$$E(x) = -\sum_i \theta_i \sum_{<s,t>_i} x_s x_t - \theta_0 \sum_s x_s - \ln\binom{N}{x_s}$$

where the levels of grey belong to range [0, N]. In our specific case N=255. This model was used by Cross and Jain (1983) for synthesis and modelling of real textures.

---

[4]. GERHARD WINKLER. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer-Verlag Berkub Geudkberg, 1995.

This Energy function will yield a greater probability of images with high levels of grey and $_s$ values (light images), owing to the product of strengths. The logarithmic factor tries to compensate this effect.

4. SIMULATION BY METROPOLIS ALGORITHMS

Having specified an energy function E(x), this univocally defines the probability distribution, that is, the Markov (or Gibbs) field. The objective set was obtaining realisations proceeding from the defined probability distribution. As the standardising constant Z is usually unknown, it is difficult to simulate. The Monte Carlo dynamic algorithms have drawn considerable attention in this field. Given the nature of our problem, 2 specific algorithms have been selected. These algorithms yield the image by means of an individual pixel updating scheme[4].

SINGLE FLIP algorithm
Given a configuration x **RUN**
START
 Generate position in image (pixel) randomly
 Generate level of grey x_s randomly x_s in range [0,255]
 Let x_{sa} be level of grey in position s in configuration x
 If $E(x_s) <= E(x_{sa})$ then
  New configuration z on replacing level of grey by x_{sa}
 If not,
  Let u be a random uniform number
  If $u > E(x_{sa})$ then
   New configuration z on replacing level of grey x_{sa} by x_s
 If not the configuration does not update z=x
END

EXCHANGE Algorithm
Given a configuration x **RUN**
START
 Generate 2 positions in image (pixels) s and t randomly $s \neq t$
 Obtain a new image z exchanging levels of grey corresponding to positions s and t
 If E(z) <= E(xs) then
  New configuration z.
 If not,
  Let u be a random uniform number
  If u > E(z)-E(x) then
   New configuration z
  If not the configuration does not update z=x
END

[4]. Gerhard Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods.* Springer-Verlag Berkub Geudkberg, 1995.

It is to be highlighted that this last algorithm conserves the grey level proportions of the initial image, which may in some cases be of considerable interest.

Also note that this algorithm only updates one position in each run. Therefore, to obtain a total scan of the image, the program must be run for at least the number of pixels in the original image. The end image will be obtained by repeating the algorithm in an n large number size multiple of the original image. The convergence of the algorithms indicated can be found in[4].

5. EXPERIMENTAL RESULTS

By using three different types of energy functions, 2 as set out above and often found in the literature, and one defined by ourselves, and employing the Metropolis algorithms indicated, certain examples were produced of textures that fully met our initial objectives.

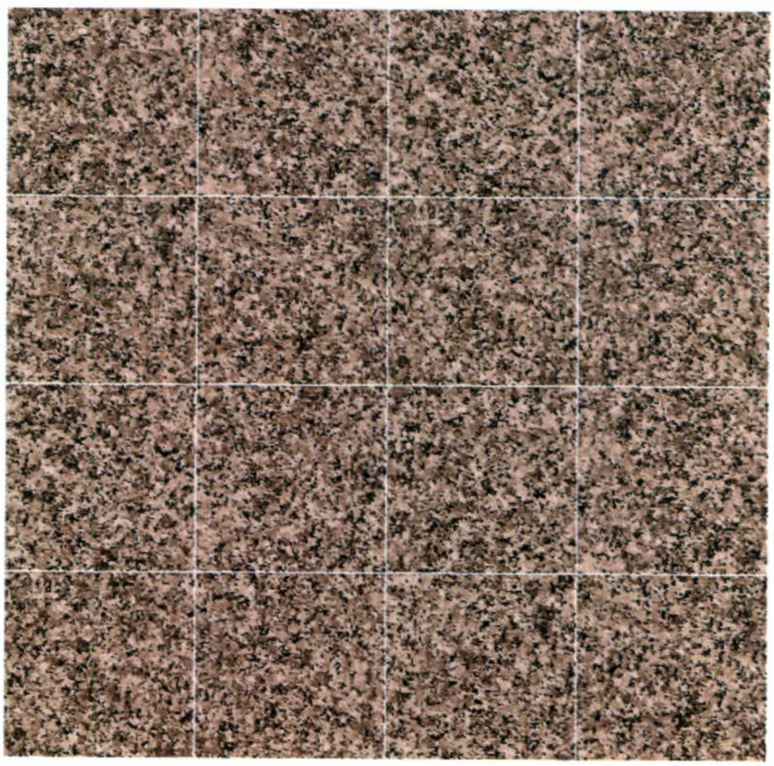

Figure A. *Texture obtained with model F. Each tile was a different realisation obtained with the same parameters*

A single texture was generated in Fig. A and this has been repeated in order to visualise the continuity effect at the boundaries.

Multiple realisations are presented in Fig. B of one same texture (each tile differs from the others). It can be observed, even in this case, that there is an effect of continuity.

---

[4]. GERHARD WINKLER. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods.* Springer-Verlag Berkub Geudkberg, 1995.

All the realisations are samples of the same texture, obtained by the simulation algorithms mentioned above, which yield a sample with 5000 iterations each after correcting the starting algorithm a sufficiently high number of times for the realisations to fit the probability distribution set as a model.

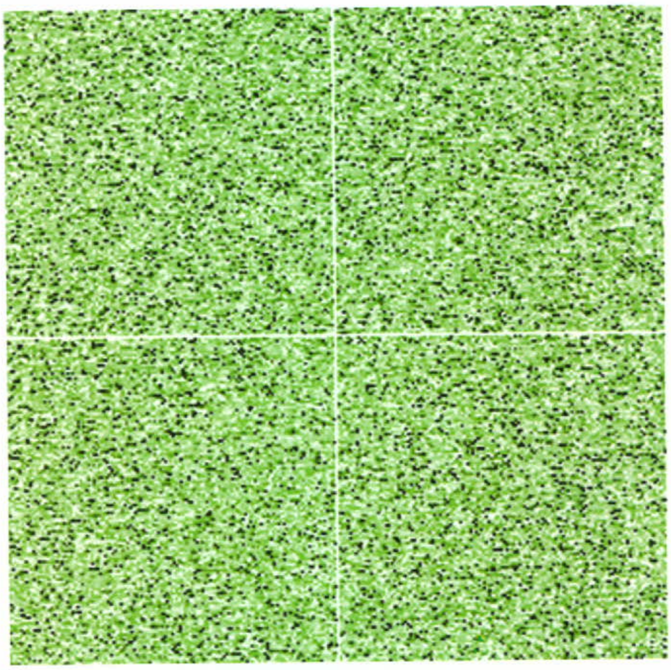

Figure B. *Texture obtained with the same model but with parameters allowing a finer granule.*

Figs. C and D show two different textures obtained by simulating the probability distribution of model f, on modifying this model's variable parameters.

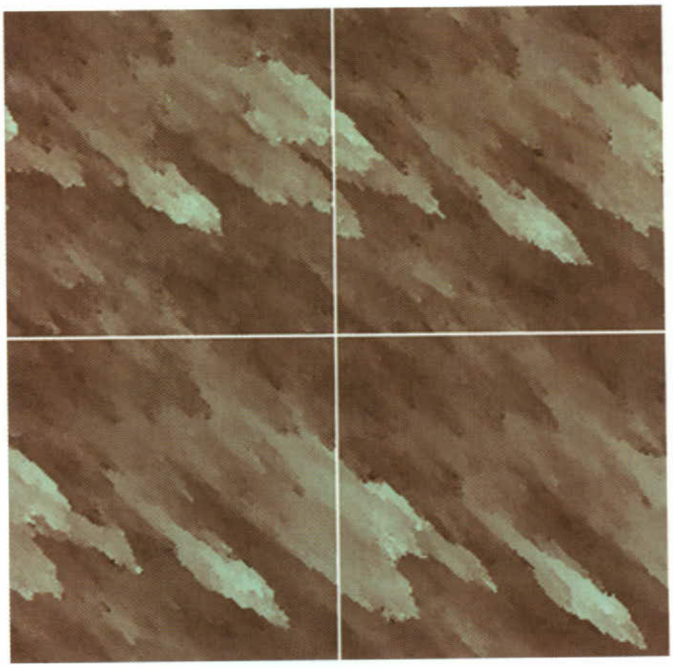

Figure C. *Texture obtained with the same model but with parameters that smoothed the texture.*

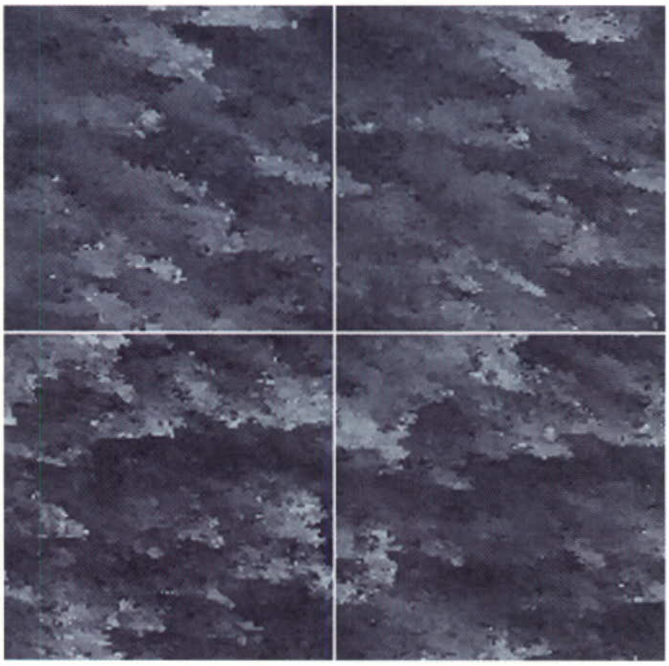

Figure D. Example of a texture generated by changing the parameters in the foregoing model.

6. CONCLUSIONS

A work method has been set out that allows obtaining different textures by specifying a local energy function. The textures synthesised by Markov Random Fields were continuous owing to the neighbourhood used. Various different realisations were also generated from a single texture, which were installed in a mosaic without breaks at the edges.

Markov Random Fields are very useful in generating textures that simulate nature's random behaviour. In this study, the objective was to set an energy function and produce an associated texture. Future studies will involve using the method in the opposite sense: given a natural structure, parameters will be sought that fit such a texture in order to imitate it.

7. ACKNOWLEDGEMENTS

Esther de Ves's work was funded by a pre-PhD grant awarded by the Generalidad Valenciana in the framework of the Research Staff Training Programme.