3D FURLIKE TEXTURE GENERATION BY A 2D AUTOREGRESSIVE SYNTHESIS.

Leila Khouas, Christophe Odet, Denis Friboulet
Creatis
CNRS Research Unit (UMR 5515)
affiliated to INSERM, Lyon, France
khouas@creatis.insa-lyon.fr

ABSTRACT

This paper presents a texture model that allows a two dimensional synthesis of 3D furlike texture. The model is based on the modelling of the texture autocorrelation function (ACF) and a two dimensional AutoRegressive synthesis (2D AR). We consider that the appearance of a furlike texture is defined by two characteristics: the orientation and the length of each individual element (i.e. filament). The values of these characteristics are constant in a stationary texture. The AR synthesis model of a given texture is defined by a number of parameters which can be computed from the texture ACF. Therefore, we have built a mathematical model of the ACF with respect to orientation and length. The model obtained consists of an analytical expression of ACF with respect to orientation. For the length, we have used a lookup table. Indeed, the ACF expression includes some parameters which are numerically estimated for several values of length. This allows to build a lookup table indexed by length and containing at each entry a set of ACF parameters. In this way, a furlike texture synthesis based on this table can be implemented. We show that with an appropriate choice of other parameters of the AR synthesis, this approach yields satisfying results, essentially for the simulation of fur appearance on 3D surfaces.

Keywords: furlike texture, texture modelling, AR modelling, texture synthesis, texture mapping, fur appearance simulation.

1 Introduction

The difficulties of furlike texture synthesis result from the large number and detailed geometry of the individual elements to process, the complex interaction of light and shadow among them and the small element width in relation to the rendered image. These problems, generally encountered while rendering objects with complex surfaces were largely studied. Some of the proposed solutions are based on a geometrical modelling of each individual element [Dalde93a][Dalde93b]. Others attempt to reproduce the complex surface appearance through 3D textures and lighting models [Kajiy89][Perli89]. These techniques yield realistic results but remain too time consuming. As an alternative to the above complex models, we propose in this work to build a 2D texture model for a simple and efficient synthesis of furlike texture. This requires that the model must produce furlike texture with arbitrary orientation and length (i.e. length of individual elements); in addition the appearance of the generated texture is desired to be quite realistic. We have used the AR approach as described in section 2 to build such a model.

A straightforward approach would consist in the estimation of different sets of AR model parameters from input images of fur with different orientation and length values.
Since we need precise knowledge of length and orientation, we have not used real fur images (which we did not have at our disposal anyway). Instead, we have used as input, images generated by a 3D texture synthesis technique based on the hypertexture modelling developed by K. Perlin and M. Hoffert [Perli89]. This method in which we have introduced orientation, allows a full control of all visual aspects of the texture but, as already said, is computationally intensive and the modelling of an oriented fur for a given shape is not trivial.
Once the AR parameters are estimated from those input images, we could build a lookup table giving the AR parameter sets for several

combinations of length and orientation values. However, dealing with arbitrary values would require a huge table including an important and complete range of orientation and length values. We therefore investigated the possibility to obtain a mathematical model that expresses the ACF with respect to orientation and length. In this way, the ACF can be computed during the synthesis process for any orientation and length values and the AR parameters can be easily deduced and used for image synthesis (see section 2.3).

The resulting furlike texture generator was applied for the simulation of fur appearance on 3D surfaces. This is done, as shown in section 3 by an appropriate mapping of the texture during the rendering process.

2 Furlike texture generation by 2D AR synthesis

2.1 Review of the AR model

2.1.1 Definition

The AR modelling of a texture is based on a supposed linear dependence of each pixel of the texture image on its neighbours. A texture image is considered as the output of a two dimensional linear filter in response to a white noise at the input. The shape and size of the neighbourhood respectively define the causality of the model and its order. The values of the combination coefficients represent the set of the model parameters. The model can be described by the following difference equation:

$$y(i,j) = w(i,j) + \sum_{(m,n)\in\mathcal{V}} a(m,n,i,j)y(i-m,j-n) \quad (1)$$

Where y is a texture image with a zero mean; w is a zero mean white noise; $y(i,j)$ is the value of a pixel (i,j) and $a(m,n,i,j)$ are the model parameters. $\mathcal{V}$ defines the prediction region, i.e. the set of neighbours on which the pixel (i,j) depends. In a stationary case, the values of the parameters $a(m,n,i,j)$ are independent from the pixel position (i,j).

2.1.2 Parameters Estimation

Given an appropriate neighbourhood $\mathcal{V}$ (shape and size), the estimation of the parameters $a(m,n)$ in the stationary case can be performed using the ACF of the texture image y. Indeed, the equation (1) can be rewritten as follows:

$$w(i,j) = y(i,j) - \sum_{(m,n)\in\mathcal{V}} a(m,n)y(i-m,j-n)$$

In our context, since we are looking for a predictive model that best fits the given texture image, we consider that $w(i,j)$ is the prediction error at each pixel. Therefore, the model parameters can be estimated by the minimisation of the variance of the noise w:

$$E\big[w(i,j)^2\big] = E\Big[\big(y(i,j) - \sum_{(m,n)\in\mathcal{V}} a(m,n)y(i-m,j-n)\big)^2\Big]$$

with respect to $a(m,n)$ parameters. This implies the solution of the set of equations:

$$\frac{\partial E\big[w(i,j)^2\big]}{\partial a(m,n)} = 0, \forall (m,n)\in\mathcal{V} \quad (2)$$

$E[.]$ denotes the average operator. By performing the partial derivation given in equation(2) and replacing $w(i,j)$ with its value, we obtain:

$$E\big[y(i,j)y(i-s,j-t) - \sum_{(m,n)\in\mathcal{V}} a(m,n)y(i-m,j-n)y(i-s,j-t)\big] = 0, \forall (s,t)\in\mathcal{V}$$

which can be written using the ACF $r(.,.)$ as:

$$r(s,t) - \sum_{(m,n)\in\mathcal{V}} a(m,n)r(s-m,t-n) = 0, \forall (s,t)\in\mathcal{V} \quad (3)$$

The parameters $a(m,n)$ of the AR model are obtained by solving the set of linear equations (3) and the noise variance is computed using:

$$\sigma_w^2 = E\big[w(i,j)^2\big] = E\big[w(i,j)y(i,j)\big] = r(0,0) - \sum_{(m,n)\in\mathcal{V}} a(m,n)r(m,n) \quad (4)$$

Thus, if the ACF is known, all the AR parameters can be derived.

2.2 AR synthesis of stationary furlike texture

Based on several input images that are generated by the oriented hypertexture model with different combinations of orientation and length, we have estimated the ACF $r(m,n)$. The choosen

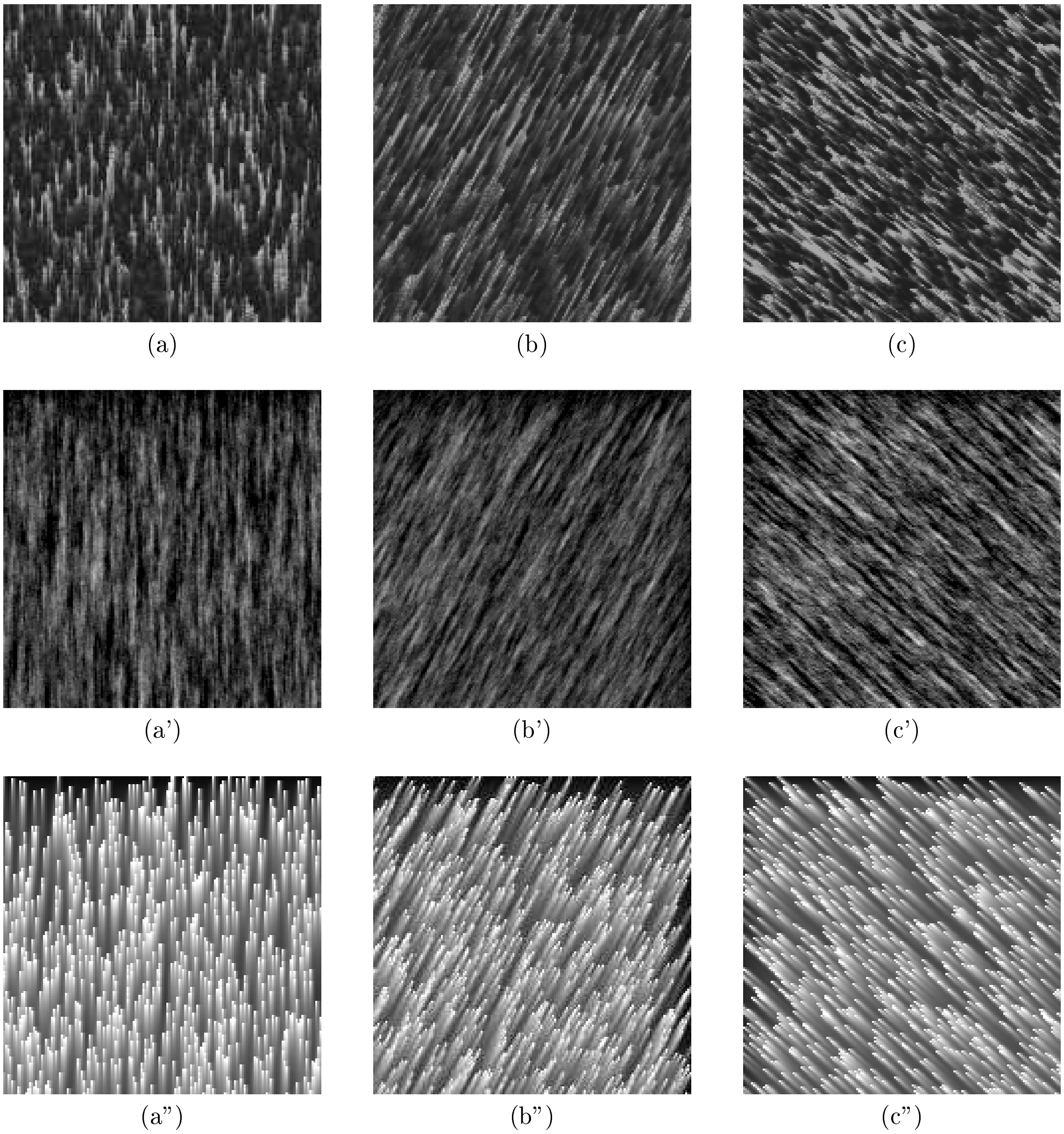


Figure 2: AR modelling, (a,b,c) original images. (a',b',c') and (a",b",c") corresponding images obtained by AR synthesis using the sizes (jx,jy)=(3,2), (4,2) and (4,2); with two different white noises at the input

neighbourhood allows a causal synthesis (all pixels in the prediction region are computed before the current pixel). Shape and size are defined by two parameters jx and jy illustrated in figure 1. Then we have computed the AR parameters using equations (3) and (4). The synthesis model in the stationary case (for one given combination of length and orientation) consists of a 2D linear filter defined by equation (1) with the corresponding AR parameters. The nature of the white noise used as filter input turns out to be decisive. An appropriate noise consists of a two dimensional random sequence of scaled impulses (of a fixed magnitude val), following a Bernoulli distribution with a given parameter p.

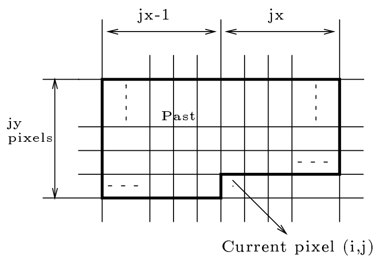


Figure 1: Neighbourhood $\mathcal{V}$

We present in figure 2 results obtained in the stationary case. The first row contains original images (a,b,c) generated with the oriented hypertexture model, while the second one presents images (a',b',c') obtained by an AR synthesis with a Gaussian white noise and parameters computed from the ACFs of images a, b and c. In the last row, we present images (a",b",c") generated like in the second row except that the input noise is a 2D sequence of impulses with parameters val=255 and p=0.033. This noise gives strong directional and relief aspects which are of a great interest for the expected applications.

To produce non stationary texture (with varying orientation and length), we could use a lookup table of AR parameters, indexed by length and orientation. This solution would require a huge table including an important and complete range of orientation and length values. Since AR parameters are computed using the ACF, as another solution, we propose to build a mathematical model that expresses the ACF with respect to orientation and length. Using such model, the AR parameters can be derived during the synthesis process according to current values of orientation and length.

2.3 Modelling of the autocorrelation function

Assuming that the length and orientation of the furlike texture are defined by two values l and α (see figure 3), we are looking for a function $\hat{r}(m,n,l,\alpha)$ that best fits the actual ACF $r(m,n,l,\alpha)$.

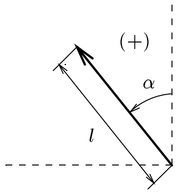


Figure 3: (α, l)

2.3.1 Texture Orientation

Since it is easier to find a good model when the ACF is separable, a simple solution is to find a model for a vertically oriented texture (α=0) and perform a rotation of the obtained model to deal with other orientations. A satisfying model is given by:

$$\hat{r}(m,n,l,0) = r_1(m,l)r_2(n,l) = C_l e^{-a_l m^2 - c_l|m|} e^{-b_l n^2 - d_l|n|}$$

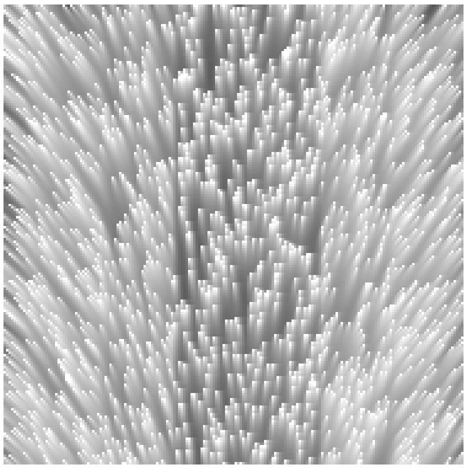
The ACF of a texture with an arbitrary orientation α (which ranges in $-\frac{\pi}{2}, \frac{\pi}{2}$) can be obtained by rotating $\hat{r}(m,n,l,0)$ at the same angle α. We then obtain the following general model:

$$\begin{aligned}\hat{r}(m,n,l,\alpha) = C_l e^{-(a_l\cos^2\alpha + b_l\sin^2\alpha)m^2} \\ e^{-(a_l\sin^2\alpha + b_l\cos^2\alpha)n^2} \\ e^{-2\cos\alpha\sin\alpha(a_l-b_l)mn} \\ e^{-c_l|m\cos\alpha + n\sin\alpha|} \\ e^{-d_l|n\cos\alpha - m\sin\alpha|}\end{aligned}$$

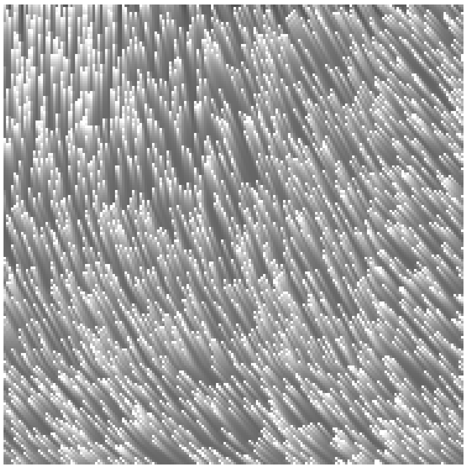
In this way, assuming that the set of ACF parameters $\{C_l, a_l, b_l, c_l, d_l\}$ are known, we have obtained an analytical expression of the ACF with respect to the orientation.

2.3.2 Texture Length

For the length parameter, we have opted for the use of a lookup table. Indeed, for different values of l, the set $\{C_l, a_l, b_l, c_l, d_l\}$ is computed from a sample of the ACF of an input texture of length l which is vertically oriented. This calculation is based on a general least mean squares algorithm



(a)



(b)

Figure 4: Non stationary AR synthesis, images (a,b) are obtained with (jx,jy)=(3,2), (4,3); val=255 and p=0.049, 0.066

to fit the sampled data to the model. Thus, a lookup table indexed by l that contains at each entry the values of $\{C_l, a_l, b_l, c_l, d_l\}$ is built. A furlike texture generator based on this table and on the ACF model is implemented.

2.3.3 Results

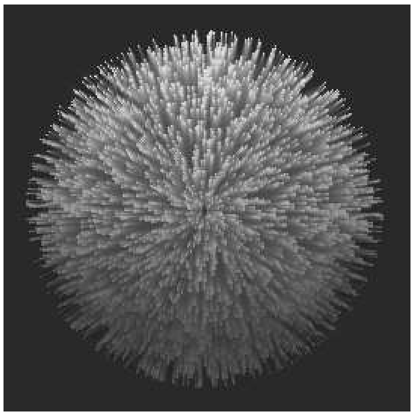
We show in figure 4 two images obtained by a non stationary AR synthesis. The AR parameters changing during the synthesis according to the variation of orientation and length are computed using the above described model of ACF and the consultation of the lookup table. In image (a), both length and orientation are varying, when only orientation changes over image (b).

3 Fur generation on 3D objects

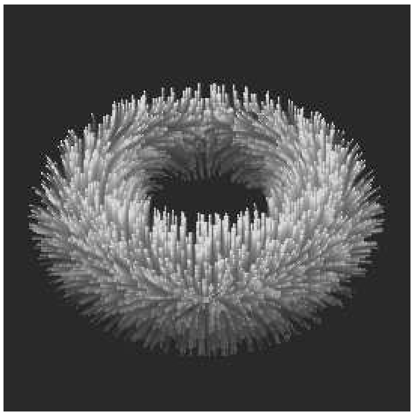
We show here how we can produce quite satisfying fur appearance by a simple mapping of a 2D texture on 3D surfaces. The mapping procedure can be described as follows:

- Given an object defined by a 3D surface, we assume that a spatial orientation map on this surface is defined, i.e. for each point in object space (x, y, z), an orientation vector is known.
- A white noise image $w_{0,1}$ of the same size as the frame to be rendered is precomputed. It is generated as described in section 2.2, with a zero mean and a unit variance.
- The surface is rendered using a general ray tracing algorithm. The 2D furlike texture is dynamically synthesized during the rendering process. For each pixel (i, j) in screen space, we perform:
  - Computation of the visible object point. A normal vector to the surface is deduced with a colour value (this value can be constant over the surface or variable according to some colour mapping).
  - Using the given orientation map, the orientation vector at that point is computed and then projected into the screen space. The resulting 2D vector is converted to polar form (figure 3) (α, l). With l, we obtain $\{C_l, a_l, b_l, c_l, d_l\}$ from the lookup table. Using $\hat{r}(m, n, l, \alpha)$ in equations (3) and (4), the AR parameters $a(m, n, i, j)$ and σ_w^2 are estimated for given values of jx and jy (these two values are fixed over all the experiments).
  - At this stage, we can compute the current value of the 2D texture. We use equation (1) with the noise value $w(i, j)$ equals to $\sigma_w^2 w_{0,1}(i, j)$. The grey level value is normalised to the unit interval and then returned as the texture value $t(i, j)$.
  - The colour (RGB components and transparency) at the current object point is scaled with $t(i, j)$. The new colour is used for all the next illumination calculations in the ray tracing process.

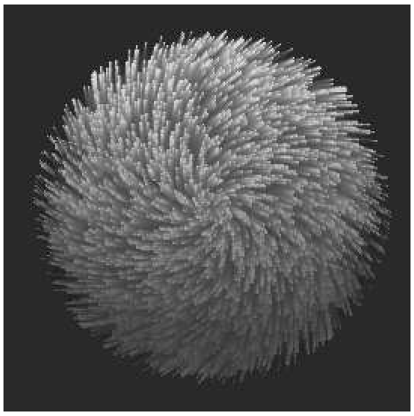
In figure 5, we illustrate the mapping procedure. In case of torus and first sphere, the orientation vector is defined as the normal vector to the surface at each point. For the second sphere, a



Sphere



Torus



Brushed sphere

Figure 5: Mapping of the furlike texture on 3D surfaces to simulate fur appearance

brush action is simulated: the orientation vector is obtained by rotating the normal vector around the z axis at a constant angle. For all images, we used $(jx, jy) = (5, 2)$, $val = 255$ and $p = 0.066$.

4 Conclusion

In this work, we are interested in realistic furlike texture generation. We have proposed an alternative to direct 3D synthesis techniques which give realistic results but are computationally intensive. The suggested solution is a 2D texture model based on the modelling of the ACF of furlike texture and on a 2D AR synthesis. We show that with an appropriate choice of the input white noise, the use of an AR model gives very interesting results even with very small sizes of neighbourhood. This makes the technique quite fast. Indeed, to obtain the texture value at each pixel, some simple operations have to be performed:

- Read the ACF parameters $\{C_l, a_l, b_l, c_l, d_l\}$ from the lookup table using the current value of l.
- Compute $\hat{r}(s-m, t-n, l, \alpha)$ for the N^2 coordinate pairs $\{(s,t),(m,n)\}$ of the neighbourhood $\mathcal{V}$. N is the number of pixels in $\mathcal{V}$. The choosen shape of $\mathcal{V}$ implies $N = (2jxjy - jx - jy)$ (figure 1). The number of these computations is reduced by half (the ACF is symetric). Furthermore, since the ACF is stationary, pairs corresponding to the same translation have the same ACF value which only needs to be computed once.
- Solve the set of linear equations (3) defined by a symetric matrix of size N^2. This gives the AR parameters $a(m,n,l,\alpha)$. σ_w^2 is deduced using equation (4).
- Perform the computation given in equation (1) to obtain the texture value.

The technique allows a 2D synthesis of images with a strong 3D directional appearance. This makes its application to realistic fur generation on

3D surfaces straightforward. In addition, the underlying model permits an easy control of orientation and could be used to represent vector fields. For the future, several aspects could be investigated such as the modelling of ACF with respect to length and a deep study of the influence of val and p on the visual aspects obtained. A parallelisation of the synthesis process is also possible. Indeed, all the operations listed above, except the last one, could be performed at the same time for all the pixels. Thus, the technique could be used in an interactive system. Finally, we are planning to develop the application of the texture model as a kernel filter for vector field visualisation and we expect to apply the technique to various data.

References

[Dalde93a] Agnes Daldegan and Nadia Magnenat Thalmann: Creating virtual fur and hair styles for synthetic actors, *Communicating with virtual Worlds, 11th International Conference*, Lausanne, pp.358-370, June 1993

[Dalde93b] Agnes Daldegan and Nadia Magnenat Thalmann and Tsuneya Kurihara and Daniel Thalmann: An integrated system for modeling, animating and rendering hair, *Eurographics '93*, Vol.12, No.3, pp.211-221, 1993

[Kajiy89] James T. Kajiya and Timothy L. Kay: Rendering Fur with Three Dimensional Textures, *Computer Graphics*, Vol.23, No.3, pp.271-280, July 1989

[Perli89] K. Perlin and Eric M. Hoffert: Hypertexture, *Computer Graphics*, Vol.23, No.3, pp.253-262, July 1989