PARAMETRIC TEXTURE SYNTHESIS FOR FILLING HOLES IN PICTURES

Anil Kokaram

Electronic and Electrical Engineering Department,
Trinity College, University of Dublin,
Ireland
email: anil.kokaram@tcd.ie

ABSTRACT

This paper presents a framework for "Filling In" missing gaps in images and particularly patches with texture. The underlying idea is to construct a parametric model of the p.d.f. of the texture to be re-synthesised and then draw samples from that p.d.f. to create the resulting reconstruction. A Bayesian approach is used to re-pose 2D Autoregressive Models as generative models for texture (using the Gibbs sampler) given surrounding boundary conditions. A fast implementation is presented that iterates between pixelwise updates and blockwise parametric model estimation. The novel ideas in this paper are joint parameter estimation and fast, efficient texture reconstruction using linear models.

Keywords: Texture synthesis, Image reconstruction, Image restoration, Filling in, Gibbs sampling, Bayesian inference, 2D Autoregressive models.

1. INTRODUCTION

The problem of filling missing gaps in pictures is a well known one in the archive film and video restoration field (See for instance Kokaram [1]). The problem of missing data also occurs in still photographs but here reconstruction of the missing image material can only rely on spatial inference. In such a case, one could consider the problem to be that of texture synthesis in that it is required to 'fill in' the missing region with some plausible texture. The film line scratch problem is amenable to such 'filling in' algorithms [1].

'Filling in' has been discussed in early literature under different names e.g. Strohmer in 1997 [2] presented a frequency domain method for "Reconstruction of irregularly sampled images", Hirani [3] in 1996 presented similar work on "Fast interactive noise removal". Recently, this problem has been receiving greater attention in the literature e.g. Efros et al[4], Bertalmio et al [5]. The process is also called 'Inpainting'. This problem of 'filling in' is related to the problem of texture synthesis or resynthesis [4, 6]. In the general case of texture synthesis, a typical problem is to take a small sample of texture and use that to cover a much larger region in another image. To relate and define the two genres of work more clearly it is possible to define 'Inpainting' or 'Filling In' as a texture generation process with boundary conditions[1] and the process

---

This work was funded in part by the AURORA and BRAVA EU Projects during 1998-2002

[1]The interpolated or filled in region should 'blend smoothly' with the rest of the image.

of Texture synthesis or resynthesis is a texture generation process without boundary conditions.

The texture generation algorithms that have been proposed thus far could be considered to fall into two classes 'parametric' and 'non-parametric'. The parametric class tends to contain deterministic algorithms e.g. Strohmer [2], Hirani et al [3], while the non-parametric class contains stochastic algorithms [7].

This paper introduces the use of 2D Autoregressive models for 'filling in' and texture generation. As such it is a **parametric** process, but by posing the problem as that of estimating the p.d.f. for the underlying texture, it is able to generate the texture **stochastically**. This avoids problems of 'flat' or 'over blurred' textural reconstruction. Typically, filling in algorithms tend to be computationally demanding, but because of the structure of this framework, a fast algorithm can be developed. The use of MCMC (the Gibbs sampler) is explored to enable a computationally trivial iterative process for interpolation.

The following sections first introduce the generative image model, then illustrate how it could be used for texture generation and 'filling in' by incorporation into a Bayesian framework.

2. THE GENERATIVE MODEL

It is assumed that the underlying image generation process is 2D autoregressive (AR). Thus a pixel in the original, clean image, $I(x)$, where x is a spatial position vector, can be modelled as follows

$$I(x) = \sum_{k=1}^{P} a_k(x)I(x + q_k) + e(x) \qquad (1)$$

A particular pixel at site x is therefore predicted by a linear combination of pixels in the current frame plus an added excitation or residual error $e(x) \sim N(0, \sigma_e^2(x))$. The P coefficients of the model are denoted a_k for $k = 1 \ldots P$. The pixels used in the prediction are called the support or neighbourhood of the model and are mapped by the P spatial offset vectors q_k. Note that the model parameters are non-stationary (they are a function of position x) in order to capture the statistical non-stationarity in interesting images. The model parameters are σ_e^2, and a (arranged in a vector of P elements).

3. JOINT SOLUTION AND RELATIONSHIPS WITH PREVIOUS WORK

The problem is posed as that of 'Filling in' missing texture in gaps in the image. This means estimating both the model parameters and the missing data itself. Consider a block of pixels containing the missing data as shown in Figure 1. Let i_k contain the known pixels, and i_u the pixels to be estimated (the inner 2×2 block in Figure 1) , or *synthesised*. Let θ represent a vector of model parameters $[a, \sigma_e^2]$.

Ultimately, we would wish to manipulate $p(i_u, \theta|i_k)$ to select an interpolant (estimate of i_u), $\hat{i}_u$ AND the model parameters θ, *at the same time*. Assuming that i_u, θ are independent variables, the required probability expression can be written as

$$p(i_u, \theta|i_k) \propto p(i_k|\theta, i_u)p(i_u)p(\theta) \qquad (2)$$

Connections with previous work can be established at this point. Efros et al employs the *empirical measurement* of the distribution $p(i_u|i_k)$ using surrounding image data. Here we use instead a parametric form for the distribution via the 2D AR model, hence we do not require as much image material to 'learn' the necessary distribution. Strohmer and Hirani use an estimate of spectral content to generate missing texture. The 2D AR model can be considered to *summarise* all spectral information *without* modelling each spectral coefficient. Additionally, we do not need a patch template (Hirani), because model estimation is built in to the problem.

To proceed, specific expressions are required for the likelihood and priors in the equation 2 above. A uniform prior is adopted for $p(a)$ and $p(i_u)$, hence $p(\theta) \propto p(\sigma_e^2)$, and $p(i_u)$ is a constant. For the residual variance, a Jefferies prior is used: $p(\sigma_e^2) \propto 1/\sigma_e^2$. The Likeihood is examined next.

3.1. The Likelihood

Consider a block of $B \times B$ pixels (containing both known and unknown sites). Within this block $e(x) \sim \mathcal{N}(0, \sigma_e^2)$ for a subset of pixel sites chosen such that the prediction support for each site does not extend outside the block. See figure 1 for $B = 4$, and a 3 tap causal AR model. By raster scanning prediction errors or residuals into e it is always possible to express this residual error in different forms. $e = Ai = A_k i_k + A_u i_u$. Therefore, the likelihood can be written in a few alternate forms since $p(i_k|\cdot) \propto p(e)$. The form needed for 'filling in' is as follows

$$p(i|a(x), \sigma_e^2) \propto \exp\left(-\frac{[A_k i_k + A_u i_u]^T[A_k i_k + A_u i_u]}{2\sigma_e^2}\right) \qquad (3)$$

4. INTERPOLATION/FILLING IN/TEXTURE SYNTHESIS

To solve for i_u, θ in the expression 2, we use the Gibbs sampler. Starting from an initial parameter estimate (which may be 0), the sampler proceeds iteratively drawing random samples for each variable in turn. Here the sampler is altered to sample for *both* a, σ_e^2 at the same time.

$$(a, \sigma_e^2) \sim p(\theta|i); \quad i_u \sim p(i_u|\theta, i_k) \qquad (4)$$
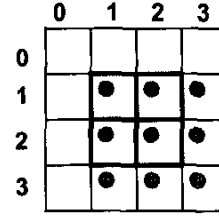

Fig. 1. A 4×4 block of pixels showing a missing 2×2 inner block. Prediction errors are observed at sites marked with •, and these errors are raster scanned to create the vector e.

These conditional expressions may be derived by integrating the posterior distribution [8] in equation 2. In addition, the first joint sample can be achieved using composition sampling which requires first drawing a sample for σ_e^2 then using that sample, drawing a sample for a as follows.

$$p(\sigma_e^2|i) = IG((N - P)/2, E(\hat{a}, i)/2) \qquad (5)$$

$$p(a|i, \sigma_e^2) = \mathcal{N}_P(\hat{a}, \sigma_e^2(I^T I)^{-1}) \qquad (6)$$

where $E(\cdot)$ is the sum squared prediction error in a block using the Least Squares estimate of the coefficients â.

The conditional for i_u can be derived by completing the square [8, 1] in the argument of the exponent of that equation 3 and yields

$$p(i_u|i_k, a, \sigma_e^2) = \mathcal{N}(\hat{i}_u, \sigma_e^2[A_u^T A_u]^{-1}) \qquad (7)$$

$$\text{where } \hat{i}_u = [A_u^T A_u]^{-1} A_u^T A_k i_k \qquad (8)$$

Hence the MAP estimate for the missing data is $\hat{i}_u$. In this case it is identical to the Least Squares interpolant. MAP reconstructions tend to look 'faded' when used in the presence of heavy texture. Instead, we are interested here in using the samples from the Gibbs sampler. This sample is used as the interpolant to fill in the missing region. The recipe for drawing a sample from a multidimensional Gaussian distribution is found in [8, 1].

Direct solution for $\hat{i}_u$ is computationally demanding because of the need to invert $[A_u^T A_u]$. For instance, a missing gap of size 8×8 pixels and a simple 3 tap causal 2D AR model, requires of the order of $64^3 + 10^2 \times 20$ operations in total. To improve efficiency, a fast algorithm is discussed next.

4.1. A pixel update algorithm

The problem is that of drawing samples for the pixel data that constitutes i_u. By using the Gibbs sampler this draw can be decomposed into draws for each pixel iteratively as follows.

$$i_{u1} \sim p(i_{u1}|i_{-u1}, \theta); \quad i_{u2} \sim p(i_{u2}|i_{-u2}, \theta); \quad \cdots$$

where i_{u1} is the first missing pixel in a scan of the data, i_{u2} the second and so on. i_{-u1} represents the current state of all the image data around the pixel to be treated regardless of whether it contains missing pixels or not. This sequence of operations converges to samples from $p(i_u|\theta, i_k)$.

Since at each iteration, only one pixel is missing, $\mathbf{i}_u$ becomes a single pixel sample i_u. Each conditional is now a *univariate* Gaussian, and so $\mathbf{A}_u \rightarrow \mathbf{a}_u$ etc. Therefore $\mathbf{A}_u^T \mathbf{A}_u$ becomes a scalar $\sum_{k=0}^{P} a_k^2$ ($a_0 = 1.0$), making inversion trivial.

5. THE FINAL ALGORITHM

To adapt to spatially varying image statistics the image is divided into blocks of fixed size and the same value for θ is used throughout each block. Therefore, the model parameter estimation process is conducted on a *block* basis and the filling process conducted on a *pixel basis*. The starting estimate for the missing data is just the mean of the pixels surrounding the 'hole'.

1. Select a block size $N \times N$, for θ estimation. Select a model order and causality. There are P pixels in the support used by the model.

2. BLOCK BASED MODEL ESTIMATION

   (a) For each block DO:

   (b) Calculate the least squared estimate for $\mathbf{a}$ using the Normal equations. Calculate the prediction error in the block using the estimated $\hat{\mathbf{a}}$.

   (c) Using these values, draw a sample for the model coefficients $\mathbf{a}_s$ and prediction variance σ_e^2 using equations 5 (Inverted Gamma), and 6 (Gaussian) *in that order*.

3. PIXEL BASED FILLING IN

   (a) At each missing pixel site DO:

   (b) Lookup the model coefficients for the corresponding block

   (c) Calculate $\hat{i}_u$ according to equation 8 using the previously generated samples for θ for the given block.

   (d) Generate a sample of the missing data at that site using $\hat{i}_u + r \times \sqrt{\sum_{k=0}^{P} a_k^2}$ where $r \sim N(0,1)$.

4. Repeat steps 2,3 for L iterations.

In practice a checkerboard image scan for filling in gives better convergence than a consecutive raster scan pattern.

6. PICTURES AND PERFORMANCE

Figure 2 shows a textured image (of size 220 rows and 310 columns) upon which three patches of blue have been drawn to simulate missing areas. The top left shows the degraded image, the top middle shows the resulting reconstruction when the algorithm above is applied and the 50th sample is used as the filled in texture. The top left is the original texture. The synthesised texture is shown to be extremely well matched to the rest of the image. A block size of 32×32 was used with a semi-causal model of 10 taps. The colour reconstruction was achieved by processing Y,U,V colour planes separately. There are about 4600 missing pixels in this image, and the (inefficiently coded) algorithm took just less than 1 sec per iteration (including both model estimation and pixel filling over the whole image) to execute on a PIII 300 MHz PC. The reconstruction is virtually indistinguishable from the original image.

The last two rows show examples with varied picture content. The middle row shows an example of *picture editing* in which the chain supporting the bicycle is removed by highlighting that area as missing. In all cases the results are convincing.

See www.mee.tcd.ie/~ack/icip2002 for more complete results including a visual illustration of convergence of the algorithm. Typically 10 iterations show good results for highly textured images with gap sizes as in the first example shown in Figure 2.

7. FINAL COMMENTS

This paper has presented a quantitative parametric framework for texture resynthesis or filling-in. It has been shown that the 2D AR model can be articulated within this framework to yield useful textural reconstruction. A fast algorithm has been examined which removes much of the computational distress caused by using matrix algebra in deriving the process.

It would be useful to allow this algorithm to adapt to the image details more flexibly than on a block basis. This would imply the solution of a simultaneous segmentation/estimation problem. Current work revolves around examining multiscale implementations in an attempt to increase speed and decrease the sensitivity to block sizes.

The scheme presented in this paper holds much future scope for generalisation or combination of the ideas presented by Strohmer, Hirani et al, Efros et al and Bertalmio et al; while still retaining the notion of 'stochastic texture sampling'.

8. REFERENCES

[1] A. C. Kokaram, *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, Springer Verlag, ISBN 3-540-76040-7, 1998.

[2] T. Strohmer, "Computationally attractive reconstruction of bandlimited images from irregular samples," *IEEE Image Processing*, pp. 540–548, April 1997.

[3] Anil N. Hirani and Takashi Totsuka, "Combining frequency and spatial domain information for fast interactive image noise removal," in *Proceedings SIGGRAPH*, 1996, pp. 269–276.

[4] Alexei A. Efros and William T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings SIGGRAPH*, 2001, pp. 341–346.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings SIGGRAPH*, 2000.

[6] David J. Heeger and James R. Bergen, "Pyramid-based texture analysis/synthesis," in *IEEE International Conference on Image Processing*, October 1995, pp. 648–651.

[7] Alexei A. Efros and Thomas K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, September 1999, vol. 2, pp. 1033–1038.

[8] J. J. O Ruanaidh and W. J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*, Springer–Verlag, Springer Series in Statistics and Computing, 1996.
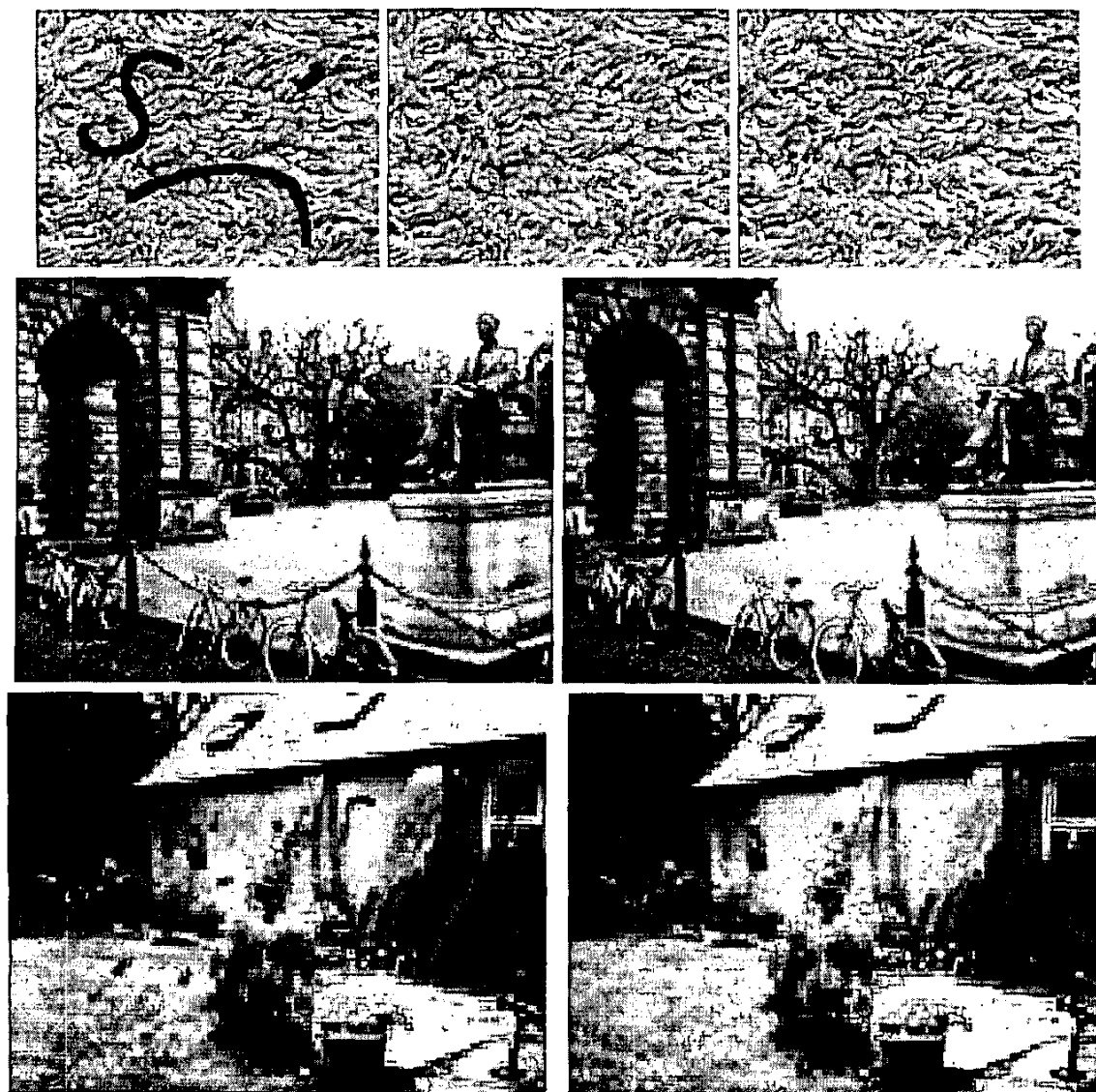
Fig. 2. Top row: Performance of the algorithm in removing the blue swathes in the top left image Middle: reconstruction, Right: Original. There is very little visible difference between the reconstruction and the original image. The other two rows show reconstruction on more typical imagery. **Note that the reproduced resolution is not as good as in the actual processed data because of the filesize limit required by ICIP.** See www.mee.tcd.ie/~ack/icip2002 for more complete results.