APPROXIMATION OF CONDITIONAL DENSITY OF MARKOV RANDOM FIELD AND ITS APPLICATION TO TEXTURE SYNTHESIS

Arnab Sinha

Dept. of Electrical Engineering
Indian Institute of Technology,kanpur, India,
Email: arnab@iitk.ac.in

Sumana Gupta

Dept. of Electrical Engineering
Indian Institute of Technology,kanpur, India,
Email: sumana@iitk.ac.in

ABSTRACT

Markov Random Field (MRF) based sampling method is popular for synthesizing natural textures. The main drawback of the synthesis procedure is the large computational complexity involved. In this paper, we propose an approximation of the conditional density description for the reduction of computational complexity required in sampling texture pixels from the conditional density. Assuming, $Y \in \Lambda$, and $X \in \Lambda^d$, we in this work studied the approximation of the conditional density function $P(Y|X)$ as $P(Y|\theta^t X)$, where $\theta \in \Re^d$, is a unit vector. We have also shown that the classical gradient based optimization method is not suitable for finding the solution of θ. We have estimated θ using Genetic algorithm. The perceptual (visual) similartiy and neighborhood similarity measures between the textures synthesized using the full conditional description and approximated description, are shown for validating the method developed.

Index Terms— Texture synthesis, MRF, Genetic Alrorithm, Quasi-Newton, approximation of conditional density

1. INTRODUCTION

Texture synthesis by MRF from a given sample of natural texture is known to be a time consuming task. The problem has been addressed in [1], and [2], through the reduction of the search space. But, if the dimension of the neighborhood (which is 360 for order 10 and in most cases requires order $\geq$ 10 for synthesizing natural textures) is not reduced, then the comparison of a huge neighborhood vector at each pixel site in the output synthesized texture, with neighborhood vectors at each pixel site from the input sample texture done several times. This is a computationally inefficient process as can be seen from figure (1). The exponential growth in the computational complexity with the increase in neighborhood order is a matter of concern. If we can approximate the conditional density through a reduced number of random variables, then it is possible to increase the speed of the synthesis process and other related approaches. The procedures, such as PCA, ICA and the nonlinear kernel versions of these concepts can not be applied in the straightforward manner here because our objective is to approximate the conditional density and not the joint density of the spatial data.

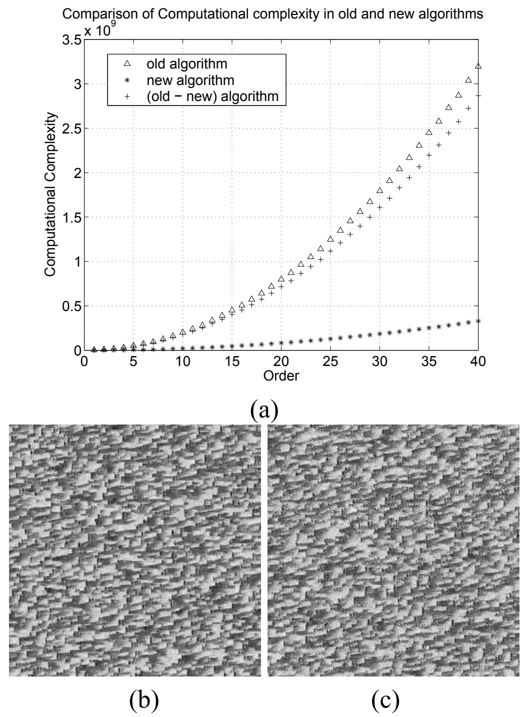


Fig. 1. (a) Explains the reduction of computational complexity of proposed algorithm with respect to old algorithms, In the second row: The textures synthesised with the (b) full conditional density, and (c) approximated conditional density with the minimized computational complexity, are compared, where original texture taken as D1 from the Brodatz album and MRF order $= 10$

In this paper we have tried to approximate the conditional density through a linear transformation of the neighborhood vector to a one dimensional random variable. This intuitive idea can be deduced from the fact that due to the Markovian nature, the neighborhood pixels are also dependent on each other and a linear transform could capture this dependency in a way such that the conditional density has not been compromised. Our contribution lies in the design and analysis of the approximation method for the specific application in texture synthesis through nonparametric Markov random field.

In section 2 we review the preliminary concepts required for further discussion. In section 3 we discuss the approximation strategy for the specific problem. Finally in section 4 we discuss the results obtained through the proposed algorithm, and finally conclude the paper in section 5.

2. MRF PRELIMINARIES IN TEXTURE SYNTHESIS

2.1. Nonparametric MRF preliminaries

MRF models have been used for different applications in different branches of science and engineering. In the present case, description of MRF is taken from the view point of lattice models. The lattice, X, is a collection of random variables (r.v. henceforth) at the sites, $s = \{i, j\} \in S$, where, $i, j = 0, 1, \ldots, M-1$. The random variables are described as $X_s \in \Lambda$, i.e., they belong to the same state space. The MRF assumption implies

$p\left(X_s = x_s | X_{(s)}\right) = p\left(X_s = x_s | X_s = \{x_r; r \in \aleph_s\}\right)$

, which describes the fact that given a neighbor set, $\aleph_s$, the r.v. at s is independent of all other sites, $(s) = S - s$ and this conditional probability is termed as local conditional pdf (LCPDF). The neighborhood system is defined with the help of two axioms, $s \notin \aleph_s$, and, $s \in \aleph_r \Leftrightarrow r \in \aleph_s$. Let us now consider the nonparametric MRF model as described in [3] in the context of texture synthesis. Assume, S_{in} and S_{out} signify the input and output texture lattices respectively, and for simplicity we also assume that X_s denote the neighborhood set of the pixel r.v. Y_s. For each pixel in the output lattice, $s \in S_{out}$, we estimate the LCPDF $P(Y_s|X_s)$, for $Y_s = 1, 2, \ldots, L$ from the data $\{X_p, Y_p\}$ where, $p \in S_{in}$. This is generated from the input texture through Parzen window estimator using the product kernels as,

$P(y_s|X_s) = \frac{\sum_p \kappa_{h_y}(y_s - y_p) \prod_{j=1}^{d} \kappa_{h_j}(X_{s_j} - X_{p_j})}{\sum_{p \in S_{in}} \prod_{j=1}^{d} \kappa_{h_j}(X_{s_j} - X_{p_j})}$

Here, $\kappa_h(Z) = \exp\left(-Z^2/2h^2\right)/(2\pi)^{1/2}$ is the Gaussian kernel function, $Z \in \Re$, and $h_z = \sigma_z\left(4/(n(2d+1))\right)^{1/(d+4)}$, is the bandwidth as described in [3], $\forall p \in S_{in}$, and $\forall s \in S_{out}$. In the following a brief description of the sampling process from the nonparametric conditional density, as described in [3], is given. Choose a new y_s, by sampling the estimated LCPDF, through either Gibbs sampler or ICM algorithm. We will first describe a simple algorithm to increase the speed of this sampling algorithm, that has been actually used by Rupert Paget in [4]. Generate a vector W according to the following equation, and then sample y_s, $s \in S_{out}$, from this constructed vector according to a uniform distribution.

$W = \{y_k; d(X_i, X_k) \leq d(X_i, X_p), \forall p \in S_{in}\}$

In the algorithm described in [4], one has to calculate only the distance between two neighborhood vectors from input and output textures and then sampling the output pixel value from the generated vector W. The distance calculation between two high-dimensional (generally more than 100 and can go upto 1000 or more), neighborhood vectors X_p and X_s, is extremely time consuming.

3. APPROXIMATION OF CONDITIONAL DENSITY

The idea of approximating conditional density is conceived from [5]. In [5], authors described the approximation through cumulative distribution function. In this paper, we have modified the approximation through density functions, according to the requirement of texture synthesis.

Let Θ be a set of d-variate unit vectors θ, $f(X)$ the density function of $X \in \Lambda^d$, where Λ is the set of gray levels (here, neighborhood vector). $F_{Y|\theta^T X}(Y|Z)$ is the conditional distribution function of Y given Z, and $f(Y|Z)$ is the conditional density function of Y, given $Z = \theta^T X$. Given subsets A and B of d-dimensional space and of the real line, respectively, define

$\pi_\theta(A, B) = \int_A F_{Y|\theta^T X}(B|\theta^T x) f(x) dx$

$\pi(A, B) = P(X \in A, Y \in B)$

The cost function or the objective function defined as in [5], is,

$S_1(\theta) = \int\int \{\widehat{\pi}_\theta(A_\alpha, B_\beta) - \widehat{\pi}(A_\alpha, B_\beta)\}^2 w(\alpha, \beta) d\alpha d\beta$

where, w is a weight function and the integral is taken over a parameterization (α, β) of (A, B), and $\widehat{\pi}_\theta$ and $\widehat{\pi}$ are the nonparametric estimators of π_θ and π respectively. Since, we are only interested in sampling the LCPDF, we have used the second definition of the cost function, as described in equation (1). This cost function eq. (1) is compuationally less complex compared to the earlier cost function and it is dependent upon the conditional density functions. In the next subsections we describe an intuitive procedure for the estimation of θ, given the sets Y_j and X_j, where, $j = 1, \ldots, n$, and $n = |S_{in}|$.

$$S_2(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{256} \left\{p(y_j|X_i) - p(y_j|\theta^T X_i)\right\}^2 \qquad (1)$$

3.1. Problem with classical optimization methodology

In classical optimization methodology we need to assume some properties of the set of variables and the corresponding objective function. The set, Θ, has to be a convex set, i.e., if θ_1 and θ_2 both belongs to Θ, it implies that, $\{\alpha\theta_1 + (1-\alpha)\theta_2\} \in \Theta$. Again, from the definition of the Θ, we can say that, $||\theta|| = 1 \leftrightarrow \theta \in \Theta$. But, it can be shown that, $||\alpha\theta_1 + (1-\alpha)\theta_2|| \neq 1$, which implies that, $\{\alpha\theta_1 + (1-\alpha)\theta_2\} \notin \Theta$. Therefore, Θ is not a convex set. Let us assume,

$A'_{p_k} = \sum_{m=1}^{n} C_{ip_k m}\widehat{C}_{im}$, for $k = 1, 2$,

$A' = \sum_{m=1}^{n} D_{jm}\widehat{C}_{im}$, $A'' = \sum_{m=1}^{n} \widehat{C}_{im}$,

$A''_{p_k} = \sum_{m=1}^{n} D_{jm} C_{ip_k m}\widehat{C}_{im}$, for $k = 1, 2$,

$A'_{p_1 p_2} = \sum_{m=1}^{n} D_{jm} C_{ip_1 m} C_{ip_2 m}\widehat{C}_{im}$

$A''_{p_1 p_2} = \sum_{m=1}^{n} C_{ip_1 m} C_{ip_2 m}\widehat{C}_{im}$

$\widehat{C}_{im} = \exp\left\{-\sum_{k=1}^{d} \theta_k^2 C_{ikm}\right\}$, $C_{ikm} = \frac{(x_{ik} - x_{mk})^2}{2h^2}$,

$C = \frac{2\theta_{p_1}\theta_{p_2}}{\pi h^2 (A^{''})^4}$, $D_{jm} = \exp\left\{-\frac{(y_j - y_m)^2}{2h^2}\right\}$, and
$D_1 = \left[3A^{'}_{p_2}A^{'}_{p_1}(A^{'})^2 + A^{''}_{p_2}A^{''}_{p_1}(A^{''})^2\right]$
$D_2 = \left[2A^{'}_{p_1}A^{''}_{p_2} + 2A^{''}_{p_1}A^{'}_{p_2}\right]$, $D_3 = \left[A^{''}_{p_1 p_2}A^{'} - A^{'}_{p_1 p_2}A^{''}\right]$
$D_4 = \frac{1}{2\theta_p}\left[-(A^{''})^2 A^{'}_p A^1 + (A^{''})^3 A^{''}_p\right]$

Therefore, the Hessian matrix for the cost function in eq. (1) can be written as,

$$H_{\{p_1,p_2;p_1\neq p_2\}} = CD_1 - CA^{'}A^{''}(D_2 + D_3) \quad (2)$$

$$H_{\{p_1,p_2;p_1=p_2=p\}} = CD_1 - CA^{'}A^{''}(D_2 + D_3) - CD_4 \quad (3)$$

Now, it can be shown that if the following conditions are satisfied then the Hessian matrix will be rank deficient. The first condition is that the random variables at the neighborehood sites become linearly dependent, and the second is that the probability of the random vector X_i is approximately zero, or, $A^{''}A^{''}_p = A^{'}_p A^{'}$, or, $\theta_p = 0$. As the conditions described above are most likely to satisfy, therefore, we have to think of other alternatives than the classical gradient based optimization method.

3.2. Genetic Algorithm based methodology

In this work we redefine the cost function mainly because of two reasons. The first reason would be the reduction of the computational cost for evaluating the Parzen window estimation of the LCPDF's $P(Y_i|X_i)$ and $P(Y_i|\theta^T X_i)$. Since we use the algorithm in [4], we have to modify our cost function according to it. In genetic algorithms one only needs to have some performance evaluation function between two samples of θ. Hence, we can easily modify the cost function according to the current requirement. In case of texture synthesis we use the algorithm described in [4], where we only need to perform the comparison between two neighborhood vectors and choose the Y_j corresponding to the minimum distance. Therefore, the new definition of the cost function would be,

$$S_3(\theta) = \sum_{i=1}^{M}\left(W_i - \widehat{W}_i\right)^T\left(W_i - \widehat{W}_i\right) \quad (4)$$

where, $W_i = \{Y_k; d(X_i, X_k) \leq d(X_i, X_m), \forall m\}$, and, $\widehat{W}_i = \{Y_k; d(Z_i, Z_k) \leq d(Z_i, Z_m), \forall m\}$. In equation (4) we have considered M number of evaluations of the inner products, i.e., we have generated M random vectors $X_i, i = 1, \ldots, M$ for the test problem. The genetic algorithm approach that we have considered, is described in [6].

4. RESULTS

For the evaluation of equation (4) we have generated $M = (d \times 100)$ random vectors. In our implementation we have kept the parameters required in GA implementation, the same for each case. The number of population members that we have considered is 8 and the number of generations is 40. From our experience we can say that, considering the time factor and computational cost requirement, the parameters chosen are reasonably adequate as the value of the objective function in each case attain a zero value, i.e., the solutions are at global minimum points. The other parameters required for the algorithm are assigned their default values. The results are shown in figure (3).

The perceptual (visual) similarity between the synthesized textures with the FCD (Full Conditional Density) and ACD (Approximated Conditional Density), for all types of textures, give us the confidence about the validity of the proposed method. For performance evaluation we have calculated the neighborhood similarity measure between the original texture sample and the synthesized one. This is defined as, the sum of (d_{ij}; s.t., $j \in S_{in}$, $i \in S_{out}$, and $d_{ij} \leq d_{ik}, k \in S_{in}$), where, $d_{ij} = ||\aleph_i - \aleph_j||^2$, and $\aleph_i$ is the neighborhood of the i^{th} pixel in some lattice. Let us define a comparitive measure between the synthesized textures (one with FCD and another with ACD) with respect to the original texture as, $d_{orig,FCD} - d_{orig,ACD}$ normalized by $d_{orig,FCD}$, i.e., the change of neighborhood similarity measure with respect to the original texture when we consider the approximated conditional density. Figure (2) describes this measure for a number of different textures and order informations.From the figure it is clear that the change is very small, (of the order of 0.04).

The comparison between the computational complexities can be stated mathematically. If we consider M as the number of pixel sampling for each texture synthesis, Q as the number of pixels in the input texture sample, and K the order of the MRF. Then the computational complexity for the old ([4]) algorithm would be, $MK\log(Q)$, and for the proposed algorithm it is, $(MK + \log(Q)K + M)$. As, Q is fixed for a given input texture; M has to be atleast equal to the number of pixels in the output texture (to sample each pixel atleast once). If we take $Q = 128 \times 128$, $M = 256 \times 256$, and vary K, we obtain the plot shown in figure (1). From the comparison plots shown in the figure (1), it is clear that the reduction of the computational cost is large, with no significant perceptual error (figure (3)) or mathematical error (figure (2)), in the output.

5. CONCLUSION

We have proposed and analysed a new method for the approximation of conditional density through a transformation of the conditional random vector to a single random variable. The approach has been modified for the specific application in texture synthesis through the sampling of the conditional density. The results obtained show promise for the validation of the approach, though there is no theoretical justification of the approximation method at present. The estimation of the unit vector is a time consuming process and computationally expensive. But, once θ is estimated for a particular sample of texture, it can synthesise the texture a number of times, (say on a 3D surface) in a much more faster way compared to the conventional sampling approaches. In this paper we have taken another approximation into consideration, due to an important ingredient of the texture synthesis algorithm [4], that is local simulated annealing. The future direction of our research will be to find an algorithm for the incorporation of local simulated annealing within the approximation of conditional density, otherwise we will not able to get the reduction in computational complexity, in practice.

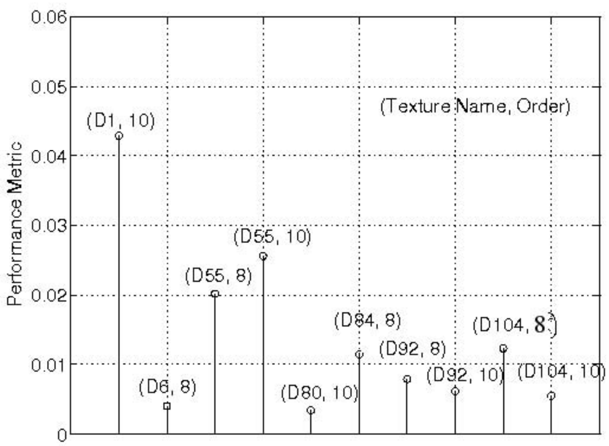
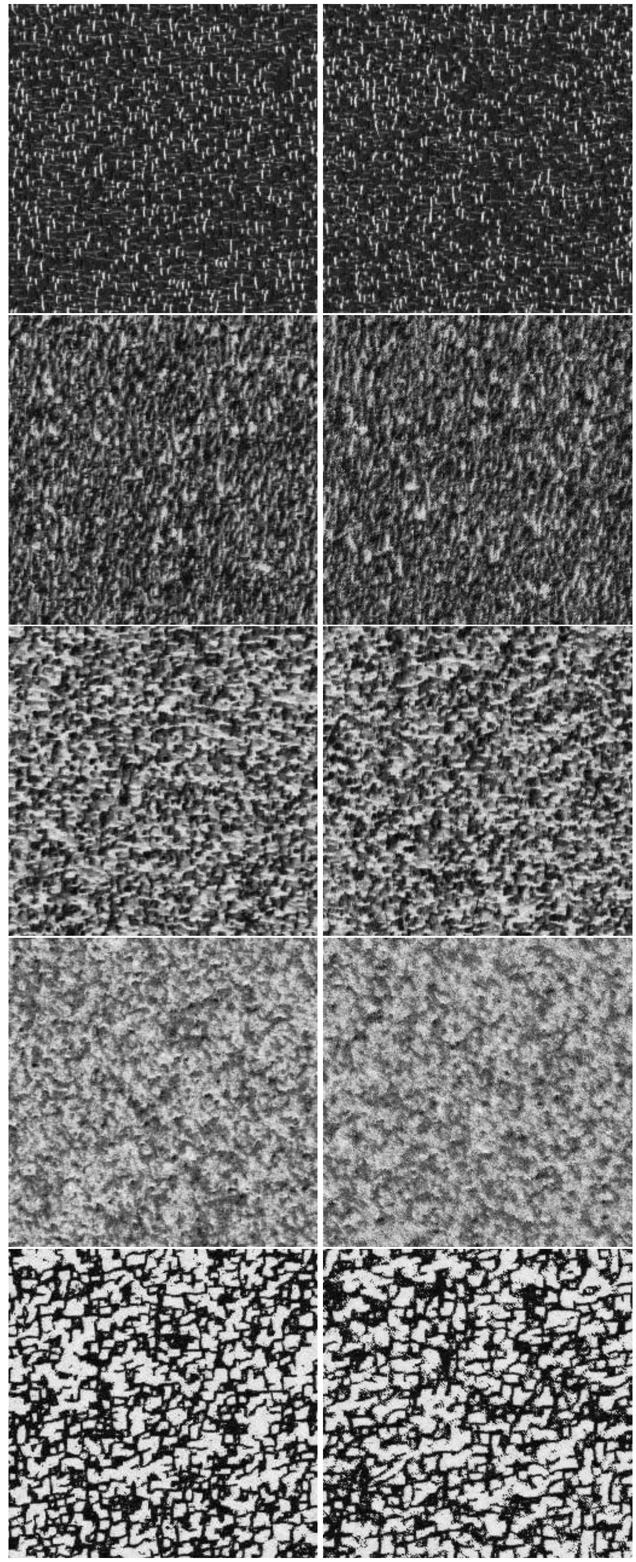


Fig. 2.



(a) (b)

Fig. 3. (a) The original synthesized texture (b) The approximated synthesized version with the estimated θ: From top to bottom the textures and the corresponding orders are (D6,8), (D80,12), (D84,8), (D92,10), and (D104,10)

6. REFERENCES

[1] L.Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of SIGGRAPH*, 2000.

[2] M. Ashikhmin, "Synthesizing natural textures," in *The proceedings of 2001 ACM Symposium on Interactive 3D Graphics*, 2001.

[3] R. Paget and I. D. Longstaff, "Texture synthesis via a noncausal nonparametric multiscale markov random field," *IEEE Transaction in Image Processing*, vol. 7, no. 6, pp. 925–931, June 1998.

[4] Rupert Paget, "Source code for texture synthesis," http://www.rupertpaget.com/.

[5] P. Hall and Q. Yao, "Approximating conditional distribution functions using dimension reduction," *The Annals of Statistics*, vol. 33, no. 3, pp. 1404–1421, 2005.

[6] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1995.