ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

APPLICATIONS OF TEXTURE ANALYSIS FOR ROI BASED STEGANOGRAPHY

N.V.Shibu,
Research Scholar, Dept of CSE
Sri Ramakrishna Institute of Technology, Coimbatore, India.

Dr.P.Malathi, Ph.d
Principal
Bharathiyar Institute of Engineering for women
Salem, India.

Ramya.S
P.G.Student, Dept of CSE
Sri Ramakrishna Institute of Technology, Coimbatore, India.

ABSTRACT

Texture synthesis is a process of generating a large user defined high quality texture image from a small input sample texture. Texture synthesis is one of the major application domains of the texture analysis. In most of the existing texture synthesis algorithm texture image is directly taken as input sample. Instead of direct texture sample, we would like to extract the texture from a digital image. For the extraction process, a digital image can be segmented into continuous region and homogeneous texture properties are extracted and can be used as input texture and patch based synthesis technique is used for synthesizing.

Keywords

Texture synthesis, Texture analysis and Patch based synthesis.

1. INTRODUCTION

Texture can be defined as a spatial arrangement of colours or intensities in an image. Texture analysis can be applied to various stages of processes such as pre-processing, feature extraction, classification. One of the problems in texture analysis is defining exactly what the texture is used for processing, i.e., texture can be either structural texture or statistical texture. In structural approach, a set of primitive telex's will be in some repeated or regular relationship. In the statistical approach, texture is a quantitative measure of intensity arrangements in a region.

Texture analysis applications are texture classification, texture segmentation, shape from texture and texture synthesis. Texture is classified as either supervised or unsupervised classification based on the prior knowledge about the texture properties. With the help of texture classification image data can be sorted into interpretable information and this information are readily used for a wide range of applications such as image retrieval, medical imaging and remote sensing. Texture segmentation is the process of partitioning an image into contiguous homogenous regions based on texture properties.

Texture synthesis is a process of generating a large user defined texture image from a small given sample texture input. Synthesizing textures can be done using different methods like tiling [10], patch based texture synthesis [2], [3], [4], pixel based texture synthesis [5], and stochastic texture synthesis. Pixel based and patch based are the two commonly used texture synthesis method. Pixel based is a successful algorithm where a texture is synthesized in a scan line order by finding and copying pixels with the similar local neighbourhood. In patch based texture synthesis method, a patch will be selected and that patch will be pasted in the output window. This is effective and faster method.

2. RELATED WORK

An improved example based texture synthesis technique is used by Guanren.et.al [2] where two improvements are made to the existing famous texture synthesis algorithm of WL00. The first proper size of the neighbourhood is selected automatically using self-similarity and those similarities are shown using histograms. The second process is to synthesis three pixels at a time rather than using one pixel as in traditional pixel based method. Three pixels are used because the synthesis

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

procedure can be done faster when comparing to original WL00 algorithm.

Most of the mobile phones have a high pixel digital camera which is being used as code detector. They use 2D bar codes for efficient and effective communication but meaningless images reduce the attractiveness of that image. To overcome that drawback Otori.et.al [3] uses texture as image code. Texture is used because texture synthesis can embed more information than the 2D image. Data coding is done based on local binary patterns. Data coding using local binary patterns first converts bit patterns of the information into coloured dotted patterns but the existing bar codes use densely arranged black and white dots. Least binary pattern is best high quality synthesis. It computes code by comparing the pixel value with its nearby pixels. Image calibration mechanism is used to detect all the corners of a squared region automatically and distortions are corrected through projective transformation. The limitation in this paper is that payload is quite smaller than the 2D bar code.

Shiguang.et.al. proposed a patch based synthesis algorithm to generate a plausible results using hybrids synthesis method [4]. When an input image is given, it is first passed into the analysis phase where an effective data structure is built and candidate sets are prepared by dividing the sample input into N X N overlapping RGB patches with same size. The overlapping regions between the candidates and their neighbours are focused. Here a PCA (Principle component Analysis) technique is used to reduce searching space. In synthesis phase nearest neighbour searching is performed and Poisson editing and deblurring is used for boundary optimization. This paper helps to improve speed of analysis and synthesis and the texture is also maintained to some degree. This work did not consider colour aspect of image when building candidate sets and in nearest neighbour searching and can be viewed as a serious limitation.

The texture synthesis process produces a large new image from an initial source image by selecting one pixel at a time. Efros.et.al used markov random field (MRF) [5] where it is assumed that the probability distribution of brightness values for a pixel in its spatial neighbourhood should be independent from rest of the image. Drawback of this paper is that some textures occasionally slip into a wrong part of the search space and will be degraded to be considered as a garbage value. This problem can be removed by choosing a bigger sample image. In this algorithm, a pixel is synthesized by selecting the neighbourhood pixels and treating one among the selected neighbour as the centre for new synthesized texture.

Dong.et.al [6] used patch based sampling algorithm which is fast and also produces a high quality real time texture synthesis. Patches are sampled according to the local conditional MRF density function. Square patches are chosen as image in and are pasted in the image out i.e. synthesised texture. To avoid mismatching between the patch boundaries we carefully choose the blocks based on patches that are already pasted in the output image. Here kd-tree is used as general approximate nearest neighbour (ANN) search technique. This algorithm is tested on main textures from regular to stochastic. This method is proven to be fast and their analysis and synthesis time is the lowest among this category.

The algorithm used by Bin.et.al [7] is a combination of jump map and patch based sampling texture synthesis method. The algorithm has two synthesis stages. First is the pre-processing stage, where patches are selected randomly from the input sample texture and jump lists are generated from that selected patch region and stored in the jump maps of patch. Second stage is the synthesis process in real time, where the selected patches are pasted in the output image with the help of jump maps. In the first stage the patches are transformed to new position and the edges are filtered to get smooth boundary. The performance can be improved by using PCA (principle components analysis) and kd-tree techniques for boundary comparison. Better performance can also be produced if number of candidate sets is limited in each patch.

Junyu.et.al [8] states that single SVM model is not enough to generate a synthesized output texture in the predicted manner. Thus the paper [8] introduced a new method by combining multiple seed blocks with SVM. First the sample texture is taken as input and features are classified and extracted to form a linear model which is used as training set for SVM. Those samples are again divided using cross validation. Each subsample is used as test data once and remaining subsamples are considered as training sets. This action is done iteratively and then output is produced by copying the upper left corner patch and pixel is calculated in each patch. Seed block is searched to check whether texture characteristic is maintained, after copying left corner patch then that patch is used as seed block for synthesizing the second patch. Image quilting technique is used to maintain the image quality. Advantage of this paper is that sample is not required during synthesis process and drawback is that computation cost is high because only one pixel is predicted by SVM model.

3. EXISTING METHOD

1. Initialize the source texture and its size is a factor of the size of kernel block (Kw x Kh).
2. Subdivide the source texture into no. of non-overlapped kernel blocks

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

3. Expand a kernel block with depth (Pd) at each side to produce source patch.

If kbi is located around boundary region ten boundary mirroring function is applied.

4. Generate candidate patches when synthesizing synthetic texture by employing a window and then travel the source texture (Sw x Sh). By shifting a pixel each time in a scan line order.

5. Generate an index table (Tpw x Tph) to record the location of source patch.

6. Produce a patch composition image by pasting source patches into a workbench.

Size of workbench= size of synthetic texture.

7. Apply the message oriented texture synthesis method.

(a) Compute mean square error (MSE) for all candidate patches of overlapped region between synthesized area and candidate patch.

(b) Rank cp according to MSE's.

(c) Select the candidate patch where its rank equals the decimal value of n-bit secret message.

8. Determine the capacity of message.

9. Recover the source texture and extract the message by generating index table and constructing candidate list at the receiver side.

4. PROPOSED METHOD

In most of the existing algorithms texture images are directly taken as input. Instead of direct texture input, we can extract texture from a digital image. In pre-processing, the digital image is taken as input and it is divided into non overlapping regions based on local texture properties. Texture properties are selected in each region and those texture properties are extracted and grouped together to form a texture image. Feature extraction can be done using various techniques such as first order histogram, co-occurrence matrix, multiscale feature, etc. Grouped properties or segmented region can be taken as input for synthesizing texture.

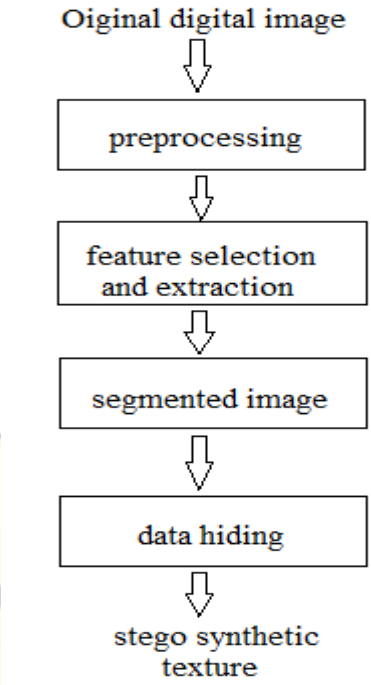

Fig.1 Block diagram of proposed system

Based on the prior knowledge about the number of textures in the images, texture segmentation can be classified into supervised and unsupervised. Patch based synthesis process method is for synthesizing texture, where a patch is selected from input texture and that patch is pasted into workbench. Size of workbench is equal to the user specified synthesized texture. Data can be hidden during synthesizing process. At last image quilting technique is used for smoothing and removal of artifacts produced.

5. CONCLUSION

Texture synthesis is a process of generating a large high quality user defined texture from a small input sample texture. Instead of taking input texture directly for synthesizing purpose, here texture is extracted from a digital image and then it is taken as input sample and patch based synthesis method is used for texture synthesis. In future pixel based synthesis method can be used with different digital images.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

6. REFERENCE

[1]Kuo-Chen Wu and Chung-Ming Wang.(2015), "Steganography Using Reversible Texture Synthesis", in IEEE Transactions On Image Processing, vol. 24, no. 1,.

[2] Guanren Huang, Mingyu Gao, Zhiwei He, Yuanyuan Liu, "An Improved Example-based Texture Synthesis Algorithm" in IEEE Second WRI Global Congress on Intelligent Systems 2010.

[3] H. Otori and S. Kuriyama, "Texture synthesis for mobile data communications," IEEE Comput. Graph.Appl., vol. 29, no. 6, pp. 74–81, Nov./Dec. 2009.

[4] Shiguang Liu, Jingting Wu, "Fast patch-based image hybrids synthesis" in 12th International Conference on Computer-Aided Design and Computer Graphics 2011.

[5] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in ICCV "99: Proceedings of the International Conference on Computer Vision-Volume 2. Washington, DC, USA: IEEE Computer Society, 1999, p. 1033.

[6] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Realtime texture synthesis by patch-based sampling," ACM Trans. Graph., vol. 20, no. 3, pp. 127–150, 2001.

[7] Bin Wang, Jun-Hai Yong&Jia-Guang Su (2004), "Real-Time Texture Synthesis with Patch Jump Maps", springer, Computational and Information Science, Volume 3314, pp. 1155-1160.

[8] Junyu Dong, Ran Wang & Xinghui Dong (2010), " Texture synthesis based on multiple seed-blocks and support vector machines", 3rd International Congress on Image and Signal Processing (CISP), Volume:6, pp.2861 – 2864.

[9] Christian Eisenacher, Sylvain Lefebvre, Marc Stamminger (2008), "Texture Synthesis from Photographs", The Eurographics Association and Blackwell Publishing Ltd., 27, Issue: 2, pp.419-428, 2008.

[10] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen, "Wang tiles for image and texture generation," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 287–294, 2003