Image Enhancement Techniques by Texture Synthesis

Ila Rathod[1], Nitin Pandya[2]
Mewar University[1]
Rajasthan,Chittorgarh, India[1]
Shankersinh Vaghela Bapu institiute Of Technology[2],
Gujarat Technological University, Gujarat.[2]
Ila_rathod08@yahoo.co.in[1], nitin.pandya@bapugkv.ac.in[2]

***Abstract*—Main rationale of image enhancement to improve some feature of image to make visibly better one. Here we introduce image should be enhancement using Different kind of techniques, some kind of features and also add proposal. Texture synthesis can be used to remove undesirable artifacts in Images such as scratches, wires, pops, or scrambled regions.**

We present a patch-based Texture Synthesis and Pixel Based Texture Synthesis for synthesizing textures from an input sample texture. Using Both of these methods noise is removed, add smooth edges, and clean the color image, Add Density more.

Keywords- Enhancement,Texture Synthesis, Pixel Based Texture Synthesis, Patch-based Texture Synthesis.

I INTRODUCTION

Texture is an everywhere understanding. We present a well again Enhancement program, that use interpolation algorithm to read in an image *z* and produce an image *z*2 of twice the size with improved factor. By this algorithm does not add any extra details of image.

Texture Synthesis:

Texture synthesis method starts from a sample small image and attempts produce a large non repetitive, a texture with a visual appearance similar to that sample. Texture synthesis techniques can be broadly categorized as either explicit or implicit [4, Chapter 2]; an explicit algorithm generates all the texture samples directly while an implicit algorithm answers a query about a particular sample without computing the whole texture.

The goal of texture synthesis can be stated as follows: Given a texture sample, synthesize a new texture that, when perceived by a human observer, appears to be generated by the same underlying process.

The major challenges are:

Modeling: How to estimate the texture generation process from a given finite texture sample. The success of modeling is determined by the visual fidelity of the synthesized textures with respect to the given samples.

Sampling: How to develop an efficient sampling procedure to produce new textures from a given model. The efficiency of the sampling procedure will directly determine the computational cost of texture generation [1].

Application:

(a) Distorted synthesis:

Using a random order of visit of the output pixels and modifying the shape of the current neighborhood according to the local distortion of the output image, the synthesis is produces images with both the texture information and the distortion one.

(b) User control:

Using a image painted by the user instead of the random Noised image, with few modifications of the algorithm of Ashikhmin or other method , it is possible to force the synthesis to collocate the different features of the texture in different places of the output image.

II. EXISTING METHODS

Tiling:

Using these method generate a large image from a sample image is to tile it. This means multiple copies of the sample are simply copied and pasted alongside. The effect is rarely suitable.[2].

Stochastic Texture Synthesis:

Texture synthesis can be useful in a variety of images Which need the replacement of large areas with stochastic textures. This technique, however, is useless for images that need the replacement of areas with structured texture. These algorithms perform well with stochastic textures only, otherwise they produce completely unsatisfactory. results as they ignore any kind of structure within the sample image.

Single purpose structured Texture Synthesis:

In this use a fix procedure to create an output image. They are limited to a single kind of structured texture. For example, a single purpose algorithm could produce high quality texture images of stonewalls; yet, it is very unlikely that the algorithm will produce any viable output if given a sample image that shows pebbles.

Pixel-Based Texture Synthesis:

A pixel-based method has a tendency to blur features or to Grow small-scale structures in synthesized textures.[3] Here Iteratively, choose a pixel next to one that is already filled in. In that find the patch from the original image most similar to the nearby filled in pixels. Here use D=distance. Then find the set of patches of distance (1+ε) D.[5]

Patch-Based Texture Synthesis:

These methods protect global structure by generating the texture on a per-patch basis, and then in most cases attempt to repair the patch overlap regions using different strategies. In scan line order, the algorithm selects, for each output patch, a similar patch of pixels from the input texture, constrained by overlap with the already synthesized result. It then performs a minimum-error-boundary-cut (MEBC) within the overlap region of adjacent texture patches to reduce artifacts.[6]

III Image Enhancement By Texture Synthesis

Image enhancement using texture synthesis combines interpolation, classification and texture synthesis to enhance low resolution imagery, particular aerial imagery. In that input as low resolution source image and several high resolutions sample textures. The output is high resolution image.

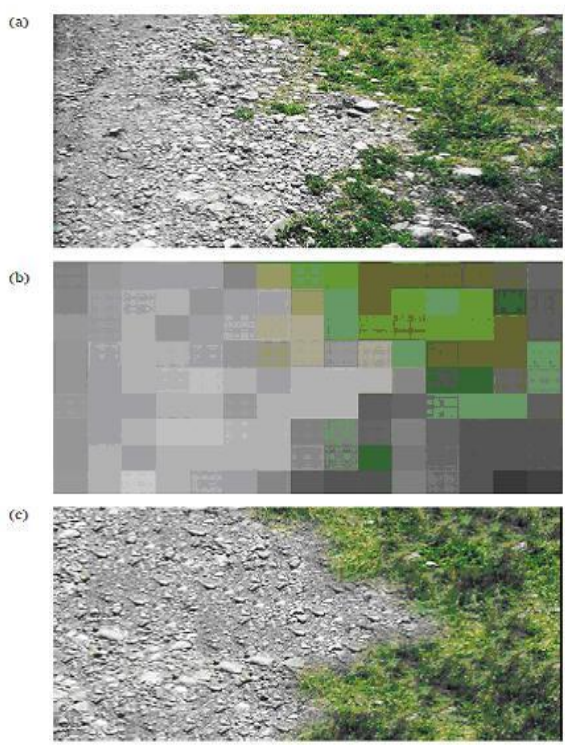


Figure 1: Image Enhancement by Texture Synthesis
(a) Original Photograph (b) Sample version of Image
(c) Enhanced version of Image

IV LITERATURE SURVEY

Various literatures have referred for basic understanding and analysis of structure with Texture Synthesis. Some of that are summarized below.

Image Replacement through Texture Synthesis

Image replacement technique based on Heeger and Bergen's texture synthesis algorithm[7]. Photographs and images often have regions which are in some sense flawed. Image replacement algorithm, *Texture-Replace,* is illustrated in below Figure .[15]

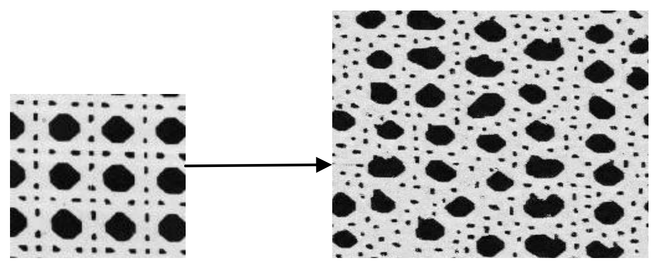
(a) Input Image (b) Output Image

Figure 2: Example of Image Replacement

Deterministic Texture Analysis and Synthesis using Tree Structure Vector Quantization

The synthesis process can be described by the following pseudo code:

1. Loop through all pixels (x, y) in the output texture in raster scan order.
2. Collect the neighborhood vector, Nx,y, of pixel (x, y).
3. Assign F(Nx,y) to be the synthesized color of pixel (x,y).



Original Texture Synthesized Texture

Figure 3: Texture Analysis/Synthesis Algorithm

Using Texture Analysis and Synthesis, generating high quality texture images the key advantage of this approach is that it can efficiently synthesize a wide variety of textures. Our algorithm is also easy to use: only an example image along with a few parameters is required to generate a new texture of any size and shape.[9]

Because the set of local neighborhoods N (*pi*) is used as the primary model for textures, the quality of the synthesized results will depend on its size and shape. Intuitively, the size of the neighborhoods should be on the scale of the largest regular texture structure; otherwise this structure may be lost and the result image will look too random.[10]

Real-Time Texture Synthesis By Patch-Based Sampling

Super texture can be synthesized in real-time. We propose Algorithm is a patch-based sampling scheme, and the advantages of patch-based sampling are Quality and Speed.

Speed: For synthesizing textures of the same size and comparable (or better) quality, our algorithm is orders of magnitude faster than existing texture synthesis algorithms, including TSVQ-accelerated non-parametric sampling. As a result, high-quality texture synthesis is now a real-time process on a midlevel PC.

Quality: The patch-based sampling algorithm synthesizes high-quality textures for a wide variety of textures ranging from regular to stochastic. Like, that is also a greedy algorithm for non-parametric sampling. For this reason, algorithm continues to synthesize high-quality textures even when and cease to be effective. For natural textures, the results of patch-based sampling look subjectively better.[11]

Real time image enhancement using texture synthesis

It combines interpolation, classification and patch based texture synthesis to enhance low resolution imagery. RETS uses as input a low resolution source image and several high resolution sample textures. The output of RETS is a high resolution image.

Image Interpolation: Interpolation is the primary technique used for image scaling. Image scaling is the process of taking a source image and extending it to create a large image. The primary problem with enlarging images using interpolation is that the large result contains the same amount of discrete data as smaller source image.

V ANALYSIS OF LITERATURE SURVEY

No	Title	Summary	Method
1	Image Replacement through Texture Synthesis	provides a new solution to the image replacement problem for a certain class of images.	Stochastic
2	Deterministic Texture Analysis and Synthesis using Tree Structure Vector	Advantage of approach is that it can efficiently synthesize a wide variety of textures. At the same time, it is simple to implement since the most complex component is tree structure VQ.	Pixel
3	Composite texture Synthesis	Segmentation uses fixed filters, which are texture- and mutually independent, while the synthesis uses an optimal texture and mutually dependent pixel pair type selection obtained during the analysis by synthesis procedure.	Pixel
4	Fast Texture Synthesis using Tree-structured Vector Quantization	Texture synthesis method is its low computational cost. This permits us to explore a variety of applications, in addition to the usual texture mapping for graphics that were previously impractical.	Patch
5	Synthesis Algorithms	Patch based approach can produce better synthesis results while requiring less computation.	Patch
6	Real time image enhancement using texture synthesis	Combination of Interpolation and patch based texture synthesis to enhance low resolution images while they are being large.	Patch
7	An efficient Texture Synthesis Algorithm based on WT	An algorithm to speed up texture synthesizing using WT technique using patch based texture synthesis.	Patch

VI METHODOLOGIES

Pixel Based Texture Synthesis

Pixel-based multi-resolution texture synthesis algorithm, which is based on a non-parametric sampling method. It also assumes a Markov random field texture model, which means a pixel value at a certain location only depends on its immediate neighborhood.[12] When choosing the value of the next pixel in the output image.

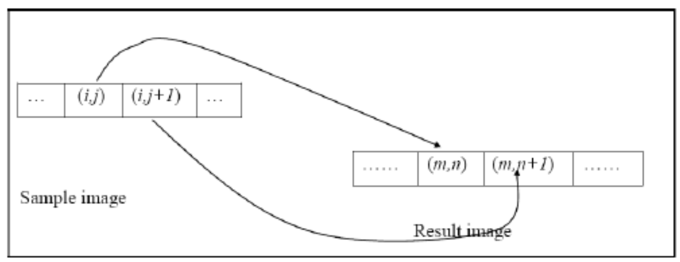


Figure 4: Match neighborhood Pixel

the algorithm uses the populated portion of the pixels neighborhood to exhaustively search for the best matched region in the sample image. synthesizing texture in raster order. If pixel (i,j) has the most similar neighborhood, the value of pixel (i,j) in the sample pyramid is assigned to pixel (x,y) in the result pyramid.

The Flow diagram of *Pixel Based Texture Synthesis* Show in below figure.

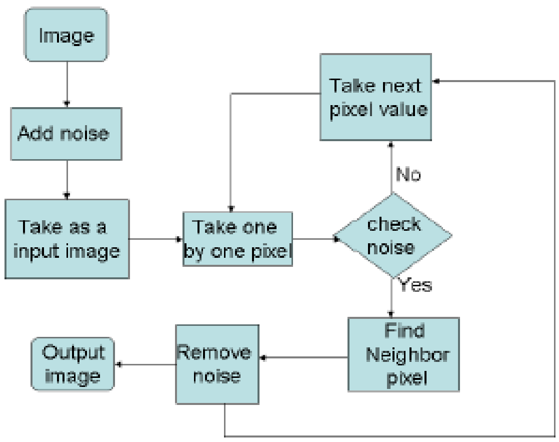


Figure 5: Flow of Pixel Based Texture Synthesis

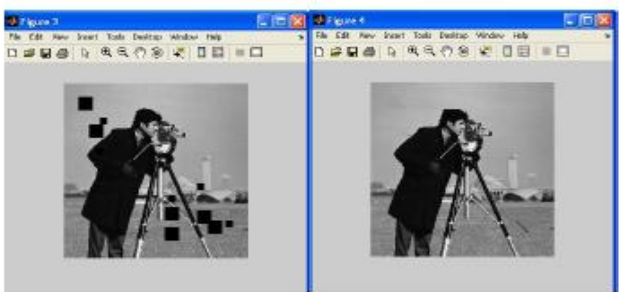
Implementation:

In that first taking a one image and then add some noise in that image and taken that image as input image. Then after removing the noise from that input image and get the outputimage. Here taken noise patch is fixed size.

A pixel value at a certain location only depends on its immediate neighborhood. In this approach, for a certain percentage of the selections we use the next column neighbor pixel.There are two common steps:

a. Searching for the best match for the current output neighborhood within the sample texture

b. Merging a patch or a pixel with the synthesized output texture. After this two steps we are getting image in which noise is not present and that is output image.



Input Image Pixel Output Image

Figure 6: Remove noise in Pixel Based

Patch Based Texture Synthesis

The method synthesists a new image by stitching together small patches from the sample image. The method in synthesists a result image block by block in raster order. Square blocks are used to capture the primary pattern in the sample texture.

The Flow diagram of *Patch Based Texture Synthesis* Show in below figure.

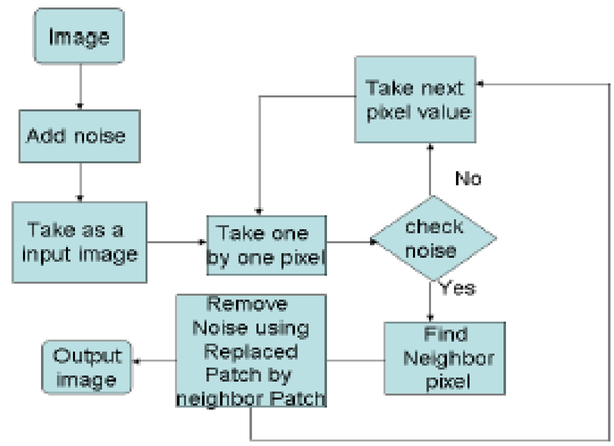


Figure 7: Flow of Patch Based Texture Synthesis

Implementation:

In patch based approach synthesis the result image by stitching together small patches selected from the sample image. In this method synthesiss a result image block by block in raster order. Square blocks are used to capture the primary patterning the sample texture.

There are two common steps:

a. Searching for the best match for the current output neighborhood within the sample texture

b. Merging a patch or a pixel with the synthesized output texture.

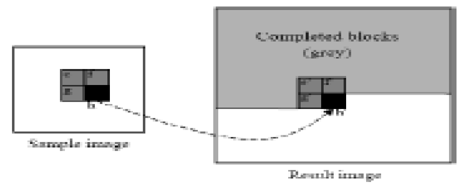
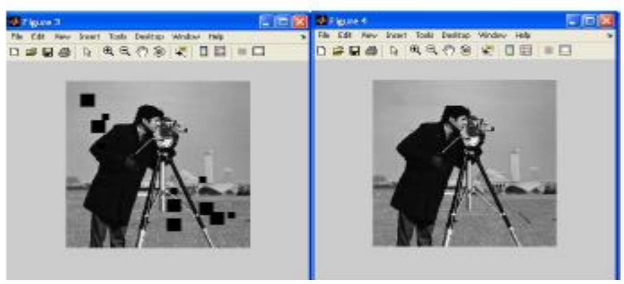


Figure 8: Pixel in Patch



Input Image Patch Output Image

Figure 9: Remove noise in Patch Based

Noise Comparison in both methods

Compare the output of both methods then we conclude that Patch Based Texture Synthesis is better than Pixel Based Texture Synthesis because in Pixel Based Texture Synthesis some noise is not completely removed so possible some information is lost. Compare to Pixel based texture Synthesis good effect is getting after removing the noise in Patch Based Texture Synthesis.

In Pixel Based Texture Synthesis some Information is lost because of different size of patch is taken. Using Patch Based Texture Synthesis information is not lost and give better output than the Pixel Based Texture Synthesis.

Program Complexity

In Pixel Based Texture Synthesis taking One by one pixel and then find its near neighbor pixel and replaced it. So, program complexity is theta (n2).In Patch Based Texture Synthesis taking a group of pixels. so not taking one by one pixel so complexity is theta(n2)/x where x is a patch size. So, Patch Based Texture Synthesis complexity is less than Pixel Based Texture Synthesis

VII CONCLUSION

Patch Based Texture Synthesis is better than Pixel Based Texture Synthesis. Using Patch Based Texture Synthesis removed noise better than Pixel Based Texture Synthesis. Sometimes some information will be loss in Pixel Based Texture Synthesis Compare to Pixel Based Texture Synthesis loss of information is less in Patch Based Texture Synthesis.

REFERENCES

[1] Li-Yi Wei. Texture Synthesis By Fixed Neighborhood Searching. PhD thesis, STANFORD UNIVERSITY, November 2001.

[2] Image Enhancement Techniques by Texture Parth Bhatt2, Prof. Kirit J. Modi3, Volume 2, Issue 1, January 2012

[3] On Pixel-Based Texture Synthesis by,Non-parametric Sampling,Seunghyup Shin Korea Advanced Institute of Science And Technology

[4] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and StevenWorley. *Texturing and Modeling A Procedural Approach*. Morgan Kaufmann Publishers, 1998.

[5] Pixel based Pixel-based Texture Synthesis Yuanliang Meng , 12-1- 2010

[6] Hybrid Texture Synthesis a thesis (diplomarbeit) submitted to the department of computer science of the technische universit¨at Darmstadt in partial fulfillment of the requirements for the degree Of diplom informatiker (dipl.-inform.).

[7] D. Heeger and J. Bergen, "Pyramid-Based Texture Analysis/Synthesis". In Computer Graphics (SIGGRAPH '95 Proceedings), volume 29, pages 229-238, 1995

[8] T. Malzbender and S. Spach, "A Context Sensitive Texture Nib". Communicating with Virtual Worlds, N.M. Thalmann and D. Thalmann, Eds. Springer-Verlag, New York, pages 151-163, 1993.

[9] LI-YI WEI. Deterministic texture analysis and synthesis using Tree structure vector quantization. Gates Computer Science Building,Stanford University,CA 94309, U.S.A., (386).

[10] Li-Yi Wei Marc Levoy. Fast texture synthesis using tree-Structured vector quantization. Stanford University.

[11] Yingqing Xu Baining Guo Lin Liang, Ce Liu and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. Technical Report, (MSR-TR-2001-40), March 2001

[12] D. Scharstein. Synthesis Algorithm, volume 1583/1999 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1999.

[13] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (TOG)*, 20(3):127– 150, 2001

[14] G. Caenen L. Van Gool A. Zalesny, V. Ferrari. Composite texture synthesis.International Journal of Computer Vision, 2004.

[15] Lucas Pereira Homan Igehy. Image replacement through texture synthesis. Com-puter Science Department,Stanford University.

[16] Vivek Kwatra Greg Turk4 Li-YiWei, Sylvain Lefebvre. State of the art in example-based texture synthesis. The Eurographics Association, 2009.

[17] J. F. Blinn and M. E. Newell. Texture and reection in computer generated images. Communications of the ACM, (19):542546, 1976.

[18] E. Catmull. A Subdivision Algorithm for Computer Display of Curved Surfaces. Phd thesis, Computer Science Department, University of Utal,, Salt Lake City, Utah, 1974.

[19] Martin Szummer and Rosalind W. Picard. Temporal texture modeling. Inter- national Conference on Image Processing, 3:823{826, sep 1996.

[20] Maneesh Agrawala Andrew C. Beers and Navin Chaddha. Rendering from compressed textures. Proceedings of SIGGRAPH 96, pages 373-378, August 1996.

[21] Alexei Efros and Thomas Leung. Texture synthesis by non-parametric sampling. International Conference on Computer Vision, 2:1033{1038, sep 1999.

[22] Homan Igehy and Lucas Pereira. Image replacement through texture synthesis. International Conference on Image Processing, 3:186{189, oct 1997.

[23] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. SIGGRAPH 97 Conference Proceedings, pages 361-368, August 1997.

[24] Darwyn Peachey Ken Perlin David S. Ebert, F. Kenton Musgrav And Steven Worley. Texturing and modeling. In A Procedural Approach. Morgan Kaufmann Publishers, 1998.

[25] Ning Zhou Weiming Dong, Ning Zhou. Optimized tile-based texture synthesis. Graphics Interface, Montreal, Canada, 2007.

[26] W. Guo Y. Meng, W.H. Li and Y.L. Liu. Particle swarm optimization method used in pixel-based texture synthesis.

[27] Aaron Bobick Nipun Kwatra Vivek Kwatra, Irfan Essa. Texture optimization for example-based synthesis.

[28] Pizzanu Kanongchaiyosy Jakrapong Narkdej. An efficient parameters estimation method for automatic patch-based texture synthesis.

[29] E. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coeffcient magnitudes. Fifth International Conference on Image Processing, 1:62{66, oct 1998.

[30] Hai-Feng Cui Xin Zheng Tong Ruan. An e_cient texture synthesis algorithm based on wt. 6:3472-3477, 2008.

[31] Pizzanu Kanongchaiyosy Jakrapong Narkdej. An efficient parameters estimation method for automatic

[32] Matthew Sorenson. Real time image enhancement using texture synthesis. November 2004.