ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Performance and analysis of texture synthesis based on Multi Seed-blocks and kernel support vector machine

[1] Syed Abdul Moeed, [2] Dr. Nagaratna Hegde, [3] Dr. Niranjan Polala
[1] Research Scholar, CSE Dept, JNTUH, Hyderabad
[2] Professor in CSE Dept, Vasavi College of Engg, Hyderabad Area, India.
[3] Professor & Head CSE Dept, KITS, Warangal

Abstract - **This paper explains numerous applications, the texture synthesis is mainly based upon multiple seed-blocks and multi-kernel support vector machine (MKSVM). In generally the MKSVM is a linear kernel that hybrid the linear kernel and quadratic kernel in order to improve the efficiency of the synthesis texture. And also this paper explains of four modules such as (i) Pre-processing, (ii) Feature extraction, (iii) Training using MKSVM and (iv) Output pixel synthesis. So At first we should give a sample image to the processing unit. Next, we should extract the features from the input preprocessed gray scale image. Now we should train the sample with a class label by using MKSVM model. Finally, in the testing process, the output pixels are synthesized. And also the experimental result shows, our proposed methodology successfully synthesize both random as well as textures.**

Keywords: Texture synthesis, seed-block, multi kernel, support vector machine, linear kernel and quadratic kernel.

1. INTRODUCTION

Generally the texture synthesis is very important in the research parts in computer graphics and in computer vision over ages. And a texture synthesis technique twitches from a sample image and efforts to yield a texture. And the key behind these texture synthesis methods is the inherent repeatability existing in textures.
Normally the Texture synthesis methods are mainly divided into two types, namely Local region-growing technique and worldwide optimization based approaches. And also these methods may also produce output textures that are greater in size than the input sample. These Texture synthesis is another way to create textures in an effective manner.

As these synthetic textures can be made of any size and visual recurrence is evaded. Texture synthesis can also yield tile able images by properly holding the boundary circumstances.

For Potential implementations of texture synthesis are also broad; some instances are like image de-noising, occlusion fill-in, and compression [1].

In Generally the Solid textures have the numerous notable compensations over 2D textures. The natural materials are, like wood and stones are may be more convincingly modeled with the help of solid textures. These solid textures are also removing the necessity for ruling a parameterization for the surface of the object to be textured. In fact, for all objects of overall topology, it's not conceivable to detect a parameterization that evades seams and/or distortion. The Parametric approaches are usually engage with the statistical models. And that and can give compacter demonstration for the illustration. Hence for texture synthesis they may have the issues to synthesize semi-structured or extremely structured textures. For contrast, methods on the basis of non-parametric models, like e.g. algorithms depended up on the "smart copying", that can contribute the finest solutions for numerous types of texture, especially for semi-structured texture [2].
The following are the various technologies which are utilized for the texture synthesis. They are mainly classified into two types such as follows Procedural texture synthesis and Example-based texture synthesis. These Procedural texture synthesis, are utilizing an algorithm means to generate a realistic illustration of natural elements such as wood, marble, granite, metal, stone, and others. The second one is Example-based texture synthesis produces a high-resolution texture from a small input example. And it pledges the continuity. As

support vector machines (SVM) is a machine learning technique, has been extensively utilized for classification and regression. For the improvement of utilizing support vector machines (SVM) is that the sample can be merely modeled using a linear model. And also the input and output mapping performance can also produce from a set of labeled training information. For the meantime, the sample is not obligatory at the time of the synthesis stage.

The proposed and implemented an efficient texture synthesis based on multi kernel support vector machine (MKSVM) in this paper. Initially we should extract the "L" shape features of the input image. Next we train the features by using multi kernel support vector machine (MKSVM). At finally in testing process, randomly select any small patch from the sample image. Now seed the patch in the output image. And as the rest of the paper is organized as follows: A brief review of some of the literature works and also texture synthesis techniques are presented in the section 2. And in section 3 the back ground of the research will be shown. And the proposed texture synthesis system is detailed in Section 4.At finally the experimental result is provided in Section 5. And the end, the conclusions are summed up in Section 6.
Here the structure of multi class SVM is shows in below figure 1.

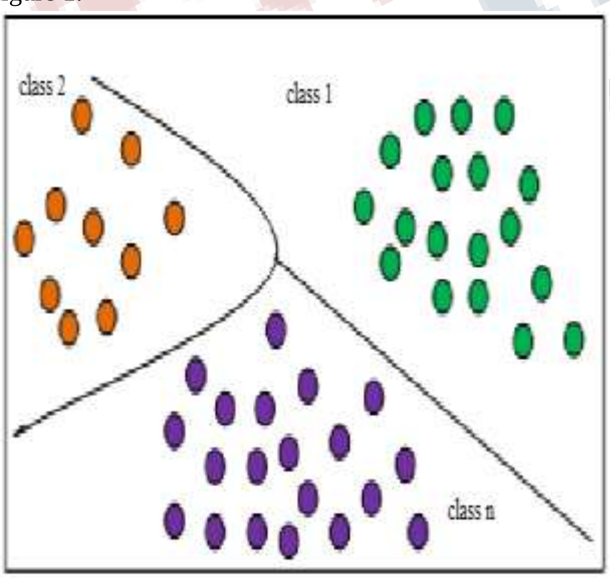


Fig.1.Multi class support vector machine structure

2. PROPOSED TEXTUR SYNTHESIS:

The basic principle of texture synthesis is basis of multiple kernel support vector machine (MKSVM). The texture synthesis is mainly associated with the production of an enormous texture image from a sample image. And it is also used to determine the value of a specific output pixel. The input pixel with the most comparable neighborhood is then assigned to the output pixel. Hence this process is repeated for every output pixel, until the whole texture is developed [4].

Here this scheme comprises of four steps like, (i) preprocessing stage, (ii) Feature extraction, (iii) Training using MKSVM, (iv) output patch synthesis.

(i)Preprocessing stage:
The main aim of this paper is to texture synthesis based on Multi kernel support vector machine (MKSVM).
Let us consider the sample imageR, which have the size of (N=64).Using this sample image R, hence it is synthesized in to (N=128). Initially we have given the image into a preprocessing stage. Because the image is not directly given to the process, it has some noises. Now we decrease the noise in the sample image, in such a way that the model training will not be afflicted by noise. Actually, here we are using a median filter, in order to reduce the noise. The gray level is 64.

(ii)Feature extraction stage:
This is the next stage of pre-processing stage. After extracting the features from the sample image this stage may use. For texture synthesis process Feature extraction is an important stage. It is mainly based on the features only. Here we can utilize all pixel values in the pre existing neighborhood in order to create a feature vector to overcome overpriced calculation of extra feature extraction.

(iii)Training based on MKSVM:
After the feature extraction stage, we can see this stage, and it is used to train the data by using multi kernel support vector machine (MKSVM). Here, every feature vector have dissimilar class label in this model. So we can

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

use multi class support vector machine intended for train the information.

$$(x_1, y_1),(x_2, y_2)\ldots\ldots(x_N, y_N)$$

Where, $x_i \in R^n$, $i = 1,2,\ldots, N$,

$$(w_k \cdot x) + b_k = 0,\; k = 1,2,\ldots.K$$

And $B_k = \left[S_1^T,\ldots,S_{k-1}^T, S_{k+1}^T,\ldots,S_k^T\right]^T$

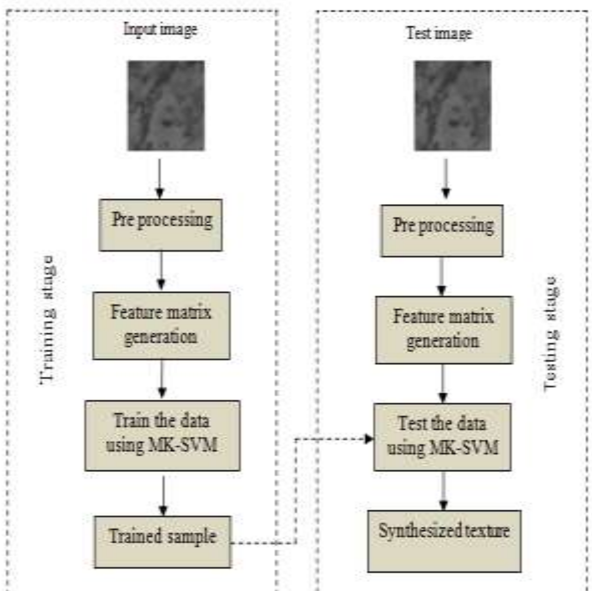


Fig 2: Overall diagram of the proposed methodology

Here, the MKSVM detects the result of the subsequent at the time of training. And it is also used to improve the influence of the synthesis output, at that time; we can add SVM along with multi kernel.
The multi kernel support vector machine function is given in equation below

The general version of the kernel is shown below

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i\, y_i \sum_{d=1}^{m} \beta_d\, K_H(U,V) + b\right)$$

The SVM invariably depends on the selection of the kernel. And also association with any two kernel performances and furthermore is adequate to provide superlative precision than that attained with help of engaging any single kernel function. There are two kernel performances are there, one is linear kernel and other one quadratic kernel performances. These are connected to yield superb performance ratios.

So, finally associated kernel function equation is given as below

$$K(U,V) = \varphi(U)^T \varphi(V)$$

And

$$K_H = \frac{1}{2}\left(\left(u^T v + c\right) + \left(1 - \frac{\|u-v\|^2}{\|u-v\|^2 + c}\right)\right)$$

So, the two kernels like the linear and quadratic are considered for the objective of recognizing the hyper plane in the enhanced Support Vector Machine .And also each data having the neighborhood value and corresponding class label also. Finally, in training stage we obtain the n number of trained data.

3. OUTPUT PATCH SYNTHESIS:

This stage is used to predict the pixels values based up on the training feature vectors. If, once the model is generated, it can be utilized for the texture synthesis only. And it is also used to detect the pixel values. And it is used to detect the class label in the output image [5].

Normally, at this time, we are randomly producing a white noise image along with a predetermined size. Hence, the image is called as initial output image. Then, select a minor patch from a sample image.

From the sample image we can also seed the patch in the output image . And also the output texture is synthesized in a raster order.

IFERP
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Single resolution texture synthesis is shown below figure;

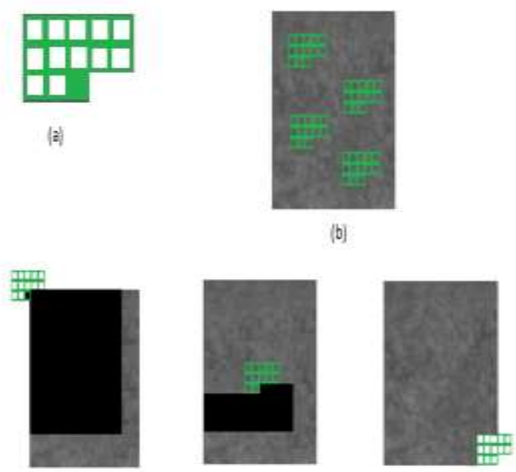


Fig.3. Single resolution texture synthesis.

(a) "L" shape neighborhood
(b) is the input texture and
(c)-(e) show different synthesis stages of the output image.

Here, the above figure 3 shows the graphical illustration of the synthesis process. It is a synthesis process of proposed methodology using "L" shape vector. Generally, the MKSVM model is used to predict pixel values. If first patch is synthesized, we can use it to search for another seed-block to be used for synthesizing the second patch also. This technique is performed again and again as far as all pixels are assigned values.

All Pixels in the output image are assigned in a raster scan ordering. The value of each output pixel "M" is determined by comparing its spatial neighborhood A with all neighborhoods in the input texture [5].

The input pixel is the most similar neighborhood. And it will be will be assigned to the corresponding output pixel. Training and Testing process of texture synthesis is shown below figure:

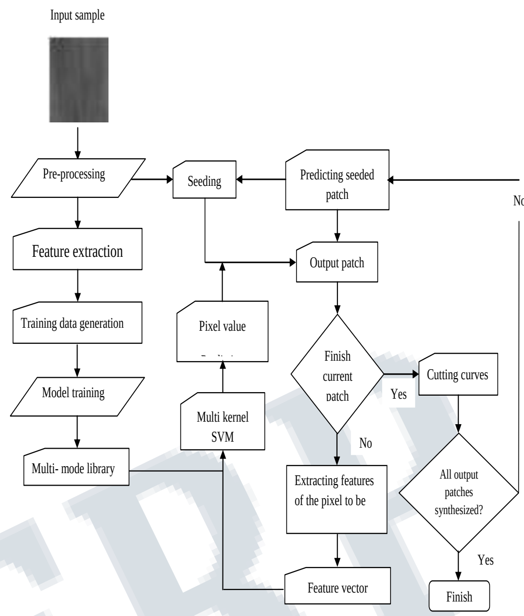


Fig.4 Training and testing process of texture synthesis

The training and testing process of proposed methodology is illustrated in figure 4 and steps involved in proposed methodology are given in table 1.

Table 1: Steps involved in proposed methodology

Input : sample image I^{in}
Output: synthesized image O_{out}
Start
Training process
1. Consider the input image I^{in}
2. Preprocessed the input image I
3. Extract the "L" shape feature
4. Generate the feature matrix
5. Train the features using MKSVM
Synthesis process
6. consider the part of sample image
7. predict the pixel value one by one
8. Repeat the process until all pixels are synthesized
9. Obtained the synthesized image size of 128×128
End.

IFERP

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

4. RESULT AND DISCUSSION

In this particular section, the experimental results of proposed texture synthesis by using multi kernel support vector machine (MKSVM).Especially, for implementation purpose we have used MATLAB version 7.12. And it includes the proposed technique is done by using windows machine having Intel Core i5 processor with speed 1.6 GHz and 4 GB RAM [6].

Here there are some of the sample images are used in proposed methodology is shows in below figure 5.

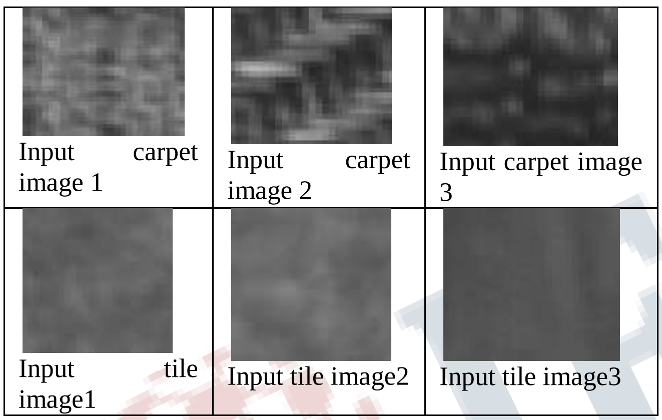


Fig 5: Experimental used images

4.1 Experimental Results

The main aim of this research paper to texture synthesis using multi kernel support vector machine (MKSVM). Initially, we extract the "L" shape feature from the preprocessed image. And this "L" shape feature has neighborhood pixels and corresponding class label. Next, we train the features using multi class with multi kernel support vector machine .In these synthesis process only; we can generate first a white noise image with a pre-defined size as the initial output. After that we can select randomly small patch from the sample image and seed the patch in the output image [7].

Next, we select the "L" shape neighborhood and predict the class label. Hence, if this process is repeated until all the pixels are synthesized.

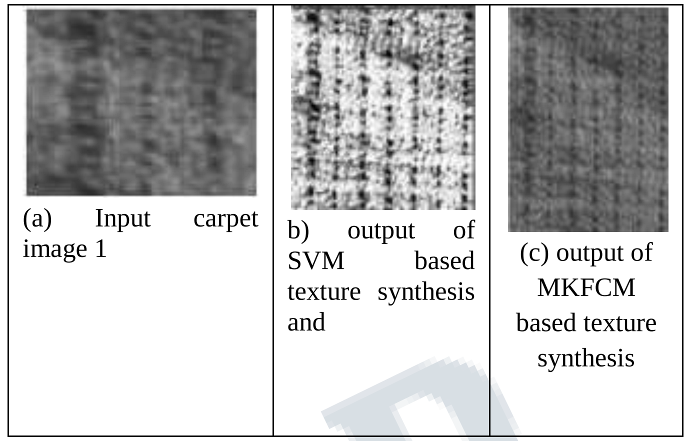


Fig 6: Visual representation of Texture synthesizes

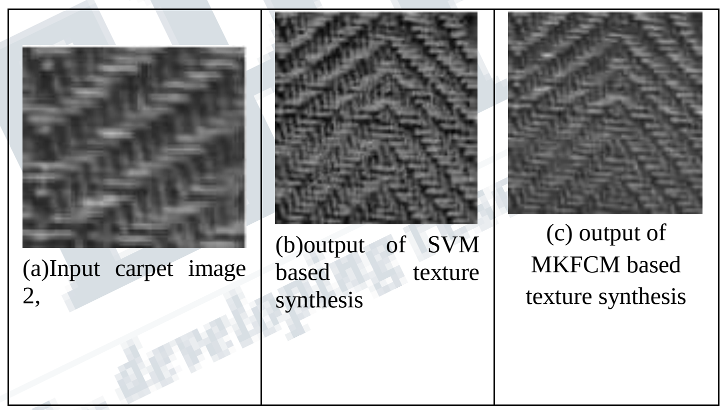


Fig 7: Visual representation of Texture synthesis

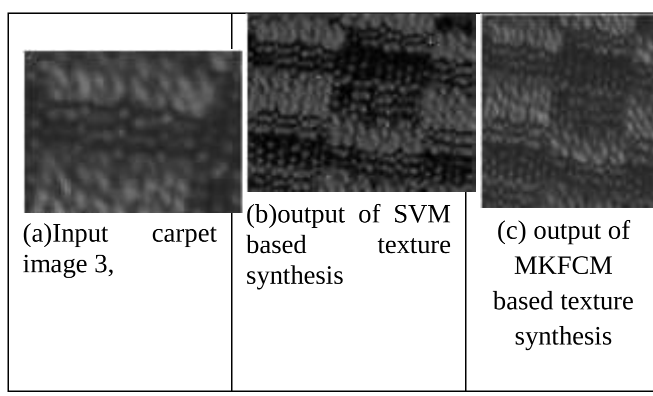


Fig 8: Visual representation of Texture synthesis

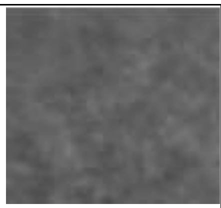
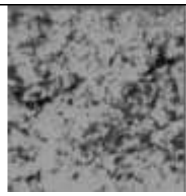
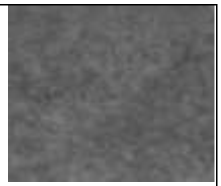
		
(a)Input tile image 1,	(b)output of SVM based texture synthesis	(c) output of MKFCM based texture synthesis

Fig 9: Visual representation of Texture synthesis

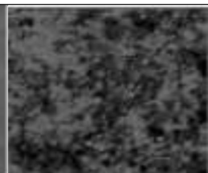
		
(a)Input tile image 2	(b) output of SVM based texture synthesis	(c) Output of MKFCM based texture synthesis

Fig 10: Visual representation of Texture synthesis

		
(a)Input tile image 3	(b) output of SVM based texture synthesis	(c) Output of MKFCM based texture synthesis

Fig 11: Visual representation of Texture synthesis

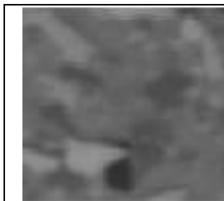
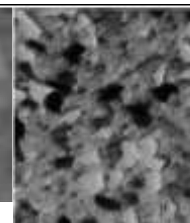
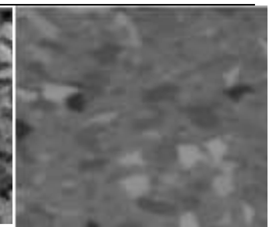
		
(a)Input tile image 4	(b) output of SVM based texture synthesis	(c) Output of MKFCM based texture synthesis

Fig 12: Visual representation of Texture synthesis

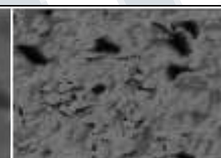
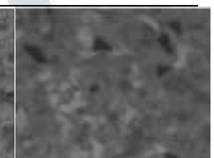
		
(a)Input tile image 5	(b) output of SVM based texture synthesis	(c) Output of MKFCM based texture synthesis

Fig 13: Visual representation of Texture synthesis

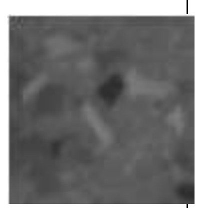
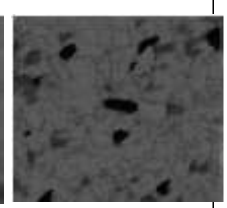
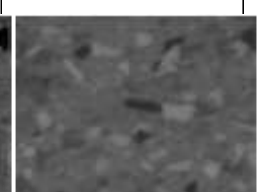
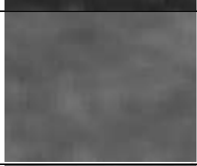
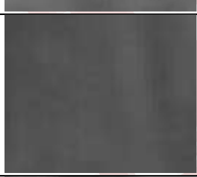
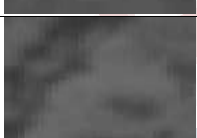
		
(a)Input tile image 5	(b) output of SVM based texture synthesis	(c) Output of MKFCM based texture synthesis

Fig 14: Visual representation of Texture synthesis

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)
Vol 5, Issue 2, February 2018

Table 2: Performance comparison of proposed approach based on entropy measure

Synthesize image	Entropy (existing)	Entropy (Proposed)
	5.394518	6.226042
	5.436837	6.559297
	5.016986	5.900048
	3.885006	4.883544
	4.316008	4.742716
	4.415489	5.136379
	4.415489	5.136379
	5.012221	5.632726
	4.812933	5.644666
	4.848186	5.48841

The above figure represents the visual results of proposed and existing methods. The results which are explained in the proposed algorithm can generate good results for semi or highly structured textures. The hybrid kernel function improves the efficiency of the system [9].Here; we can get the maximum entropy of 6.559297 which is 5.436837 for using SVM based texture synthesis. Hence, from the above result analysis, we clearly understand our proposed approach achieves the better result compare to the existing work [8].

5. CONCLUSION:

Finally, in this paper the multiple seed block and multi kernel support vector based texture synthesis was proposed. Initially, we extract the "L" shape feature from the input image. And each feature vector form based on the region pixels and corresponding class label. Hence, the generated feature vectors are trained by using multi kernel support vector machine (MKSVM) and also Support vector machine (SVM). And also we can see in the testing model, we can randomly select a small patch from the sample image and it may seed with the patch in the output image.

This process is repeated until all the pixels are synthesized. Our new results show that the method can successfully model and synthesize semi or highly structured texture and also we can notice that there is only one drawback of this method is the high calculation cost. So, finally we can hope the Future work may improve the computation efficiency of the entire system.

REFERENCES:-

[1] D. J. Heeger, J. R. Bergen, "Pyramid-based texture analysis/synthesis", AGM Proceedings of SIGGRAPH, pp. 229-238, 1995.

[2] J. S. De Bonet, "Multi-resolution sampling procedure

for analysis and synthesis of texture images", ACM Proceedings of SIGGRAPH, pp. 361-368, 1997.

[3] G. R. Cross and A. K. Jain, "Markov random field texture models", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, no. 1,pp. 25-39, 1983.

[4] Efros and T. Leung, "Texture synthesis by nonparametric sampling", IEEE International Conference on Computer Vision, vol. 2, pp. 1033-1038, 1999.

[5] Li- Yi Wei and MaercLevoy, "Fast texture synthesis using tree-structured vector quantization", ACM Proceedings of SIGGRAPH, 2000

[6] Y. Q. Xu, B. N. Guo and H. Shum, "Chaos mosaic: fast texture synthesis", Microsoft Research Report, 2000.

[7] P. J. Burt and E. H. Adelson, "Multi-resolution spline with application to image mosaics", AChl Transactions on Graphics, vol.2, no.4, pp.217-236, 1983.

[8] Chongyang Ma, Li-Yi Wei, BainingGuo and Kun Zhou, ''Motion Field Texture Synthesis'', ACM Transactions on Graphics, 2009

[9] Ebert, Musgrave, Peachey, Perlin and Worley, "Texturing and Modeling: A Procedural Approach ", Morgan Kaufmann Publishers Inc., 2002.

[10] Jiang Niand and Henry Schneiderman, "An Improved Texture Synthesis Algorithm Using Morphological Processing with Image Analogy", 2014.