The 3rd International conference on Big Data Applications and Services, The Korean Big Data services society, pp.188-195 (2016)

Shape from Textures Phenomenon for 3D Modeling Applications with Big Data Images

Kalaivani Thangamani , Takashi Okuma, Ryosuke Ichikari and Takeshi Kurata

Service Sensing, Assimilation, and Modeling Research Group,
Human Informatics Research Institute,
National Institute of Advanced Industrial Science and Technology (AIST),
1-1-1 Umezono, Tsukuba 305 8568, Japan
{thangamani.kalaivani, takashi-okuma, r.ichikari, t.kurata}@aist.go.jp

Abstract. This paper discusses the shape from textures phenomenon with proposed features and finds its application in the interactive 3D modeling works. Two different themes are discussed in this paper; the proposed algorithm and its integration with the 3D modeling environment. There are many 3D modeling tools which builds 3D model out of the 2D image collections, by requiring much of human interactions for the generation of the 3D objects. This paper applies the shape from textures phenomenon with additional features and propose an algorithm to simplify the process of the 3D planes generation. The proposed algorithm analyzes the textures in the single image and results the orientation and shape details for the corresponding textures. The result of the algorithm is integrated with the 3D modeling environment. There are mutual exchanges between the modeling works and the proposed shape from textures algorithm. The texture orientation details from the algorithm are complemented with the depth measures from the modeler. The algorithm statistically analyzes the collective orientation of the superpixels and suggests the user to go for the nearby matching templates. The proposed algorithm together with the 3D modeling measures provides the flexible environment for the big data image collections.

Keywords: 3D Modeling, Big-Data Images, Shape from Textures, Fourier Transform, Image Processing, 3D Geometry, User Interaction, Template-Matching.

1 Introduction

Shape from texture research finds its major application in the field of three-dimensional (3D) design. Early works on shape from textures were presented by Gibson [1] and Garding [2, 3]. The computation of local surface orientation from the textures by Malik et al. [4] is considered as the initial steps in proposing the shape from texture algorithm in this paper. The concept of finding the affine transform between the neighboring patches in the single image [4] is adapted for the proposed

algorithm. This paper also applies the idea of superpixels [5] and uses one or more superpixels to define the objects in the 2D image.

Figure 1 shows the position of the proposed work in the modeling application. In the field of 3D modeling where every single 2D image in the big data is subjected to the 3D modeling, and the collection of the individual models contributes to the complete 3D environment. The aim of this paper is to introduce flexibility and reduce the repeated human interventions during the modeling stages. The interactive 3D indoor modeling tool [6] which finds its application in the virtual field simulators [7] and service sectors [8] are considered for testing and integrating the proposed works. The modeling tool [6] allows the user to create the 3D model with simultaneous view correspondences between the 2D image and the 3D modeling space. The proposed algorithm is integrated in the 3D planes generation and editing stages of the modeler.

Figure 1 briefly explains the shape from textures phenomenon by finding the texture distortion parameters for the superpixels. One such superpixel is enlarged and its surface normal computation measures are shown in Figure 1. The resultant surface normal for the superpixels (enclosed by the red polygon) are shown as white lines projected over the red circles. These surface orientation features are used to suggest the 3D plane position in the interactive modeling processes. The next section explains the proposed shape from textures algorithm in detail. The following sections explain the integration of the algorithm with the modeling tool. The later sections explain the mutual data exchanges between the algorithm and the modeler. The accuracy test and the results analyses are included in every section.

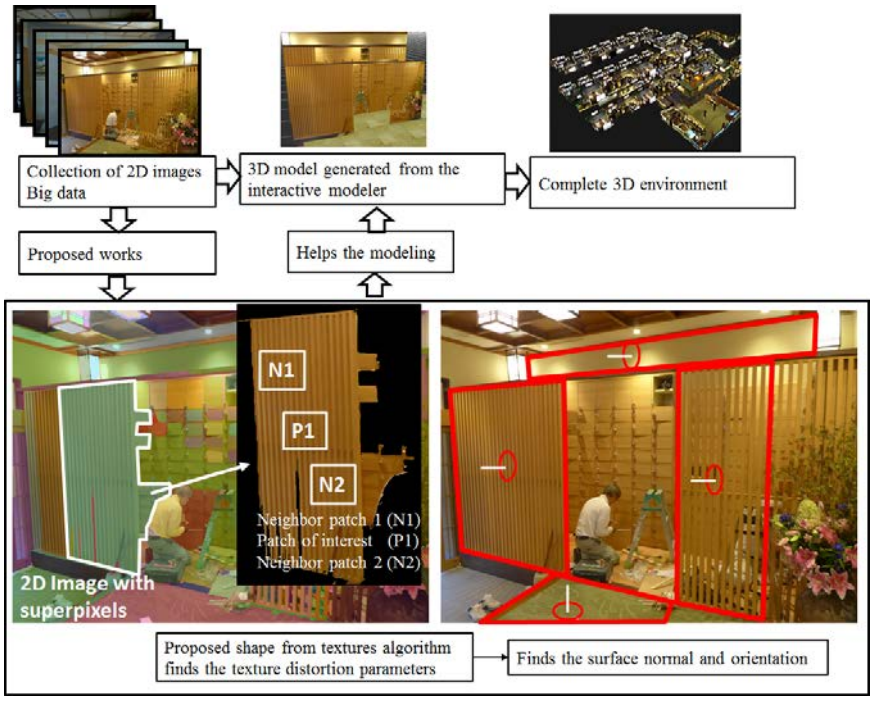


Fig. 1. Overview of the proposed work and its application in the 3D modeling

2 Shape from Textures Phenomenon

This section describes the proposed works with the shape from textures phenomenon. The relation between the texture gradient and the scene geometry [4] is used for finding the texture distortion parameters. Slant and tilt as the orientation parameters and another three shape parameters for defining the curvatures, comprises the total five distortion parameters. Slant is the angle between the line of sight and the surface normal. Tilt is otherwise called as the direction of slant and the projection of the slant direction over the image plane [9]. Tilt is usually measured in the positive x axis and the texture gradient. Figure 2 explains the proposed shape from texture algorithm in detailed stages,

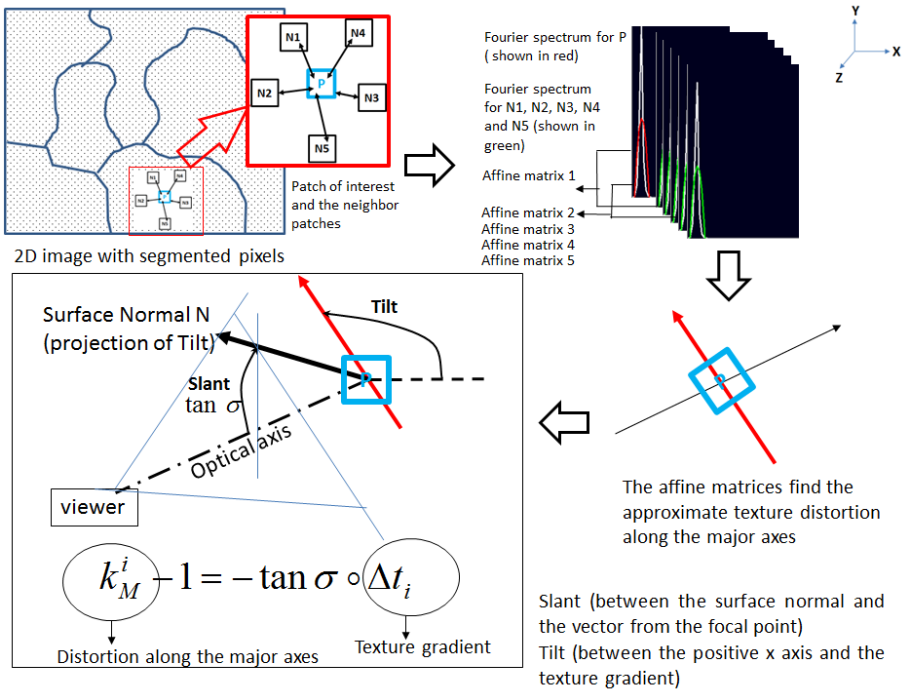


Fig. 2. Shape from textures algorithm stages. Every patch (~80 x 80 pixels) is considered as the patch of interest and their Fourier spectrum is compared with the neighboring patches. Affine transform is calculated between the two spectrum frequencies. Solving the affine transform sets by Singular Value Decomposition, gives the slant and tilt angles and the surface normal.

2.1 Superpixel Stage

The 2D image is segmented into superpixels [5] and the collection of the neighboring superpixels define the various objects in the image. The superpixels are treated as the square patches (e.g. 80 x 80 pixels) and subjected to the preprocessing measures such as the normalization process to have the same gray scale ranges. Every patch acts as

the patch of interest and contacts the neighboring patches for finding the affine transform.

2.2 Fourier Analysis and Affine Transform

The patch of interest and the neighboring patches are subjected to the discrete Fourier transform and their spectrum magnitudes are analyzed for finding the affine transform between the patches. The spectrum of one patch differs from the other by 2x2 affine transform. Figure 2 shows the Fourier spectrum of the patches and the neighboring patches. The Fourier spectrum is smoothened by the second order polynomial type curve fitting and then the fitted curve is considered for finding the affine transformation matrix. If the spectrograms of two patches at frequency ω are taken as $F_2(\vec{\omega})$ and $F_1(\vec{\omega})$ then the affine transformation between the two is given by the following relation,

$$F_2(\vec{\omega}) - F_1(\vec{\omega}) \approx \nabla\vec{F}_1 \circ \Delta A\vec{\omega} \,. \qquad (1)$$

where,

$\nabla\vec{F}_1$ is the gradient of the spectrogram at frequency ω,

ΔA is the affine transformation matrix,

Equation 1 gives the linear equation relating two spectrums at frequency ω. For every frequency point in the spectrum, a relating linear equation can be formed. This ends up with the over-determined number of linear equations (more number of equations than the solutions). These equations are solved by SVD (Singular Value Decomposition) and the singular values provide the initial estimation for the texture distortion features. The singular values are related to the magnitude of texture changes along the minor (k_m) and major (k_M) texture axes. The patch of interest (P) mapped against the initial estimation is stated in Figure 2.

2.3 Slant and Tilt Findings

The set of affine transforms are calculated between the patch of interest and the neighbor patches and they are related by Equation 2.

$$\begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ . & . \\ \Delta x_n & \Delta y_n \end{bmatrix} . \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} k_M^1 - 1 \\ k_M^2 - 1 \\ . \\ k_M^n - 1 \end{bmatrix} . \qquad (2)$$

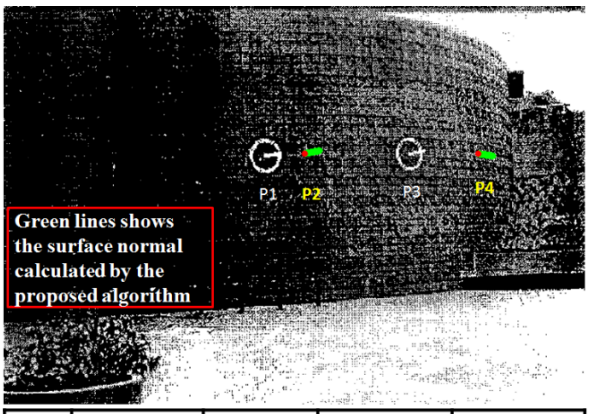
where k_M refers to the magnitude change along the major texture axis. Δx_n and Δy_n refers the distance between the two patches under test. The t_x and t_y represents the tilt direction and the set of k_M arrays represents the initial

estimation values that are obtained in the previous stages. Solving Equation 2 once again, by SVD provides the tilt direction and substituting the values of t_x and t_y in Equation 3, gives the slant angle $\tan\sigma$,

$$k_M^i - 1 = -\tan\sigma \circ \Delta t_i \ . \tag{3}$$

Equation 3 shows the trigonometric relation between the texture gradient and the initial singular values that are found in the previous stages. The detection of slant and tilt leads to the surface normal which will be used by the modeling tool for generating the 3D planes in the 3D space.

2.4 Accuracy Testing for the Proposed Algorithm



	(P1) Shown in [4]	(P2) By the proposed algorithm	(P3) Shown in [4]	(P4) By the proposed algorithm
Slant	19	19.900	40	33.345
Tilt	-8	32.619	-11	20.75

The proposed algorithm is tested over the example used by Malik et al. [4]. Since the proposed algorithm adapts its startup from [4], the accuracy testing over the same example holds the relative comparison. The Table inserted in the Figure 3 compares the slant and tilt values estimation by [4] and by the proposed method. Irrespective of the constraints such as the texture quality and the neighbor patch requirements, the proposed algorithm detects the surface orientation within the reasonable tolerance limits.

Fig. 3. Accuracy test for the proposed algorithm
(Image courtesy: Malik et al. [4])

3 Algorithm Adaptability with the 3D Modeler

The adaptability of the proposed algorithm is tested over the images that are modeled interactively by the users in the 3D modeler [6]. The surface normals of the various planes in the 3D model are considered as the ground truth for checking

Fig.4. Top row: Needle diagram shows the estimated normal (green lines) and the ground truth normal (white lines). Bottom row: Corresponding heatmap; Green zone represents less difference between the normals; More red more deviation from the ground truth.

the adaptability of the algorithm. The superpixels in the images are subjected to the algorithm and the resultant surface normals are compared against the ground truth. Figure 4, shows the test images with surface normals/ground truth detected by the modeling tool (marked in white lines) and the surface normals estimated by the proposed algorithm (marked in green lines). The heatmaps for the comparison are shown in the second row of Figure 4. The heat map is organized between the two colors, green and red; ranging the differences between 0 and 90 degrees. The green region denotes the patches with less difference and the red shaded region with more difference between the ground truth and the estimated normal angles. More the green shaded region states that the estimated result stands closer to the ground truth values. The testing shows that the algorithm can be adapted to the interactive modeling tool and used for suggesting the 3D plane orientation in the model generation stages.

4 Integration with the Modeler

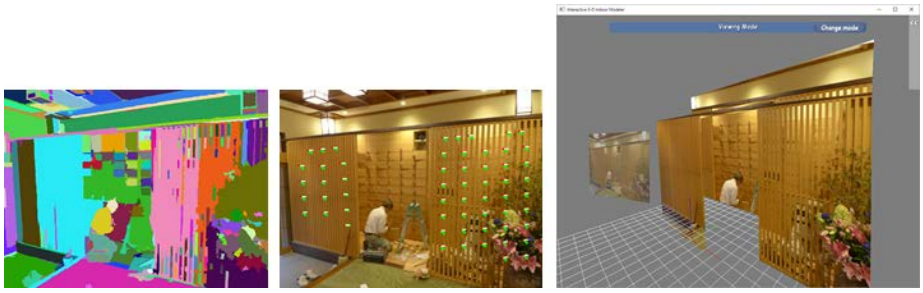


Fig.5. Algorithm integration in the 3D modeling. Figure shows the superpixel, the surface normal detected by the algorithm and the orientation guidance for the 3D modeling process.

The algorithm is integrated in the plane generation stages of the interactive modeler [6]. One such modeling example is shown in Figure 5. The user interaction is

accessed for grouping the nearby superpixels that defines the individual planes. The superpixels defining the individual planes are subjected to the algorithm and the detected normal vectors for every single patch is shown in green lines. The collective direction of all the patches are used for suggesting the direction of the plane of interest. The third subplot in Figure 5 shows the view correspondence in the modeler that guides the user with the algorithm outputs to fix the 3D planes in the 3D model.

The depth values of the generated ground plane is used by the algorithm to enrich the superpixels with depth information. The depth embedded superpixels is used for defining the height differences between the horizontal planes and the depth alignment for the successive vertical planes. The mutual data exchanges between the algorithm and the modeler features brings flexibility in the modeler and increases the degree of automatic modeling.

4.1 Statistical Analysis of the Surface Normals and Template Matching

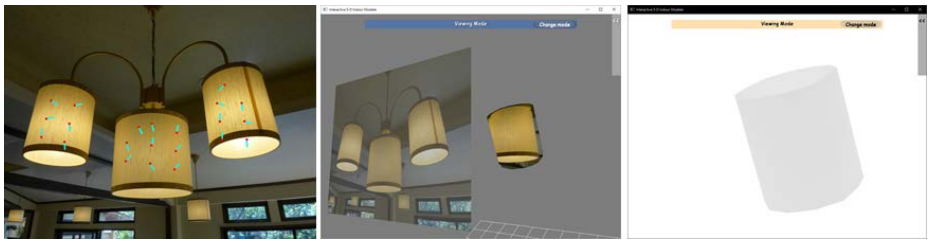


Fig.6. Template-Matching feature. The 2D image with the detected surface normal. The detected normals are guided to the nearby matching template. The depth map of the 3D model generated by the cylindrical template.

Extending the analysis of the surface normal of the individual patches, leads to the design of the template matching feature. The template matching feature helps to relate the collective orientation of the superpixel to the nearby shape templates. The template matching feature is aimed for the cases, where the textures are not rich enough to provide the accurate normal estimation. In those cases, the user interaction is adopted for guiding the orientation to the nearby matching template. The templates can be ranging from the simple 3D plane to the traditional 3D shapes and complex structures.

Figure 6 shows the example of the cylindrical template which helps to fix the target texture with the estimated orientation. The first subplot in Figure 6, shows the 2D image with the estimated surface normals. The second subplot shows the cylindrical template complements the 3D design for the targeted texture. The third subplot shows the depth map for the generated 3D model with the template matching assistance. The standard 3D templates such as the spheres, cubes, polygons and the traditional shapes such as the tea pot, flower vase and other interior shapes are the future targets to get tested in the modeling environment. The texture lags in the templates caused by the occlusion issues are expected to be handled with the inpainting treatment [10]. The modeler features enable the user to move and scale the

templates in groups, interactively. Also, there are options to reuse the textured templates in the models, especially in the case of 3D indoor modeling applications.

5 Conclusion and Future Works

Shape from texture algorithm with additional features to support the 3D modeling environment is proposed and discussed. The algorithm detects the surface orientation of the segmented texture clusters in the 2D image. The output of the algorithm is integrated to the 3D modeling applications and steps are taken to improve the efficiency of the modeling. The mutual data exchanges between the proposed algorithm and the modeling tool helps to boost the flexibility in the 3D design. To compare the modeling time and efficiency with and without the presence of the proposed works will strengthen this research and the pros and cons leads to our future works. The qualitative and the subjective analyzes will be conducted in the future for exploring the optimization processes in the proposed works. There are accuracy constraints encountered, with the limited texture details, textures with shading and reflection issues. To address these constraints and make the algorithm robust to handle the variety of textures, pave the way for the future works.

References

1. Gibson, J.: The Perception of the Visual World. Boston. Houghton Mifflin (1950)
2. Garding, J.: Shape from Texture and Contour by Weak Isotrophy. J. of Artificial Intelligence. vol. 64(2), pp. 243-297 (1993)
3. Garding, J.: Shape from Texture for Smooth Curved Surfaces in Perspective Projection. J. of Mathematical Imaging and Vision. vol. 2(4), pp. 327-350 (1992)
4. Malik, J., and Rosenholtz, R.: Computing Local Surface Orientation and Shape from Texture for Curved Surfaces. International Journal of Computer Vision. vol. 23(2), pp. 149-168 (1997)
5. Felzenswalb, P.F., and Huttenlocher, D.P.: Efficient Graph Based Image Segmentation. Int. J. of Computer Vision. vol. 59(2), pp. 167-181 (2004)
6. Ishikawa, T., Thangamani, K., Kourogi, M., Gee, A., Mayol-Cuevas, W., Hyun, J., Kurata, T.: Interactive 3-D Indoor Modeler for Virtualizing Service Fields. Int. J. of Virtual Reality. SI: Mixed and Augmented Reality. vol. 19(2), pp. 1-21 (2011)
7. Okuma, T., Kurata, T.: Service Field Simulator: Virtual Environment Display System for Analyzing Human Behavior in Service Fields. Int. Conf. on Serviceology, pp. 55-60 (2014)
8. Fukuhara, T., Tenmoku, R., Okuma, T., Ueoka, R., Takehara, M., Kurata, T.: Improving Service Processes Based on Visualization of Human-Behavior and POS Data: A Case Study in a Japanese Restaurant. Int. Conf. on Serviceology, pp. 1–8 (2013)
9. John Krumm and Steven A. Shafer.: Shape from Periodic Texture Using the Spectrogram. IEEE (1992)
10. Thangamani, K., Ishikawa, T., Makita, K., Kurata, T.: Fast and Texture-Structure Preservable Inpainting and its Application for Creating Virtualized-Reality Indoor Models. Int. J. Computer science and informatics, vol. 1(3), pp. 85-101 (2012)