

# A Three-dimensional Scene Simulation System for Evaluating Textures of Fabrics

Ruyuan Dun

College of Electrical Engineering, Zhejiang University  
Hangzhou, 310000, China  
E-mail: dryshuai@163.com

Zhen Fan

College of Electrical Engineering, Zhejiang University  
Hangzhou, 310000, China  
E-mail: fanzhen0095019@gmail.com

Huilin Yao

Department of Electrical Engineering and Automation,  
Luoyang Institute of Science and Technology  
Luoyang, 474023, China  
E-mail: Yhl64@163.com

Senlin Zhang

College of Electrical Engineering, Zhejiang University  
Hangzhou, 310000, China  
E-mail: slzhang@zju.edu.cn

**Abstract**—In this paper, a method of displaying 3D model on the web end based on Three.js library which can simulate the wrinkle and distortion of fabric and support the replacement and modification of texture is realized. The visual effect of a specific fabric texture can be evaluated after texture mapping from the custom area in the model. At first, extract the texture image of 3D model and model it by using the finite element mesh algorithm. Using cubic convolution interpolation algorithm to solve the texture mapping anti aliasing problem can effectively smooth the jags after texture mapping so that the result of fabric simulation is more accurate and realistic. The luminance information of the scene is preserved based on fusion algorithm to make texture mapping more stereo. Introduce the processed texture and 3D model into the Web end for 3D display, which can help to evaluate the effect of different fabrics in different scenes.

**Keywords**—three.js; three-dimensional display; texture mapping; luminance fusion; cubic convolution interpolation

## I. INTRODUCTION

For the general methods of simulation of scene[1], the first step is to model the virtual scene. And the next step is to establish the surface model according to the form of the fabric in the 3D space. Then, the texture is mapped to the model with illumination model concerned. Finally, the fabric scene is generated. The 3D scene simulation system of this paper is divided into two parts, which are the 2D simulation system of fabric texture and the 3D display system. The function of the former one is to import the texture of 3D model used for display into computer and to define regions on the texture for mapping a specific texture. After computing, the pattern of the custom regions is replaced by the texture provided by the user. The latter one's function is to import the processed texture and 3D model into the web end and to construct virtual 3D scene. Compared to the general simulation, the operation is more convenient for replacing the 2D texture of a 3D model.

## II. THE DEFINITION OF CUSTOM AREA FOR MAPPING

The first thing of the system is to define the regions[2] to be processed. For example, in Figure 1, a business suit of a 3D model of a person can be divided into different parts such as buttons, pockets, collar and so on. For each part, a closed polygon area can be represented by a number of points.

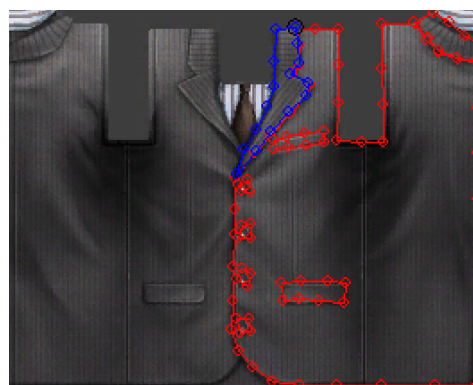


Figure 1. The definition of regions

## III. DIVIDE SURFACE MESH

The boundary curve of a grid whose shape can be arbitrary should be defined before dividing the grid. Because the four boundary lines of the grid must be divided before obtaining the grids in the grid, the curve fitting of the boundary curves is needed. So the method of accumulative arc length spline curve[2] is employed.

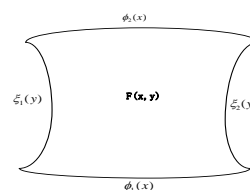


Figure 2. The parameter curves

There are many methods for the finite element meshing in the planar region, and Haber has proposed a bilinear interpolation method[3]. In Figure 2, the four arbitrary parameter curves  $\phi_1(x)$ ,  $\phi_2(x)$ ,  $\xi_1(y)$ ,  $\xi_2(y)$  which form a quadrilateral. Using the parameters (x,y) as coordinates, the 2D space F(x,y) can be constructed. For any point F(x,y) in the quadrilateral, the following bilinear interpolation formula can be used to calculate its coordinates:

$$F(x, y) = (1 - y)\phi_1(x) + y\phi_2(x) + (1 - x)\xi_1(y) + x\xi_2(y) - (1 - x)(1 - y)F(0,0) - xyF(1,1) - (1 - x)yF(0,1) - x(1 - y)F(1,0) \quad (1)$$

According to formula (1), the uniform segmentation of the boundary curve is firstly needed. Since the accumulated arc length spline curve is arc length, it can be divided easily. And then the coordinates of the interpolation points of the quadrilateral will be obtained by bilinear interpolation. In general, divide the quadrilateral into M\*N meshes. Firstly, divide the two opposite sides into M parts to get (M+1) points, which are, F(i,0)...F(i,N) (i=0,1,2...M). And (N+1) points can be obtained by the same way, which are, F(0,j)...F(M,j) (j=0,1,2...N). According to the bilinear interpolation formula (2), any point F(i,j) in the quadrilateral can be calculated through the 2\*(M+N) points.

$$F(i, j) = (1 - \frac{j}{N})F(i, 0) + \frac{j}{N}F(i, N) + (1 - \frac{i}{M})F(0, j) + \frac{i}{M}F(M, j) - (1 - \frac{i}{M})(1 - \frac{j}{N})F(0, 0) - \frac{i}{M}\frac{j}{N}F(M, N) - (1 - \frac{i}{M})\frac{j}{N}F(0, N) - \frac{i}{M}(1 - \frac{j}{N})F(M, 0) \quad (2)$$

where  $0 \leq i \leq M, 0 \leq j \leq N$ .

Attaching point F(i,j) and point F(i+1,j) ( $0 < i < M, 0 < j < N$ ) results in the latitudinal lines of the meshes, and the longitudinal lines of the meshes on the pseudo surface will be formed after attaching point F(i,j) and point F(i,j+1) ( $0 < i < M, 0 < j < N$ ). The expansion of the texture of the sleeve of the model of a person is shown in Figure 3.

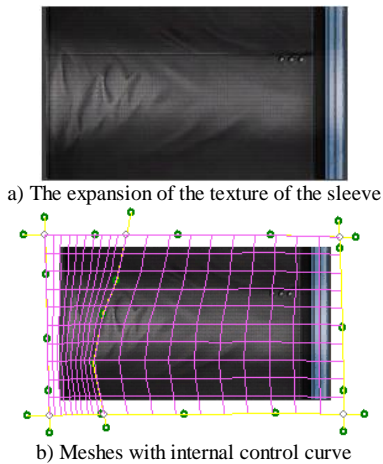


Figure 3. The boundary definition and mesh insertion

Because the texture pattern is not flat in many cases and some regions even have obvious folds, so simple replacement of texture does not work well. It is necessary to divide each defined region into meshes of the pseudo surface to obtain a better effect[4]. The folds and distortion in regions will be well simulated. Modifying the mesh boundary according to the fold trend and shape of the texture makes the simulation more realistic. In this paper, inserting control line within the mesh can overcome the problem of simulating folds and distortion. The internal control curve must intersect with the two horizontal or vertical lines of the curved quadrilateral to divide the original area into several parts. Each part is a small quadrilateral. Since the internal control curve is fitted to spline curves, so the meshes can be inserted by using bilinear interpolation directly.

#### IV. TEXTURE MAPPING

According to the size of unit grid of the curved mesh, divide the original texture to make sure that the unit square on the texture which is used for replacement corresponds to the unit grid of it. As is shown in Figure 4, texture mapping realizes the mapping from a square to an arbitrary irregular quadrilateral[5].

In Figure 4, (a,b,c,d) is the target quadrilateral. It is necessary to calculate the corresponding point PP(xx,yy) in (aa,bb,cc,dd) of an arbitrary point P(x,y) in (a,b,c,d) to realize texture mapping. And the point here means pixel.

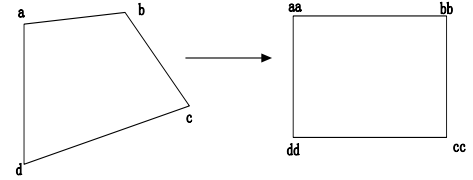


Figure 4. Texture mapping from source region to target region

Any point P(x,y) in the target quadrilateral can be expressed as follows:

$$x = (1 - \beta)(1 - \alpha) \times a_x + (1 - \beta)\alpha \times d_x + (1 - \alpha) \times \beta \times b_x + \alpha \times \beta \times c_x \quad (3)$$

$$y = (1 - \beta)(1 - \alpha) \times a_y + (1 - \beta)\alpha \times d_y + (1 - \alpha) \times \beta \times b_y + \alpha \times \beta \times c_y \quad (4)$$

where  $0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$ .

According to the above two formulas,  $\alpha$  and  $\beta$  is easy to be calculated. Then put  $\alpha$  and  $\beta$  into formula (5) and formula (6) to get the coordinates of the corresponding point PP(xx, yy).

$$xx = (1 - \beta)(1 - \alpha) \times aa_x + (1 - \beta)\alpha \times dd_x + (1 - \alpha) \times \beta \times bb_x + \alpha \times \beta \times cc_x \quad (5)$$

$$yy = (1 - \beta)(1 - \alpha) \times aa_y + (1 - \beta)\alpha \times dd_y + (1 - \alpha) \times \beta \times bb_y + \alpha \times \beta \times cc_y \quad (6)$$

The  $xx$  and  $yy$  here are both float. So the exact pixel cannot be obtained directly with this method. But this problem can be solved by interpolation. The concept of interpolation is to calculate the pixel value of a point based on a number of points around it by using interpolation formulas. The three most popular methods are: nearest point method, cubic polynomial interpolation method and cubic convolution interpolation method[6].

The nearest point method is to round up the coordinates of  $xx$  and  $yy$ . It is simple and fast. But the drawback is also obvious. Because of cutting the fractional part of the data, the mapping information of the pixel point is lost so that the scene after mapping is jagged and not smooth enough. So this method is not considered in 3D display depends much on details. The cubic polynomial interpolation method takes eight points around the target pixel into account. It makes use of the mapping information of the pixels to keep better quality of the details compared with the first method. The cubic convolution interpolation method[7] considers the effect of the adjacent 16 pixel values. It enhances the effect of edge and keeps the tiny structure of the image. Compared with cubic polynomial interpolation method, although its processing speed of the picture is slower, but the texture quality is better because of the better use of the mapping information of pixels.

In order to solve the deformation problem to obtain a better effect of 3D scene display, cubic convolution interpolation method is proposed in this paper. As is shown in Figure 5, assume that  $f(i,j)$  is the pixel value of the pixel  $P(i,j)$ , and  $(x,y)$  are the coordinates calculated from formula(5) and formula(6). So  $f(x,y)$  represents the pixel value of  $PP(x,y)$ . The coordinates  $x$  and  $y$  can be expressed as  $(i+u)$  and  $(j+v)$  respectively, where  $i$  and  $j$  represent the integer part and  $u$  and  $v$  represent the fractional part.

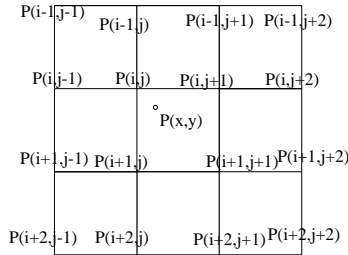


Figure 5. The 16 pixels around the target point

According to the continuous signal sampling theorem, if interpolation function  $S(x) = \sin(\pi x) / (\pi x)$  is used to interpolate the sample values, the original function can be restored accurately. It means that the value of any point between sample points can be obtained accurately. The fact of cubic convolution interpolation method is to use a three time polynomial to approximate the best theoretical interpolation function  $S(x)$  as formula (7) shows.

$$S(x) = \begin{cases} 1 - (|x| + 3) \times |x|^2 + (|x| + 2) \times |x|^3 & (|x| < 1) \\ -(4 \times |x| + 8 \times |x|^2 - 5 \times |x|^3) & (1 \leq |x| \leq 2) \\ 0 & (|x| \geq 3) \end{cases} \quad (7)$$

Take the  $f(i,j)$  and 15 pixel values around it as shown in Figure 5. In this paper, set firstly which is in formula (7), then pixel value of  $f(i+u,j+v)$  can be obtained by the following formula:

$$f(i+u, j+v) = A * B * C \quad (8)$$

where  $A = [S(u+1) \ S(u) \ S(u-1) \ S(u-2)]$ ,

$$B = \begin{bmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) & f(i-1, j+2) \\ f(i, j-1) & f(i, j) & f(i, j+1) & f(i, j+2) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) & f(i+1, j+2) \\ f(i+2, j-1) & f(i+2, j) & f(i+2, j+1) & f(i+2, j+2) \end{bmatrix},$$

$$C = \begin{bmatrix} S(v+1) \\ S(v) \\ S(v-1) \\ S(v-2) \end{bmatrix}.$$

After texture mapping, this method eliminates the jags of texture and makes the color transition smooth on the image. In Figure 6, the results of the cubic polynomial interpolation algorithm and the cubic convolution interpolation algorithm are compared. It is clear that the cubic polynomial interpolation is less fine than the cubic convolution interpolation about reduction of texture because the color of a part of the pixels is different from ones around them and the texture pattern is fuzzy compared with the cubic convolution interpolation. The cubic convolution interpolation algorithm has a higher degree of reduction of texture color and improves the definition of texture mapping and even smoothes the jags effectively.

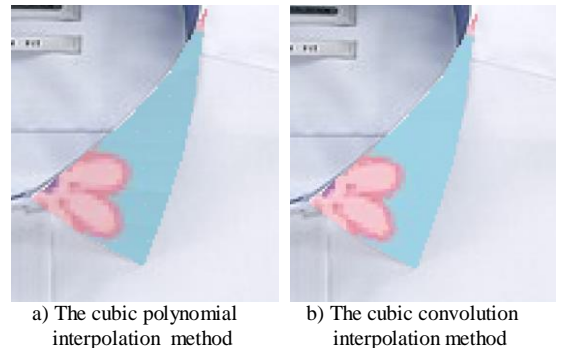


Figure 6. The texture mapping rendering of two interpolation methods:

## V. TEXTURE FUSION BASED ON LUMINANCE

It is possible to map any fabric texture to the target region of the original image with the obtained curved mesh. But without the luminance information of the original image, the result will be lacking a sense of reality and lost some information of the original pattern. Therefore, it is needed to transfer the brightness information that decides

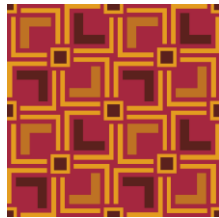
both the shadow and the fold in the original image to the texture pattern. Because calculating the luminance information reversely is a very difficult task in the field of computer vision[8], an approximate method is adopted in this paper.

HSI color model[9] is a color description method. It has three elements: the H (Hue), S (Saturation) and I (Intensity). Since each element of HSI is interrelated, directly replacing I component is easy to make the texture distortion after texture mapping. So both the intensity of fabric texture and original image should be taken into account. Firstly, calculate  $H_0, S_0, I_0$  value according to the original image's  $R_0, G_0, B_0$  value. Secondly, calculate the  $H, S, I$  value of the mapped points obtained by the cubic convolution interpolation method. Finally, adjust I value based on the intensity of original image and fabric texture and keep the  $H, S$  value. According to formula (9), obtain the final  $R', G', B'$  value from the final  $H', S', I'$  value.

$$\begin{cases} H' = H \\ S' = S \\ I' = (1-k)I + kI_0 \end{cases} \quad (9)$$

K the weight of intensity determines the rate of the luminance value of the fabric texture and the original image. And the bigger k is, the closer the intensity of the mapped texture and the shadow is to the original image[10]. In this paper, it is set to 0.5. And the visual effect is very well in most cases.

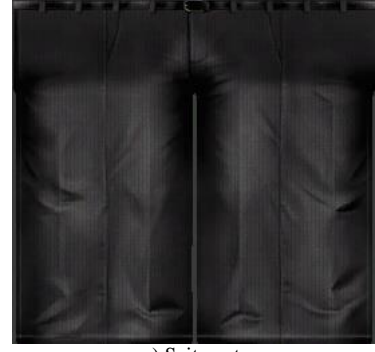
So far, all processing of the texture of material required by 3D scene is finished. The picture b) in Figure 7 shows the simulation of the sleeve in Figure 3 with internal control line based on intensity fusion and cubic convolution interpolation. It can be clearly observed that texture mapping restores the tensile of the texture in the original image. And observe in the texture mapping to the pants more clearly that the shadow and the folds of the original image are fairly well restored. The two mapped images have a sense of stereo.



a) A fabric texture



b) Texture mapping on a sleeve



c) Suit pants



d) Texture mapping on suit pants

Figure 7. The simulated picture based on the intensity fusion

## VI. BUILDING THREE-DIMENSIONAL SCENE

Due to the security restrictions of reading file, JavaScript has no authority to directly read the local file from the HTML page. It is necessary to configure a Web application server on the computer and Tomcat server is selected in this paper. The Tomcat server belongs to the lightweight application server, and is widely used in the medium and small systems.

When the Tomcat server is ready for 3D display, build 3D scene in Chrome browser with Three.jslibrary. Three.js[11] is a light weight cross-browser JavaScript library used to create and display animated 3D computer graphics on a Web browser. Three.js uses WebGL. The source code is hosted in a repository on GitHub.

At first, an empty HTML frame page should be created, and imported. Three.js file in head <head> tag. "OBJMTLLoader.js" and "MTLLoader.js" is essential to import OBJ model and MTL material file. The display of a 3D scene is separated from three elements, namely, the scene, the camera and the renderer. The scene is a container that can contain models, coordinate axes and lights and other elements. Whenever a new element is created that needs to be displayed in the scene, it can be added directly to the scene. The camera defines the projection mode of the 3D space to the two-dimensional screen, which determines what the user can see from the page. By setting the camera position and target coordinates, the object can be simulated in reality from different angles and different distances. And rendering is the essential process of the dynamic display of the model data on the screen. Through the rendering process, the results of the previous rendering can be modified, including color and location information. That is, by rendering, the imported model file, the location of the lights, color and other

information can be modified and can even keep the object in the scene moving randomly.

## VII. THREE-DIMENSIONAL SCENE DISPLAY

In order to observe the model from all angles, it is necessary to add the appropriate control mode for the scene and the "OrbitControls" control mode is adopted. When using this control mode, pressing left mouse or right mouse can rotate the camera around the center of the scene. Roll mouse wheel to zoom in and zoom out the object in the scene. The observation point and the size of the object to be observed can be adjusted if needed.

The effect of the 3D model of a man mapped by the texture in Figure 7 is shown in Figure 8. After adopting the "OrbitControl" control mode, observe the effect of the fabric texture on a man from any angle. In order to get a better visual effect, change the type of light source, color and position. Furniture and house model can even be imported to get the visual effect in various occasion[12].



Figure 8. 3D display for evaluating the texture of fabric

In picture a) of Figure 9, it can be seen that after adding light sources behind the model, the original dark place becomes very bright and we can observe it clearly. And in picture b) of Figure 9, evaluate the effect of the human model with a couch behind. Add and delete any object in the 3D display scene if needed.



a) Add light sources in the scene



b) Add a couch in the scene

Figure 9. Add objects in the scene for a better effect

The frame of the 3D scene simulation system for evaluating texture of fabric is shown in Figure 10.

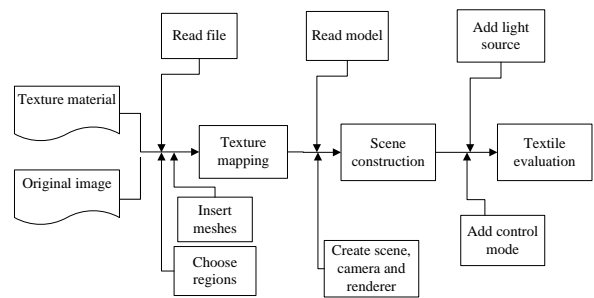


Figure 10. The frame of the 3D scene simulation system

The 3D scene simulation system consists of texture mapping, scene construction and textile evaluation. First of all, import an original image and some fabric images, and then divide the former one into several regions. Next, insert meshes for each region by the bilinear interpolation algorithm to obtain target regions for texture mapping. Then, calculate the pixel value based on the cubic convolution interpolation algorithm and intensity fusion. Finally, construct 3D scene after importing models and textures for evaluating a specific fabric texture.

## VIII. CONCLUSIONS

The three-dimensional scene simulation system for evaluating texture of fabric realized in this paper can simulate the visual effect of specific fabric texture in a specific scene. After the texture replacement of the custom region, the original image can be reduced to the folds and stretching and the accuracy of the simulation is high. Using the cubic convolution interpolation algorithm for texture mapping, the simulation results are more accurate, the texture is clearer, and the color reduction is better. The simulation effect is more stereo and real by using the fusion algorithm based on luminance. When using Three.js library for 3D display in the web page, continue to add the environmental elements if necessary. The whole scene is under control all the time and can be observed from any angle. The system implemented in this paper is suitable to evaluate the fabric texture designed by a textile enterprise, and it is also suitable for showing the wear effect of the clothing to customers for selling. But designers are required to design specific 3d models.

#### ACKNOWLEDGMENT

This project is supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant 2013BAH58F02.

#### REFERENCES

- [1] Zeng F T. Texture-mapping Technology Based on OpenGL[J]. Computer Knowledge and Technology (Academic Exchange), 2007, 3: 121.
- [2] Wang M Z, Wang F M, Wang S Y, et al. Simulation and computer realization of the three dimension of fabric in virtual environment based on mesh model[J]. Journal of Donghua University (Natural Science), 2006, 6: 72-76.
- [3] Liang X K. Step-up accumuative bowstring & arc length regenerated quadratic parameter spline curve[J]. Computer and Information Technology, 2001, 2: 36-39.
- [4] Haber R, Shephard M S, Abel J F, et al. A general two - dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings[J]. International Journal for Numerical Methods in Engineering, 1981, 17(7): 1015-1044.
- [5] Xu Q X, Zhang S L. Software development for textile 2D environment simulation[J]. Journal of Jiangnan University (Natural Science Edition), 2007, 6: 314-318.
- [6] Stoer J, Bulirsch R. Introduction to numerical analysis[M]. 3rd, Vol.12, New York: Springer, 2002: 38-51.
- [7] Fu X, Guo B L. Overview of image interpolation technology[J]. Computer Engineering and Design, 2009, 1: 141-144.
- [8] Meijering E, Unser M. A note on cubic convolution interpolation[J]. IEEE Transactions on Image Processing, 2003, 12(LIB-ARTICLE-2003-008): 477-479.
- [9] Guo Y W, SUN H Q, PENG Q S, et al. Mesh Optimization Based Image Texture Replacement[J]. CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-, 2007, 30(9): 1580-1587.
- [10] Rogers D F. Procedural elements for computer graphics[M]. New York: McGraw-Hill, 1985.
- [11] Jin M J, Jiang H W. Fusion Image Quality Evaluation Algorithm Based on Human Visual Features[J]. JISUANJI YU XIANDAIHUA, 2013(3): 184-188.
- [12] Jos Dirksen. Learning Three.js: The JavaScript 3D Library for WebGL[M]. Birmingham B3 2PB, UK : Packt Publishing Ltd, 2013 : 7-20.
- [13] Sawicki B, Chaber B. Efficient visualization of 3D models by web browser[J]. Computing, 2013, 95(1): 661-673.