

Image Information Extraction Model Based on Multi-features of K-nearest Neighbors

Anmin Xu^{1,a}

1.School of Computer Science & Technology,Shandong University, Jinan, 250101, China

a.Email: aminxu1128@gmail.com

Keywords:K-nearest neighbor; multi-feature; BRDF modeling; information extraction

Abstract.Bidirectional Reflectance Distribution Function (BRDF) and Bidirectional Texture Function (BTF) are two major methods used to describe the reflectance of light at a surface under different illuminations and different views. However, due to a large number of measurement data, an efficient compression method is needed. Meanwhile, the requirements of real-time rendering should be met. Under this background, this paper mainly does some exploratory research on BRDF modeling methods based on the qualitative measurements of textures. By conducting BRDF modeling on the measurements derives with different acquisition devices, it obtains the illumination properties of textures. Finally, combined with the property information of textures on the surface of object, under a new light source, from a new viewpoint, it renders the relighting effect of the object under a new light source.

Introduction

Animations, TV and films, advertising, product CAD, fashion CAD and other fields all require a highly realistic 3D representation of objects. A highly realistic representation of objects depends on their geometric modeling and texture modeling. Geometric models describe the shape of objects, while texture modeling describes the reflection of objects under illuminations. Due to different surface geometries, materials, densities, colors and other properties, different objects with the same geometric model present complex and rich optical phenomena under different illuminations. Geometry reconstruction enables us to obtain the geometrical structure of objects, with active reconstruction methods, such as 3D structure light scanners, 3D laser scanners or passive reconstruction methods, such as reconstructing 3D models from multi-angle pictures. By contrast, it is very difficult to reconstruct the properties of textures on the surface of object. This is because measurements of the accurate surface reflection properties of the whole object need to be described with very high dimensional data. In the case of high dimensional data, data collection, modeling and real-time rendering will bring great difficulties. Therefore, how to realize texture modeling efficiently has been a hot issue of research and a focus of attention in academic and application fields. Although in recent years, research on surface texture modeling and rendering has made substantial progress, the modeling of complex surface textures, especially the modeling of the surface reflection properties of objects still face great challenges.

Support Vector Machine (SVM) and Improved K-nearest Neighbors Algorithm

Support Vector Machine Algorithm

Support vector machine algorithm (SVM) is an epitome of machine learning and research, which is applicable to the solution of small sample, high dimensional and other nonlinear decomposition problems. The goal of recognition is to find an optimal separating hyperplane, separate all data points and try to make the distance between points and hyperplanes the farthest when all constraints are met, as shown in Figure 1.

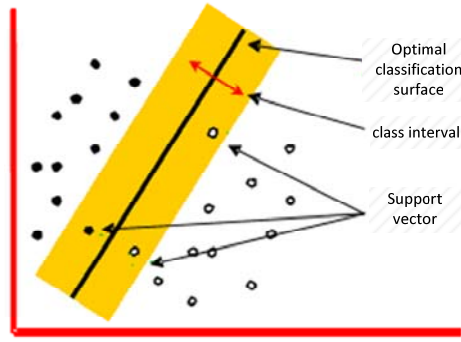


Figure 1 The optimal separating plane

Give a data set $\{x_i, y_i\}, x_i \in R^n, i=1, 2, \dots, n$, x_i represents the recognition features of measurements, y_i represents the pattern of measurements. According to the principle of risk minimization, the optimal hyperplane of SVM is represented as:

$$y = \omega^T \Phi(x) + b \quad (1)$$

Where ω is the weight vector and b is the offset vector.

It is difficult to solve a linearly inseparable problem directly using SVM. Therefore, a non-negative relaxation factor is introduced to transform the problem into a quadratic optimization problem, namely

$$\begin{aligned} \min J(w, \xi) &= \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \dots, n \end{aligned} \quad (2)$$

Where ξ_i is the training error of the i th sample; C is the penalty parameter.

Formula (2) is a typical convex quadratic optimization problem. It can be solved by introducing Lagrangemultiplier and being transformed into a dual problem, namely

$$\begin{aligned} \max W(\alpha) &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \alpha_i \alpha_j y_i x_i \cdot x_j \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^l \alpha_i y_i = 0 \\ C \geq \alpha_i \geq 0 \\ i = 1, 2, \dots, l \end{cases} \end{aligned} \quad (3)$$

Where α_i stands for the Lagrangemultiplier.

The value of α_i can be obtained by solving Formula (3). Then the weight vector ω can be represented as:

$$\omega = \sum \alpha_i y_i \Phi(x_i) \cdot \Phi(x) \quad (4)$$

In this way, SVMclassificationdecision function is transformed into:

$$f(x) = \text{sgn}(\alpha_i y_i \Phi(x_i) \cdot \Phi(x) + b) \quad (5)$$

It is difficult to perform dot product operation in a high-dimensional space directly. Replace the dot product $\Phi(x_i) \cdot \Phi(x_j)$ with a kernel function. The decision function can be transformed into:

$$f(x) = \text{sgn}(\alpha_i y_i k(x, x_i) + b) \quad (6)$$

The radial basis function (RBF) only needs to determine the width of the kernel function, σ , thus the present study chooses RBF function as the kernel function of SVM. Finally, the SVMclassificationdecision function based on RBF function is:

$$f(x) = \text{sgn}(\alpha_i y_i \exp\left(-\frac{\|x - x_i\|}{2\sigma^2}\right) + b) \quad (7)$$

Improved K-nearest Neighbors (KNN) Algorithm

K-Nearest Neighbors (KNN) Algorithm is a classification method based on statistics. The basic principle is to give an unknown sample x and find out k nearest training samples as k neighbors. Classify the unknown sample x to a category containing the most KNNs.

Suppose k_1, k_2, \dots, k_c are K -nearest samples of the unknown sample x , $d_i(x)$ is the discrimination function of w_i category. $d_i(x) = k_i$, $i = 1, 2, 3, \dots, c$. The discrimination criterion is: if Formula (8) is met, then $x \in w_m$. When $d_m(x) \geq m-1$, the discrimination is regarded as correct.

$$d_m(x) = \max\{d_i(x)\}, i = 1, 2, 3, \dots, c \quad (8)$$

The confidence level of the KNN classifier is defined as:

$$T = d_m(x) / k(m+1)T \quad (9)$$

Where (x) is the number of neighbors of the sample through category judgement with KNN and k is the number of neighbors of the sample, d_m .

In actual applications, the distribution of image data is uneven. Since KNN chooses k samples with maximum similarity from the whole training set as a reference, the chance for large category samples to be selected is larger than that of small category samples, which affects the classification accuracy of the KNN classifier. In Figure 2, it seems that the test sample x should belong to Category 0, but the distribution density of Category 1 is greater than that of Category 0. When judging the category that x belongs to by defining $k=9$, the classifier discriminates the test sample x as Category 1 and gets a wrong judgment.

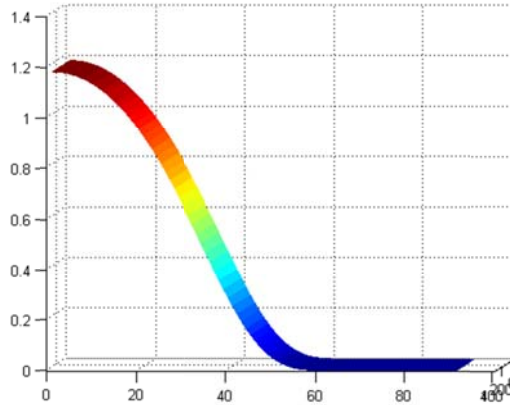


Figure 2 The Impact of Sample Non-uniformity on KNN Performance

To eliminate the negative impact of uneven distribution of illumination training samples, a weight factor is used to adjust, i.e., to increase the contribution of a sparse sample area to its category and reduce the contribution of a dense sample area to its category. In the training set D , $\forall d_j \in D$ is used as the average similarity (sample density) between d_i and its KNNs, then

$$p(d_j) = \frac{1}{k} \sum_{d_j \in KNN} \text{sim}(d_i, d_j) \quad (10)$$

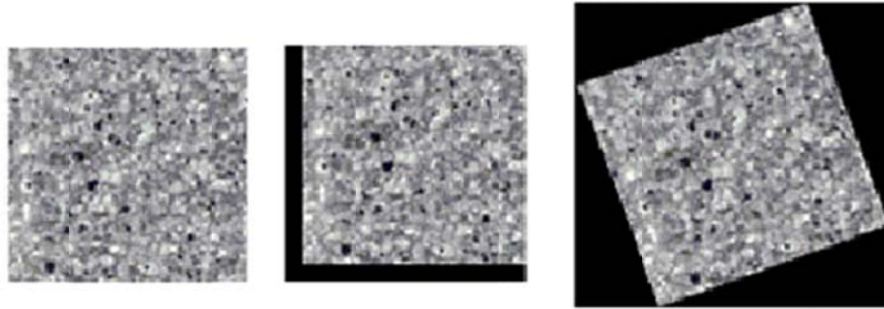
Adjusting with the reciprocal of the density of inter-category samples $w(d_i) = 1 / \rho(d_j)$ as the weight factor can reduce the contribution of a dense sample to its category and increase the contribution of a less dense sample to its category correspondingly, thus reducing the impact of the unbalanced distribution of training sets effectively.

Experiments and Analysis

Simulation Experiment

The accuracy and reliability of algorithms in this paper is verified by speckle images with

known deformation parameters. The calculated results and calculating time are compared with a commonly-used illuminationsubpixelalgorithm in related calculation, i.e., Newton-Raphsoniterationmethod. Figure 3 (a) is a reference speckle image. Translate it 10.5 pixels in the positive direction of x and y, respectively. The image after translation is shown in Figure 3 (b). Rotate the reference image clockwise by $\theta=20^\circ$. The resulting image is shown in Figure 3 (c). The accuracy of the resulting value of u , v , u_x , u_y , v_x and v_y and angle values is verified by translating and rotating the image.



(a) A reference speckle image
(b) The speckle image after translation
(c) The speckle image after a clockwise rotation of 20°
Figure 3 speckle image

One-way Fatigue Tensile Test

To verify the validity of the method, a one-way fatigue tensile test is conducted on a sample with a round hole in it (the tensile speed is set to 2 mm/min). Calculate its full field deformation. Before test, it is necessary to spray lacquer and make artificial speckles on the surface of sample. Spray white flat lacquer first, as the background color and then spray a little black flat lacquer to form speckles randomly. Clamp the sample onto a tensile machine. The images of sample surface before and after deformation are collected with a CCD camera. Figure 4 (a) is the image of sample surface before deformation; Figure 4 (b) is the image of tested sample surface after deformation.

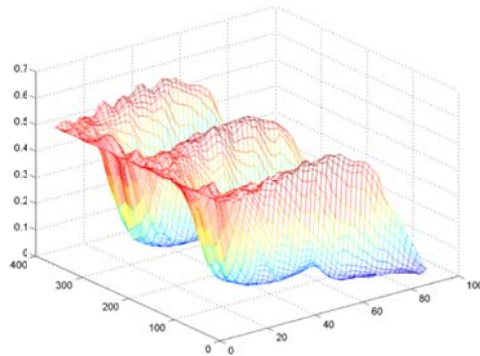


Figure 4 (a) The image before deformation

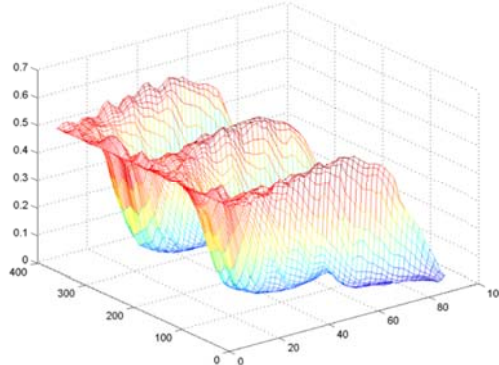


Figure 4 (b) The image after deformation

Related operation is performed on images before and after illumination deformation. The subarea of calculation is 31×31 pixels. The distance between two adjacent calculation points is 5

pixels. Interpolation is conducted on the derived sparse displacement field. The resulting full-field deformation image is shown in Figure 5. Since the load is in the x direction, the deformation in the x direction is obviously bigger than that of the y direction. The stress in the hole is concentrated. This is consistent with the actual situation.

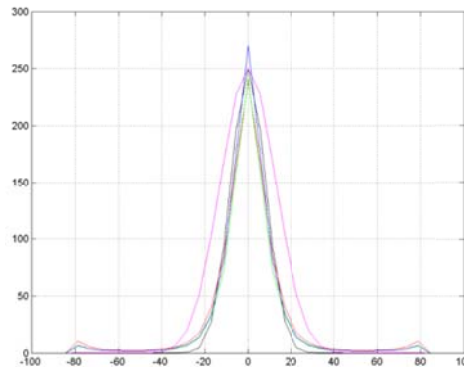


Figure 5 (a) The displacement field in the x direction

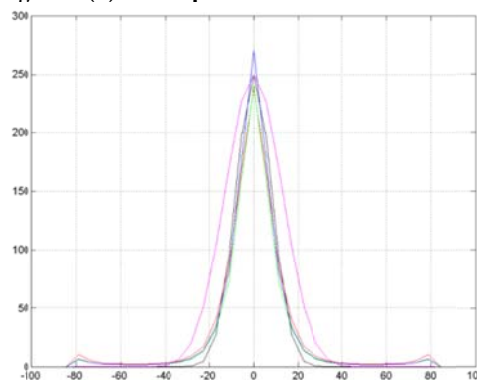


Figure 5 (b) The displacement field in the y direction

Conclusion

In current SVBRDF fitting methods, textures are often two-dimensional planar textures. Generally texture pixels shouldn't have inter-reflection, self-shadowing, self-shielding and other effect. Many acquisition methods require textures to be isotropic. And BTF materials usually have uneven textures and tend to present very strong self-shadowing, self-shielding and other effect. For this reason, common SVBRDF fitting methods don't have an ideal fitting effect on BTF data. First of all, BTF-acquired data are divided into two parts, geometrical features and illumination properties. Next the effect of indirect illumination in BTF data is eliminated and parameters are analytically fit, using the decomposed geometric features.

Reference

- [1] Cover T M, Hart P E. Nearest neighbor pattern classification[J]. Information Theory, IEEE Transactions on, 1967, 13(1): 21-27.
- [2] Ngan, Addy, Frédo Durand, and Wojciech Matusik. "Experimental Analysis of BRDF Models." *Rendering Techniques 2005* (2005): 16th.
- [3] Menzel N, Guthe M. g - BRDFs: An Intuitive and Editable BTF Representation[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2009, 28(8): 2189-2200.
- [4] Tatarchuk N. Advances in real time rendering, part II[C]//ACM SIGGRAPH 2015 Courses. ACM, 2015: 2.
- [5] Klein R. Possibilities and Limitations of the Bidirectional Texture Function as Appearance Representation[C]//Color and Imaging Conference. Society for Imaging Science and Technology, 2015, 2015(1): 1-1.