Journal of Electromagnetic Interference and Compatibility(2019) Vol. 1: 1-4 DOI: 10.23977/jeleic.2019.11001
Clausius Scientific Press, Canada ISSN 2523-9031

Asymmetric user relationship strength calculation integrated into network structure and social habits

Wang Peng[1,a,*]

[1]*School of Economics and Management, Dalian University, No.10, Xuefu Avenue, Economic & Technical Development Zone, Dalian, Liaoning，The People's Republic of China（PRC）*
a. email：wangpeng1@dlu.edu.cn

Keywords: *Asymmetry, network structure, social habits, user relationship strength*

Abstract: Along with the creation and exchange of a large amount of user content, large-scale interactive data and complex user relationships have emerged in the social network platform, which has attracted more and more researchers' attention. However, the existing research on relationship strength is mostly from user feature attributes. Similarity and social mobility are carried out, ignoring the influence of network structure on relationship strength, and does not consider the directionality and habitual problems of social power. Based on this, this paper proposes an asymmetric social network user relationship strength calculation. Method, the error method combines the user feature attribute similarity, the network structure connection strength, the social interaction strength three dimensions to comprehensively calculate the user relationship. When calculating the network topology connection strength, not only considers the number of neighbor nodes between users, but also considers The directionality and habituation of the social interaction force of the neighbor nodes will affect the user's perception of the relationship strength. Therefore, this paper calculates the contribution weight of different social forces, and finds the user's social strength from the interaction. The proposed method of asymmetric user relationship strength can Research findings and information dissemination mechanism microblogging opinion leaders in customer relationship strength forecast accuracy helps.

1. Introduction

3D rendering research is continuously striving to come closer to a physically realistic representation of realworld surfaces, as it is ever more widely applied to fields where a high degree of realism is required (e.g., the 3D games industry) or where rapid prototy pingis applied in early stages of design (e.g., in the automotive and textile industries). The approach of measuring ABTFs (approximate bi-directional texturing functions) is one way to avoid the need for synthetic design of material models, as it used to be the standard for a long time even though requiring significant manual effort. ABTFs were first proposed, who used a quarter light arc with incident illumination from point lights, mounted at angles ranging from zero elevation (grazing angles onto the material surface) up to nearly vertically incident light directions. The acquisition of optical material behavior, while already spatially varying due to the matrix sensor of the camera, relied on the assumption of isotropy, meaning that the material sample response is not affected by its rotation around its surface

normal. The resulting ABTF data consisted of two spatial dimensions (within the sample surface) and one dimension for light variation, leading to three dimensions in total. Many materials cannot be faithfully represented by assuming isotropy, as they are a combination of many different materials each with different physical structure, and thus reacting differently at each individual surface point to changes of the incoming light direction around the surface normal. Thus, an additional dimension of incoming light direction was added. By using a turntable which rotates the sample around its surface normal under the camera, 4D ABTF model results from combining the two lateral dimensions with two angular dimensions describing the hemisphere of incident light directions. Also, the acquisition process was entirely automated and reduced to only a few minutes per sample. Effectively, the combinatorial use of rotary and quarter light arc leads to an illumination hemisphere with the camera at its center, looking vertically down on the sample. The ABTF material model acquired and rendered with the technique represents actually captured real-world material behavior as a 4D texture that can be rendered in real time. By only considering one perspective vertically above the sample, it provides an abstraction of the higher dimensional 6D BTF (bidirectional texturing function) that captures the spatially varying material behavior of flat materials, discretized to the resolution of a matrix sensor (2D), for all combinations of incoming light (2D) and outgoing observer direction (2D), while preserving the dependence on the incoming light direction. [1]

2. Technical approach

2.1. Texture synthesis and periodization

The algorithm starts with an empty destination texture matching the input image dimensions. It consists of four phases.

Phase 1: Patches are placed at all corners to ensure periodicity, which is achieved by choosing a random source patch and splitting it vertically and horizontally through its center for transfer to diagonally opposing corners of the target texture. As the inner patch split edges coincide with the respective outer texture edges, appending the target texture at any edge reunites the original patch, thus leading to artifact-free periodicity for the corner pieces.

Phase 2: A similar idea is followed to fill the horizontal edges of the target texture, with two differences. Firstly, the source patch to be transferred is now horizontally split into two halves which then are transferred to vertically opposing edges of the target texture, ensuring periodicity as in Phase 1. Secondly, to avoid deterministic behavior, the patch to be transferred is now chosen at random from a set of best matching patches[2].

2.2. Visual similarity metric

The metric used to determine optical similarity to guide the process of finding best matching patchesfor transfer (see Section 2.3) is a monochrome error image I, covering exactly the area of overlap currently considered. The intensity for each pixel is computed as the sum of squared differences between source and destination image space for the three color channels evaluated at that pixel position. For 2D textures, the texture is the source image space for this computation, while for an ABTF dataset, consistency throughout the 4D texture must be maintained to avoid visual artifacts caused by inconsistencies between texture layers captured with different lighting directions, so a suitable image space must be chosen that represents the entire ABTF dataset.

2.3. Seamless transfer

Finding the free-form boundary cut following the path with the minimum cumulative error in I is a path-finding problem which can be solved using Dijkstra's algorithm. The algorithm for finding the free-form boundary cut presented here computes many path candidates in a set of sub-regions of I. Every pixel in a sub-region represents a node in the Dijkstra graph. Interconnections are generated for all adjacent pixels (only horizontally and vertically, not diagonally) using the average error (i.e., intensity value in I) between both pixels as cost. The inner region is masked out as this part of the patch is adopted unchanged and must thus not be crossed by the boundary cut. At first, only the sub-regions at the left, top, right, and bottom edges are considered. All combinations of start and end points on the yellow lines for each of the four highlighted regions are evaluated using Dijkstra's algorithm, leading to the path with minimum cost for the next step. The endpoints of the selected paths are then connected through the corner-regions which requires only one evaluation of Dijkstra's algorithm per corner. The full cyclic path represents the minimum error boundary cut[3].

3. Calculation process

3.1. Variation synthesis and rendering

Periodic regular textures can be mapped to large surfaces back to back without visible repetition artifacts, but in the case of irregular and stochastic textures, highly noticeable periodic patterns as in appear. This problem can be handled by generating tile sets, within which certain combinations of elements are periodic, and the variation of different combinations can be used to fill a regular grid and thus create a textured surface without visible seams and repetitions. Many tile sets are based on Wang tiles that define different types of edges and compute a tile for every combination of edge types so that there is always a compatible tile for a specific neighborhood of adjacent edges. Only tiles with the same type

3.2. Fitting

The smallest possible tile set using V = 2 corner types(V = 1 is equivalent to a single periodic texture) leadsto 24 = 16 texture variations in total.The total memory consumption depends on the imageresolution W × H and the sampling density of the virtual lighting hemisphere, expressed by R rotations of the sample and E discrete elevations of the light source, and amounts to M = 3W ×H ×R×E bytes. The data sets used in this work range from 300 MB to 1 GB, which leads to a total memory consumption of 4.7–16 GB including all texture variations. This amount exceeds typically available graphics memory and also would result in poor frame rates because of the lack of memory locality, as the renderer needs to access a widespread range of data when the surface is rendered under varying illumination. The key to realtime processing of large tile sets is thus compression. We use a fitting scheme we designed explicitly for this challenge. We limit the memory consumption for V = 2 corner types, or 16 texture variations, per ABTF data set to under 2 GB by fitting an analytic model based on the HSL color space. The choice of this specific color space for fitting enables reconstruction of the reflectance behavior for each single pixel on the sample surface without visual degradation.

4. Results

The synthesis method for measured ABTF material models developed in this work is based on image quilting which assembles a new texture by transferring small patches from an input image of

identical dimensions to different new locations within the new image while maintaining optical similarity within overlapping regions of neighboring patches. We extended it by adding initial reassembly phases during which the target texture is made periodic at its boundary regions so that it can be seamlessly concatenated border to border in both dimensions. Most importantly, we use the output of the algorithm not just to generate a new patch placement distribution for one texture, but instead for a non-deterministically generated set of patch transfer prescriptions applied to all ABTF data set layers, which both guarantees consistency for different light angles and provides a random patch placement for every texture variation to avoid repetition artifacts. The consequence of using an entire set of texture variations for all different illumination angles is respective increase in memory consumption. Exploiting the fact that only one dimension is significantly affected in HLS color space by changes in the direction of incident illumination, we compensate for the increased memory consumption by a novel fitting scheme. Consequently, the reflectance behavior for each single pixel on the sample surface is reconstructed without visual degradation, while 3D rendering of models mapped with texture variationbased ABTF data sets can still be done in real time.

5. Conclusions

We have removed the most significant limitation of an existing 4D texture-based system for acquiring spatially varying optical material behavior of real object surfaces, in that now neither textureseam artifacts nor repetition artifacts disturb the compelling visual experience when rendering the material applied to arbitrary 3D geometries, even if the textures are mapped to large surfaces with many repetitions. As future improvements, we see firstly that the quality of synthesized tile sets can be automatically adjusted according to the occurence of prominent patterns within the texture that draw the observer’s attention, since periodicity artifacts depend strongly on the texture class. Secondly, to avoid noisy rendering for very dense tiling due to poor surface sampling, mipmaps adjusted to texture variations can be used that store a texture pyramid with increasing resolution to adaptively react to the sampling resolution.

Acknowledgements

This article was specially funded by Dalian University's 2019 Ph.D. Startup Fund (20182QL001) and 2019 Jinpu New District Science and Technology Project.

References

[1] Pouria Sadeghi-Tehran, (2017) Multi-feature machine learning model for automatic segmentation of green fractional vegetation cover for high-throughput field phenotyping, Plant methods, 1， 96-108

[2] Sung Kyun Park. (2015) Construction of environmental risk score beyond standard linear models using machine learning methods: application to metal mixtures, oxidative stress and cardiovascular disease in NHANES, Environmental Health, 1, 356-374.

[3] Tanchanok Wisitponchai. (2018) AnkPlex: algorithmic structure for refinement of near-native ankyrin-protein docking. BMC Bioinformatics, 2, 119-136.