Joint Virtual Reality Conference of EuroVR - EGVE (2011)
S. Coquillart, A. Steed, and G. Welch (Editors)

Short Paper: Acquisition and Management of building materials for VR Applications

P. Westner, M. Bues

Fraunhofer IAO Stuttgart, Germany
philipp.westner@iao.fraunhofer.de, matthias.bues@iao.fraunhofer.de

---

Abstract

This paper describes a workflow and system to digitize, manage and render large and fast changing sets of building materials, e.g. brick, concrete, ceramic tiles, for VR architectural visualization. We describe the use case of VR-based sampling of detached houses, i.e. the process of choosing the interior and exterior materials of the house to be built. This use case implies some specific requirements: The main material database is very large, in the range of several thousands of individual materials, and is subject to frequent change due to regular changes in the material manufacturer's product collections. In addition, a relatively large number of materials have to be rendered simultaneously in the VR visualization. These requirements imply some limitations on both the material acquisition process, which has to be handled by end users, and the rendering methods useable for this application, which have to deal with the available graphics memory and rendering performance resources.

The solution we propose in this paper combines a simple and easy to handle image acquisition process with an automatic generation of image resources and associated parameters, which together form the resources for a shader-based real-time rendering. Despite not being physically correct, this rendering provides a high fidelity in representation of the relevant materials. In addition to the description of our concept, we also discuss the results of its implementation and productive use for a manufacturer of detached houses.

Categories and Subject Descriptors (according to ACM CCS): I.2.0 [Virtual Reality]: —, I.2.4 [GPU Programming]: —, I.2.4 [Color, Shading, Shadowing and Texture]: —, I.2.4 [Architecture]: —,

---

1. Introduction

VR is already an established tool in archtecture visualization. Recently, manufacturers of detached houses show increasing interest in VR visualization, not only for customer acquisition, but also to improve the process of sampling. For example having the customers choose the materials for their house, by using a fully configurable VR model, in which the materials of most surfaces can be chosen from the actual material catalog and immediately visualized in the VR model. Obviously, a high fidelity visual representation of the surface materials is a crucial success factor for such a solution. Equally important, the full catalog of materials offered by the manufacturer has to be presentable to the customer, which leads to a quite large and frequently changing material database. The VR system for this use case must be able to render an at least moderately complex geometric model with a relatively large number of different materials at interactive frame rates.

2. Motivation

Our experience from a variety of VR architecture visualization projects shows a common problem in collecting all the visual materials needed for high-quality visualization, often leading to a fallback to lookalike materials as described above. In our particular use case of customizable buildings, this problem is even harder: If we want to realize a true VR-based building customization, we do not only need a rendering quality way above simple textured geometry, but also a process for acquisition and generation of the necessary rendering resources, for exactly the materials which the house manufacturer wants to offer - lookalikes are not acceptable. In addition, in order to be productively implementable in the relevant industry, this process must be manageable and usable by computer graphics laypersons. Once established, such a process can generate a material database which is usable for a variety of visualization applications and target media. Many projects of architecture visualization, also those

DOI: 10.2312/EGVE/JVRC11/083-086

with less need for customization functionality, can benefit from such a process and the resulting database.

3. Related Work

High-fidelity representation of surface materials is one of the key factors to visual realism, and, consequently, one of the most important research areas in computer graphics; [DRS08] provides an overview of techniques for material representation. As our use case requires handling of a large number of different physical materials, we concentrated on representation methods which can use acquired image data. A common method of digital representation of surface reflectance characteristics is the bidirectional reflectance distribution function (BRDF, [Nic65]), which defines the reflectance characteristics of a point on an opaque surface. As the specular portion of reflectance is depending on both the lighting and viewing angle, the BRDF is a four-dimensional function. For real materials, the BRDF can be represented numerically as a set of surface reflectance samples measured from different lighting and viewing angles, e.g. using a gonioreflectometer as described by Foo [Foo97]. For a sufficient sampling density, this results in large numerical databases. Extending the BRDF concept to materials where the BRDF varies over the surface, which is the case for most materials in our use case, leads to the the bidirectional texture function (BTF, [JNvGK96]), which further increases the amount of data to be stored to represent real materials. For example, the material samples in the BTF database Bonn [BTF11] were acquired for 81 lighting and viewing directions, resulting in a set of 6561 images with a resolution of 256x256 pixels each, for samples with a maximum physical size of 80x80 mm. For the acquisition of this imagery, two setups were used, one employing a moving camera and moving light source, the other, avoiding any moving parts, consisting of an array of 151 cameras and light sources in a half-spherical arrangement.

To reduce the amount of data for real-time rendering usage, various approximation and compression techniques for BTF scanner data have been proposed. For example, Kautz [Kau05] proposes a BTF approximation using only ten image samples for different lighting directions and a fixed viewing direction, assuming isotropy and only small-scale features in the surface texture, and not representing specular characteristics in the acquired BTF data. Compression methods like the one proposed by Havran et al. [HFM10] can reduce the amount of data to be stored by a two or even three orders of magnitude, but at the cost of a computationally intense decompression during rendering.

4. Application Requirement Analysis

The input of the following groups was used to specify the requirements for our material acquisition and rendering process:

Material Manufacturer (MM) The MM has direct access to the physical material, as well as new materials even before they go into mass production.

Industry Software Manufacturer (ISM) Produces a software tool which manages 3D geometry of buildings and material assignment, probably along with production management data. Often a SME with special knowledge of industry sector.

VR Software Provider (VSP) Provides a VR visualization software with high quality visualization.

Application End User (AU) Uses the software and material data for the products and projects which make up their business. If the AU is a house manufacturer, it has also access to physical material.

As a result of our analysis, we found the following constraints and requirements:

- The total number of materials for a single AU can quickly reach several thousands.
- The acquisition of material image data has to be cost-effective and simple, using low-cost acquisition hardware.
- For cost considerations, it should be possible to use existing image data provided by the MM.
- For the materials that cannot be provided by the MM the resources have to be acquired in another way, e.g. material samples photographed by the AU, who has them available in most cases.
- As a result from the above, the process has to be able to get along with photographic image data.
- Existing image data from MM was not useable in most cases.
- A wide range of different material types (e.g. wood, stone, ceramic tiles) must be covered.
- Visual parameters that cannot be automatically collected have to be specified, e.g. relief depth measure.
- For correct 1:1 scale visualization, the actual material scale must be recorded.
- There must be a way to synthesize materials that could not be captured or are not tileable. e.g. wood based materials.
- Meta-materials are required to describe surfaces which consist of multiple different elements (tiles), such as checkerboard layered floor tiles.
- Similar material resources should be usable by multiple materials.

Based on these requirements we designed the necessary guidelines, tools and a VR application which uses the materials from the process within a productive use case.

5. Our Approach

From the above requirements, it becomes clear that a full BTF-based approach is not suitable for the task at hand: Image acquisition is too complex and too costly, and either the amount of resulting data is too large to be manageable for the number of materials we must handle in acquisition and processing, or, if compression methods are applied, the runtime decompression is too costly to render all the materials occurring simultaneously in a typical scene. Thus, we had to trade physical correctness for manageability, while still providing a high-fidelity material rendering.

The basic idea of our approach is to render the materials using algorithms which can be efficiently implemented in GPU shaders, and which require only surface-related image resources such as diffuse texture, bump and normal maps. We analyzed and categorized the required range of material categories and developed a shader for each material category which yields the best representation of the particular material. This way, we could also incorporate material characteristics not represented in the BTF, such as occlusion effects from geometric surface relief, being important for materials like brick.

We further defined a standardized process of acquiring color images from the material samples, and use automatic or semi- automatic postprocessing to approximate the necessary shader resources, such as diffuse texture, height and normal maps, from these raw image data.

5.1. Acquisition of Image Resources

Most imagery initially available from MM and AU was taken under inappropriate lighting conditions and lacked the necessary scaling information and were thus not usable for our purposes. Thus we developed a material acquisition process with reproducible constraints to digitize the materials using a simple photographic setup. This process is described in a guideline for the AU, helping to ensure its correct application. It contains correct lighting condition setup, surface preparation, camera setup and image capturing. The area containing the material sample must be segmented from the image taken, which should happen automatically whereever possible. To that end, we designed a gauge consisting of three parts, each equipped with an AR-Toolkit [KB99] marker. Having placed these at the corners of the material sample, we are able to extract the actual material image area as well as its physical size in the same step.

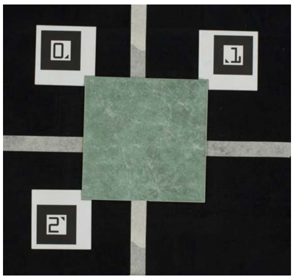

Figure 1: Example picture for image acquisition of a single ceramic floor tile.

Additional information not represented in the image, such as surface roughness of the material, is approximated by weighted parameters which are entered by the user through an electronic form associated with the image. These data are used to indirectly parameterize the shaders, as described below.

5.2. Image Postprocessing and Import Process

After acquisition and cropping, the other image maps which may be required by the shaders, such as height or normal maps, are generated from the original photographic image. Obviously, this process does not generate physically correct surface informations, but it gives a sufficient approximation in most cases. References to the resulting images, along with scaling and the other additional parameters are then inserted into the material database. Even for a limited number of materials, due to the high number of dimensions of the parameter space, a huge amount of data would result if we would pre-generate all occurrences of each material. To avoid redundant material generation, including generation of the required texture resources from the raw images, is deferred to the moment of request. The result is cached so that subsequent requests won't delay the generation process.

5.3. Categorization of Materials and Representation in Shaders

We categorized the materials into basic groups such as ceramic tiles, paint, or plaster, for two reasons. At first for shader development, in order to select the combination shading algorithms yielding the best visual result for a particular material. Secondly to provide the end user with a set of material parameters which are described in terms common to the user instead of exposing the raw shader parameters. For example, the user gets a parameter for surface roughness which controls both bump and parallax mapping of a brick shader. Each material category has its own particular set of parameters describing it completely. The mapping of these user-space parameters to the actual shader tweakables is described in a per-material data structure called the Shader Map. For the current implementation, we developed 21 different shader effects to efficiently use the memory and computing power of up-to-date graphics hardware, while using techniques like relief mapping (see [AMHH08]) to achieve maximum visual realism. The categorization of materials allows us to apply the best-fitting shading algorithms to each material. Environment maps for reflection mapping are provided by the rendering engine. For complexity reduction as well as to reduce artifacts from repetition for tiled materials, we incorporate a patterning technique inspired by [LN03] within the shaders. We store the shader resources for one or more base tiles, and use an additional small index texture, which requires a resolution of one texel per pattern tile, to construct arbitrary tile patterns, requiring texture space only for the base tiles and the index texture. In addition, we use this technique to pseudo-randomly rotate repeated tiles by multiples of 90/180 degrees for quadratic/rectangular tiles;

this considerably reduces repetition artifacts. Another degree of freedom handled within the shaders is rotation of tile patterns, e.g. for diagonal tile alignment. The shaders are implemented in the Cg language [MSK*03] and encapsulated in the associated CgFX effect format. Color plates 2, 3, 4, 5 show some examples of materials as rendered by our VR application.

6. Results

We implemented all of the previously mentioned features of the process for the German brick house maker Bauunion1905 in a special VR-setup [BHK11] which consists of a 3m x 2.4m stereo projection in a cinema-like ambience. For the planning and calculation of the houses, the integrated industrial software solution VI2000 from the German supplier Softwareparadies [SWP11] is used, which is capable of generating a simple, textured 3D view of the house being planned. The VR application developed by us is based on VRfx [BWDB07], with a user interface customized to the requirements of this particular use case. Our VR application communicates bidirectionally with VI2000 over network, obtaining the 3D geometry and material assignment information and providing information about material changes carried out during the customization session. The brick house maker acquired about 2000 material images and parameters, resulting in about 5800 distinct materials from 13 material categories, which are rendered using 21 different shaders. For a typical average house model with 422k triangles and decorated with 12 different material shaders, we achieve a frame rate of about 20 Hz on the productive system with two NVIDIA Quadro FX 5800 cards (one per eye).

The evaluation focus for us was the usefulness of the VR-based sampling process itself rather than the VR user interface, as the latter is operated by a trained user and thus is less important for the customer's user experience. Until today, Bauunion has led about 200 clients through the virtual building customization process. Bauunion presents a satisfaction questionnaire to every customer and statistically evaluates the results; this was extended by specific questions about the VR-based material choosing. About 80% say that VR-based sampling was helpful in choosing the materials for the building, compared to the traditional process of examining isolated material samples in a pyhsical exhibition. About 90% of the customers reported that the real house which was built later met or even exceeded their expectations they had from the virtual sampling process.

7. Future work

These results indicate that the visual quality we can obtain using the proposed process is sufficient for the use case. Nevertheless, further improvement of visual quality by incorporating new shading techniques is an ongoing process. The material acquisition process itself, while having proven to be applicable for the large amount of materials we encountered, still has to be made more end user friendly, by streamlining the current prototypic user interfaces for data import and database maintenance. Also, the integration of recent material image processing tools like Allegorithmic [ALL11] or CrazyBump [CRA11] is investigated, which is facilitated by the component-based structure of our material processing workflow.

References

[ALL11] Allegorithmic software, 2011. http://www.allegorithmic.com. 4

[AMHH08] Akenine-Möller T., Haines E., Hoffman N.: *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA, 2008. 3

[BHK11] Das Bauherrenkino, die weltweit einzige dreidimensionale Hausbemusterung - Bauunion 1905, 2011. http://www.bauherrenkino.de. 4

[BTF11] BTF Database Bonn, 2011. http://btf.cs.uni-bonn.de. 2

[BWDB07] Bues M., Wenzel G., Dangelmaier M., Blach R.: Vrfx: a user friendly tool for the creation of photorealistic virtual environments. In *Proceedings of the 4th international conference on Universal access in human-computer interaction: ambient interaction* (Berlin, Heidelberg, 2007), UAHCI'07, Springer-Verlag, pp. 624–632. 4

[CRA11] Crazy bump software, 2011. http://www.crazybump.com. 4

[DRS08] Dorsey J., Rushmeier H., Sillion F.: *Digital Modelling of Material Appearance*. Elsevier Science and Tecnology, 2008. 2

[Foo97] Foo S. C.: *A Gonioreflectometer for measuring the bidirectional reflectance of material for use in illumination computation*. Master's thesis, Ithaca, NY, USA, 1997. 2

[HFM10] Havran V., Filip J., Myszkowski K.: Bidirectional texture function compression based on multi-level vector quantization. *Computer Graphics Forum 29*, 1 (jan 2010), 175–190. 2

[JNvGK96] J.Dana K., Nayar S. K., van Ginneken B., Koenderink J. J.: *Reflectance and Texture of Real-World Surfaces*. Tech. rep., New York, 1996. 2

[Kau05] Kautz J.: Approximate Bidirectional Texture Functions. In *Graphics Gems 2*. ACM Press, 2005, ch. 11. 2

[KB99] Kato H., Billinghurst M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *in IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality* (1999), IEEE Computer Society, p. 85. 3

[LN03] Lefebvre S., Neyret F.: Pattern based procedural textures. In *ACM-SIGGRAPH Symposium on Interactive 3D Graphics (I3D)* (2003), ACM, ACM Press. 3

[MSK*03] Mark W. R., Steven R., Kurt G., Mark A., Kilgard J.: Cg: A system for programming graphics hardware in a c-like language. *ACM Transactions on Graphics 22* (2003), 896–907. 4

[Nic65] Nicodemus F.: *Directional reflectance and emissivity of an opaque surface*. Tech. Rep. 7, 1965. 2

[SWP11] Vi2000 application, 2011. http://www.softwareparadies.de. 4