CGI2012 manuscript No.
(will be inserted by the editor)

An Artist Friendly Material Design System

Wen-Chieh Lin · Jen-Hao Liao · Tsung-Shian Huang · Jung-Hong Chuang

Abstract Material design is an important issue in game and animation industry. It often takes artists a lot of efforts to create an elaborate material. Although there are already several successful material editing systems, it is still hard for artists to design a material on their own since these systems usually have neither an intuitive and sufficiently sophisticated material representation nor an artist friendly editing system. We propose a simple material creation flow and a material design system based on an intuitive material representation. With the system, an artist can bring their creativity into play to create materials that are realistic or even do not exist.

Keywords Material Design and Editing · Texture

1 Introduction

Materials such as brick, wood or cloth are everywhere in game and animation. It is an important feature to enrich a virtual world and makes objects look alive. Thus, creating an appealing rendering result of a material is crucial. Note that an appealing rendering result is not necessarily photo-realistic, since the world in game and animation is often full of imagination.

Material design is not an easy task, which often involves the collaboration of artists and technical artists. An artist designs the base textures and a technical artist designs the lighting phenomena of the material using the base textures as the input. Some parameters are then tuned by the artist for finer rendering results. Although several successful material editors have been developed, it is still hard for the artists to really manage these tools since the control of the appearance of a material is mainly restricted to the parameters provided by the technical artist. As there are no intuitive material representations and artist friendly editing tools to design the appearance of a material, many passes back and forth between the artist and the technical artist are usually needed to design an appealing material.

When considering the rendering of a material in reality, Bidirectional Texture Function (BTF)[2] is often a good choice. BTF is a six-dimensional function, involving lighting, viewing direction, and position. Since BTF data are captured from a real material, it can produce photo-realistic appearance. Many methods have been proposed to compress the BTF data. Some of the methods also come with some editing capabilities which are, however, usually limited. Such editing tools usually lack the editing abilities of a drawing system, which is much more friendly for an artist.

Recently, Menzel and Guthe [9] proposed a novel representation called geometric BRDF (g-BRDF), which decomposes BTF data into meso- and micro-scale properties represented by images (texture maps). Although some BTF editing results are demonstrated, their work focused on BTF modeling problem. More importantly, the input to their system are BTF data of materials, which are not conveniently available. These observations motivate us to develop a system that allows artists to design general materials via a single or a few images and requires less effort from the technical artists.

We propose a simple material creation flow and a material design system based on an intuitive material representation. The representation consists of the geometrical part, diffuse part and specular part of the material. Such intuitive representation allows artists to

W.-C. Lin · J.-H. Liao · T.-S. Huang · J.-H. Chuang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
E-mail: wclin@cs.nctu.edu.tw, howardlyliao@gmail.com, csie9624@gmail.com, jhchuang@cs.nctu.edu.tw

design a material without much difficulty. Given a single or a few images of a material, the artist can first generate the textures that represent the material by using some image processing tools and then manipulate and refine each texture using the proposed design system. Our design system is implemented as a plugin of an image manipulation program, thus all the manipulations can be viewed in real-time and the editing tools are much friendly to the artist. The capabilities of the editing tools are not restricted to what are currently provided in our system. Any other plugins can be added to strengthen the design process. With the proposed system, artists can bring their creativity into play to design complex materials intuitively with flexible controls over the appearance of the materials.

2 Related Work

Material Design. In game and animation industry, software and application programming interfaces, such as Unreal Engine, Frostbite Engine and RenderMan, provide a node-based shader synthesis tool to edit the appearance of a material. Users can design the appearance of a material in a visual programming way by connecting some operators (e.g., multiply, add, etc.) and operands (e.g., texture, texture coordinates, constant, etc.). Such connectivity is called a node expression, from which a shader code is automatically generated to render the material. Thus, users can design and render a material without knowing how to write a shader code. They can create a node expression that is well tailored to a specific material; however, it is difficult for an artist to design such expression since knowing how to manipulate all the operators and operands to create a material requires a lot of experiences and computer graphics knowledge. Moreover, the manipulation of the operands itself is not readily provided in these systems, e.g., painting the texture or warping the geometry. Artists need to tune numerical parameters to modify the appearance of the materials, which is nonintuitive and can be very tedious sometimes. Some basic texture editing, such as changing the color of a region, which can be done very easily through an image manipulation program, would be troublesome by using the node expression.

Recently, Dong et al. proposed an interactive material modeling system[3]. Given a single image lit by a known directional light source, their system lets the user interactively separate the image into an albedo map and a shading map, and then reconstruct the normal from the shading map to recover the geometric features. Although this system can reproduce the appearance of a variety of materials, it cannot handle complex specular phenomena, such as anisotropic reflection. Besides, the system does not provide more sophisticated editing abilities, which are crucial for material design.

BTF Compression and Rendering. To represent a material in reality, Dana et al. [2] introduced the BTFs. A BTF is obtained by taking lots of photos of a material under different lighting and viewing directions. Hence, a BTF of a single material may need several gigabytes for storage. Such enormous data size is not suitable for rendering a BTF on current graphics hardware. Thus several methods have been proposed to compress the BTF data, e.g., principal component analysis (PCA), matrix factorization, optimized Spherical Radial-Basis Functions (SRBF)[15]. These approaches reduce the high-dimensional input data to low-dimensional subspaces containing most of the information. Thus, BTF can be rendered in real time while preserving good visual quality. The compressed data do not offer meaningful parameters to control the appearance of the material. For a nice survey on BTF modeling and compression, please refer to [4] and [11].

BTF Editing. BTF editing has its own practical demands due to the inconvenience and difficulty of BTF acquisition. Existing BTF editing approaches can be roughly divided into two categories: direct editing approach [7,17] and fitting approach [9,16]. Kautz et al. proposed an out-of-core data management for editing raw BTF data [7]. They provided sophisticated operators to edit BTF appearance such as shadows, specularity and meso-structure. A drawback of this approach is that if the meso-geometry of the material is changed, the corresponding feature (e.g., shadows) does not change accordingly. Xu et al. proposed a novel editing propagation scheme to edit BTF data[17]. View-independent features, such as normals and reflectance reconstructed from each view, are used to guide the propagation process. Although their editing approach does not rely on explicit geometry, this also implies that it cannot edit the meso-geometry of a BTF material.

The fitting approaches adopt a BRDF model to fit the BTF data. These approaches are advantageous since BRDF models provide physically meaningful parameters for users to control; however, they usually lose accuracy as they either fit the BTF data using a single BRDF model or do not consider the underlying meso-geometry of the surface into account. Recently, Wu et al. [16] proposed a Sparse Parametric Mixture Model (SPMM) to represent general BTFs. They adopted the stagewise Lasso algorithm to fit a BTF to different types of multiple, rotated analytical BRDFs. This representation preserves the appearance while the storage size remains compact. The editing task is accomplished by changing the parameters of the analytical models;

however, without modeling the meso-geometry of a material, the control over the meso-geometry of the material cannot be achieved.

In general, to accomplish the editing task, most BTF editing approaches need a specific user interface tailored for a specific algorithm. Although a such user interface provides simple editing, e.g., changing color, adjusting specular intensity or surface roughness, these editing operations are usually global. There are almost no artist friendly tools that support editing for a certain local region. Moreover, as the user interface is not drawing-based, editing is less intuitive for artists. Recently, Menzel et al. [9] proposed a novel way called g-BRDF to represent a BTF. The BTF data is decomposed into a height map, which describes the meso-geometry, and some other maps that store the BRDF parameters. This method provides high compression rate and good rendering quality; however, as g-BRDF addressed the BTF modeling and editing problem, it is not intuitive for artists without technical background to use it for designing a material directly. Furthermore, it is also not easy to acquire the BTF of a material. This inspires us to develop a material design system that is intuitive and friendly for artists and other novice users.

3 Our Approach

Our material design system is built upon a material representation that is physically meaningful. Artist-friendly material creation and editing can be achieved by directly applying image editing and drawing tools to texture maps that constitute the representation of a material. We introduce our material representation model and material design system in this section.

3.1 Material Representation

We modified the g-BRDF proposed by Menzel and Guthe to represent a material for its physically-meaningful representation and photo-realistic rendering quality [9]. The g-BRDF is originally proposed for representing BTF data. For a given BTF, g-BRDF first extracts the meso-geometry (height map and normal map) from the BTF using the graph cut stereo technique [12] and then fits the BTF to a BRDF model with the meso-geometry information considered. Figure 1 illustrates the g-BRDF model. The g-BRDF employed Ashikhmin's distribution-based BRDF [1] as its BRDF model since it provides a good approximation for many real world materials

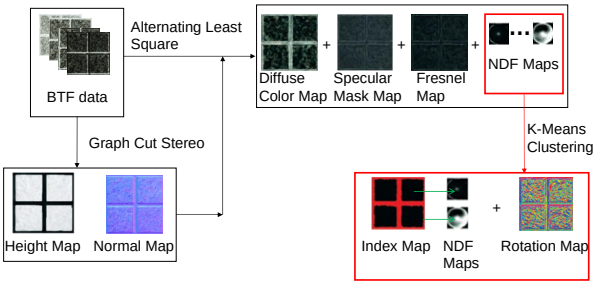


Fig. 1 g-BRDF decomposition workflow.

with intuitive meaning. The BRDF model is defined as follows:

$$\rho(L,V) = \frac{C_d}{\pi} + \frac{C_{s\_int} P(H) F(L \cdot H)}{L \cdot N + V \cdot N - (L \cdot N)(V \cdot N)}, \quad (1)$$

where L and V are the lighting and viewing direction, respectively, C_d is the diffuse color, C_{s_int} is the specular intensity, $P(H)$ is the normal distribution function (NDF) depending on the half vector $H = (L+V)/|L+V|$, and $F(L \cdot H)$ is Schlick's Fresnel term defined by

$$F(L \cdot H) \approx r_0 + (1 - r_0)(1 - L \cdot H)^5, \quad (2)$$

where r_0 is the reflectance at normal incidence.

As a BTF can be viewed as a combination of spatially-varying BRDF and meso-geometry, the parameters in Eq. 1 vary at different pixel locations. These parameters can be obtained by fitting the BTF data using an alternative least square approach and then stored in texture maps, which include a diffuse color map, a specular intensity map, a Fresnel map and several NDF maps. Each NDF map represents the NDF at each pixel location of the BTF (for an BTF with 256×256 image resolution, the number of the NDF maps would be 256×256). To reduce the storage size of NDF maps, all the NDF maps are approximated by some NDF maps with different rotations around the center using the K-Means clustering. An index map and a rotation map are additionally defined to indicate which NDF Map is used and how much rotation is needed for each pixel, respectively. To reproduce the NDF map of a pixel, the corresponding NDF map is retrieved according to the index stored in the index map, and then rotated around the center with the degree stored in the rotation map.

In g-BRDF, the same NDF is used for different color channels. The NDF function describes the percentages of the micro-facet normal at every direction on a hemisphere. It is stored in a parabolic map [6] and accessed by the half vector (more specifically, the inclination angle and the azimuth angle of H). To obey the law of conservation of energy, the sum of probabilities over hemisphere cannot exceed one. However, it is obscure for artists to manipulate NDF while satisfying this constraint.

To model more complex specular effects and provide better control, we use different NDFs for different regions of a material. Because $P(H)$ represent the micro-facet normal direction that can reflect the light into the viewing direction, the NDFs indeed is the specular color. Therefore, we replace the gray-level NDF maps by the RGB specular color maps in our representation. Also, we multiply the specular color maps by *Specular Mask Map* to allow a user to specify the specular regions easily. Finally, we introduce the *Rotation Map* and *Tilted Reflection Map* to control the reflection of the material. With these modifications to g-BRDF, the BRDF model in our material representation is expressed as

$$\rho(L,V) = \frac{C_d}{\pi} + \frac{C_{s\_mask} C_s(I_s, \tilde{H}_{RT}) F(L \cdot H_T)}{L \cdot N + V \cdot N - (L \cdot N)(V \cdot N)}, \quad (3)$$

where C_{s_mask} is the specular mask value; $C_s(I_s, \tilde{H}_{RT})$ is the specular color depending on a map index I_s, a rotated and tilted half vector H_{RT} ($\tilde{H}_{RT}$ denotes the inclination and azimuth angles of H_{RT}). H_T represents a tilted half vector. Eq. 3 denotes the BRDF at a pixel location. To encode the spatial variation of C_d, C_{s_mask}, I_s, rotation angle of the half vector, and the Fresnel terms, we store these material-dependent parameters into texture maps to represent the diffuse and specular appearance of a material. Figure 2 illustrates our material representation. We describe more details of these maps in the following section.

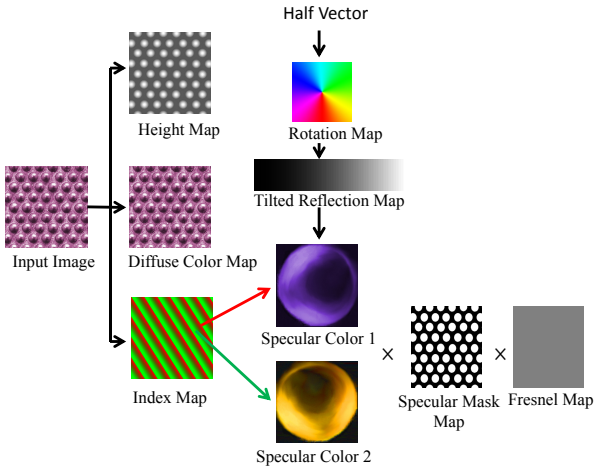


Fig. 2 Our material representation model.

3.2 Material Design System

Based on the proposed material representation model, we develop a material design system. Given a single or a few images for a material, our system constructs the texture maps that represent the material's meso-geometry, diffuse part, and specular part. Then a user can manipulate each part through image editing or drawing to design a desired material. For ease of user interface development, we implement our system as a plugin of GIMP(GNU Image Manipulation Program), which provides many standard image processing and drawing tools. We describe the construction and modification processes of the texture maps for representing a material and the intuitive interpretations of these processes.

3.2.1 Building and Editing Meso-geometry

Normal map and height Map. Some materials may contain self-shadowing and parallax effects due to its meso-geometry. To obtain meso-geometry information without the BTF data, we can either use photometric stereo (for the input with more than one images) or use the gray-level intensity of the image (for the input with a single image) to get the approximated height map and normal map.

In the case of multiple input images, the photometric stereo method proposed in [13] is adopted to recover the normal map by assuming that the material is a Lambertian surface and illuminated from a distant point light source. The normal integration method proposed by Frankot and Chellappa is then applied to recover the height map [5].

In the case of a single input image, the gray-level intensity of the image is a good approximation of the meso-geometry of those materials with less specularity and color variations. To use the gray-level intensity of an image as the height map of a material, it is important that the material is lit evenly from all directions to avoid the shadowing and specular effects. As the approximated meso-geometry may not be very accurate and a user may also want to make the material smoother or rougher, our system allows the user to edit the meso-geometry by manipulating the height map of the material; however, simply editing the height map may cause the inconsistency between the normal and height maps. To solve this problem, our system only stores the height map and the normal map is calculated by applying 3×3 Sobel filters to the height map at runtime. By convoluting the height map with the Sobel filters, we can obtain the derivatives along the x direction (dx) and y direction (dy), respectively. The derivative along the z direction (dz) can be calculated via $1 - \sqrt{(dx^2 + dy^2)}$. Thus the normal at a pixel location would be (dx, dy, dz). Since we calculate the normal map at runtime, the user can see the changes on the fly without manually rebuilding the normal map.

3.2.2 Building and Editing Diffuse Part

Diffuse color map. For the diffuse part of a material, we assume the surface is perfectly diffuse such that incident light scatters uniformly in all directions. Under this assumption, we can simply use the same input image as the diffuse color map of the material. If the surface for the input image is not perfectly diffuse, our system decreases the intensity of the diffuse color map to exclude the specular intensity. If the user is not satisfied with the diffuse color map, further editing may be applied to achieve the desired result.

3.2.3 Building and Editing Specular Part

To model complex specular effects, such as anisotropic reflection, and provide intuitive editing of specular effects, such as adjusting the shape, region, and angular dependency of specular highlights, we represent the specular part as a combination of an index map, specular color maps, a specular mask map, rotation map, a Fresnel map, and a tilted reflection map. Index map, specular mask map, Fresnel map, and rotation map are position dependent, i.e., the map value varying with spatial location, while the specular color map is angle dependent. In particular, index map can be considered as a rough classification of the specular properties of the material. For each class, a specular color map is used to describe the shape and color of specular highlights under different half vectors. The specular mask map is used to define if a region has any specular effects. The rotation map and tilted reflection map affect the orientation of highlight areas in anisotropic reflection. These maps not only provide a sophisticated model for specular effects but also have intuitive interpretations for artist friendly material design.

Index map. From the index map of different materials in [9], we observed that regions of a material having similar diffuse colors often share the same index. Thus we can use the color selection, edge detection, color mapping/remapping tools or any other image processing tools to segment the input image of the material into parts of similar color. Each segmented part is then indexed by an index map. Note that there are multiple specular color maps in our representation. An index map may contain multiple channels, where each channel stores the weight for each specular color map.

Specular color map. After segmentation, the next step is to design the specular color map for each region. The specular color is stored in a parabolic map and accessed using the half vector. Although using the half vector to access the specular color is straightforward, it may be still difficult for artists to design a parabolic map. It is not clear for artists to determine which pixel in the parabolic map is accessed from given lighting and viewing directions. To tackle this problem, our system provides a special designed window with nine viewports. Each viewport shows the appearance of the material at different lighting and viewing directions over the upper hemisphere (Figure 3). To let the user know which pixel in a specular color map corresponds to which lighting and viewing directions, we mark the lighting and viewing direction of each viewport as a red point on the 'Specular Position' block at the bottom right. The user can draw the parabolic map and receive the feedback in real-time. For artists, the specular color map can be intuitively considered as the shape of specular highlights. They can manipulate it by drawing on the specular color map. Figure 4 shows some examples.

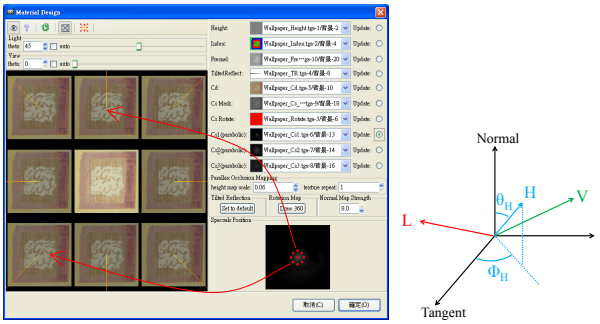


Fig. 3 Left: Design interface for specular color maps. Right: The half vector can be represented by θ_H and ϕ_H

Specular mask map and Fresnel map. For a material that is lit evenly from all directions, brighter regions tend to have lower ambient occlusion or higher reflection. Hence, we use the intensity of the diffuse color map as the initial specular mask map, and the user can change its brightness and contrast via our design system for better results. The Fresnel term expresses

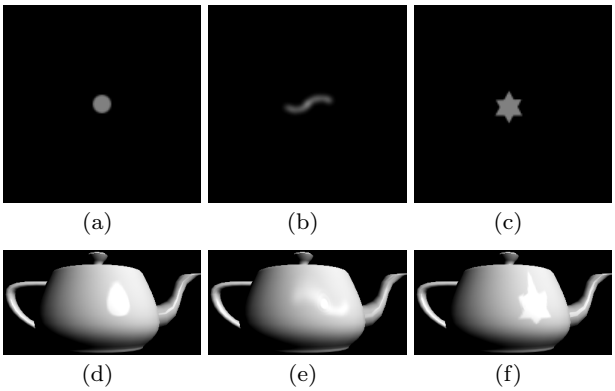


Fig. 4 The shape of specular highlight can be controlled by the specular color map.

the reflection of light on material. We adopt Schlick's Frensel term, which increases from r_0 to 1 as the angle between L and H increases from 0 to 90 degrees. The materials become brighter when they are illuminated at the grazing angle and viewed from the opposite side. Therefore, similar to the construction of the specular mask map, we also use the intensity of the diffuse color map as the initial Fresnel map. Its brightness and contrast can be adjusted later for better results. The specular mask map can be roughly considered as the location of the specular highlights.

Rotation map and Tilted reflection map. When dealing with materials with anisotropic reflection, i.e., the highlight changes continuously over the phi angle (e.g., disk and metal surface), it is hard to use a lot of discrete specular colors for representation. To solve this problem, the rotation map is introduced. Each pixel of the rotation map stores a rotation angle $\triangle\phi$ for the corresponding specular color map. Given a half vector H, whose direction is (θ_H, ϕ_H) as defined in Figure 3, the rotated half vector is represented as $\tilde{H}_{RT} = (T(\theta_H), \phi_H + \triangle\phi)$. Here $T(\theta_H)$ is called a tilted reflection map, which is indeed a 1D mapping representing a tilting to the half vector. $T(\theta_H) = \theta_H$ if no tilting is applied. The same tilt reflection map can also be applied to the Fresnel term through $H_T = Vec(T(\theta_H), \phi_H)$, where $Vec(\theta, \phi)$ is a unit-length vector with direction (θ, ϕ)

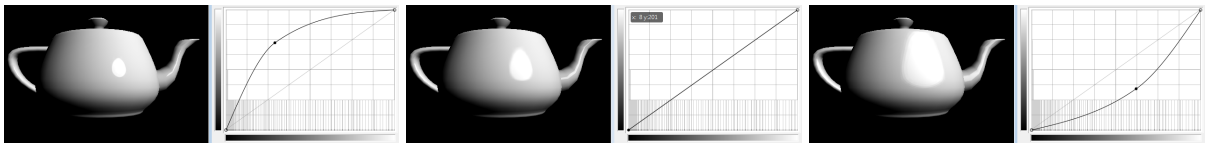
The rotation map is encoded as hue component in HSV color space, i.e., the hue component is used to define the rotation angle in our material editing system. Our system provides a function allowing the user to specify arbitrary rotation at any region by drawing, so a rotation map can be generated easily. The purpose of the tilted reflection map is to allow artists to control the visual roughness of surface by adjusting the size of hightlight area. To achieve this, the tilted reflection map remaps the angle between half vector and surface normal, i.e., θ_H, to a new value. By remapping θ_H, an artist can visually tune the highlight area as shown in Figure 5. The effect controlled by tilted reflection map is similar to the shininess of Phong lighting model. The default tilted reflection map is no tilting, i.e., a linear mapping (a straight line of slope one).

4 Experimental Results

We implement the proposed design system using OpenGL with GLSL and integrate it with the GIMP as a plugin. Considering the real-time feedback, we carefully manage the resources by sharing the same texture objects and shader context for the object preview window and specular design window. Also, to reduce texture accesses time, all the specular color maps are combined into a 2D texture array. When changes are applied to a specific specular map, we only update the corresponding layer of the 2D texture array rather than regenerate the whole texture array. For real-time applications, the normal map is precomputed from the height map and stored with the height map as an RGBA texture. Also, the specular mask map, Fresnel map, and the rotation map are stored as an RGB texture. All the specular color maps are merged into a 3D texture to reduce the storage. In our experiments, three specular color maps along with the index map, specular mask map and rotation map are sufficient to represent most materials. If a material is very complex, an additional index map can be used to indicate more specular color maps. In that case, all the index maps are combined into a 3D texture. Besides, parallax occlusion mapping [14] is applied to simulate the self-shadowing effect of the material. On an Intel Core 2 Dual E8400 3.0GHz with Nvidia GTX 260 desktop, our system achieves 1450 fps when rendering 10 viewports (a 512×512 object preview viewport and nine 170×170 specular design viewports) at the same time while a 256×256 texture is continuously updated. All the materials in our results are created from a single photo or edit from an image. Unless otherwise specified, the input image of each material is the diffuse color map in the experiment. Please see also the accompanying videos for demonstration of our system and results.

4.1 Comparison with BTF and texture mapping

To verify the effectiveness of our material design system, we used a single image chosen from a BTF of a material as an input of our system. Figure 6 shows the rendering results using the designed material, texture mapping of the input image, and the original BTF data (rendered with Local PCA [10]). Note that, as the BTF data did not record the appearance of the material when the θ angle is higher than 80 degrees, the lit regions that have no corresponding BTF data are marked with white dotted lines. In these regions, the BTF data is approximated by those with $\theta = 80$ degrees. In Figure 6, the material has a large variation in height. Without the meso-geometry information, the texture mapping result looks flat and quite different from the BTF result. On the other hand, although the shadow in our result is not reproduced very well by the parallax occlusion mapping, our result is much similar to the BTF result. Given that only one image is used, our material design system demonstrates a great strength in material creation and design.



(a) Small focused highlight area (b) Default setting of linear mapping (c) Enlarged smoothed highlight area

Fig. 5 The highlight area is controlled by tilted map. The plots at right are the the mapping curve.

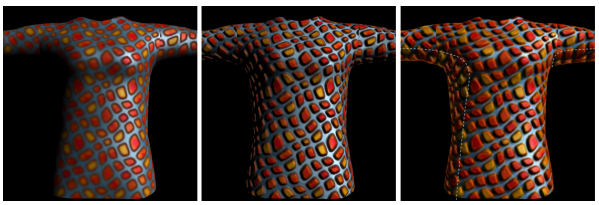
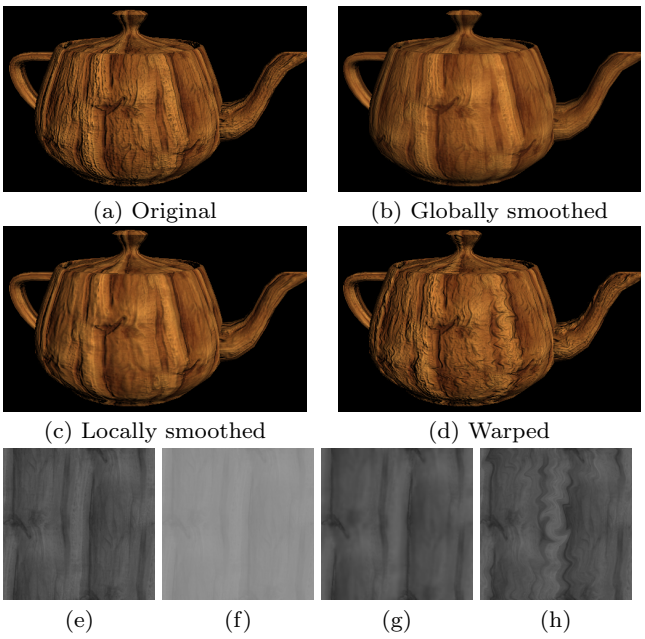


Fig. 6 The comparison of different rendering methods for WL_cool. Left: texture mapping. Middle: BTF rendered using local PCA. Right: our result.



(a) Original (b) Globally smoothed (c) Locally smoothed (d) Warped

(e) (f) (g) (h)

Fig. 7 Material editing results by changing the height map of wood. (e)-(f) are the height map used to generate (a)-(d), respectively. (e) Original height map. (f) Changing the brightness and contrast to smooth the height map globally. (g) Using 'Blur/Sharpen' brush to smooth the bright color part. (h) Using 'Interactive Warp' to warp the height map counterclockwise.

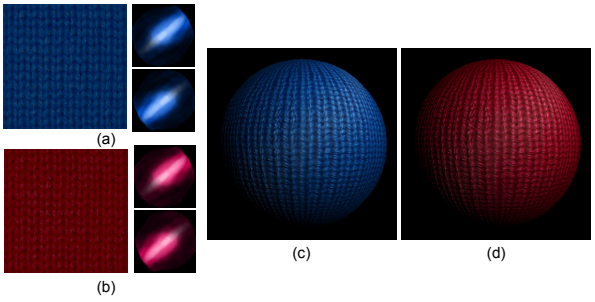


Fig. 8 Changing the color of the material by editing diffuse color map and specular color maps. (a) Original; (b) Edited; (c) and (d) are the rendering results of (a) and (b), respectively.

4.2 Material Design and Editing

As our system represents a material using texture maps, material editing can be achieved easily by manipulating these texture maps as images. Moreover, since our system is built upon GIMP that provides handy image processing and drawing tools, a user can modify these maps and see the edited material immediately. In Figure 7, we change the brightness and contrast of the height map to make the material's surface smoother globally or locally (Please see the supplemental results for the material representation of this Wood sample). We can also use warping tools to create artistic examples. In Figure 8, the color of the material is changed by editing the diffuse color map and specular color maps of the material.

A specular color map stores the specular color under different lighting and viewing directions. It can be designed to hide some marks in the material which can only be observed in some specific viewing or lighting directions (Figure 9(a)). This kind of appearance mostly appears on sports coat or aluminum paper with embossed patterns.

Rotation map can be utilized to produce special effects. In Figure 9(b), we inverse the rotation angle of the 'CGI' text, thus the text would have the specular phenomena that is opposite to the other regions, making the text darker when the other region is brighter and vice versa. Moreover, the specular color maps shown in Figure 9(b) have a brighter intensity near the periphery region. Such distribution creates the glowing effect of the corduroy that can be observed on the chest of the model. In Figure 10, anisotropic reflection is achieved by editing specular color maps with rotation map. In the accompanying video, one can see the highlight rotates when lighting or viewing direction changes.

It is also very easy to create a glossy material using our system. For a glossy material, we can see the highlights when the viewing direction is almost equal

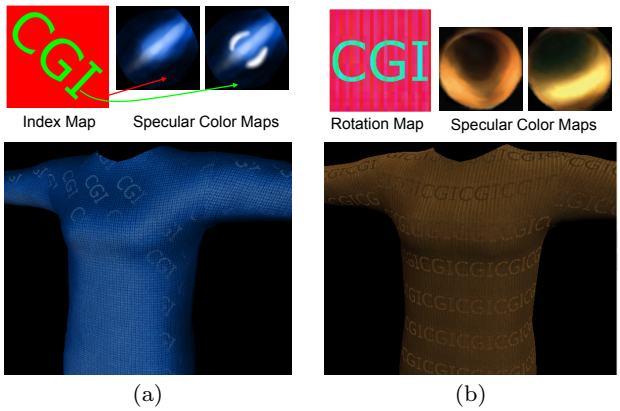


Fig. 9 (a) Specially designed specular color maps and their rendering result. The 'CGI' text can only be viewed under certain viewing directions. (b) Another example of designing rotation map and specular color maps.

to the reflecting direction. The half vector in such cases would be almost equal to $(0, 0, 1)$, which corresponds to the center of the specular color map. Therefore, we can simply draw a bright white dot at the center of the specular color map and apply the 'Blur/Sharpen' painting tool to this dot to soften the highlights. Such effect can also be observed in Figure 11 .

Our system can also be used to create materials that do not exist in reality. In Figure 12, we specially design the specular color map such that the light blue dots of the material have blinking effects. Figure 13 are created from the same input image of the Wood sample in Figure 7(a). In Figure 13, we add some lather on the wood. To avoid drawing the lather pattern on different maps, we add a transparent layer on the diffuse color map and draw the lather pattern on this layer. Then we copy the lather pattern and paste to all the other maps with some modifications. For the specular mask map and the Fresnel map, we adjust their intensity to make them darker since the lather does not have strong specular effects; For the height map, the lather region has a higher intensity since it is higher than the wood region; For the index map, we assign blue color to the lather region, which corresponds to the specular color map 3. Since the specularity of the lather may vary spatially, we add HSV noise to the specular color map to produce such an effect.

Figures 14 and 15 show two examples designed by an artist. In both examples, imaginary materials with complex (or not physically correct) anisotropic reflection are generated. Please see the accompanying video for better demonstration. Note that the light source in Figure 15 is single directional light, but multiple specular highlights can be seen in the result. These two examples show the promising of our material design system.

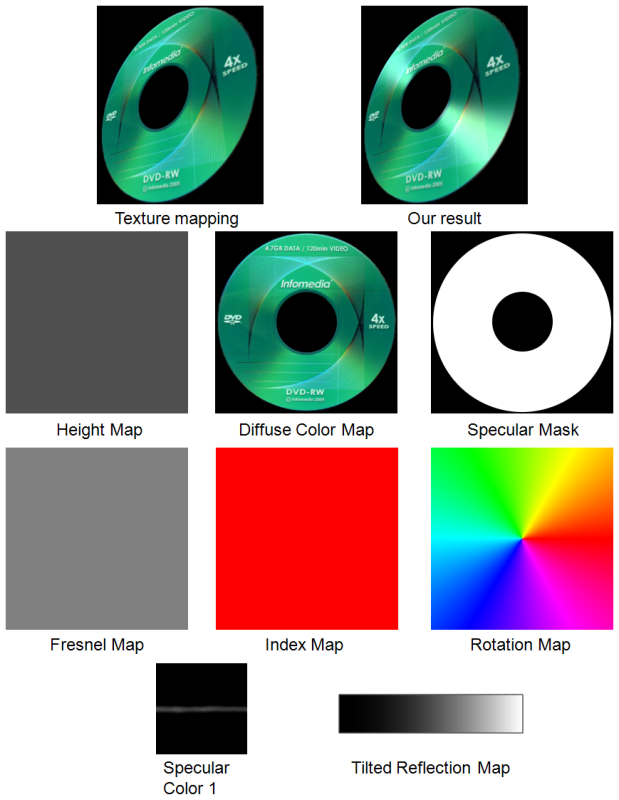


Fig. 10 Disk.

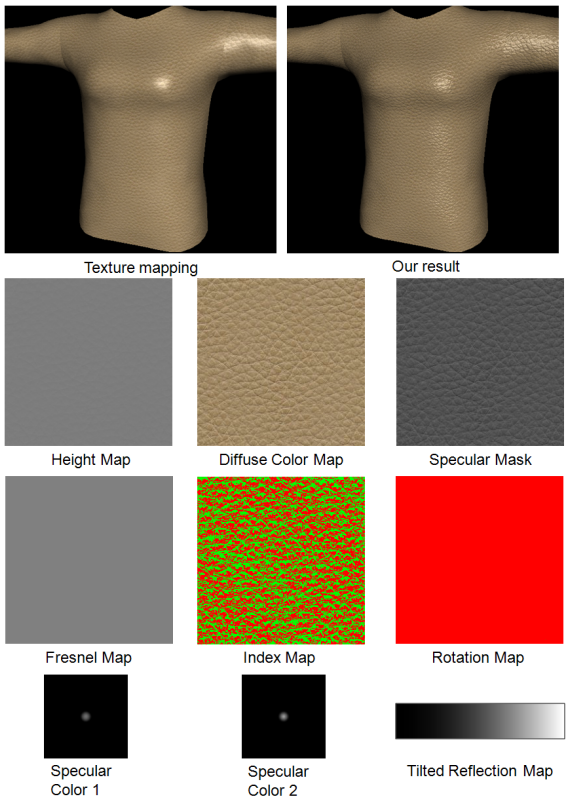


Fig. 11 Beige glossy leather.

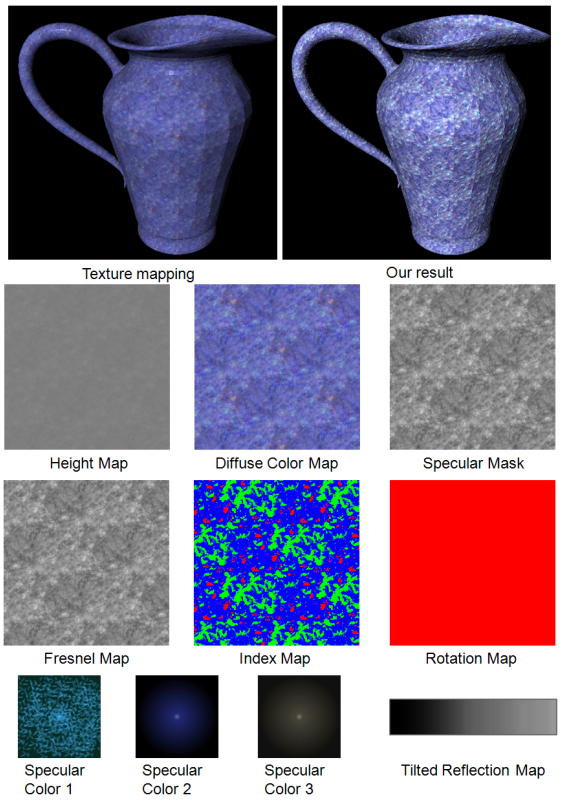


Fig. 12 Artificial porcelain vase.

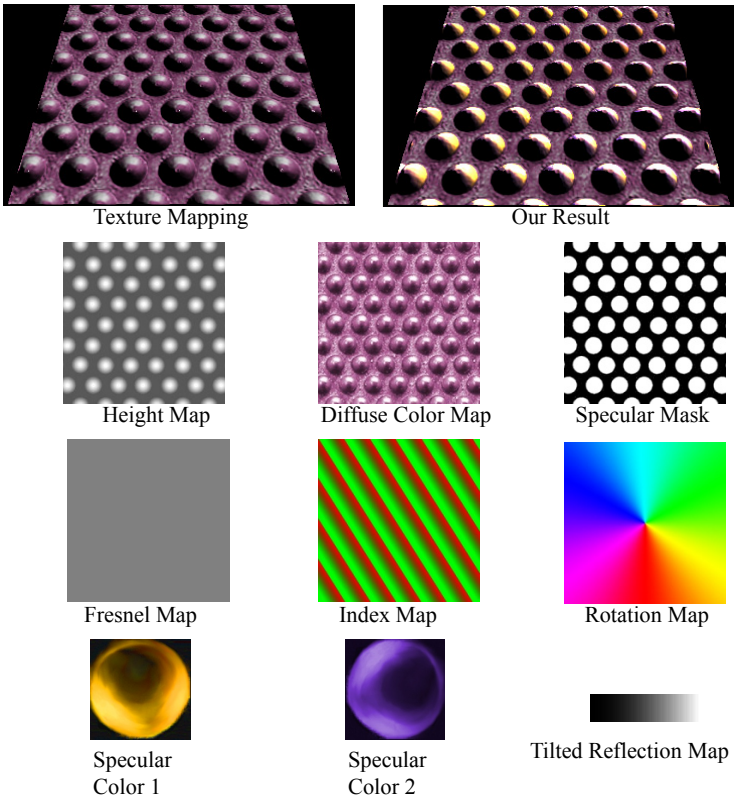


Fig. 14 An imaginary material made by an artist.

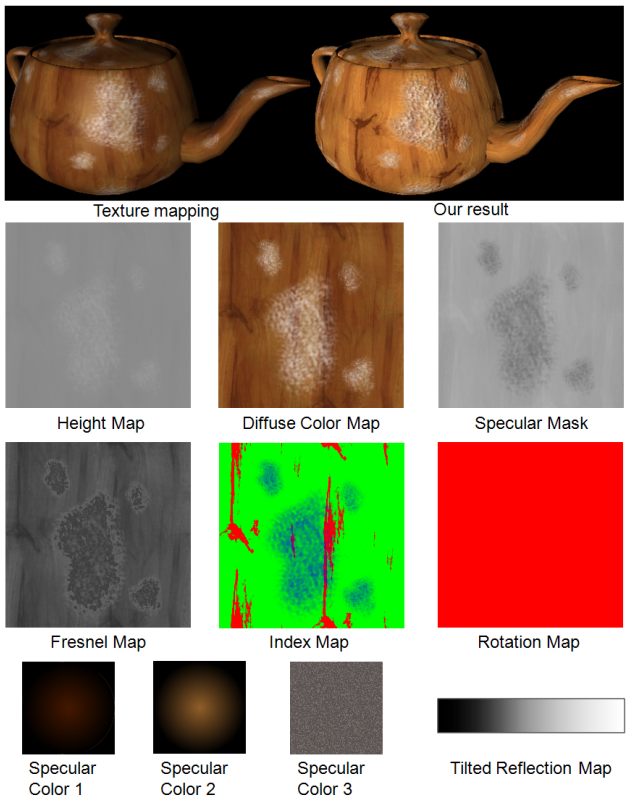


Fig. 13 Wood with lather.

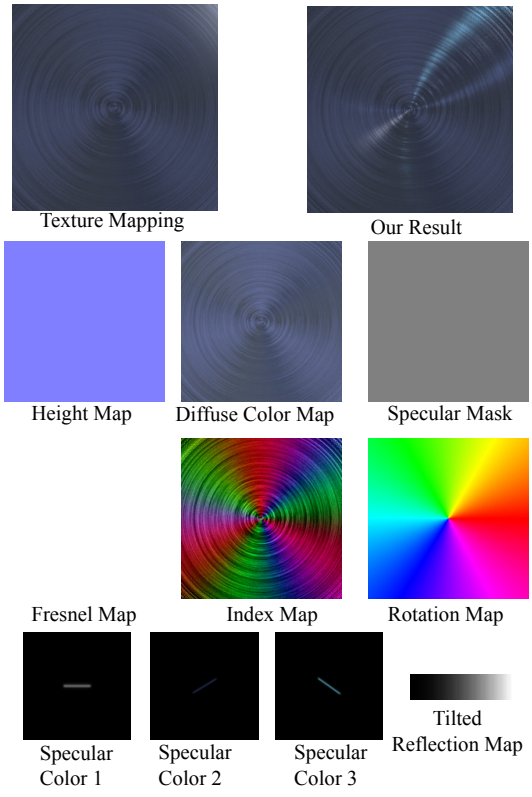


Fig. 15 Another imaginary material made by an artist. Note that the light source in this example is just a directional light, but multiple highlights appear on the material.

4.3 Limitations

Although our system can create many general materials using the proposed material representation, it is hard to design translucent materials because the BRDF model we used cannot handle the subsurface-scattering and inter-reflection phenomena. To extend our material design system for more versatile materials, a node based shader synthesis tool can be incorporated. Another limitation about our material creation flow would be the geometry part. Although the meso-geometry we derived using photometric stereo or gray-level intensity of the image can produce realistic visual effects, it is not accurate compared to the meso-geometry of materials in reality. For accurate meso-geometry information, a reconstruction approach in [3] or other depth acquisition hardware can be applied.

5 Conclusion and Future Work

We propose a simple material creation flow and an artist friendly material design system for artists and novice users in this paper. Artists can create appealing results from a single or a few images intuitively. Our system generates the texture maps to representing the material's meso-geometry, diffuse part, and specular part and user can design and edit a material by manipulating these texture maps.

The material editing capability of our system is not restricted to the image editing tools provided in our system. Artists can combine our system with any other useful plugins or image editing tools to help them design and edit materials. As a result, our system has a great strength in material editing. It can well produce the appearance of general isotropic and anisotropic phenomena for real or imaginary materials. Complex specular phenomena, such as anisotropic reflection, can also be produced easily.

The material representation of our system stores all the lighting phenomena of a material in texture maps. Thus given two to three materials, maybe we can find a reasonable way to create a new in-between material using the texture transition idea as in [8].

6 Acknowledgments

We thank the anonymous reviewers for their valuable comments and the artists in International Games System CO., LTD for their thoughtful feedback and creative examples. This work was supported in part by International Games System CO., LTD and the UST-UCSD International Center of Excellence in Advanced Bioengineering sponsored by the Taiwan National Science Council I-RiCE Program under Grant Number: NSC-100-2911-I-009-101 and NSC-100-2628-E-009-002-.

References

1. Ashikhmin, M.: Distribution-based BRDFs. (2006). URL http://jesper.kalliope.org/blog/library/dbrdfs.pdf
2. Dana, K.J., Van-Ginneken, B., Nayar, S., Koenderink, J.: Reflectance and texture of real-world surfaces. ACM Transactions on Graphics **18**(1), 1–34 (1999)
3. Dong, Y., Tong, X., Pellacini, F., Guo, B.: Appgen: Interactive material modeling from a single image. SIGGRAPH Asia (2011)
4. Filip, J., Haindl, M.: Bidirectional texture function modeling: A state of the art survey. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1921–1940 (2009)
5. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **10**, 439–451 (1988)
6. Heidrich, W., Seidel, H.P.: View-independent environment maps. In: ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware, pp. 39–45 (1998)
7. Kautz, J., Boulos, S., Durand, F.: Interactive editing and modeling of bidirectional texture functions. ACM Trans. Graph. **26**(3), 53 (2007)
8. Matusik, W., Zwicker, M., Durand, F.: Texture design using a simplicial complex of morphable textures. In: ACM SIGGRAPH, pp. 787–794 (2005)
9. Menzel, N., Guthe, M.: g-BRDFs: An intuitive and editable BTF representation. Computer Graphics Forum **28**(8), 2189–2200 (2009)
10. Meseth, J., Müller, G., Klein, R.: Preserving realism in real-time rendering of bidirectional texture functions. In: OpenSG Symposium 2003, pp. 89–96 (2003)
11. Müller, G., Meseth, J., Sattler, M., Sarlette, R., Klein, R.: Acquisition, synthesis and rendering of bidirectional texture functions. Computer Graphics Forum **24**(1), 83–109 (2005)
12. Roy, S., Cox, I.J.: A maximum-flow formulation of the n-camera stereo correspondence problem. In: ICCV, pp. 492– (1998)
13. Rushmeier, H., Taubin, G., Guéziec, A.: Applying shape from lighting variation to bump map capture. In: Eurographics Rendering Techniques, pp. 35–44 (1997)
14. Tatarchuk, N.: Dynamic parallax occlusion mapping with approximate soft shadows. In: I3D, pp. 63–69 (2006)
15. Tsai, Y.T., Fang, K.L., Lin, W.C., Shih, Z.C.: Modeling bidirectional texture functions with multivariate spherical radial basis functions. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 1356–1369 (2011)
16. Wu, H., Dorsey, J., Rushmeier, H.: A sparse parametric mixture model for BTF compression, editing and rendering. Computer Graphics Forum **30**(2), 465–473 (2011)
17. Xu, K., Wang, J., Tong, X., Hu, S.M., Guo, B.: Edit propagation on bidirectional texture functions. Computer Graphics Forum **28**(7), 1871–1877 (2009)