Techniques for an artistic manipulation of light, signal and material

Sarah El-Sherbiny*
Vienna University of Technology

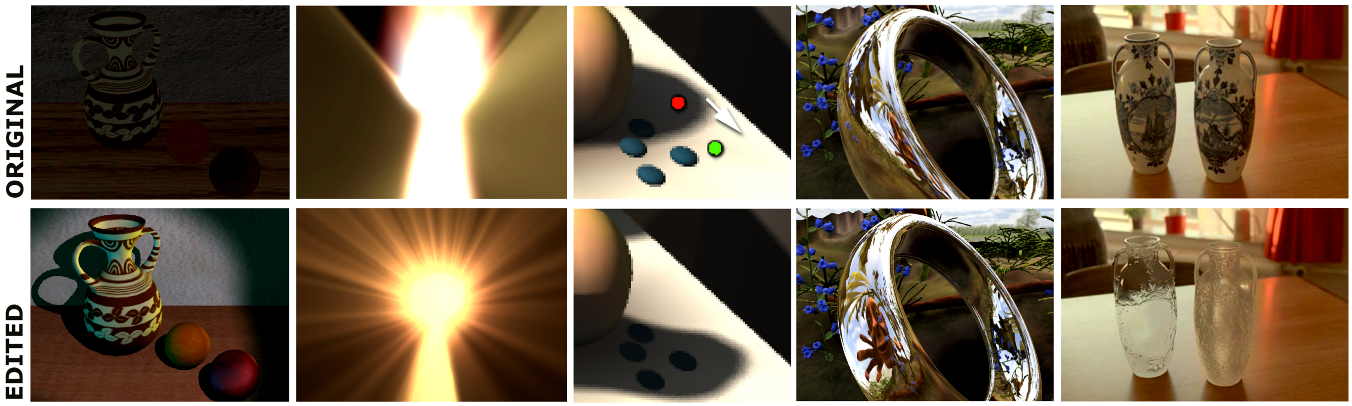

Figure 1: Beginning from the left side the images in the first column show an editing of the lighting. The scene was manipulated by painting with light and automatic refinements and optimizations made by the system [Pellacini et al. 2007]. A result of volumetric lighting can be seen in the second column. The top image was generated by static emissive curve geometry, while the image in the bottom was created by animated beams that get shaded with volumetric effects [Nowrouzezahrai et al. 2011]. In the third column the shadows were modified. The user can set constraints and manipulate surface effects via drag and drop [Ritschel et al. 2010]. The images in the fourth column illustrate the editing of reflections. The original image was manipulated so, that the reflected face gets better visible on the ring [Ritschel et al. 2009]. The last images on the right show the editing of materials by making objects transparent and translucent [Khan et al. 2006].

Abstract

This report gives an outline of some methods that were used to enable an artistic editing. It describes the manipulation of indirect lighting, surface signals and materials of a scene. Surface signals include effects such as shadows, caustics, textures, reflections or refractions. Some of the methods are physically based while others just emulate a realistic view. These methods are aim to mimic a photo-realistic scene or create scenes that fit better depending on the context. Artistic manipulation is also relevant for multimedia production.

Most of the used techniques are based on popular algorithms like photon mapping or path tracing. Researchers extend or alter the algorithms or decouple specific stages to bring the possibility of editing into it. These methods include modifications by defining distance fields, setting constraints or transforming objects to different spaces. Some approaches start with physically correct systems and allow the manipulation of parameters in a non-physical way to be able to freely manipulate the scene. Other methods calculate physically correct results.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and realism—Color, shading, shadowing, and texture

Keywords: global illumination, light transport, volumetric lighting, curving light rays, cinematic relighting, signal deformation, painting, appearance, shadows, reflections, material editing, post-processing, artistic control

*e-mail: sarah.el-sherbiny@student.tuwien.ac.at

1 Introduction

Global illumination is important for getting more realistic images. It considers direct illumination that occurs when light falls directly from a light source on a surface. Also indirect illumination is taken into account, when rays get reflected on a surface. In diffuse reflections the incoming ray gets reflected in many angles while in perfect specular reflections the incoming ray only gets reflected in one direction. Most of the surfaces are a combination of diffuse and specular.

Caustics is an effect that can appear on surfaces, when the reflected light gets bundled in one point. Mostly this is the case on curved surfaces. The light rays act as a tangent on the accrued caustic.

When there is an object on which a light ray falls before it gets on a surface, then shadows appear on that surface. Shadows appear on surfaces that are not lightened when a lightened object gets placed in front of that surface and casts shadows on it. In global illumination shadows are also defined as the absence of light.

Materials of objects can be influenced by properties like the color or texture and the lighting of a scene. With the *bidirectional reflectance distribution function (BRDF)* reflections of light rays can be defined. It describes the incoming and outgoing light rays and their directions.

The goal is to edit all these effects that occur on surfaces like light, shadows, caustics or materials in an artistic way to get better looking results. It should for example be possible to move or replace these effect, enlarge them or remove the unwanted ones of them from a scene.

Artistic editing is a powerful method that can be applied as a post-processing effect in film production, for a better visualization of more important parts of images or the creation of more realistic scenes. Scenes must not be physically correct to look realistic. Also physically incorrect methods are used to achieve a better view. Some approaches start from a physically based method and allow editing it in a non-physical way.

In most of the cases the human eye cannot distinguish whether a scene is edited or not. Some researchers like Ritschel [2009; 2010] also made user studies about that.

This report gives an outline of techniques that enable the editing of scenes in an artistic way. This gets realized by manipulating effects like the lighting, shadows, reflections or materials. It will have its focus on the used methods and not on the interfaces of the presented systems.

Overview of some work in artistic editing

Schmidt [2013] presented a physically based rendering technique that enables it to manipulate multiply-refracted caustics, indirect lighting, reflections, and shadows. The operation is based on path space, not on a specific shading phenomena. Depending on the user input it starts by selecting and filtering a transport effect. The two methods path retargeting and path proxy linking were introduced to enable editing the light transport.

Path retargeting allows a direct manipulation on paths, while path proxy linking indirectly modifies the path space when the scene gets edited by the user. A list of transport paths is generated and denoted in parametrized regular expressions, so that the user can select from it. The primary goal was to create a physical based system without limiting the artistic latitude. It should also enable a consistent rendering of the manipulated light transport and apply an automated clustering based on light paths and their surroundings.

Nowrouzezahrai [2011] introduced a method that starts from a physical based algorithm and allows the user to manipulate it by reducing the physical parameters. It also supports the editing of non-physical parameters like the color. The system separates the modeling stage from the shading stage by using physically-based photon beams in a modified way. The goal was to enable a volumetric lighting in an intuitive and natural manner.

A non-physically based method was created by Ritschel [2010]. It works with constraints and is based on a previous system [Ritschel et al. 2009] that allows editing reflections. It allows the modification of more effects, which show up on the surface. Shadows, caustics, reflections, or 3D-textures can be manipulated via drag-and-drop by creating a mapping from the surface location of the effect to another location. Another non-physically approach was introduced with a lighting model by Kerr [2010]. It allows the control over light rays in a non-linear way by reshaping splines.

For static volumes Klehm [2014] presented a system that allows to change the emission or single-scattering. By drawing the appearance of a scene for some view-points, the physically based properties and the illumination in the environment can be automatically optimized. For example it is possible to add colors or single scattering to static volumes.

In the work of An [2008] the user defines the rough edits by painting to manipulate the appearance of a scene. These edits will then be automatically propagated over the whole image and refined by the system. It allows the modification of low- or high-dynamic range images, and materials.

Pellacini [2007] introduced a method where the user can directly paint lighting effects into a scene. The system uses a non-linear optimizer to find the best setting for each parameter that matches the painting of the user. It is possible to modify the colors, light shapes, shadows, highlights, and reflections of a scene by painting.

For cinematic relighting Hašan [2006] created a technique with multi-bounce indirect illumination based on a hierarchical algorithm and on photon mapping. Pellacini [2005] presents a cinematic lighting system using *RenderMan* shaders [PIXAR 2005]. It supports a numerical estimation of surface response, image-space caching and deferred shading.

Obert [2010] allows the editing of shadows in lighting environments by storing information of the lighting and the material of a scene. It is for example possible to remove or curve shadows of scene objects. Khan [2006] presents an approach that allows the editing of materials. Starting from a high dynamic range image, the system can make objects transparent or translucent or apply any other kind of material to scene objects.

Colbert [2006] made it possible to create bidirectional reflectance distribution functions (BRDF). This is realized through positioning and editing highlights on a spherical canvas by painting. Ben-Artzi [2008] introduced a technique for manipulating BRDF with global illumination.

2 Surface light and signal deformation

2.1 Light transport

When a light ray gets emitted from a light source, it typically falls on many surfaces. Depending on the material of the surface a certain percentage of the light gets reflected or absorbed before it finally reaches the eye. Specular surfaces reflect a light ray only in one direction while diffuse surfaces spread the ray in many directions. Light transport describes the amount of light that receives an object and the way that it gets reflected from one surface to another depending on the material of the surface.

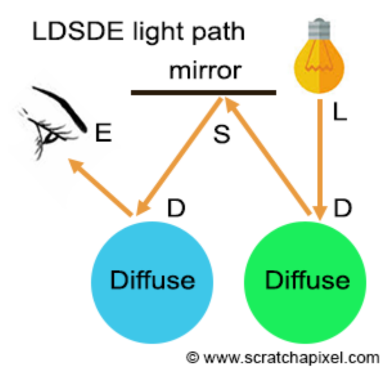

Figure 2: An example light path with Heckbert's notation [1990] can be denoted as *LDSDE*. First a ray gets out of a light source *L*, then it gets reflected by a diffuse surface *D*, mirrored by a specular surface *S* and again reflected by a diffuse surface *D* until it finally falls into the eye *E* [Scratchapixel].

Heckbert [1990] presented an intuitive notation which shows a possible way to describe the path of a light ray from the light source to an eye (Figure 2). The regular expressions $L[D|S]^*E$ can be used to describe all possible light paths. Every light ray that gets emitted from the light source L, can be reflected by diffuse surfaces D or get mirrored by specular surfaces S before it reaches the eye E. Normally most of the surfaces will be a combination of diffuse and specular. The shortest light path can be denoted by LE. In this case the light do not fall on any surfaces. It shines from the light source L and directly hits the eye E.

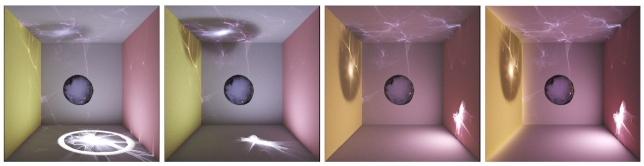

Figure 3: Starting from the left side the first image represents the original state. In the second image path proxy linking was used to change the size of the shadow on the wall. As the system is physically based this also influences the caustics on the floor. The third image moves the caustics from the floor to the right side of the wall using path retargeting. In the last image the indirect lighting from the left wall gets stretched [Schmidt et al. 2013].

Using Heckbert's notation Schmidt [2013] demonstrated two methods for manipulating light transport based on global illumination (Figure 3). While path retargeting can enable an editing in a directly way, path-proxy linking does it indirectly by using additional objects as a proxy. These objects are not visible, but affect the lighting of the scene. Brightness manipulation and hue editing is also supported for both methods as described by Obert [2008]. The aim was to create a manipulation, which is physically based without narrowing the possibility of editing the scene in an artistic way.

Path retargeting allows the user to directly manipulate the lighting by moving caustics or reflections of a scene. This happens by choosing a light path or an eye path and then moving on its endpoints. To modify the scene indirectly by path proxy linking, objects can be linked or connected to it. For example it is possible to connect a big sphere that is not visible, to a smaller one in the scene to get a bigger shadow of the small sphere. Therefore it can be made more appropriate depending on the context.

2.2 Volumetric lighting

Kajiya [1986] and Immel [1986] represented the rendering equation

$$L_o(x,\omega) = L_e(x,\omega) + \int_\Omega L_i(x,\omega')f_r(\omega,x,\omega')(\omega'\cdot n)\,d\omega'$$

which illustrates the amount of light that gets emitted from a surface point x in the direction of the vector ω. L_o describes the outgoing light of a point x in direction ω, L_e the emitted light of the point x in direction ω and L_i the incoming light of the point x in direction ω'.

The integral stands for the sum of all the incoming light rays in every direction ω' over all surfaces in the scene. The term $f_r(\omega,x,\omega')$ represents the BRDF that shows the fraction of the incoming light that gets reflected in the point x with the incoming direction ω' and the outgoing direction ω. The cosine of the angel between ω' and the normal vector n is shown by the dot product $(\omega'\cdot n)$.

This integral cannot be solved in an algebraic way, so it must be approximated. A possible way of doing this is the Monte Carlo method. It is based on random sampling of numerical results. First a domain of possible inputs must be defined, then random inputs should be generated and computed in a deterministic way. Finally the results found should be combined.

Volumetric photon mapping is a two pass algorithm that was defined by Jensen [1996] and is based on the Monte Carlo method. In the first pass information about the light is needed. So many photons should be brought into the scene by using light sources. When they get destroyed, their position and direction can be saved using photon maps. The second pass is for rendering and stores the intensity of pixels using their locations from the photon map.

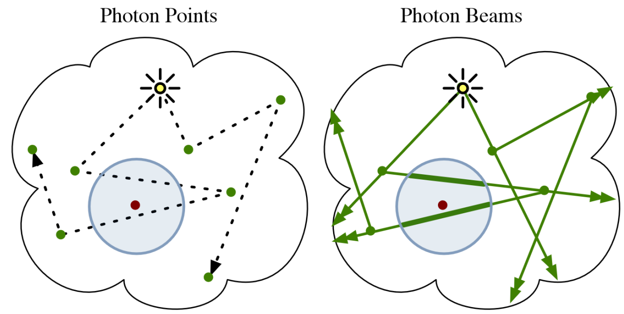

Figure 4: While photon points store the locations of photons in the photon map, photon beams store their paths. Photon beams can lead to a better quality because it is more probable that a path lies in a specific location than a point [Jarosz et al. 2011].

Nowrouzezahrai [2011] extended this technique to photon beams instead of photon points (Figure 4). In contrary to photon points the paths of the photons get stored not their locations [Jarosz et al. 2011]. Therefore the estimation of the radiance gets more exact because it is more probable that a path is inside a specific location than a point. To enable an artistic manipulation of scenes the modeling and the shading part get separated. It is possible to modify the behavior of beams by editing the first pass of shooting. Also the geometry of beams can be influenced so, that they are not restricted to be physically based anymore (Figure 5). For the shading stage the sum of all beams in the scene gets calculated.

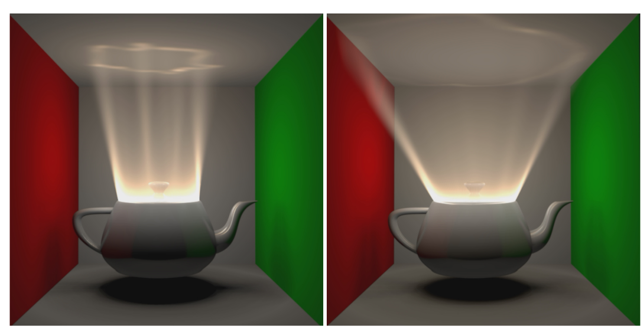

Figure 5: Generated photon beams can produce reflections and indirect light on surfaces [Nowrouzezahrai et al. 2011].

2.3 Signal deformation

Signals that occur on a surface like caustics or indirect light can be edited on a 2D surface although they are in a 3D space. Shadow signals and global illumination are visualized by 1D lines that can also be edited with the approach of Ritschel [2010].

When the user defines a new position for surface signals via drag and drop the signals get transformed from the initial location to another one. The system is based on mass spring to allow the setting of point constraints (Figure 6). User can define a region to limit the influence of the scene which can be realized by the Nearest Neighbor algorithm. The deformation of the points happens by a mapping function that considers the constraints that are set by the user. These function should also be able to deform points that lie in the neighborhood and locate these points on the surface of a scene object.

Then a mapping from the initial position of the points to the deformed position is needed which can be realized by the inverse of the mapping function from before. Finally a distance field gets constructed and stored in a hierarchical data structure to map every point to the closest point on the surface of an object.

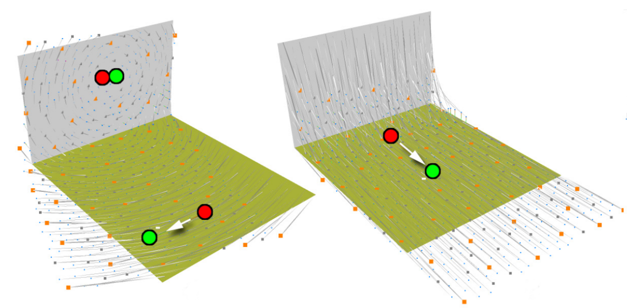

Figure 6: This example shows two possible ways for defining constraints and visualizes the surface signals due to the deformation. In the left image two constraints were set. One for the origin of the rotational field and the other one determines the exact rotation that happens about this point. The right image only includes one constraint that defines a translation [Ritschel et al. 2010].

2.4 Curving light rays

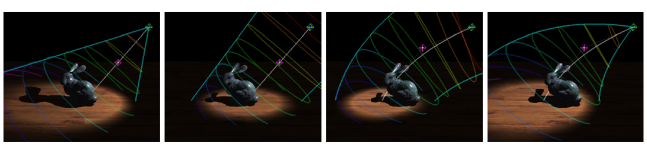

Figure 7: Direct illumination can be manipulated with bendy lights. The first image on the left side shows the original state using a spotlight. In the second image a constant radius is set around the scene object. The light rays of these object get bend using the defined radius in the third image. Again the light rays get bend in the last image but this time it uses the radius from the first image [Kerr et al. 2010].

For artistically reasons it may be necessary to curve rays. This can be done by reshaping spline paths in a non-linear way. Kerr [2010] presented a light model that can bend shadows and directions of the incoming light (Figure 7). The model is based on spot-lights and uses quadratic Bezier-splines. The manipulation happens by moving on the control points of the Bezier-curve.

In this approach the direction of the incident light and an appropriate function for the visibility must be defined for every point in the scene. The direction of the light is defined as the tangent of the light tube at every point location. The visibility function is calculated by the inverse of a non-linear light path. The geometry in the scene

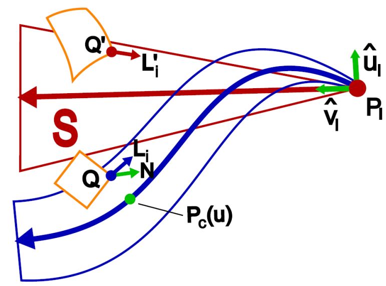

Figure 8: The red spotlight S is defined by a green coordinate system. It determines the first control point of the blue curved light which is represented by a quadratic spline with three segments. Instead of calculating the scene geometry Q' in an non-linear way on the original geometry, Q can be determined in a linear way on the deformed geometry [Kerr et al. 2010].

needs to be transformed in a non-linear way. A non-linear transformation of shadows on the original geometry can also be achieved by calculating a linear computation of shadows on the deformed geometry (Figure 8).

3 Painting and lighting with paint

3.1 Volume stylization

Klehm [2014] presented a system for editing properties of static volumes. It allows for example the modification of the emission or single-scattering of volumes in an intuitive way. After changing the view-point, the system still remains smooth and coherent. The approach uses *tomographic reconstruction*. This method works with some view-points that get defined by the user. These view-points get evaluated by the system so that the scene can be adapted due to them (Figure 9).

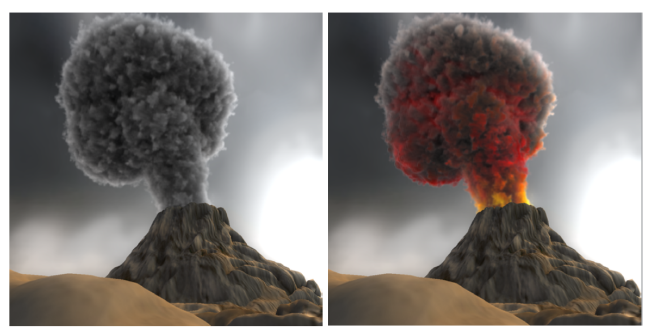

Figure 9: A static volume was stylized by using a color gradient [Klehm et al. 2014].

The system can also optimize the environmental lighting appropriate to the appearance of the scene without modifying the volume properties. It uses the volumetric rendering equation in an inverted way. This describes the interaction between the light and the volume. It forms a problem that can be solved by the inversion of a large scale linear system. The user only needs to define the appearance on some of the view-points and the system automatically optimizes the appearance of the volume.

3.2 Manipulation by painting

In the work of An [2008] an approach was introduced that manipulates the scene by painting. To edit the scene the user can paint rough changes directly into it. Then these edits get automatically propagated over the whole image and refined by the system. User edits that are similar are applied to spatially-close regions which look similar. The approach is based on a non-parametric representation and an approximation algorithm that spreads the defined edits over the whole image. This method supports high- or low-dynamic range images, measured materials and many appearance samples.

Pellacini [2007] created a method that allows the user to directly paint with light into a scene (Figure 10). It is based on a non-linear optimizer to find the best setting for each parameter that matches the painting of the user and allows modifying the color, light shape, shadows, highlights, and reflections of a scene by painting. Therefore effects can be achieved that could not be created with a physically based system.

Figure 10: The light in the scene was directly painted by the user in an artistic way [Pellacini et al. 2007].

4 Cinematic relighting, shadow and reflection manipulation

4.1 Cinematic relighting

Hašan [2006] introduced an approach with multi-bounce indirect illumination. It works from a fixed view point and uses a frame buffer. This buffer contains a set of view samples. View samples represent the indirect illumination of points. The indirect illumination of the samples gets calculated by the direct illumination of many gathered samples from the scene. This phenomena was named *direct-of-indirect-transfer*. It forms a large, linear transformation matrix because it must fit all view samples and all the gathered samples. So calculations with it will be much expensive. Therefore the authors simulate these transformation with a set of some few matrices. They realized this by using the wavelet space in a hierarchical way for encoding the matrices. Then these matrices get mapped to the GPU by an image-based approach.

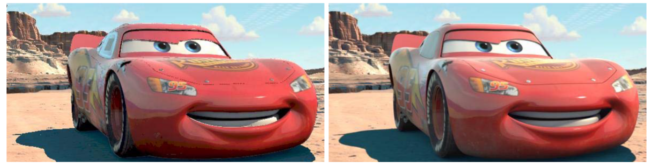

Figure 11: The left image was rendered by *Lpics* and the right one by a software renderer. Users need to wait for about 0.1s after moving one light to get a respond from *Lpics* while it takes 2000s with the software renderer [Pellacini et al. 2005].

Another system for cinematic lighting was presented by Pellacini [2005]. It is named *Lpics* (Figure 11) and uses *RenderMan* shaders [PIXAR 2005]. In comparison with final renders, it has only a small approximation which was achieved by numerical estimation of the surface response. For complex geometry it is more effective when the duration of an algorithm is independent from the size of the scene. When it depends on the size of the image it terminates faster.

The execution of surface shaders has also a long render time when not using ray-tracing. This is because of the part of generating patterns. Pellacini [2005] encoded the response of surface to the lighting in a function that gets used by the shader based on some parameters that get computed while pattern generation. This function is similar to the BRDF but it also supports higher dimensions.

4.2 Shadow manipulation

Shadows help to get more realistic images. To be able to modify them Obert [2010] presented a system that separates the shading stage from the lighting stage. So the shadows can be modified without changing the lighting. The changes of the shadows by the user get localized by a shadow selection algorithm in the domain of the light.

The approach first calculates and stores information about the visibility in the scene. This happens separately from the lighting and BRDF. Beginning from the rendering equation on, all the terms that do not affect the visibility were grouped together to get separated from the visibility terms. The non-visibility terms form the transport coefficients and include the lighting and BRDF. They can be calculated beforehand in two different ways depending on the working setting. It is possible to work either with a fixed viewpoint with any possible BRDF or with a freely viewpoint and a BRDF that is limited to diffuse surfaces.

The authors first take the two methods per-vertex sampling and per-texel sampling into account. Vertex sampling would be more intuitive but scene objects and shadows with a high frequency would become undersampled. So they realized it through per-texel sampling. The objects get sampled corresponding to their UV-coordinates, which allows a finer control than per-vertex sampling.

4.3 Reflection manipulation

Ritschel [2009] allows the editing of reflections with a non-physically technique. Reflections can be manipulated in a view independent way by directly editing the reflection ray. This happens by setting point constraints as described before (Section 2.3).

The constraints can be defined by the point from which light gets reflected, and the point on which the reflected light ray will fall (Figure 12). Both of these points can be modified via drag and drop.

All the objects in the scene should be influenced depending on the user edits. So the system spreads the edits as rotations of reflection directions over a 3D surface. This was realized by the moving least squares deformation that was defined by Schaefer [2006]. In the work of Schaefer [2006] translations and rotations were interpolated for 2D pixels. This was extended by Ritschel [2009] to manipulate rotations of the direction of reflections over a 3D surface. Finally the spreading of a user deformation over a domain leads to a minimization problem. A possible solution was a weighted combination of user edits and reflections of the scene. Then the orthonormal vectors that span the surface should be found.

While this system do not support bendy light, shadows, bounces and caustics the work of Ritschel [2010] based on it and also allows the modification of these effects using surface signals.

Figure 12: The top left image visualizes the constraints and regions defined by the user to manipulate the reflection of the object. After editing, the head of the dragon gets reflected in the mirror instead of its tail [Ritschel et al. 2009].

5 BRDF and material editing

BRDFs are functions that were formalized for the first time by Nicodemus [1965]. They describe the properties of materials and the way that light behaves on them. Surfaces with different materials act in different ways with light rays, even when they are lightened in the same way or with the same light sources. The way a surface interacts with light depends on the attributes of the light source and the material of the surface. When a light ray falls on a surface it can get reflected, absorbed or transmitted. So the amount of the incident light forms the sum of the reflected, absorbed and transmitted amount of light. BRDFs only take the reflected light into account. The transmitted light can be considered by the *bidirectional transmitted distributive function (BTDF)* [Wynn].

Diffuse and specular surfaces reflect light in different ways. While diffuse surfaces spread light consistent in different directions, (ideal) specular materials only reflect light in one direction. This behavior can be described by the BRDF. This function determines the amount of light that gets reflected by a surface with a specific wavelength and can be denoted as $f_r(\omega_i, x, \omega_r)$. A fraction of the incoming light gets reflected in the point x with the incoming direction ω_i and the outgoing direction ω_r.

When a BRDF is physically based, it has more attributes. It must be positive so that $f_r(\omega_i, \omega_r) \geq 0$ is fulfilled. It must accomplish the Helmholtz-reciprocity. Therefore the BRDF should not be affected when the angle of the incoming light direction and that of the outgoing light direction interchange: $f_r(\omega_i, \omega_r) = f_r(\omega_r, \omega_i)$. Finally the BRDF must conserve energy. So the amount of energy leaving a material must be less or equal to the amount of energy reaching it [Colbert et al. 2006].

Normally material parameters are defined by a numerical input. Users usually do not know exactly how the numerical input will look on a surface and how they must modify the parameters to achieve the property of a specific material. Colbert [2006] presented a more intuitive approach for creating physically correct BRDFs. It works with the direct control of materials through painting on a spherical canvas and supports the manipulation of the shape and highlights of materials (Figure 13).

It is also possible to manipulate the way incoming light gets reflected. To simulate the reflections, a lighting model was needed. So the system is based on the Ward illumination model [Ward 1992]. The model gets extended by a multiplication of the outgoing vector with a transformation matrix to be able to create highlights that are not close to the point of an ideal specular reflection.

Highlights that appear in a circular shape on the spherical canvas while illumination with a point-light, were named *circular highlights*. A rotation of the surface do not influence the BRDF, so these circular highlights are independent from the direction. They can be placed everywhere on the spherical canvas. With the transformation matrix included in their lighting model the highlights can be transformed to the desired position. The rendering of the scene happened with a multi-pass approach. The first pass was for the diffuse part of the BRDF using a spherical environment mapping. Afterward the highlights that were created by the users were rendered. For rendering the spherical canvas, they approximated the environment by placing a point light at the brightest location of the environment. The highlights were rendered by an approximation using the Monte Carlo method.

Khan [2006] presented an approach that allows the editing of materials. The system gets a high dynamic range image as an input and is able to automatically change the materials in it. For example it is possible to make objects transparent or translucent or apply any other kind of materials as nickel or aluminium to specific objects.

The method is not interested in physically correct changes, so it concentrates on the visual look. It uses the fact that the human eye reacts tolerant due to several changes and do not even notice them while it is sensitive to others.

Beginning from the rendering equation, the term L_e represents the amount of light that gets emitted from a surface. It can be set to zero when there are not any emitting surfaces in the scene. Light that falls on a pixel can be approximated by the corresponding nearest point on the surface. A hemisphere gets build by the normal vector of these pixel. All the light rays that are inside the hemisphere get then reflected towards these pixel. The amount of the reflected light is defined by the BRDF. To get objects that simulate the appearance of different materials, the pixel values of the outgoing light L_o should be modified. As the rendering equation contains undefined parameters it still has to be simplified. This should happen in a way that the results stay plausible, even if they are not physically based. So the hemisphere and the surface normal should be approximated by following this manner.

The approximation of the incident light can happen in a course way for diffuse surfaces because the human eye do not react to it in an intolerant way or notice inconsistencies with it. This was for example confirmed by Ostrovsky [2005]. There are still unknown values in the term that represents the incident light. These can be classified in values that are outside the image and values that come from behind the object that should be modified. The number of unknown values is not the same for both classes. So different solutions get approximated for each category. Afterward the image can be rendered with any arbitrary BRDF to get a new material on it.

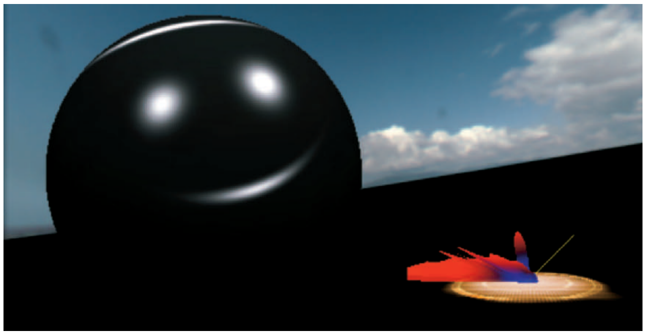

Figure 13: The generated BRDF based on the Ward model. Highlights were defined by a single-point light source [Colbert et al. 2006].

6 Conclusion

The presented methods cover the editing of effects like light, shadows, reflections, refractions, caustics or materials (Table 1). Most of the methods are not physically based but also produce realistic results. The physically based techniques concentrate on the manipulation of the light transport, the editing of static volume properties or the creation and editing of BRDFs.

Comparison of the techniques

An approach for editing volumetric effects is able to change the attenuation or color of light beams. It is based on a physically system but also allows an editing of parameters in a non-physically way. The user has the possibility to choose if he wants to edit a scene in a physically based way or not. A non-physically based editing gives the user more freedom in the manipulation process.

There are two non-physically based methods for cinematic relighting. While the first one works with multi-bounce indirect illumination and some view samples, the other one calculates a numerical estimation of the surface response.

One of the techniques concentrates on editing shadows, while another one focuses on the editing of reflections but also supports the manipulation of highlights or refractions.

Another method focuses on changing the lighting in a scene or editing volume properties such as emission or single scattering in a physically correct way. It can manipulate the environmental lighting independently from the volume properties or combine the editing of both effects.

All presented techniques had the aim to manipulate images in an artistic way, but they differ in the details. Some of them edit many effects with the same system. Other methods only concentrate on special effects. Some are physically based while others just emulate a realistic view. Some were defined for special cases like the cinematic relighting or the material editing approach. Finally it depends on the situation and input whether a method achieves good results or not because every method has its limitations.

Table 1: An overview of the presented systems. *PB* denotes if a method is physically based *(Y)* or not *(N)*, or if it supports both *(B)*. *Editing* lists the main effects that can be modified by the system, and *Year* represents the year on which the associated paper was published.

Category	Method	PB	Editing	Year
Surface light and signal deformation	Light transport	Y	Shadows, reflections, caustics	[2013]
	Volumetric lighting	B	Volumetric effects	[2011]
	Signal deformation	N	Shadows, caustics, reflections, 3D texture	[2010]
	Curving light rays	N	Lighting effects, shadows, highlights	[2010]
Painting and lighting with paint	Volume stylization	Y	Volume properties, lighting	[2014]
	Appearance editing	N	Light, spatially-varying materials	[2008]
	Lighting with paint	N	Lighting effects, shadows	[2007]
Cinematic relighting, shadow and reflection manipulation	Cinematic relighting	N	Lighting	[2006]
	Cinematic relighting	N	Lighting	[2005]
	Shadow manipulation	N	All-frequency shadows	[2010]
	Reflection manipulation	N	Reflections, highlights, refractions	[2009]
BRDF and material editing	BRDF creation	Y	BRDF, highlights	[2006]
	BRDF editing	Y	Material properties	[2008]
	Material editing	N	Material properties	[2006]

References

AN, X., AND PELLACINI, F. 2008. Appprop: All-pairs appearance-space edit propagation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '08) 27*, 3.

BEN-ARTZI, A., EGAN, K., DURAND, F., AND RAMAMOORTHI, R. 2008. A precomputed polynomial representation for interactive brdf editing with global illumination. *ACM Transactions on Graphics 27*, 2.

COLBERT, M., PATTANAIK, S., AND KŘIVÁNEK, J. 2006. Brdf-shop: creating physically correct bidirectional reflectance distribution functions. *IEEE Computer Graphics and Applications 26*, 1, 30–36.

HAŠAN, M., PELLACINI, F., AND BALA, K. 2006. Direct-to-indirect transfer for cinematic relighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '06) 25*, 3, 1089–1097.

HECKBERT, P. S. 1990. Adaptive radiosity textures for bidirectional ray tracing. *ACM SIGGRAPH Computer Graphics (Proceedings of SIGGRAPH '90) 24*, 4, 145–154.

IMMEL, D. S., COHEN, M. F., AND GREENBERG, D. P. 1986. A radiosity method for non-diffuse environments. *ACM SIGGRAPH Computer Graphics (Proceedings of SIGGRAPH '86) 20*, 4, 133–142.

JAROSZ, W., NOWROUZEZAHRAI, D., SADEGHI, I., AND JENSEN, H. W. 2011. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics 30*, 1, 5:1–5:19.

JENSEN, H. W. 1996. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, Springer-Verlag, 21–30.

KAJIYA, J. T. 1986. The rendering equation. *ACM SIGGRAPH Computer Graphics (Proceedings of SIGGRAPH '86) 20*, 4, 143–150.

KERR, W. B., PELLACINI, F., AND DENNING, J. D. 2010. Bendylights: Artistic control of direct illumination by curving light rays. *Computer Graphics Forum (Proceedings of EGSR '10)*, 1451–1459.

KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '06) 25*, 3, 654–663.

KLEHM, O., IHRKE, I., SEIDEL, H.-P., AND EISEMANN, E. 2014. Property and lighting manipulations for static volume stylization using a painting metaphor. *IEEE Transactions on Visualization and Computer Graphics 20*, 7, 983–995.

NICODEMUS, F. E. 1965. Directional reflectance and emissivity of an opaque surface. *Applied Optics 4*, 7, 767–775.

NOWROUZEZAHRAI, D., JOHNSON, J., SELLE, A., LACEWELL, D., KASCHALK, M., AND JAROSZ, W. 2011. A programmable system for artistic volumetric lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '11) 30*, 4, 29:1–29:8.

OBERT, J., KŘIVÁNEK, J., PELLACINI, F., SYKORA, D., AND PATTANAIK, S. 2008. icheat: A representation for artistic control of indirect cinematic lighting. In *Proceedings of the Nineteenth Eurographics Conference on Rendering*, Eurographics Association, EGSR '08, 1217–1223.

OBERT, J., PELLACINI, F., AND PATTANAIK, S. 2010. Visibility editing for all-frequency shadow design. *Computer Graphics Forum (Proceedings of EGSR '10) 29*, 4, 1441–1449.

OSTROVSKY, Y., CAVANAGH, P., AND SINHA, P. 2005. Perceiving illumination inconsistencies in scenes. 1301–1314.

PELLACINI, F., VIDIMČE, K., LEFOHN, A., MOHR, A., LEONE, M., AND WARREN, J. 2005. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '05) 24*, 3, 464–470.

PELLACINI, F., BATTAGLIA, F., MORLEY, K., AND FINKELSTEIN, A. 2007. Lighting with paint. *ACM Transactions on Graphics 26*, 2.

PIXAR, 2005. The renderman interface. `http://renderman.pixar.com/products/rispec/rispec_pdf/RISpec3_2.pdf`. Accessed: 2nd May 2015.

RITSCHEL, T., OKABE, M., THORMÄHLEN, T., AND SEIDEL, H.-P. 2009. Interactive reflection editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia '09) 28*, 5.

RITSCHEL, T., THORMÄHLEN, T., DACHSBACHER, C., KAUTZ, J., AND SEIDEL, H.-P. 2010. Interactive on-surface signal deformation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '10) 29*, 4.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Transactions on Graphics*, 533–540.

SCHMIDT, T.-W., NOVÁK, J., MENG, J., KAPLANYAN, A. S., REINER, T., NOWROUZEZAHRAI, D., AND DACHSBACHE, C. 2013. Path-space manipulation of physically-based light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '13) 32*, 4.

SCRATCHAPIXEL. Light transport. `http://www.scratchapixel.com/lessons/3d-basic-rendering/rendering-3d-scene-overview/introduction-light-transport`. Accessed: 14th April 2015.

WARD, G. J. 1992. Measuring and modeling anisotropic reflection. *ACM Transactions on Graphics (Proceedings of SIGGRAPH '92) 26*, 2, 265–272.

WYNN, C. An introduction to brdf-based lighting. `http://www.cs.princeton.edu/courses/archive/fall06/cos526/tmp/wynn.pdf`. Accessed: 4th May 2015.