A Flexible Pinhole Camera Model for Coherent Non-Uniform Sampling

Voicu Popescu, Bedrich Benes, Paul Rosen, Jian Cui, Lily Wang

Abstract — We present a pinhole camera model that allows modulating the sampling rate over the field of view with great flexibility. This flexible pinhole camera or FPC is defined by a viewpoint (i.e. eye) and by a sampling map that specifies the sampling locations on the image plane. The sampling map is constructed from known regions of interest with interactive and automatic approaches. The FPC provides an inexpensive 3-D projection operation which allows rendering complex datasets quickly, in feed-forward fashion, by projection followed by rasterization. The FPC supports many types of data, including image, height field, geometry, and volume data. The resulting image is a coherent non-uniform sampling (CoNUS) of the dataset that matches the local variation of the importance of the dataset. We demonstrate the advantages of CoNUS images in the contexts of remote visualization, of focus-plus-context, and of acceleration of expensive rendering effects such as rendering of surface geometric detail and of specular reflections.

Index Terms — camera models, non-uniform sampling, interactive rendering, remote visualization, specular reflection rendering, surface geometric detail rendering, volume rendering, focus-plus-context visualization.

1 Introduction

Most computer graphics and visualization applications employ images computed using the planar pinhole camera (PPC) model. The PPC is a good approximation of the human eye which makes it uniquely well suited for applications where the goal is to show users what they would see during an actual exploration of the scene. However, there are applications where the reduced field of view, the single viewpoint, and the uniform sampling rate limitations of the PPC model are a severe disadvantage.

In this paper we address the uniform sampling rate limitation of the PPC model. We introduce the flexible pinhole camera or FPC which allows for adjustments of the sampling rate according to the local importance or complexity of the data imaged. Like the PPC, the FPC is defined by a viewpoint (i.e. center of projection or eye) and an image plane. However, the sampling locations are not defined by a uniform grid but rather by a sampling map that allows shifting sampling locations from one region of the image plane to another. The FPC image provides a coherent non-uniform sampling (CoNUS) of the dataset. The CoNUS image in Figure 1, left, samples the five faces at a higher rate. The underlying sampling map is shown in Figure 4. The sampling map has the topology of a 32×32 regular rectangular mesh but it is distorted to implement the sampling rate modulation.

FPC CoNUS images preserve the advantages of conventional images. A CoNUS image can be computed quickly with the help of GPUs. Data access remains constant time, with the small additional cost of the sampling map indirection. A CoNUS image has good pixel to pixel coherence and conventional image compression algorithms apply. Finally, a CoNUS image remains a single-layer 2-D array of samples which defines connectivity implicitly.

The sampling map underlying the FPC can be constructed from known regions of interest in a variety of ways. We build sampling maps in one of three ways: (1) interactively, using a physics-based mass-spring system, (2) by composing multiple sampling maps together, or (3) analytically. The FPC provides fast 3-D projection which allows rendering CoNUS images quickly, in feed-forward fashion, by projection followed by rasterization, from many types of data. We demonstrate FPC rendering of CoNUS images from image, height field, geometry (i.e. 3-D triangle meshes), and volume data. We explore the use of CoNUS images in the contexts of remote visualization, of focus-plus-context visualization, and of acceleration of expensive effects such as surface geometric detail and specular reflection rendering. We also refer the reader to the accompanying video.

Consider the example of a high resolution portrait photograph that has to be downsampled to be sent over the internet (Figure 1). The recipient is likely to want greater detail on the faces, which, of course, cannot be provided by zooming into the conventional downsampled image. If instead the server sends a CoNUS image of same size but with a higher sampling rate at the known regions of interest, the user can zoom in with better results. Consider a second example where a height field is visualized remotely. The server sends periodically height field sections corresponding to the current user location. If instead the server sends CoNUS height fields, the fidelity of the output frame increases considerably (Figure 5). Like any height field, the CoNUS height field samples the ground plane orthogonally, avoiding occlusions. However, the sampling pattern is defined analytically to match the sampling rate requested by the output frames.

Many techniques employ depth images in order to accelerate expensive rendering effects. In relief texture mapping a depth image is used to enhance a surface with geometric detail. In specular reflection rendering the environment mapping approximation errors are avoided by modeling objects close to reflectors with depth images. The main reason depth images accelerate these effects is that one can compute the intersection between a ray and a depth image faster than one can compute the intersection between a ray and the original geometry. A CoNUS depth image brings sampling flexibility (Figures 2 and 6), without increasing the cost of the intersection operation.

Whereas in the examples presented so far the need for non-uniform sampling is to improve an auxiliary data representation from which conventional output images are computed, in the case of focus-plus-context visualization the CoNUS image is shown directly to the user. The FPC approach enables a versatile focus-plus-context visualization technique (Figures 3 and 7) that can handle any type of data and that provides good control over the focus regions. The

1. *Voicu Popescu and Jian Cui are with Computer Science Department in Purdue University, E-mail:{cui9, popescu@purdue.edu}*
2. *Bedrich Benes is with Computer Graphics Technology Department at Purdue University, E-mail bbenes@purdue.edu*
3. *Paul Rosen is with the Scientific Computing and Imaging Institute at the University of Utah, E-Mail: prosen@sci.utah.edu*
4. *Lili Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, the School of Computer Science and Engineering of Beihang University, E-mail: wanglily@buaa.edu.cn*

Fig. 1. Coherent non-uniform sampling (CoNUS) image that allocates more samples to the face regions (left), output frame reconstructed from CoNUS image (middle), and output frame reconstructed from a conventional image of same size (right), for comparison.

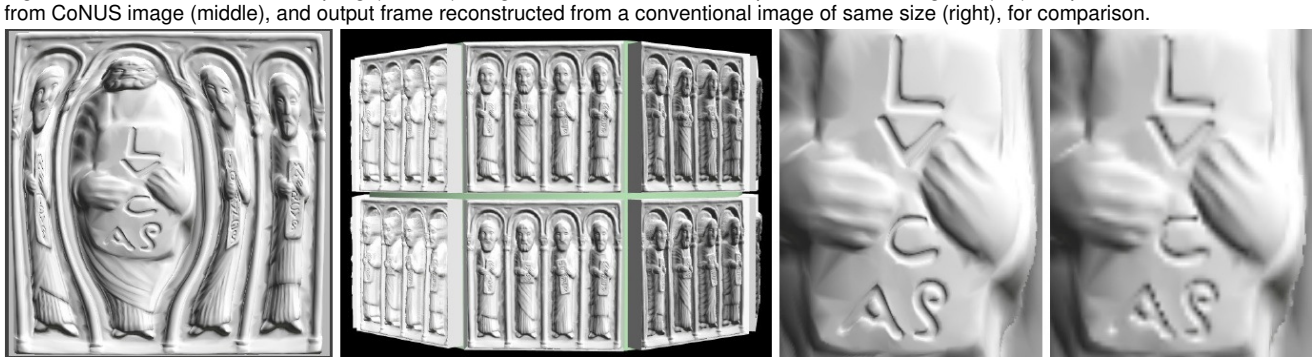

Fig. 2. CoNUS relief texture that allocates more samples to a tablet of interest (left), relief texture mapping (middle), and comparison between frames that zoom in on tablet of interest and are rendered with CoNUS and with conventional relief textures of same size (right).

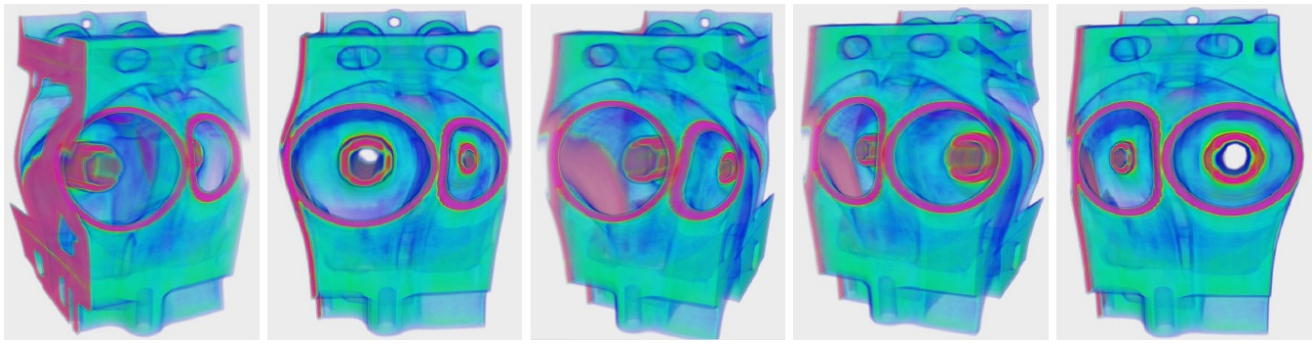

Fig. 3. CoNUS focus-plus-context volume rendering visualization emphasizing the left (1-3) and then the right (4-5) cylinder housings.

CoNUS images are rendered directly from the dataset (e.g. volume data, geometry) using the FPC, and they are not obtained by first rendering and then downsampling a high resolution conventional image.

2 Prior Work

We first review prior efforts aimed at removing the uniform sampling rate constraint of conventional images, and then review prior work in the application contexts where our paper examines the benefits of FPC rendered CoNUS images.

Non-uniform sampling rate. Hierarchical spatial partitioning schemes such as kd-trees do improve representation efficiency by stopping subdivision in regions where data is sampled accurately. One could define the FPC sampling map with such a partitioning scheme. Compared to our distorted grid approach (Figure 4), the hierarchical approach has the advantage of supporting a wider range of sampling rates, but it has the important disadvantages of sampling rate discontinuity, of lack of contiguity, and of greater construction (i.e. rendering) and usage (i.e. lookup) complexity.

Images with a non-uniform sampling rate were first obtained as a side effect of techniques for removing the field of view limitation of conventional images. For spherical panoramas the sampling rate variation was an unwanted side effect and they were replaced by cube maps with a more uniform sampling rate. Recently, single image panoramas have received renewed attention, as the programmability of graphics hardware enables sampling patterns that avoid the earlier undersampling problems [1]. Researchers have also been addressing the single viewpoint limitation of conventional images with innovations at the camera model level such as the general linear camera [2] and the occlusion camera [3]. The FPC CoNUS is complementary to these approaches, providing sampling rate flexibility to panoramic and non-pinhole cameras.

Irregular sampling patterns have also occurred in the contexts of image-based rendering by 3-D warping [4] and of shadow antialiasing [5]. In both cases depth images are re-projected to novel views where the forward mapped samples are irregular. The granularity with which sampling is controlled in the FPC CoNUS approach is insufficient to sample the shadow map precisely at the locations where the output image does, as needed to completely eliminate shadow aliasing. However, shadow aliasing could be reduced by using a CoNUS shadow map with a higher sampling rate in regions that are magnified in the output image.

The most general pinhole camera defines each ray independently with its own image plane point [6]. Such a camera model has the theoretical importance of maximum generality, allowing for any sampling pattern given n rays, but it has no practical use. First, the rays are not organized in a 2-D array and therefore the resulting

Fig. 4. Visualization of sampling map for Figure 1

image is an unsorted list of color samples which cannot be easily displayed. Second, rendering such an image is expensive as it would require tracing each ray independently. The practical implementation of the general pinhole camera that the researchers use [6] restricts the sampling rate variation to a rectangular region *R* of the image plane. A smaller rectangle *r*, concentric with *R*, provides a higher resolution sampling of the scene at that region of the image. The higher resolution region *r* is sampled with a planar pinhole camera and hence it is distortion free (i.e. 3-D scene lines map to 2-D image lines). The region *R-r* is used to transition from the low sampling rate outside *R* to the high sampling rate inside *r*. The sampling locations in *R-r* are chosen with a quadratic or cubic function to achieve C^0 or C^1 continuity. Several regions of higher resolution are supported as long as they are disjoint.

The FPC amounts to a different specialization of the abstract general pinhole camera model. The FPC CoNUS approach has two fundamental advantages over the previous specialization. First, the FPC provides far greater flexibility in defining the sampling locations. Second, as explained in the next sections, the FPC sampling map provides fine grain control of the sampling rate while keeping the amortized cost of the fundamental image point distortion and undistortion operations constant. In contrast, a general planar pinhole camera with multiple rectangular regions of high resolution requires checking each of the regions for the distortion / undistortion of a point, which does not scale with the number of regions.

In texture mapping non-uniform sampling has been pursued through compression, atlasing, enhancement with explicitly modeled high frequency features (e.g. edges), and distortion. We only discuss the last two approaches as they are closest to our work. Textures enhanced with edges modeling shadow silhouettes [7] or abrupt changes in color [8] are more robust to magnification. The approach is compatible with CoNUS textures. When edges are derived from vector graphics primitives the edges have to undergo the sampling map distortion (Figure 4), and long edges have to be split. For edges derived from the texture, the CoNUS texture can be used directly. Space-optimized textures [9] distort textures with a similar mechanism to our sampling map. Our FPC work extends nonuniform sampling to more types of data and applications.

Remote visualization. As the size of acquired and computed datasets continues to increase, so will the importance of remote visualization which is called upon to provide access to remote datasets for clients with no high-end storage or visualization capabilities. One approach is to reduce the dataset on the server to a size that can be transmitted to and visualized by the client. Many techniques can be used to reduce the dataset size on the server, including data compression (e.g. [10]), feature extraction (e.g. [11]), and level of detail (e.g. [12]). A second approach is to compute the visualization at the server and send images to the client (e.g. [13]). The client only needs a simple terminal that can display images, but network bandwidth limits visualization resolution and frame rate.

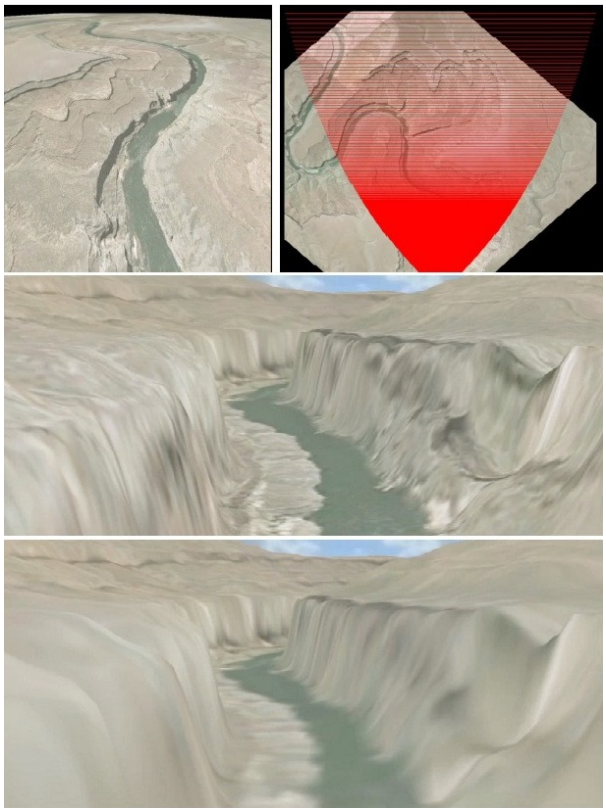

Fig. 5. CoNUS height field and its sampling pattern (top), output frame rendered from CoNUS (center), and from conventional height field (bottom).

A hybrid approach is to transfer from the server to the client images that have more data than what is needed for the client's current frame. Such an enhanced image should be sufficient for a quality reconstruction of a sequence of frames at the client, without any additional data from the server. Images enhanced with per pixel depth [14] and with additional samples at the center [6] have been used to allow translating and zooming in at the client. Remote visualization based on transferring CoNUS images falls in this third, hybrid category. A CoNUS image that samples known regions of interest in greater detail anticipates the user's intention to zoom in on those regions. A CoNUS height field that samples the ground plane orthogonally yet at a higher rate close to the user supports six degrees of freedom navigation at the client in the neighborhood of the current view.

Rendering acceleration using depth images. Depth images are powerful geometry approximations used for acceleration in many contexts including rendering of complex geometric surface detail, of specular reflections, of refractions, and of ambient occlusion. We limit the discussion of prior work to the first two contexts which are used in this paper to illustrate the benefits of our method.

Relief texture mapping is a technique for adding geometric detail to surfaces. The technique produces correct silhouettes and correct interactions between relief and other relief and non-relief geometry (e.g. intersections, casting and receiving shadows) [15]. The relief texture is a depth image attached to a base box. Rendering the box triggers intersecting the eye ray with the depth image at every pixel. The intersection computation is performed by projecting the ray onto the depth image and following the ray projection until the first intersection is found.

Specular reflections are challenging for the feed-forward 3-D graphics pipeline because one cannot easily compute the image plane projection of reflected vertices. We group specular reflection rendering techniques into four categories: ray tracing [16], approximations of the projection of reflected vertices (e.g. [17]),

Fig. 6. CoNUS depth image emphasizing all 4 engraved tablets (top left), scene setup (top right), and reflection details rendered with CoNUS (bottom left) and conv. (bottom right) depth image.

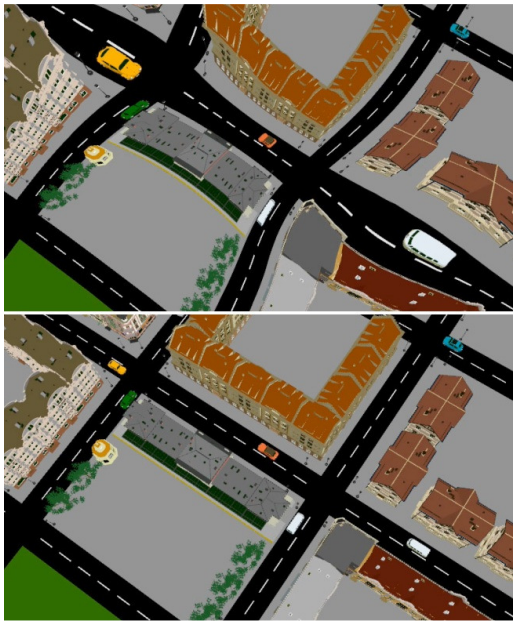

Fig. 7. CoNUS focus-plus-context visualization emphasizing the yellow and white cars (top), and conventional image (bottom).

image-based rendering (e.g. [18]), and approximations of the reflected scene. We only discuss the fourth category since it is the category where the CoNUS specular reflection rendering method falls. The most drastic approximation is undertaken by environment mapping [19], where the reflected scene is assumed to be infinitely far away from the reflector. Environment mapped reflections are incorrect for objects close to the reflector. Approximating these objects with billboards or depth images [18] improves reflection accuracy. Using CoNUS depth images as relief textures or to approximate reflected objects brings sampling flexibility without a considerable increase of the cost of ray / depth image intersection.

Focus-plus-context visualization. The visualization of complex scenes can benefit from highlighting the scene region that is more important in the context of the application. The pipeline of such focus-plus-context visualization has multiple stages, including finding the regions of interest, finding the best view for a region of interest, and highlighting the region of interest by assigning it a salient color, by assigning it more pixels, and by managing occlusions through cutaway, transparency, or non-pinhole camera techniques. For example, finding the best viewpoint for a region of interest can be done automatically by analyzing the region feature distribution in an information-theoretic framework [20]. We refer the reader to an excellent survey [21] of the state-of-the-art methods for the various stages of the focus-plus-context pipeline, and we limit the discussion to the problem of highlighting the region of interest by allocating more pixels to the region of interest, which is where the FPC makes it contribution to focus-plus-context visualization.

An important challenge stems from the fact that displays have a uniform pixel resolution (with the exception of special focus-plus-context screens [22]). Consequently, the focus-plus-context image cannot be displayed directly and it has to be mapped to displays with uniform resolution by introducing distortions between the focus and context regions. Focus-plus-context visualization is typically applied to 2-D data (e.g. to hierarchies [23], graphs [24], and maps [25]). Applying the technique to 3-D data can be done either by distorting the dataset and then visualizing it with a conventional camera [26], or by distorting the camera model [6], [27]. FPC focus-plus-context visualization falls in the second category. Like the general pinhole camera, the *volume lens* [27] defines one or a few regions of interest with higher resolution. The ray perturbation employed does not provide closed form projection and the method is restricted to volume rendering and ray tracing.

3 The Flexible Pinhole Camera

3.1 Camera Model

The goal is to define a camera whose rays pass through a point and that renders an image with a variable sampling rate, i.e. a CoNUS image. The camera has to be flexible, to allow defining the desired sampling rate for sub-regions of the CoNUS image, and it has to be fast, in order to render the CoNUS image quickly from a variety of types of input data. We implement the sampling rate variation with a sampling map that defines a distortion of a regular 2-D mesh. The distorted mesh has the same topology as a regular 2-D mesh, but with quadrilateral cells that are larger where a higher sampling rate is desired (Figure 4). The sampling map is encoded as a 2-D array of 2-D points. Each point defines a node of the distorted mesh.

Given an (undistorted) image point (u, v), the corresponding (distorted) CoNUS image point (u_d, v_d) is found by looking up the sampling map using bilinear interpolation as shown in Algorithm 1 and Figure 8. The input point is first converted to sampling map coordinates (u', v') (line 1). Then the distorted point is computed by bilinear interpolation of the four distorted mesh points stored in the sampling map at the 2×2 neighborhood containing (u', v') (line 2).

Camera model definition. We define the flexible pinhole camera model *FPC* with a conventional planar pinhole camera *PPC* and a sampling map *SM* that distorts the *PPC* image as described in Algorithm 1.

Projection. A flexible pinhole camera *FPC*(*PPC, SM*) projects a 3-D point *P* to its CoNUS image plane by first projecting *P* with *PPC* to obtain the undistorted coordinates (u, v) and then by distorting (u, v) to (u_d, v_d) (Algorithm 1).

Camera rays. The *FPC*(*PPC, SM*) ray through (u_d, v_d) is the *PPC* ray through (u, v). Consequently, in order to compute the camera ray, one needs to invert the distortion, which poses two challenges. First, one has to find the quadrilateral cell of the distorted mesh that

Algorithm 1: FPC::*Distort*(u, v) // FPC distortion

input: undistorted image resolution (w, h), undistorted location (u, v), and sampling map *SM* of resolution (w_0, h_0)

output: distorted location (u_d, v_d)

1: $(u', v') = (uw_0/w, vh_0/h)$

2: $(u_d, v_d) = SM$.BilinearLookup(u', v')

Algorithm 2: FPC::*Rays*() // Computation of FPC rays

input: *FPC* of resolution $w \times h$, defined by *PPC* and by *SM* of resolution $w_0 \times h_0$
output: *FPC* rays
1: Initialize 2-D mesh *QM* of resolution $w_0 \times h_0$
2: **for all** (i, j) where $0 \le i < w_0$, $0 \le j < h_0$ **do**
3: Vertex coordinates $QM.v_{i,j} = SM_{i,j}$;
4: Texture coordinates $QM.(s, t)_{i,j} = (i/w_0, j/h_0)$;
5: **end for**
6: **for all** triangles q in *QM* **do**
7: **for all** pixels p covered by q **do**
8: $(u, v) = (ws_p, ht_p)$
9: $ray_p = PPC.GetRay(u, v)$
10: **end for**
11: **end for**

contains (u_d, v_d). A naive approach would examine all quads. A better approach would use a hierarchical subdivision of the CoNUS image (e.g. using a kd-tree or a BSP-tree) to quickly find the quad that contains (u_d, v_d), but constructing the subdivision remains laborious. Second one needs to solve the quadratic equations of the inverse bilinear interpolation that computes x and y from (u_d, v_d), $SM_{i,j}$, $SM_{i+1,j}$, $SM_{i,j+1}$, and $SM_{i+1,j+1}$.

We bypass these challenges by leveraging two observations. First, CoNUS applications do not need to compute an individual ray of the *FPC*, but rather all rays iteratively. Second, bilinear interpolation inversion can be avoided by splitting the distorted mesh quads into two triangles and by replacing the quad bilinear interpolation with two triangle barycentric interpolations. This modification does not reduce the sampling rate flexibility of the FPC. We find all rays of the modified *FPC* efficiently as shown in Algorithm 2.

The rays of the *FPC* are found by rasterizing the distorted mesh *QM* defined by the sampling map (line 3). *QM* has the topology of a 2-D regular mesh, but its vertices are displaced according to the desired sampling rate variation (see Figure 4). Each distorted mesh vertex carries its undistorted coordinates as texture coordinates (line 4). The ray at the current pixel p is found by first finding the undistorted coordinates (u, v) of p from its texture coordinates (s_p, t_p) (line 8), and then by computing the regular planar pinhole camera ray at the undistorted coordinates (line 9). The rays are found at the cost of rasterizing the $2 \times w_0 \times h_0$ triangles of the distorted mesh, which is small since the resolution of the sampling map is much smaller than the resolution of the CoNUS image. We compute the rays on the GPU with a trivial fragment shader that executes lines 8 and 9. Algorithm 2 provides the rays of the FPC camera, one at a time, at a small amortized cost. The algorithm is not used as is, but it is rather specialized as needed to render a CoNUS image from a regular image or from volume data, as described below.

3.2 Rendering CoNUS images with the FPC

The FPC allows rendering CoNUS images efficiently from a variety of types of input data.

Geometry data. A CoNUS image is rendered from a 3-D triangle mesh with the steps shown in Algorithm 3. The 3-D triangle mesh T is projected by projecting its vertices with *FPC* (i.e. *PPC* projection followed by distortion with Algorithm 1, Section 3.1). Then the projected triangles are rasterized conventionally. The projected triangles have to be small enough such that conventional rasterization provides a good approximation of the nonlinear projection induced by the sampling map. Most datasets have small triangles and conventional rasterization is acceptable without further subdivision. When subdivision is needed, an offline approach is preferred in order to avoid the performance bottleneck of issuing a large number of primitives in the geometry shader.

Algorithm 3: FPC::*Render*(*T*) // render from geometry

input: FPC *FPC* and triangle mesh T
output: CoNUS image I
1: **for all** vertices v of T **do**
2: $v' = FPC.Project(v)$
3: **end for**
4: **for all** projected triangles t' of T **do**
5: Rasterize t'
6: **end for**

Image and height field data. A CoNUS image is rendered from a conventional input image by modifying line 9 of Algorithm 2 as shown in Algorithm 4. Once the undistorted location (u, v) is known, the input image is looked up to set the current CoNUS image pixel (u_d, v_d). A CoNUS image has fewer pixels than the original image. The original image provides the maximum resolution over the entire field of view, which is preserved in some regions of the CoNUS image, whereas the other regions of the CoNUS image are at lower resolution. A CoNUS height field sampled orthogonally to the base plane is constructed similarly with the exception that the pixel is setup by looking up the depth in the original height field instead of (or in addition to) looking up the color.

Volume data. A CoNUS image is rendered from volume data by tracing the FPC rays through the volume. The rays are determined with Algorithm 2.

Algorithm 4: FPC::*Render*(*I*) // render from image

input: FPC *FPC* and image I
output: CoNUS image I'
// identical to Algorithm 2 except for line 9
…
9: $I'(u_d, v_d) = I(u, v)$ // difference with Algorithm 2
…

3.3 Resampling of regular image from CoNUS image

We have described rendering a CoNUS image from a regular image. However, some applications, such as remote visualization, use the CoNUS image as an intermediate representation from which they have to resample a conventional image to be presented to the user. A regular image I_1 is resampled from a CoNUS image I_0 with the steps shown in Algorithm 5. The rays that sample I_1 are defined by a planar pinhole camera PPC_1. Given an I_1 pixel (u_1, v_1), the corresponding CoNUS image pixel (u_d, v_d) is computed in two steps. First, one computes the corresponding point (u_0, v_0) on the image plane of PPC_0, as shown by lines 2 and 3. This correspondence is computed by generating the 3-D point P corresponding to (u_1, v_1) by unprojection with PPC_1 and then by projecting P with PPC_0. The unprojection followed by projection can be combined into a single matrix multiplication followed by perspective divides. Second, the corresponding (u_d, v_d) is computed by distortion leveraging Algorithm 1.

3.4 Sampling map construction

We construct sampling maps in one of three ways. One way is through the use of an interactive physics-based 2-D mass-spring system. The image is covered with regularly distributed particles connected with springs to form a quadrilateral mesh. All particles have the same mass and all springs have the same resting length (set to 10% of the initial particle distance in our implementation). The user perturbs the system interactively by adding repulsive forces between particles with a circular brush (Figure 9). The force magnitude decreases from the center towards the periphery of the brush exponentially. The equilibrium state is computed by tracking the position of each particle over time until all particle velocity vectors have negligible magnitude. For each time step, the forces acting on each particle are computed first using Hooke's equation for

Algorithm 5: FPC::*CoNUS2Regular*(I_0) // Resampling

input: CoNUS image I_0, $FPC(PPC_0, SM)$, PPC_1
output: Conventional image I_1 for PPC_1

1: **for all** pixels (u_1, v_1) in I_1 **do**
2: $P = PPC_1.Unproject(u_1, v_1)$
3: $(u_0, v_0) = PPC_0.Project(P)$
4: $(u_d, v_d) = FPC.Distort(u_0, v_0)$ // see Algorithm 1
5: $I_1(u_1, v_1) = I_0(u_d, v_d)$
6: **end for**

harmonic oscillators $F_i = -kx_i$, where F_i is the force applied to the particle by spring i connected to it, k is the spring constant, and x_i is the particle displacement along the spring direction. Then the particle velocity $\boldsymbol{v}$ and displacement $\boldsymbol{x}$ are updated with equations $\boldsymbol{v} = \boldsymbol{v} + \Delta t\, \boldsymbol{F}/m$ and $\boldsymbol{x} = \boldsymbol{x} + \Delta t\boldsymbol{v}$, where $\boldsymbol{F}$ is the resultant force acting on the particle. A mesh of 256×256 particles is updated at 30 fps and a stable state is reached in less than 2s. The sampling map is defined by the final position of the particles, and it can have a lower resolution than the particle mesh.

Sampling maps can also be generated through a linear combination of the distortion vectors of existing sampling maps, as shown in the equation below. $SM_{i,j}$, $SM_{i,j}^{0}$, and $SM_{i,j}^{k}$ are elements (i, j) of the new sampling map, of the undistorted sampling map, and of the input sampling map k. s_k and t_k are the scale factor and the translation vector of map k.

$$SM_{i,j} = SM_{i,j}^{0} + \Sigma_k (s_k (SM_{i,j}^{k} - SM_{i,j}^{0}) + t_k)$$

A third approach is to do away with the discrete representation and define the distortion analytically as shown in Figure 5, top right, and as described in Section 4.

4 Remote Visualization

The resolution of digital cameras continues to increase faster than network bandwidth. It is also the case that workstation displays now have a lower resolution than the simplest digital cameras attached to cellular phones (e.g. Apple's 4MP 30" LCD and 8MP iPhone 5S camera). Consequently, even if the image is transferred at full resolution, it is most likely going to be downsized for viewing. Often not all pixels in a digital image have the same relevance for the application. For example faces in a portrait photograph are more important than the furnishings in the room. Moreover faces are found automatically by digital cameras for focusing purposes. In the context of an online geographic atlas, pixels sampling famous locations or locations marked by other users as interesting have higher relevance. In the context of remote scientific visualization, some image regions might be known to be of higher interest to scientists, such as regions showing receptors targeted in drug molecule design.

In such contexts, the CoNUS capability of the FPC could help reduce bandwidth requirements and improve interactivity as follows. The server FPC renders a CoNUS image that samples the regions of interest at a higher rate. Then the CoNUS image is transferred to the client, where it is resampled to a conventional image (Algorithm 5). The application tours the CoNUS image, showing the regions of interest in detail.

We have also investigated the use of the FPC CoNUS approach in the context of remote terrain visualization. Given a height field H at the server and a current view PPC at the client, the goal is to resample H to a CoNUS height field that has all and only the samples needed to provide a quality visualization of the height field from views in the neighbourhood of PPC. First, a reference view PPC_0 is constructed by enlarging the field of view of PPC, to support view rotations, and by increasing the resolution, to support zooming in and forward translation. Then a CoNUS height field is constructed with a sampling rate that matches the requirements of PPC_0. CH should have more samples close to the viewpoint and fewer at a distance, as illustrated in Figure 5, top right. We construct the CoNUS height field CH with an analytical distortion function as described in Algorithm 6.

Algorithm 6: *HeightFieldCoNUS*(H, PPC_0)

input: Height field H, client reference view PPC_0
output: CoNUS height field CH

1: **for all** samples (u_d, v_d) in CH **do**
2: $(u, v) = PPC_0.Ray(u_d, v_d) \cap H.g$
3: $CH(u_d, v_d) = H(u, v)$
4: **end for**

The CoNUS height field sample (u_d, v_d) is looked up in the original (undistorted) height field H at location (u, v) which is

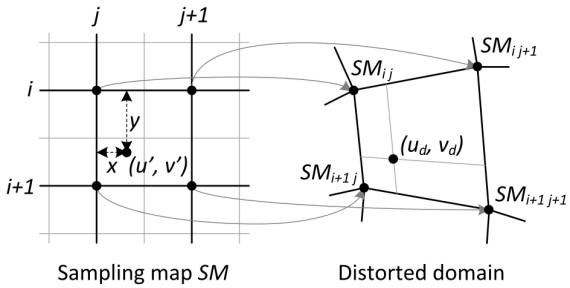

Fig. 8. Piecewise bilinear image distortion using a sampling map.

computed by intersecting the ray at pixel (u_d, v_d) in the client view PPC_0 with the ground plane $H.g$ of H. This construction applies the perspective foreshortening of PPC_0 while maintaining the orthogonal sampling of H, which avoids disocclusion error problems that would occur if one actually rendered the geometry of H from PPC_0. CH is sent to the client where it is transformed in a 3-D triangle mesh that is rendered for each frame. A CH sample is converted to a 3-D triangle mesh vertex by computing the ground plane point P corresponding to (u_d, v_d) (line 2 in Algorithm 6) and by offseting P by $CH(u_d, v_d)$ above the ground plane.

Quality. The CoNUS image shown in Figure 1 allows rendering all five faces in great detail. The CoNUS height field produces frames that are comparable to frames rendered from the original high-resolution height field (Figure 5).

Performance. For Figure 1, once the FPC model is known, rendering the CoNUS image takes negligible time; the FPC sampling map was designed interactively using the spring-mass system. For the example in Figure 5, we use a CoNUS height field of 1,024 ×1,024 resolution, which is rendered and used at over 400 and 100 frames per second, respectively.

Limitations. The FPC CoNUS approach increases the sampling rate of the regions of interest at the expense of the rest of the image. When high frequencies are present outside the regions of interest, the undersampling can become noticeable (Figure 10). The approach does not address occlusions. Whereas occlusions do not occur for images or orthogonally sampled height fields, the FPC CoNUS approach will have to be integrated with an occlusion alleviation scheme such as a non-pinhole camera to support six degrees of freedom remote visualization of general 3-D data.

5 Depth Image Rendering Acceleration

A depth image is a powerful method for approximating geometry: the depth image is computed quickly with the help of graphics hardware, and a depth image can be quickly intersected with a ray. Because of these important advantages depth images have been used to accelerate the rendering of complex effects such as specular reflections, refractions, ambient occlusion, and relief texture mapping. Eliminating the uniform sampling rate constraint of conventional depth images using the FPC CoNUS approach could benefit all these techniques provided that the efficiency of depth image construction and of ray intersection is preserved. CoNUS depth images can be rendered efficiently from height field or geometry data using the FPC as discussed in Section 3.

Fig. 9. Mass-spring system used to define sampling maps interactively. The user defines regions of higher resolution using a circular brush (yellow).

A conventional depth image *DI* is intersected with a ray by projecting the ray to the image plane of *DI* and by tracing the projection with one pixel steps until an intersection is found [15]. In the case of a CoNUS depth image, the projection of the ray is no longer a line segment, but rather a curve segment. The ray cannot longer be projected solely by projecting its endpoints. Instead, the ray has to be subdivided into segments, each projected with the *FPC* as described in Section 3. The fundamental advantage of depth images of 1-D intersection with a ray is preserved, at the cost of a slightly more complicated projection of the ray. We integrated CoNUS depth images into relief texture mapping and into specular reflection rendering, where the CoNUS depth image is intersected with eye rays and with reflected rays, respectively.

Quality. The sampling flexibility afforded by CoNUS depth images allowed improving the clarity of the engraved tablets (Figure 2) and of their reflection (Figure 6).

Performance. For both conventional and CoNUS depth images, the performance bottleneck for relief texture mapping and specular reflection rendering is the depth image / ray intersection computation. Intersecting a ray with a CoNUS depth image brings the additional cost of distorting a 2-D point at every step along the ray. However, CoNUS distortion is fast, and we measured an average frame rate penalty of only 5%. For applications where the CoNUS depth image is intersected with a large number of rays, it might be advantageous to undistort the CoNUS depth image at the client to a higher resolution conventional depth image using Algorithm 5, which results in straight ray projections and avoids the cost of per step distortion.

Limitations. CoNUS depth images inherit the occlusion limitations of conventional depth images. The sampling tradeoff can lead to visual artifacts outside regions of interest.

6 Focus-Plus-Context Visualization

The FPC CoNUS approach is well suited for focus-plus-context visualization because it offers good control over the sampling rate, which allows precisely designing one or multiple focus regions, and because CoNUS images can be rendered quickly, which supports dynamic scenes and the interactive change of focus region parameters. The CoNUS image is shown directly to the user thus no decoding is needed. The CoNUS image can be rendered efficiently from a variety of data as described in Section 3. The only remaining challenge is sampling map construction.

Unlike for the previous applications of CoNUS images, in focus plus-context-visualization the sampling map has to be constructed online, once for every output frame, which precludes the use of the mass-spring approach. We construct sampling maps by composing canonical circular sampling maps, one for every focus region. We demonstrate the approach in the context of volume rendering (Figure 3), where the user manipulates focus region and view parameters interactively to examine a volume dataset, and in the context of a city scene modelled with triangle meshes (Figure 7), where focus regions track moving cars. The focus region location is computed by projecting the center of the tracked car in the output view.

Quality. The CoNUS approach enables high quality focus-plus-context visualization for a variety of data types. The focus regions have strong magnification and low distortion. Focus region parameters can change and focus regions can merge and then separate again without abrupt changes in the output visualization. Focus plus context visualization is particularly robust to undersampling outside the focus region--users are likely to focus on the region that they themselves selected as important, and focus regions can be shifted interactively to visualize any region in more detail.

Performance. In our experiments FPC volume rendering was on average 7% slower than conventional volume rendering. The cost of volume rendering by ray casting is dominated by the traversal of the volume, thus computing the perturbed rays for the CoNUS approach has no impact on performance. We attribute the slight performance decrease to a larger output image footprint for the distorted volume, and to more rays being focused on the center of the dataset where volume traversal distances are longer. The vertex distortion performed when rendering CoNUS images from triangle meshes had no measurable performance impact.

Limitations. Since the CoNUS approach does not alleviate occlusions, tracked objects of interest can become hidden and the user has to change the view to reveal the object. As future work we will examine changing the view automatically to keep the tracked object visible.

7 Conclusions and Future Work

We have presented a general method for removing the uniform sampling rate constraint of conventional images. CoNUS images can be rendered efficiently from image, height field, geometric, and volume data. Like a conventional image, a CoNUS image has a single layer and good pixel to pixel coherence, thus conventional image compression algorithms can be readily applied. The underlying sampling map can be constructed from known regions of interest in a variety of ways including using a mass spring system, by composing multiple input sampling maps, and analytically.

The sampling map is a powerful tool for assigning more pixels to some regions of the image plane. For example, for the image in Figure 1, the maximum sampling rate increase is 8.13×, respectively, which was measured by finding the largest quadrilateral cell of the sampling mesh, and by dividing its area to the area of an undistorted cell. The sampling map does not create new pixels--the sampling rate is increased by decreasing the sampling rate in other regions deemed of lesser importance.

For a sampling map of resolution $w_0 \times h_0$, for regions of interest occupying k cells, and for a minimum sampling rate of the context regions of c×, the upper bound for the sampling rate increase is $z = w_0h_0(1\text{-}c)/k + c$. For example, if $w_0 \times h_0 = 1{,}024$, $k = 64$ and $c = 1/2$, then $z = 8.5$×. If the application tolerates downsampling the context to 1/8, z increases to 14.125×. If there is a single region of interest that fits in one cell, i.e. $k = 1$, then, even for a negligible downsampling of the context regions by $c = 0.95$×, the sampling rate

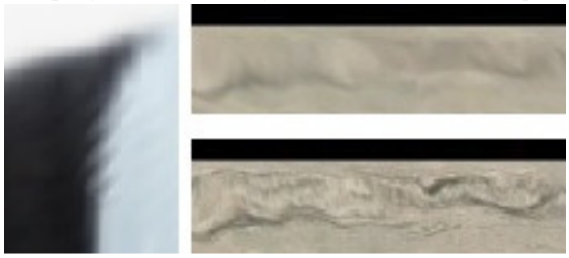

Fig. 10. Sampling artifact outside the regions of interest in a frame reconstructed from the CoNUS image in Figure 1 (left), and undersampling of distant mountain by CoNUS height field (right, top) compared to original height field (right, bottom).

of the region of interest can reach $z = 52.15\times$.

Possible directions for future work include exploring other uses of CoNUS images (e.g. geometric simplification, acceleration of additional rendering effects), investigating the benefit/cost tradeoff of higher order interpolation of the sampling map to achieve C^1 sampling rate continuity, and developing automatic sampling map constructors. This paper describes how to sample at a non-uniform rate. We are particularly interested in tightly coupling the FPC CoNUS approach with automatic techniques for determining what to sample in more detail, such as automatic geometric complexity analysis, object recognition, eye tracking, and saliency maps.

We foresee that FPC-rendered CoNUS images will have wide applicability as they are compatible with virtually all contexts where images are used.

8 Acknowledgement

We would like to thank the anonymous reviewers and the associate editor for their help with improving this manuscript. We would also like to thank the United States National Science Foundation for supporting the work of Dr. Popescu and of Jian Cui through grant No.1217215, as well the National Natural Science Foundation of China for supporting the work of Dr. Lili Wang through projects 61272349, 61190121, and 61190125.

References

[1] J.-D. Gascuel, N. Holzschuch, G. Fournier, and B. P´eroche, "Fast nonlinear projections using graphics hardware," in Proceedings of the 2008 symposium on Interactive 3D graphics and games. ACM, 2008, pp. 107–114.

[2] J. Yu and L. McMillan, "General linear cameras," in Computer Vision-ECCV 2004. Springer, 2004, pp. 14–27.

[3] C. Mei, V. Popescu, and E. Sacks, "The occlusion camera," in Computer Graphics Forum, vol. 24, no. 3. Wiley Online Library, 2005, pp. 335–342.

[4] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. ACM, 1995, pp. 39–46.

[5] G. S. Johnson, J. Lee, C. A. Burns, and W. R. Mark, "The irregular z-buffer: Hardware acceleration for irregular data structures," ACMTransactions on Graphics (TOG), vol. 24, no. 4, pp. 1462–1482, 2005.

[6] V. Popescu, P. Rosen, L. Arns, X. Tricoche, C. Wyman, and C. M. Hoffmann, "The general pinhole camera: Effective and efficient nonuniform sampling for visualization," Visualization and Computer Graphics, IEEE Transactions on, vol. 16, no. 5, pp. 777–790, 2010.

[7] P. Sen, M. Cammarano, and P. Hanrahan, "Shadow silhouette maps," in ACM Transactions on Graphics (TOG), vol. 22, no. 3. ACM, 2003, pp. 521–526.

[8] G. Ramanarayanan, K. Bala, and B. Walter, "Feature-based textures," Cornell University, Tech. Rep., 2004.

[9] L. Balmelli, G. Taubin, and F. Bernardini, "Space-optimized texture maps," in Computer Graphics Forum, vol. 21, no. 3. Wiley Online Library, 2002, pp. 411–420.

[10] L. Lippert, M. H. Gross, and C. Kurmann, "Compression domain volume rendering for distributed environments," in Computer Graphics Forum, vol. 16, no. s3. Wiley Online Library, 1997, pp. C95–C107.

[11] Y. Livnat, S. G. Parker, and C. R. Johnson, "Fast isosurface extraction methods for large image data sets," in Handbook of medical imaging. Academic Press, Inc., 2000, pp. 731–745.

[12] S. P. Callahan, J. L. D. Comba, P. Shirley, and C. T. Silva, "Interactive rendering of large unstructured grids using dynamic level-of-detail," in Visualization, 2005. VIS 05. IEEE. IEEE, 2005, pp. 199–206.

[13] S. Stegmaier, M. Magall´on, and T. Ertl, "A generic solution for hardware-accelerated remote visualization," in Proceedings of the symposium on Data Visualisation 2002. Eurographics Association, 2002, pp. 87–ff.

[14] E. J. Luke and C. D. Hansen, "Semotus visum: a flexible remote visualization framework," in Proceedings of the conference on Visualization'02. IEEE Computer Society, 2002, pp. 61–68.

[15] F. Policarpo and M. M. Oliveira, "Relief mapping of non-height-field surface details," in Proceedings of the 2006 symposium on Interactive 3D graphics and games. ACM, 2006, pp. 55–62.

[16] T. Whitted, "An improved illumination model for shaded display," in ACM SIGGRAPH 2005 Courses. ACM, 2005, p. 4.

[17] E. Ofek and A. Rappoport, "Interactive reflections on curved objects," in Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, 1998, pp. 333–342.

[18] L. Szirmay-Kalos, B. Asz´odi, I. Laz´anyi, and M. Premecz, "Approximate ray-tracing on the gpu with distance impostors," in Computer Graphics Forum, vol. 24, no. 3. Wiley Online Library, 2005, pp. 695–704.

[19] J. F. Blinn and M. E. Newell, "Texture and reflection in computer generated images," Communications of the ACM, vol. 19, no. 10, pp. 542–547, 1976.

[20] I. Viola, M. Feixas, M. Sbert, and M. E. Groller, "Importance-driven focus of attention," Visualization and Computer Graphics, IEEE Transactions on, vol. 12, no. 5, pp. 933–940, 2006.

[21] S. Bruckner, M. E. Gr¨oller, K. Mueller, B. Preim, and D. Silver, "Illustrative focus+ context approaches in interactive volume visualization." in Scientific Visualization: Advanced Concepts, 2010, pp. 136–162.

[22] P. Baudisch, N. Good, and P. Stewart, "Focus plus context screens: combining display technology with visualization techniques," in Proceedings of the 14th annual ACM symposium on User interface software and technology. ACM, 2001, pp. 31–40.

[23] J. Lamping and R. Rao, "The hyperbolic browser: A focus+ context technique for visualizing large hierarchies," Journal of Visual Languages & Computing, vol. 7, no. 1, pp. 33–55, 1996.

[24] N. Wong, S. Carpendale, and S. Greenberg, "Edgelens: An interactive method for managing edge congestion in graphs," in Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on. IEEE, 2003, pp. 51–58.

[25] E. Pietriga and C. Appert, "Sigma lenses: focus-context transitions combining space, time and translucence," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2008, pp.1343–1352.

[26] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia, "Distortion viewing techniques for 3-dimensional data." in InfoVis, vol. 96, 1996, pp. 46–53.

[27] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman, "The magic volume lens: An interactive focus+ context technique for volume rendering," in Visualization, 2005. VIS 05. IEEE. IEEE, 2005, pp. 367–374.

Voicu Popescu received the BS degree in computer science from the Technical University of Cluj-Napoca, Romania in 1995, and the PhD degree in computer science from the University of North Carolina at Chapel Hill in 2001. He is an associate professor with the Computer Science Department of Purdue University. His research interests lie in the areas of computer graphics, computer vision, and visualization. His current projects include camera model design, remote visualization, aggressive and exact visibility computation, and applications of computer graphics in education.

Bedrich Benes is an associate professor in the Computer Graphics Technology department at Purdue University and Purdue Faculty Scholar. He obtained his Ph.D. and M.S. degree from the Czech Technical University. His research is primarily in the area of procedural modeling, real-time rendering, and 3D computer graphics in general. He is a director of Purdue University High Performance Computer Graphics Laboratory.

Paul Rosen is a Research Assistant Professor at the University of Utah with appointments in the Scientific Computing and Imaging (SCI) Institute and the School of Computing. Dr. Rosen received his PhD from the Computer Science Department of Purdue University where his dissertation was about Camera Model Design, a problem-solving paradigm which advocates designing dynamic, application specific camera models for solving problems in computer graphics, visualization, and computer vision. Dr. Rosen's research interests include topics in Scientific Visualization, such as vector field and uncertainty visualization, and Information Visualization, such as parameter space and software performance visualization.

Jian Cui received his BS in computer science from the Harbin Institute of Technology, China in 2009. He is a PhD candidate in computer science at Purdue University. His research interests span computer graphics and computer vision. His current work focuses on image generalization through camera model design to overcome the single viewpoint and uniform sampling rate limitations of conventional images.

Lili Wang received the BS degree and the Ph.D. degree from the Beihang University, Beijing, China. She is an associate professor with the School of Computer Science and Engineering of Beihang University, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include real-time rendering, realistic rendering, global illumination, soft shadow and texture synthesis.