A NEURAL NETWORK REPRESENTATION OF THREE-DIMENSIONAL GEOMETRICAL SCENE DESCRIPTIONS

Christian-A. Bohn

Dept. of Visualization and Media Systems Design
German National Research Center for Information Technology
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
Phone: +49-2241-14-2230, -2040(fax), e-mail: bohn@gmd.de

ABSTRACT

We present a new kind of representation of three-dimensional virtual geometries, which is realized through the internal structure of an artificial neural network.
Based on this representation we propose an alternative to a finite element algorithm coming from the field of computer graphics, termed radiosity, which calculates the global light flow in virtual environments.
Like many finite element methods, radiosity adheres to the problem of finding an optimal meshing of the domain under consideration. We leave this to the self-organization capabilities of an incremental growing cell structures network. Sets of light rays are taken as training samples and generate a topology of radial basis functions, which finally is interpreted as a "neural mesh" for a finite element computation.

KEYWORDS

Artificial neural networks, incremental growing cell structures, radial basis functions, self organization, computer graphics, global illumination, radiosity, rendering.

INTRODUCTION

An important direction in the field of computer graphics focuses on the question: given a database defining a set of three-dimensional virtual objects and light sources, which color would each virtual surface point have if surfaces and light sources were real, and if the arrangement is observed from a specific view point. In this context, color is not only determined by light sources, but also by indirect illumination of light refracted from the objects in the environment.

Global illumination algorithms try to calculate this information by simulating the flow of light due to its physical behavior formulated by two relationships — the *bidirectional reflection distribution function* (BRDF), on one hand, which describes the reflection properties of the objects' surface points, and the *rendering equation*, on the other hand [1], which defines the propagation of light between separate instantiations of BRDFs.
The BRDF on a certain point is a four-dimensional function of the incident and reflected direction of light. Due to the amount of possible BRDF instantiations to guarantee an adequate result, approximations of the BRDF and the rendering equation are required. One one hand, these approximations should reduce the amount of computing power to a practicable level, on the other hand, they have to be chosen carefully, since they reduce the accuracy of the results, i.e., they decrease the quality of the generated pictures, in the sense that their degree of realism is diminished.

Radiosity algorithms are a widely accepted kind of those approximation methods, since they deliver a good compromise between quality and required computing resources.

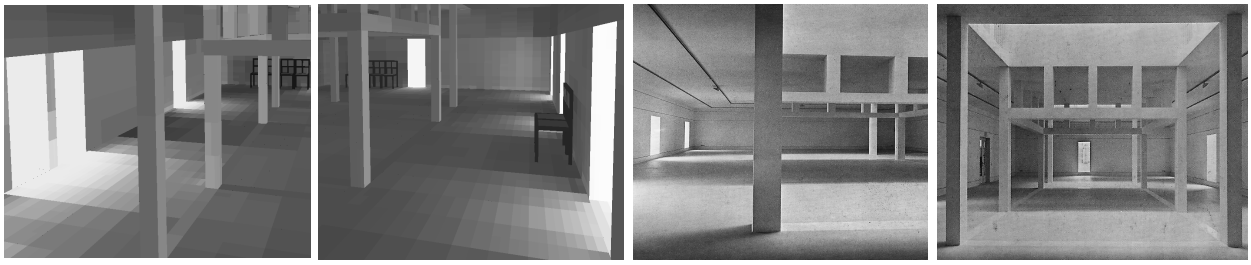

Fig. 1: The pictures to the right are photographs taken at the museum of architecture in Frankfurt, those to the left are radiosity computer simulations of a museum model.

First, the surfaces' BRDFs are simplified to just a coefficient $\rho \in \mathbb{R}$ which defines how much energy is absorbed regardless of the incoming and outgoing directions of light; its physical interpretation is the limitation of the environment to ideal diffuse reflecting and emitting objects only. Second, the light flow is approximated by a *finite element method* (FEM) which utilizes coherence in the light transfer by computing "certain averages over similar light rays". An illustrating result calculated by a radiosity approach can be seen in figure 1.
Through the diffusely defined BRDF, the rendering equation is written as the *radiosity integral equation*,

$$B(\vec{y}) = E(\vec{y}) + \int_{\mathcal{S}} \mathcal{K}(\vec{x}, \vec{y}) B(\vec{x}) d\vec{x}, \tag{1}$$

with

$$\mathcal{K} = \rho \cdot G \cdot V, \quad \text{and} \quad G(\vec{x}, \vec{y}) = \frac{\cos \phi_{\vec{x}} \cos \phi_{\vec{y}}}{\pi r^2_{\vec{x}\vec{y}}}. \tag{2}$$

$B : \mathbb{R}^3 \to \mathbb{R}$ denotes a continuous function which delivers the intensities[1] of points $\vec{y} \in \mathbb{R}^3$ on the surface set $\mathcal{S}$ defining the three-dimensional environment. The *kernel operator* $\mathcal{K} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ describes the effects concerning the light transfer, which an emitter point $\vec{x} \in \mathbb{R}^3$ bears on a receiving point $\vec{y}$. It is defined by a physically determined light transfer function $G : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$, $\phi_{\vec{x}}$ and $\phi_{\vec{y}}$ are angles between the surface normals at $\vec{x}$ and $\vec{y}$ and the connecting line from $\vec{x}$ to $\vec{y}$, $r_{\vec{x}\vec{y}}$ is the distance between $\vec{x}$ and $\vec{y}$. $V : \mathbb{R}^3 \times \mathbb{R}^3 \to \{1, 0\}$ is called the *visibility term* which delivers the value 1 if $\vec{x}$ and $\vec{y}$ are mutually visible, i.e., not occluded by other objects, otherwise zero. The reflection coefficient ρ (the diffuse BRDF) determines the amount of energy which is absorbed by refraction on a surface point, and $E : \mathbb{R}^3 \to \mathbb{R}$ stands for the energy which a surface point emits in case of being a light source.
To solve equation (1) it is considered a Fredholm integral equation of the second kind. The operator $\mathcal{K}$ weighs and accumulates the radiosity B on the surfaces defining the amount of energy which receives on the surface points from the whole geometry. Multiple applications of $\mathcal{K}$, which can be seen as multiple separate reflections of light on surfaces, deliver the finite Neumann series $B = (I + \sum_l \mathcal{K}^l)E$ which converges due to the definition of $\mathcal{K}$.
For an approximate solution equation (1) is transformed into its FEM representation, commonly done by cutting the surface domain $\mathcal{S}$ into N subpatches and assuming one value $b_i, i = 1..N$ as an average intensity defined on the area of the corresponding patch. The continuous functions B, E, and $\mathcal{K}$ change to their discrete approximations b_i, e_i, and k_{ij}, $i, j = 1..N$, and the radiosity integral equation (eq. 1) to its discrete summation [2]

[1] The computation of one intensity value delivers monochrome results. To account for the whole spectrum, radiosity approaches commonly calculate three different color bands separately and combine the results to a colored picture.

[2] FEM and radiosity approaches commonly split the domain under consideration into a set of certain *base functions*. Thus, a subpatch with constant radiosity can be seen as *box* base function over the range of the subpatch. Regarding general bases, radiosity approaches can be considered from a more principal view, i.e., higher order bases [2] can be applied to reduce the number of coefficients, since they account for the color bleeding with its shape. Also hierarchical bases have successfully been used [3]. To keep this work comprehensive, we limit our considerations to just examining coefficients.

$$\forall j : b_j = e_j + \sum_{i=1}^{N} k_{ij} b_i. \tag{3}$$

Thus, the transfer function changes to separate transfer coefficients k_{ij} which are calculated by numerical integration (eq. 4) over the expansion of each pair of patches,

$$k_{ij} = \iint \mathcal{K}(\vec{x}, \vec{y})\, d\vec{x}\, d\vec{y}, \quad i, j = 1 \,..\, N. \tag{4}$$

To solve equation (3) a great variety of *relaxation* methods can be applied such as *Jacobi*, or *Gauss-Seidel* iteration. Detailed information can be found in [4, 5].

CHALLENGE

Finding an efficient FEM representation for radiosity is driven by the search for an adequate FEM *meshing*, i.e., the number, the location, and the size of the subpatches to be generated have to be determined. This is crucial, since the number of subpatches, N, commonly equals several thousands and leads to $O(N^2)$ coefficients k_{ij} (eq. 4). The amount of memory and computing resources to calculate and to memorize these coefficient is generally the limitation of almost all radiosity approaches. Thus, the primal aim should be to keep the number of coefficients k_{ij} as small as possible, in other words, to find an approximation of the operator $\mathcal{K}$ by discrete coefficients which is as little redundant as possible.
Generally, the *coherence* of $\mathcal{K}$ has to be detected and utilized for an approximation model which additionally is suitable to be transformed into a system like equation (3) for a numerical solution of equation (1).
Classical radiosity approaches mostly deal with the problem that the development of such an approximation is driven by regarding the geometrical definition of *separate* surfaces. A pair of surfaces defines a subspace of the domain of $\mathcal{K}$, and thus, a coherence analysis is limited locally to each pair of these separate subspaces. The reason for this narrow coherence analysis is the difficulty to find an analytical model which is capable of handling sets of separate geometrical defined surfaces.
Since common virtual geometries contain several thousands of surfaces, and thus, several millions of default partitions of $\mathcal{K}$, the coherence analysis through classical radiosity approaches turned out to be inefficient in that sense. Consider, for example, two walls defined each by a hundred of small patches. Although its effect concerning the light transfer is quite unique, the kernel approximation will consist of 10000 default coefficients.
The following algorithm transforms three-dimensional geometrical data into a representation suitable for a general coherence analysis.

ALGORITHM

Analyzing coherence commonly leads to the term *self-organizing artificial neural networks*. Due to its inherent learning capabilities which are superior in analyzing systems that can not, or hardly be modeled analytically, we apply a recent development in this area, the *incremental supervised growing cell structures* (ISGCS) [6]. It is derived from Fritzke's *supervised growing cell structures* (SGCS) [7] which can be seen as an alternative to the classical *Kohonen self-organizing map* (SOM) [8]. Both are able to adapt their internal structure to the distribution of input samples (*clustering*). SGCS are also suitable for being trained by supervision. They have been extended by the ISGCS approach which is additionally capable of automatically resampling the input domain according to the approximation accuracy and the clustering task [6], and of accounting for the coherence of a continuous goal function.
We approximate the light transfer operator $\mathcal{K}$ by an ISGCS trained by sample instantiations of $\mathcal{K}$ — single six-dimensional "rays" (start and endpoints $\vec{x}, \vec{y} \in \mathbb{R}^3$). The output of the network is supervised by the according function value of $\mathcal{K}$. During training, the domain of possible rays is

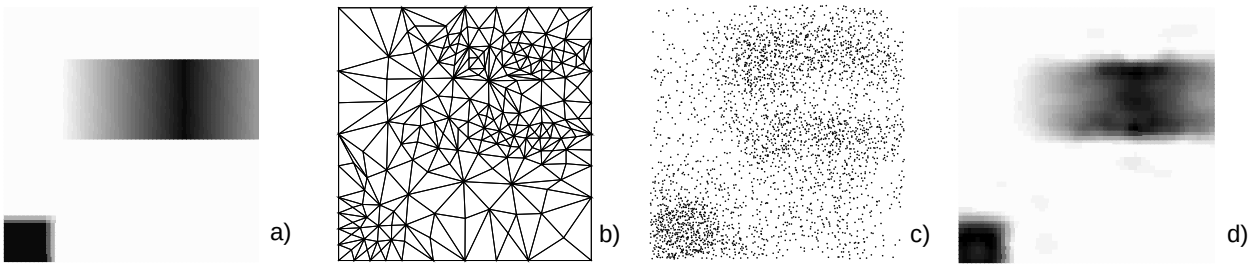

Fig. 2: A two-dimensional goal function (a), the approximation by a two-dimensional ISGCS (d), its topology (b) and sample distribution (c).

resampled to increase the sample distribution at locations of highly varying light transfer (*importance sampling*). After training, the ISGCS deliver an approximation model for $\mathcal{K}$. Due to the self-organization facilities of the ISGCS the efficiency of the resulting model is nearly optimal.
The internal network structure is a set of *radial basis functions* (RBF) which are created through a self-organizing evolutionary growth process. The RBFs, located at six-dimensional centers of ray clusters from the input domain, can be seen as *reference rays* and are interpreted as coefficients k_{ij} (eq. 4). [3] In the following we roughly explain the general features of growing cell structures.

Incremental supervised growing cell structures

An ISGCS network contains two layers. The first is instantiated by a set of n-dimensional radial basis functions called *cells*, the second accumulates each of the output *activations* of the RBFs to form the m-dimensional output vector of the network. It realizes a function $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^m$ which serves as an approximation of a goal function $f : \mathbb{R}^n \to \mathbb{R}^m$. At each state of the training process the network generalizes the function f over its input space to a certain accuracy.
An important feature of SGCS is the combination of supervised and unsupervised learning through different learning strategies for the first and the second layers. The network is trained by presenting input/output pairs $(\xi, \zeta) \in (\mathbb{R}^n \times \mathbb{R}^m)$. The unsupervised part is accomplished by moving the cells of the first layer according to the input ξ to find centers of clusters in the input data. Concurrently, the second layer is adapted to deliver the intended output ζ.
Moving the RBFs also accounts for a neighborhood relation between the single cells. It creates a topological structure of predefined dimensionality k on the training data. In the two-dimensional case, the final network of RBFs depicts a *map* of reference cells, similar to Kohonen's self-organizing feature map. Generally, the network consists of a mesh of predefined basic elements, such as triangles, quadrilaterals, tetrahedrons, etc., according to the underlying k.
Each cell of the network contains a *resource term* which holds particular information about the samples presented to the network. It delivers a local compound information about the training accuracy and sample distribution, and it is utilized to insert or to delete cells from the network. The ISGCS grows at locations with a high resource term and shrinks at places where the resource term signalizes a sufficient approximation accuracy.
In the ISGCS approach, the resource term is also used for a resampling strategy. Resampling should avoid selecting too many *initial* samples from the goal function f in order to circumvent undersampling at locations where the goal function's shape is highly varying. A high resource term shows a low local approximation accuracy, i.e., a locally insufficient representation by training samples (low coherence of the goal function). Resampling is done by randomly generating new input samples which are evaluated only if they lie in a high resource term place and added to the training sample set. For detailed explanations see [7, 9, 6, 10].
Figure 2 shows an example two-dimensional network trained with a two-dimensional goal function (fig. 2a). The network structure (fig. 2b) and sample distribution (fig. 2c) clearly accounts for the coherence of the goal function, and the network delivers the approximation in figure 2d.

[3] The interpretation as coefficients requires some further considerations neglected in this work. See [9] for details.

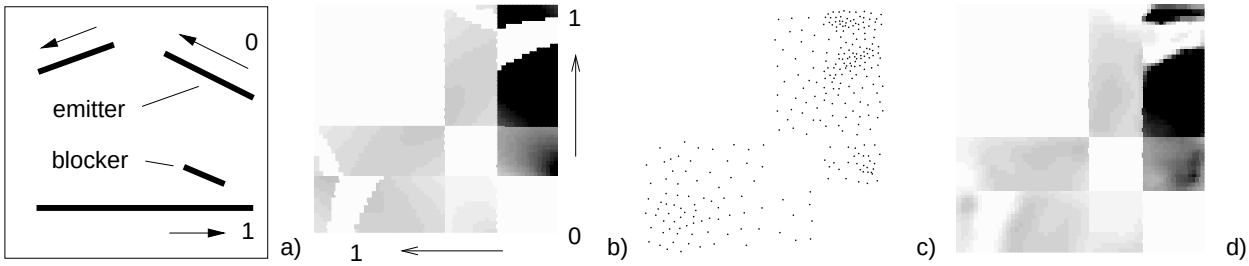

Fig. 3: (a) shows a two-dimensional exemplary geometry, figure (b) visualizes the light transfer goal function $\mathcal{K}$, (d) its approximation by the ISGCS and (c) the final distribution of RBFs.

RESULTS

Due to a convenient visualization, in the following we refer to two-dimensional geometries. Surfaces change to one-dimensional lines defined in a two-dimensional space. The geometrical term G (eq. 2) changes to $G(\vec{x},\vec{y}) = \cos\phi_{\vec{x}} \cos\phi_{\vec{y}}/(2\pi r_{\vec{x}\vec{y}})$. The network consists of four-dimensional reference RBFs which cluster four-dimensional rays between a pair of two-dimensional points.
Figure 3a shows a two-dimensional geometry defined by four lines. A visualization of the kernel operator $\mathcal{K}$ can be seen in figure 3b. Start and end points ($\vec{x},\vec{y} \in \mathbb{R}^2$) of one ray are drawn according to the one-dimensional axes. The position of a point is determined by a reparameterization on the one-dimensional line space of the geometry. In this parameterization, the edges of the scene are represented by consecutive sub-intervals of the interval $[0,1]$. A parameter value $x \in [0,1]$ uniquely identifies an edge and a point on it. A pair of points as required for the kernel is given by two parameter values s and t. Thus the kernel can be considered a real function over the unit square in the s-t-coordinate system. The assignment of the edges to intervals is arbitrary.
The kernel is weighted by the emitter's energy, shown through the darker area at the upper right of figure 3b. Here, one can also observe the bright fissure which comes from the effect of the blocker in figure 3a.
The approximation of the kernel goal function through the ISGCS can be seen in figure 3d. The distribution of RBFs is plotted in figure 3c. It is clearly shown that the distribution of RBFs is strongly related to the kernel coherence and thus efficient. The distribution of RBFs is generated independently of the real geometry, i.e., even in case of having several hundreds of surfaces defining the principle four lines from figure 3a — a very common case for modeling programs — the distribution of RBFs and the approximation accuracy would be equal. The approximation does not depend on the geometry definition due to the examination of the geometry by single rays. Coherence is detected without accounting for surface boundaries.
Figure 4a shows a result from the simulation of one reflection through the radiosity equation. In the foreground the analytically calculated kernel $\mathcal{K}$ is used, in the background the ISGCS approximation. For the visualization a two-dimensional geometry is spread to three dimensions by creating areas with a constant z-component from the two-dimensional lines. Blockers are marked by the grey frames in front of the surfaces and generate the shadow effects on the surfaces.
Figures 4b,c show a three-dimensional geometry, computed by a six-dimensional ISGCS. The light source is not displayed. The black frame stands for the blocker which causes the shadow on the wall. Figure 4b is calculated by the analytical kernel, figure 4c by the ISGCS approximation.

CONCLUSION

We have presented a new efficient representation of three-dimensional geometrical environments by an incremental supervised growing cell structures network.
The representation is designed to optimize the solution of the radiosity method which calculates the global illumination of virtual three-dimensional environments. Basically, radiosity is a finite element problem, and we explained how the proposed representation can be interpreted as the according finite element meshing. This enables the detection of global coherences of the light transfer, which

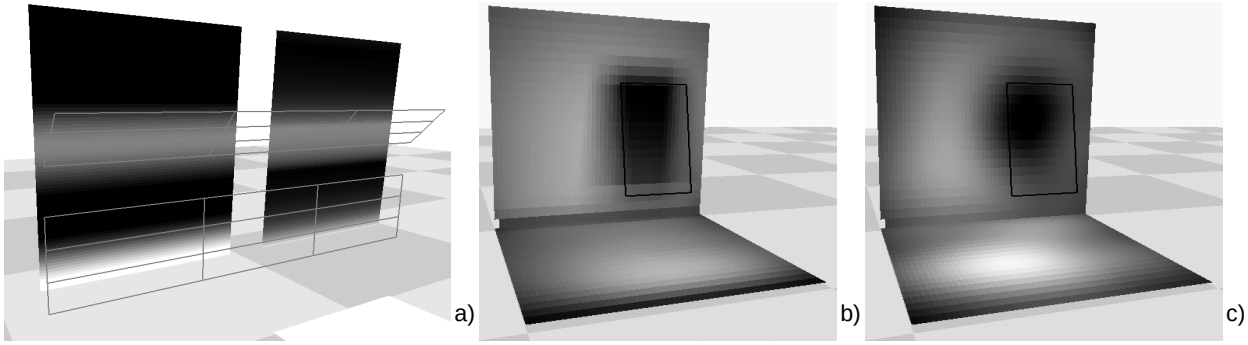

Fig. 4: The kernel approximation applied to calculate the light transfer. (a) shows a two-dimensional scene spread to three-dimensions (four-dimensional ISGCS input), in the foreground the analytically calculated shading, in the background the geometry calculated with the ISGCS approximation. (b) and (c) show the same for a three-dimensional scene (six-dimensional ISGCS input). Light sources are not displayed. Blockers are marked by a frame in the foreground.

is a problem of classical radiosity approaches since they only account for the coherence of the light transfer between single surfaces.
This approach leads to a very compact representation of the global light flow and enables a fast calculation of huge three-dimensional environments due to the finite element meshing through a "neural mesh".

REFERENCES

[1] James T. Kajiya. The rendering equation. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.

[2] Harold R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 213–220, 1993.

[3] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 221–230, 1993.

[4] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Professional, San Diego, CA, 1993.

[5] François Sillion and Claude Puech. *Radiosity and Global Illumination.* Morgan Kaufmann, San Francisco, 1994.

[6] Christian-A. Bohn. An incremental unsupervised learning scheme for function approximation. In *Proceedings of the International Conference on Neural Networks (ICNN'97)*, Houston, June 1997.

[7] Bernd Fritzke. Growing cell structures - a self-organizing network for unsupervised and supervised learning. Technical Report ICSI TR-93-026, International Computer Science Institute, Berkeley, CA, May 1993.

[8] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, pages 59–99, 1982.

[9] Christian-A. Bohn. Efficiently representing the radiosity kernel through learning. In Xavier Pueyo and Peter Schröder, editors, *Rendering Techniques '96*, pages 123–132, Wien, Austria, 1996. Springer-Verlag Wien.

[10] Christian-A. Bohn. Finite element mesh generation using growing cell structures networks. In *Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN '97)*, Stockholm, June 1997.