A Neural Network Based Finite Element Grid Generation Algorithm for the Simulation of Light Propagation

Christian-Arved Bohn [1]

Fraunhofer Institute for Media Communication [2]

Abstract

Calculating the transfer of light in three-dimensional virtual environments is an inevitable feature of modern computer graphics systems. Usually it is accomplished by a *finite element* (FE) method where the three-dimensional surfaces of the scene geometry are cut into several sub-surfaces — a *surface meshing* is generated.

We present a new way of creating this surface mesh: the internal structures of two neural networks trained by sample surface points and sample light rays are interpreted as such a mesh on which the finite element method is executed.

The presented approach avoids several drawbacks arising with classical methods. The "neural meshing" outperforms standard techniques in terms of memory requirements and accuracy. Additionally this work presents several novel ideas of interpreting a neural network skeleton in terms of a virtual three-dimensional geometry and as a representation of light energy propagating through a three-dimensional polygon-based scene.

Key words: Computer graphics, artificial neural networks, realistic image synthesis, rendering, radiosity, finite element method, growing cell structures, self-organizing mapping

1 Introduction

Realistic looking pictures of three-dimensional virtual environments are increasingly demanded in the field of computer graphics. It has been shown that computing such images commonly requires the simulation of a physically

[1] bohn@imk.fraunhofer.de, http://viswiz.de/
[2] http://www.imk.fraunhofer.de/

adequate model of the flow of light. Due to the complexity of such a model [10], the *radiosity* technique [4] has been proposed accounting for an excerpt of the overall phenomena of the physical light model. Radiosity simplifies the general scattering properties of surfaces to ideally diffuse (*Lambertian*) reflection and emission. Due to its convincing results [3], on the one hand, and due to the moderate amount of computing resources required, on the other hand, radiosity has found broad acceptance in fields like architecture, virtual reality, and film production.

1.1 The Radiosity Equation

Mathematically, radiosity is denoted by the *radiosity integral equation*

$$B(\mathbf{y}) = E(\mathbf{y}) + \rho(\mathbf{y}) \int_{\mathcal{S}} G(\mathbf{x}, \mathbf{y})\, B(\mathbf{x})\, d\mathbf{x} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are three-dimensional points lying in the two-dimensional surface space $\mathcal{S}$ and which are defined by a polygonal scene description. The radiosity value $B(\mathbf{y})$ at a point $\mathbf{y}$ represents the light *intensity*[4] determined by the weighted accumulation (integral in equation (1)) of the radiosity $B(\mathbf{x})$ emitted from all points $\mathbf{x}$ in the scene geometry. The *geometric term*

$$G(\mathbf{x}, \mathbf{y}) = \frac{\cos\phi_{\mathbf{x}} \cos\phi_{\mathbf{y}}}{\pi \left\|\mathbf{x} - \mathbf{y}\right\|^2} \cdot V(\mathbf{x}, \mathbf{y}) \tag{2}$$

describes the geometric properties of the light transport from a point $\mathbf{x}$ to another point $\mathbf{y}$. It depends on the distance between the points and on the angles which the normals at their locations with the connecting ray build (see figure 1 and [13] for a detailed derivation of equation (2)). V is called the *visibility term*. Its value depends on the mutual visibility of two points and equals zero if the direct view between them is occluded by a scene object, otherwise one. ρ is the *reflectance term* which determines the portion of light which is absorbed at a reflection at a point $\mathbf{y}$. ρ defines the color (in this case a grey tone) of the particular surface. E is the self-emittance at each point of the environment. It equals zero at surface points which are not light sources, otherwise it represents light emission and is greater than zero.

[3] see figure 2 for example

[4] For simplicity, B is assumed being a one-dimensional intensity value, which is sufficient for the description of monochrome light. The transition to colors is commonly accomplished by calculating three separate color bands and accumulating the results.

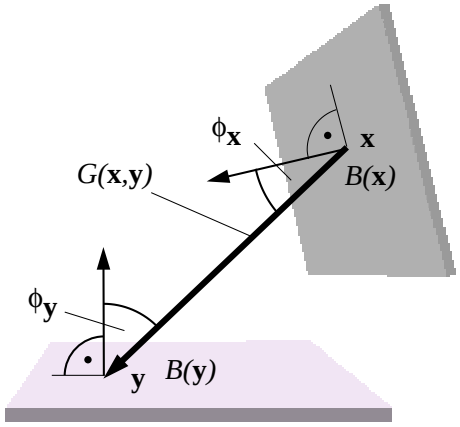


Fig. 1. The geometric term (eq. (2)) of the radiosity integral equation describing the diffuse light reflection behavior between two points in a virtual environment.

1.2 Finite Elements and the Radiosity Equation

B, E, ρ, and G in equation (1) are continuous functions defined on an infinite number of points describing the surface domain. In such cases, commonly, a *finite element (FE)* method is applied transforming the continuous functions into approximations by discrete linear function bases. Due to the usually local support of the base components this discretization is often termed *meshing*.

The discretization usually originates at B in equation (1), which is transformed into an approximation $B(\mathbf{x}) = \sum_{i=1}^{n} b_i N_i(\mathbf{x})$. Here N_i are the n components of the chosen function base, which are located on the surfaces of the scene geometry. If we also apply this scheme to E and ρ, then, equation (1) can be projected onto the function base $\mathbf{N} = (N_1, N_2 \dots N_n)$ (see [2] for an explicit description) resulting in a linear system

$$\mathbf{b} = \mathbf{e} + \mathbf{K}\mathbf{b}. \tag{3}$$

Here the vectors $\mathbf{b}$ and $\mathbf{e}$ contain the coefficients for each of the according base components and $\mathbf{K}$ is called the *discrete transport operator* consisting of single *transport coefficients*

$$k_{ij} = \int_{\mathcal{S}} N_i(\mathbf{y})\rho(\mathbf{y}) \int_{\mathcal{S}} G(\mathbf{x},\mathbf{y}) N_j(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y},\ i,j = 1 \dots n. \tag{4}$$

Thus, with equation (3) we developed a discrete form of equation (1). A *finite* set of coefficients $\mathbf{b}$ distribute their energy through a *finite* set of transfer

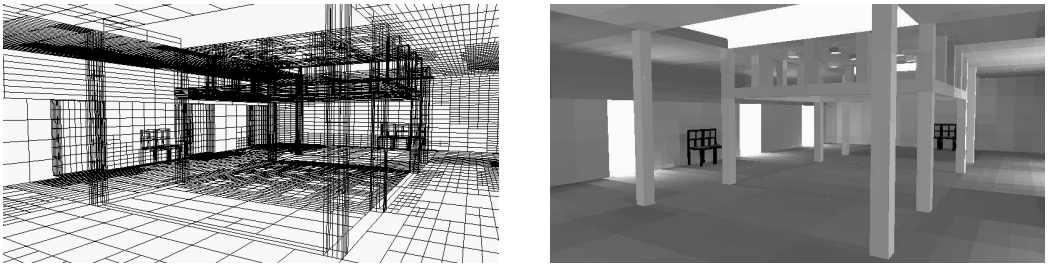


Fig. 2. On the left hand side, the subdivided surfaces of a virtual model of the *German Museum of Architecture* in Frankfurt/Main, on the right hand side, the resulting radiosity simulation of the light distribution.

coefficients $\mathbf{K}$. In contrast, in equation (1) an *infinite* number of intensity points B interact through an *infinite* number of rays K. Finally, equation (3) enables us to solve the radiosity equation numerically by common iteration methods for linear systems.

According to the formulation in equation (4) and for the following considerations we rewrite equation (1) like the general form of a *Fredholm integral equation of the second kind*

$$B(\mathbf{y}) = E(\mathbf{y}) + \int_{\mathcal{S}} K(\mathbf{x}, \mathbf{y})\, B(\mathbf{x})\, d\mathbf{x} \tag{5}$$

with the *kernel* K combining all geometric relationships from equation (1).

See figure 2, for example. Here, the scene is subdivided into smaller subpatches. Considering the last paragraphs, in this case the mentioned base components are constant box functions with support limited to each of the subpatch boundaries. A constant color value is assumed for each of them — the coefficients $\mathbf{b}$. This more intuitive view of avoiding the formal definition of base functions has been applied in many classical approaches.

1.3 Numerical Solution of the Radiosity Equation

The shape of equation (3) delivers a hint for a possible solution method — the calculation of the finite *Neumann series*

$$\mathbf{b} = (\mathbf{I} + \sum_{k=1}^{\infty} \mathbf{K}^{(k)})\, \mathbf{e} \tag{6}$$

where the components of $\mathbf{e}$ are initialized with the emission energy of the according surfaces projected on the function base $\mathbf{N}$. Intuitively, equation (6)

can be evaluated by unfolding it into a recursion scheme

$$\mathbf{b}^{(0)} = \mathbf{e},\ \mathbf{b}^{(k+1)} = \mathbf{e} + \mathbf{K}\mathbf{b}^{(k)},\ k = 1 \ldots \infty \tag{7}$$

which converges due to the *spectral radius* of $(\mathbf{I} - \mathbf{K})$ being less than one (see for example [4], pgs. 110-111). Every new $\mathbf{b}^{(k)}$ in equation (7) defines a further approximation of the radiosity and replaces the $\mathbf{b}$ on the right side for the following iterations. Each evaluation of equation (7) — each element of the Neumann series — can be seen as another reflection of the propagated light.

This type of solution by a *relaxation method* is most common in the field of radiosity since general matrix inversion techniques mostly fail due to the size of the linear system.

1.4 Motivation

Geometries coming from applications like, for example, virtual reality nowadays contain up to 50000 polygons. To represent the color shading on these surfaces adequately, they commonly are subdivided into hundreds of smaller patches each, and thus, the solution of the linear system of equation (3) easily goes beyond the scope of commonly available computing hardware.

> *The subdivision algorithm is vital point in solving the radiosity equation. The solution efficiency, and moreover, the viability of a computation strongly depend on the efficiency of the subdivision scheme.*

Subdivision efficiency means the same like in general approximation tasks, namely, accounting for coherence of the underlying goal function — smooth locations require less support by base components than locations of high variation (like sharply edged shadows in case of radiosity). The invention of subdivision algorithms for radiosity is mainly challenged by two issues:

a) **Default meshing.** It is hardly possible to blast the boundaries of the scene surfaces, i.e., to define radiosity base functions with a larger support than just the area of one single polygon of the scene definition. Thus, commonly the initial number of transfer coefficients k_{ij} (see equation (4)) equals the square of the number of polygons of the scene definition. This property is called *initial linking* and it arises independently from the subsequent subdivision scheme.
b) **Sequencing issue.** Whereas the unknown radiosity — the color shading on the surfaces — should be available for an adequate discretization of the scene, the radiosity can only be calculated if this discretization of the scene already is available.

With classical radiosity approaches the above topics are well-known. This work proposes a workaround like follows:

a) The direct reference to the scene definition, i.e., to single polygons will be avoided, and instead, a sampling based algorithm examines the functions to be approximated. An additional representation of the surface topology, which is not attached to the originating polygons is developed.
b) In order to develop an efficient approximation model the sequencing issue postulates to regard the *light flow* (the term KB (eq. (5))) as a whole and not the radiosity B separately, from which a surface discretization is derived.

In contrast to the latter item, classical radiosity approaches do not regard the light flow KB in the first place. Instead, they focus on the surface geometry by developing a radiosity base without accounting for the light flow which virtually is responsible for the particular meshing requirements.

The reason is self-evident: ideally accounting for KB is hardly possible if the geometric definition of surfaces must be regarded. The existence of polygons hinders the application of well-known efficient mathematical approximation methods for analyzing KB directly, and thus, it prohibits a completely free examination of the light transfer relationships.

> *Considering these issues, we propose a self-organizing algorithm which smoothly adapts to the problem under consideration without a predefined subdivision topology and without sticking to single surfaces. The algorithm is characterized by a fluently adapting set of neural networks, which automatically draws samples (points and rays) from the scene description and which grows to a final "neural mesh" on which the FE calculations are executed.*

Previous work. In *classical radiosity* (CR) [8] an arbitrary base on the surfaces is created "by hand" (subdivision into subpatches). Since the subdivision does not regard the final (not yet known) intensity bleeding, a common aid has been to estimate [5] the solution B and to develop the meshing accordingly. *Progressive radiosity* approaches (PR) [3] like they have been implemented in many commercially available software packages compute *direct shadows* [6] from which an initial discretization is derived. *Hierarchical radiosity* approaches (HR) [9] are the most promising attempts up to now. They start with a coarse discretization of the surface domain, and then, refine patches depending on the particular size of the according transfer coefficients.

[5] intuitively, with the knowledge of an experienced user
[6] shadows generated by the *direct* emission of the light sources

2 A Neural Network Model for Solving the Radiosity Equation

2.1 Incremental Supervised Growing Cell Structures

This work utilizes the *incremental supervised growing cell structures* (ISGCS) approach presented in [1] and based on [6] (see also [7]). It is regarded a specific sort of general artificial neural network which can efficiently be used for clustering and function approximation tasks.

In this work, one ISGCS serves as an approximation model of the radiosity kernel K (eq. (5)) and delivers the required light flow discretization which we have been asking for in the previous section. Additionally, a further ISGCS taken as discretization (meshing) of the geometry is utilized for the representation of the radiosity itself.

An ISGCS can be seen as an extended *Kohonen Self-Organizing Feature Map* (SOM) [11] which is well-known in the field of general *competitive learning* using artificial neural networks.

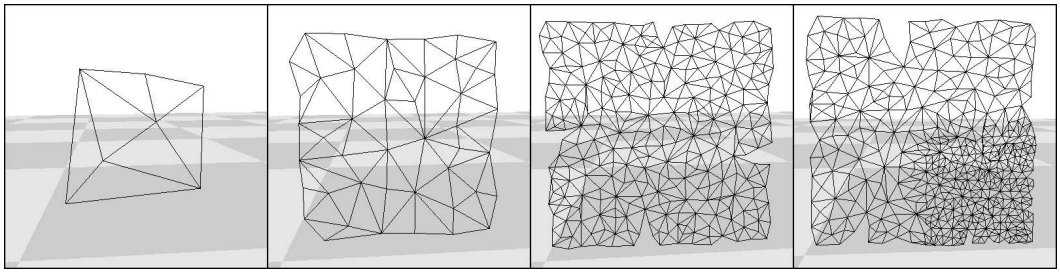
Competitive Learning — the basic principle of competitive learning renders an algorithm which adapts a set of n-dimensional *reference vectors*[7] to a significantly bigger set of n-dimensional training samples in a way that the distribution of the reference vectors *match* the sample distribution in the n-dimensional space. For example, a culmination of input samples at certain regions in the input space (*clusters*) can be noticed by a similar culmination of reference cells.

A plain iteration method accomplishes the following: samples of the training set are randomly presented to the set of randomly distributed reference cells. Each time the best matching reference cell (the *"best matching unit"* (BMU)) — the cell which is most similar to the input, i.e., which lies at the smallest Euclidian distance — is determined. Then, this cell is moved into the direction of the input according to a certain moving strength parameter.

The process is repeated several times while steadily decreasing the moving strength. Finally, the result is a set of reference units which, first, are accumulated at those locations where the training sample distribution is high, and second, each of which mimics a kind of "average vector" of its surrounding training samples.

Principally, competitive learning is applied in most iterative clustering ap-

[7] commonly called *reference units* in the field of neural networks or *reference cells* in the growing cells area



(a) (b) (c) (d)

Fig. 3. Pictures (a-c) show the growing of a two-dimensional ISGCS which is trained by three-dimensional points lying in 3D space. Image (d) is generated by training through a non-uniform sample distribution.

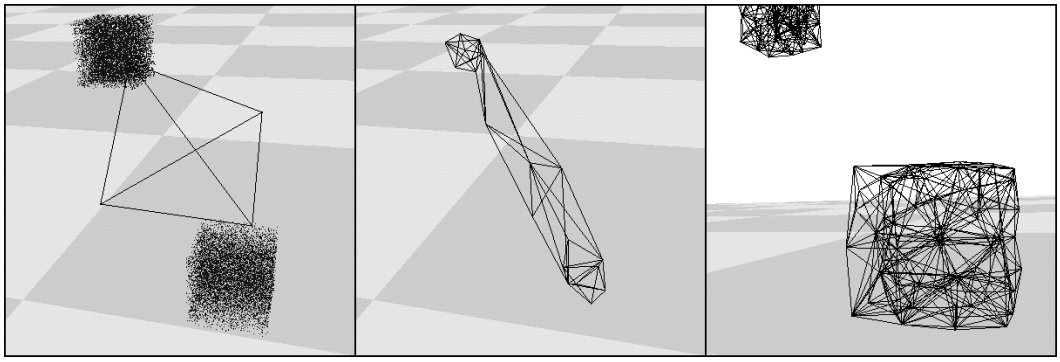
proaches, like for example *k-means clustering* or the *Linde-Buzo-Gray algorithm* [12]. For an overview of the fundamental literature see [5].

SOM — a SOM adds an additional topological constraint at the general competitive learning rule. Like in the latter, iterations start with a random distribution of reference cells, but in contrast, now the cells are connected by an additional k-dimensional topology — in case of $k = 2$ this topology forms a two-dimensional mesh. During adaption of the reference cells regarding the sample distribution, this mesh is also accounted for in a way that neighboring reference units are moved similarly but with a strength lower than that of the best matching cell. It leads to the rise of a map representing a kind of ordering of the reference cells, and herewith, an ordering of the training samples itself.

In figure 3, for example, a two-dimensional map is shown in (d), which adapts to a (not visible) sample distribution in 3D space. In the lower right quarter, the particular sample distribution is higher than in the rest of the displayed square. This property is accounted for by a higher granularity of the shown network.

ISGCS — general competitive learning strategies and specifically the SOM algorithm are based on a predefined set of initial cells which are to be adapted to a certain sample distribution. In contrast, the ISGCS does not consist of a predefined set, moreover, it starts with one minimal element and *grows* to a network of elements. This makes it more flexible concerning the underlying sample structure. To particularize further differences:

- The ISGCS topology is not fixed — it grows from an initial n-dimensional *simplex* to a network of simplices and it may shrink if the underlying data is represented redundantly. Figure 3 shows the growth of a typical ISGCS structure. Figure 4 exposes the training of a three-dimensional sample set distributed in a way describing two separate, cube-wise point clusters. It can be seen that the network deletes superfluous parts of its topology at



(a) (b) (c)

Fig. 4. Training (from (a) to (c)) of two separate 3D point clusters (drawn in image (a)). The network adapts to the sample distribution during training and the network cells vanish at places where the sample distribution is sufficiently low.

those locations where samples do not exist.

- The ISGCS is capable of *resampling* the goal function according to a certain learning criterion. The resampling feature can be observed in the results section of this work.
- The ISGCS training is a combination of supervised and unsupervised learning — the sample distribution *and* the goal function approximation accuracy determines the network growth and the above mentioned resampling process.

After training, the resulting model is a grown set of units with lateral connections forming a network of simplices of a certain dimension. For example, in case of a two-dimensional topology, simplices are triangles.

Each of the network vertices carries a *Gaussian radial basis function* (Gaussian RBF) which altogether define a linear function base approximating the training goal function. An example for the approximation capability is shown in figure 11.

For the realization of the ISGCS method, additionally a counter is attached to each of the ISGCS cells, which stores the number of selections of the particular cell as a best matching unit together with the approximation error at that cell compared to the training goal function. Regarding this counter — called *resource term* — new elements are introduced into the mesh in a way that the basic simplex structure holds. For a detailed description see [1].

2.2 Radiosity on Growing Cell Structures

See figure 5 for an overview of the basic scheme of this work. On the right hand side (pictures A and B), an example geometry is shown — a "ceiling patch"

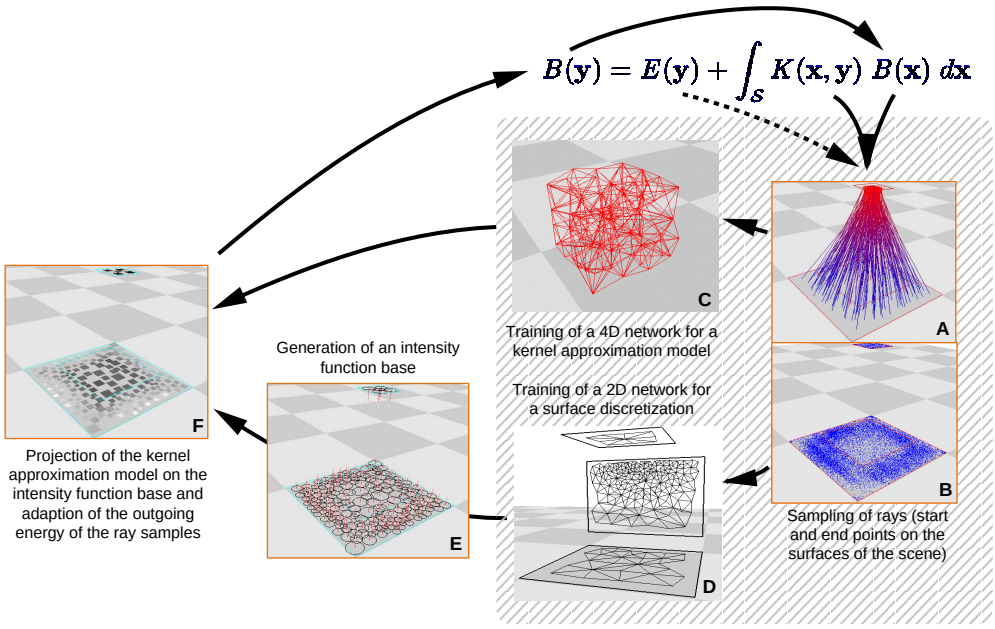


Fig. 5. Overview of the complete approach.

as light source which emits light on a "floor patch". Our approach trains two neural networks, the first one is charged with learning sample rays (picture A) of the light flow, the second one iterates on the rays' end points (picture B) to create a neural surface topology. Samples are generated randomly at the start of the algorithm. During the iterations, new rays are selected by the ISGCS' resampling feature according to the model's approximation accuracy. The center of the algorithm is a permanently growing set of samples.

Pictures C and D expose the resulting networks, the *kernel network* and the *shading network*, respectively. The shading network approximates the geometry through its training with three-dimensional point samples, the kernel network approximates the light flow by examining emission-weighted, six-dimensional rays.

Since the surface domain is two-dimensional, the shading network is a two-dimensional ISGCS. The kernel network approximates the space of all possible rays between two points on a two-dimensional surface domain. Thus, we train a four-dimensional kernel network. Nevertheless, for visualization purposes in figure 5, we draw a three-dimensional ISGCS as kernel network.

The internal representations of the two networks deliver the required discretization of the radiosity integral equation. The adaptive discretization process of the kernel and the surface network is interconnected and thus guarantees the continuous adaption of the surface meshing regarding the computed light phenomena. After a certain number of training steps, a Gaussian linear function base is formed on the shading network topology (picture E) and the light energy is propagated through the kernel network structure by calculating

the integrals from equation (4) on the networks of Gaussians.

Thereafter, the solution of one reflection of light is calculated through the surface base model (see picture F) and the result is used to adapt the emitted energy $B(\mathbf{x})$ and to adjust the training sample set. This back-coupling of the results of one integration to adjust the sample set relates to the classical iterative solution of equation (3) by calculating further elements of the finite Neumann series (eq. (6)). Without the adjustment of the training sample set, only the first element of the series would be the result.

> *Propagating the energy whereas constantly adapting the surface meshing, mainly focusing on an approximation model of the light flow and not only the surface function, and using an efficient neural network approximation model are the main ideas of this approach and result in the efficiency of the proposed solution method.*

2.3 Radiosity Light Transfer through Growing Cell Structures

The following section gives an impression of how the radiosity equation can be translated to neural network structures. The detailed algorithm can be derived from these steps or be read in the original work [2].

Consider the ISGCS representation of the radiosity kernel through a linear function base $\mathbf{\Psi} = \sum_{m=1}^{z} v_m \, \Omega_m(\mathbf{x}, \mathbf{y})$ of z six-dimensional Gaussian RBFs with Gaussian base components Ω_m and their coefficients v_m determined through the ISGCS training. $\mathbf{x}$ and $\mathbf{y}$ are two points in the scene geometry. The training set consists of randomly drawn sample rays of the geometry, which are weighted by the energy emitted from the surface at the rays' starting points. The radiosity equation is converted to

$$B(\mathbf{y}) = E(\mathbf{y}) + \int_{\mathcal{S}} \mathbf{\Psi}(\mathbf{x}, \mathbf{y}) \, d\mathbf{x}$$

and by projection to the radiosity base $\mathbf{N}$ it becomes a linear system like

$$\mathbf{b} = \mathbf{e} + \left[\left\langle \mathbf{N}(\mathbf{y}) \middle| \int_{\mathcal{S}} \mathbf{\Psi}(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \right\rangle \right]$$

with single lines

$$b_i = e_i + \int_{\mathcal{S}} N_i(\mathbf{y}) \int_{\mathcal{S}} \mathbf{\Psi}(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \, d\mathbf{y}, \; i = 1 \ldots n.$$

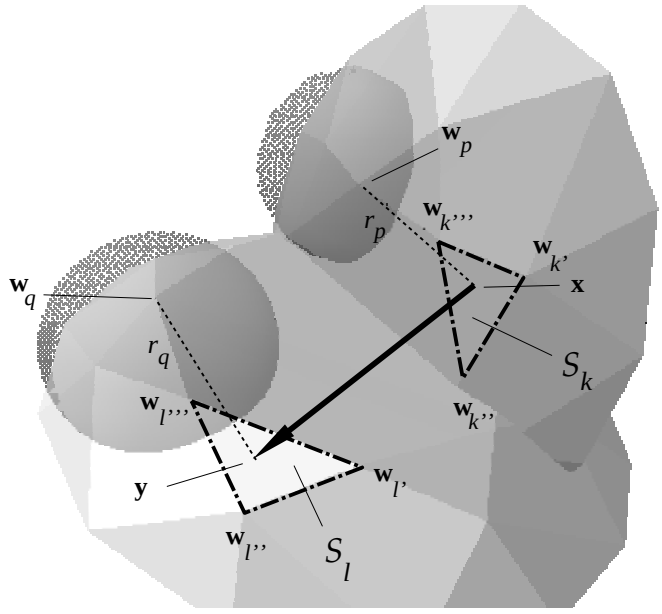


Fig. 6. Parameterization of the integration points $\mathbf{x}$ and $\mathbf{y}$ on two triangles (simplices) of the shading network to calculate the transfer coefficient of two Gaussians, $\Lambda_i(r_p), \Lambda_j(r_q)$, centered at $\mathbf{w}_p$ and $\mathbf{w}_q$.

The shading network is trained concurrently by sample ray end points and delivers the n Gaussian base components with coefficients

$$b_i = e_i + \int_{\mathcal{S}} N_i(\mathbf{y}) \int_{\mathcal{S}} \mathbf{\Psi}(\mathbf{x}, \mathbf{y}) \sum_{j=1}^{n} b_j N_j \; d\mathbf{x} \; d\mathbf{y}, \;\; i = 1 \ldots n.$$

This leads to "transfer coefficients"

$$k_{pqo} = \iint_{\mathcal{S}} \Lambda_p(\mathbf{x}) \; \Lambda_q(\mathbf{y}) \; \Omega_o(\mathbf{x}, \mathbf{y}) \; d\mathbf{x} \; d\mathbf{y}, \;\; p, q, o = 1 \ldots n \tag{8}$$

where Λ_p and Λ_q are Gaussian RBFs of the shading network and Ω_o a Gaussian component of the kernel network. After orthonormalization of the surface base we extract the last step on the way to the final solution by considering two components Λ_p and Λ_q centered at the corresponding cell centers $\mathbf{w}_p$ and $\mathbf{w}_q$. The integration is accomplished over all pairs of triangles of the approximate geometry defined by the shading network (see figure 6 as an example geometry) leading to

$$k_{pqo} = \iint_{\mathcal{S}} \Lambda_p \; \Lambda_q \; \Omega_o \; d\mathbf{x} \; d\mathbf{y} = \sum_{1 \le k,l \le M} \int_{\mathcal{S}_k} \int_{\mathcal{S}_l} \Lambda_p \; \Lambda_q \; \Omega_o \; d\mathbf{x} \; d\mathbf{y} \tag{9}$$

where M is the number of triangles of the shading network. One such pair is denoted by subscripts k and l in figure 6 and bounded by the vertices $\mathbf{w}_{k'}$,

$\mathbf{w}_{k''}$, $\mathbf{w}_{k'''}$, and $\mathbf{w}_{l'}$, $\mathbf{w}_{l''}$, $\mathbf{w}_{l'''}$, respectively. For integration, we assume two points $\mathbf{x}$ on $\mathcal{S}_k$ and $\mathbf{y}$ on $\mathcal{S}_l$ and define a two-dimensional parameterization $\mathbf{x}, \mathbf{y} : \mathbb{R}^2 \to \mathbb{R}^3$ of both triangles,

$$\begin{aligned} \mathbf{x}(s,t) &= (1-s-t)\,\mathbf{w}_{k'} + s\,\mathbf{w}_{k''} + t\,\mathbf{w}_{k'''}, \quad \text{with} \quad s+t \leq 1, \\ \mathbf{y}(u,v) &= (1-u-v)\,\mathbf{w}_{l'} + u\,\mathbf{w}_{l''} + v\,\mathbf{w}_{l'''}, \quad \text{with} \quad u+v \leq 1. \end{aligned}$$

The distances $r_p, r_q : \mathbb{R}^2 \to \mathbb{R}$ of $\mathbf{x}$ and $\mathbf{y}$ from the centers $\mathbf{w}_p$ and $\mathbf{w}_q$, respectively, can be written as

$$r_p(s,t) = \|\mathbf{w}_p - \mathbf{x}(s,t)\|, \quad \text{and} \quad r_q(u,v) = \|\mathbf{w}_q - \mathbf{y}(u,v)\|, \tag{10}$$

and the distance $r_o : \mathbb{R}^4 \to \mathbb{R}$ of the concatenation $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^6$ from the center $\mathbf{w}_o \in \mathbb{R}^6$ of the kernel Gaussian Ω_o is

$$r_o(s,t,u,v) = \sqrt{r_p^2(s,t) + r_q^2(u,v)}. \tag{11}$$

Replacing equations (10) and (11) at equation (9) leads to the integral

$$\begin{aligned} k_{pqo} = &\sum_{1 \leq k,l \leq n} \int_{\mathcal{S}_k} \int_{\mathcal{S}_l} \Lambda_p\, \Lambda_q\, \Omega_o\, d\mathbf{x}\, d\mathbf{y} = \\ &\sum_{1 \leq k,l \leq n} \int_0^1 \int_0^{1-v} \int_0^1 \int_0^{1-t} e^{-{d_p}^{-2} r_p^2}\, e^{-{d_q}^{-2} r_q^2}\, e^{-{d_o}^{-2} r_o^2} \\ &\qquad \left\| \frac{\partial \mathbf{x}}{\partial s} \times \frac{\partial \mathbf{x}}{\partial t} \right\| \left\| \frac{\partial \mathbf{y}}{\partial u} \times \frac{\partial \mathbf{y}}{\partial v} \right\| \, ds\, dt\, du\, dv \end{aligned}$$

where the constant $J_{pqokl} = \left\| \frac{\partial \mathbf{x}}{\partial s} \times \frac{\partial \mathbf{x}}{\partial t} \right\| \left\| \frac{\partial \mathbf{y}}{\partial u} \times \frac{\partial \mathbf{y}}{\partial v} \right\|$ can be drawn out of the integral, resulting in a term of the form

$$\boxed{k_{pqo} = \sum_{1 \leq k,l \leq n} J_{pqokl} \int_0^1 \int_0^{1-v} \int_0^1 \int_0^{1-t} e^{P_2(s,t,u,v)}\, ds\, dt\, du\, dv} \tag{12}$$

with a polynomial $P_2(s,t,u,v) = A\,s^2 + B\,t^2 + C\,u^2 + D\,v^2 + E\,s + F\,t + G\,u + H\,v + I\,st + J\,uv + K$.

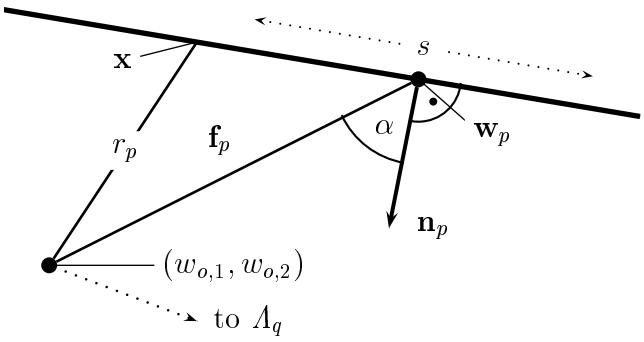


Fig. 7. The Flatland geometry for integration. Only the upper part of a kernel reference ray is shown.

2.4 Analytical Derivation of the Light Transfer

Equation (12) could be calculated numerically, similar to almost all classical approaches. Nevertheless, in this work we propose a symbolic solution by utilizing the particular integration properties of Gaussians *if they are integrated to infinity*.

For that we need to account for two constraints concerning the structure of the shading network, i.e., the structure of the geometry defined by the shading network. First, the shading network has to be "sufficiently flat" such that we get a unique integration domain at the according Gaussians. In fact, we easily can make this assumption for practical cases like proven in [2]. Second, Gaussians which participate in the symbolic, infinite integration must not be located at boundaries of surfaces, since in that case, integration must be stopped at the boundaries. Thus, we integrate the border Gaussians numerically and the rest which is the majority symbolically. This solution does not change the overall complexity of the algorithm, and practical tests have proven not to increase the computation time significantly.

The assumption of an infinite integration domain leads to the following analytical derivation. See figure 7, we assume Λ_p and Λ_q being one-dimensional functions. The geometry consists of lines instead of surfaces. The kernel is a two-dimensional function, since rays begin and end at points on lines. This way of thinking is classically termed "Radiosity in Flatland" and has been proven to be very useful in the development of radiosity methods. Herewith, first the lower dimensional, less complex Flatland solution is created, which then is transposed to three dimensions. In the following we utilize this kind of transfer also.

Consider parameters s and t are the distance arguments of the functions Λ_p

and Λ_q, respectively. The parameter r of the particular kernel Gaussian is derived from $\mathbf{x}$ and $\mathbf{y}$ through s and t and from the reformulation of the distance r of the kernel RBF center $\mathbf{w}_o$

$$r = r_p^2 + r_q^2 \quad \text{with} \quad r_{\{p,q\}}^2 = f_{\{p,q\}}^2 + \{s,t\}^2 - 2\{s,t\} f_{\{p,q\}} \cdot \sin \alpha_{\{p,q\}}.$$

The $\alpha_{\{p,q\}}$ are the angles between "surface" normals $\mathbf{n}_{\{p,q\}}$ and the vectors $\mathbf{f}_p = \mathbf{w}_p - (w_{o,1}, w_{o,2})$ and $\mathbf{f}_q = \mathbf{w}_q - (w_{o,3}, w_{o,4})$ with $f_{\{p,q\}} = \|\mathbf{f}_{\{p,q\}}\|$, and $\mathbf{f}_p$ and $\mathbf{f}_q$ are vectors defined by the centers of the surface RBF and the top and bottom of the kernel RBF center (ray), respectively.

Inserting the radii from above as parameters of Λ_p, Λ_q, and Ω_o in the two-dimensional form of equation (9) delivers

$$k_{pqo} = \iint_{\mathbb{R}} e^{P_2(s,t)} \, ds \, dt \tag{13}$$

with a polynomial $P_2(s,t) = -A_p \, s^2 - A_q \, t^2 + B_p \, s + B_q \, t - C$, and the substitutions $A_{\{p,q\}} = {d_o}^{-2} + {d_{\{p,q\}}}^{-2}$, $B_{\{p,q\}} = {d_o}^{-2} \cdot f_{\{p,q\}} \cdot \sin \alpha_{\{p,q\}}$, and $C = {d_o}^{-2} \cdot (f_p^2 + f_q^2)$.

With $\int_{\mathbb{R}} \exp\left(-As^2 + Bs - C\right) ds = \frac{\exp\left(\frac{B^2}{4A} - C\right)\sqrt{\pi}}{\sqrt{A}}$, equation (13) can be rewritten as

$$k_{pqo} = \frac{\exp\left(\frac{B_p^2}{4A_p} + \frac{B_q^2}{4A_q} - C\right) \pi}{\sqrt{A_p}\sqrt{A_q}}. \tag{14}$$

Switching back to three-dimensional geometries is identical to integrating twice over s and t with different parameters but the same coefficients $A_{\{p,q\}}$, $B_{\{p,q\}}$, and C. Thus, the non-constant arguments of the exponential function in the solution (eq. (14)) are doubled and the coefficients squared, leading to the following analytical solution.

$$\boxed{k_{pqo} = \iint_{\mathbb{R}^2} e^{P_2(s,t)} \, ds \, dt = \frac{\exp\left(\frac{B_p^2}{2A_p} + \frac{B_q^2}{2A_q} - C\right) \pi^2}{A_p A_q}} \tag{15}$$

3 Results

Figure 8 shows a simple example geometry. It is built from a floor surface, a wall, and a ceiling, the latter serves as light source. On the left hand side,

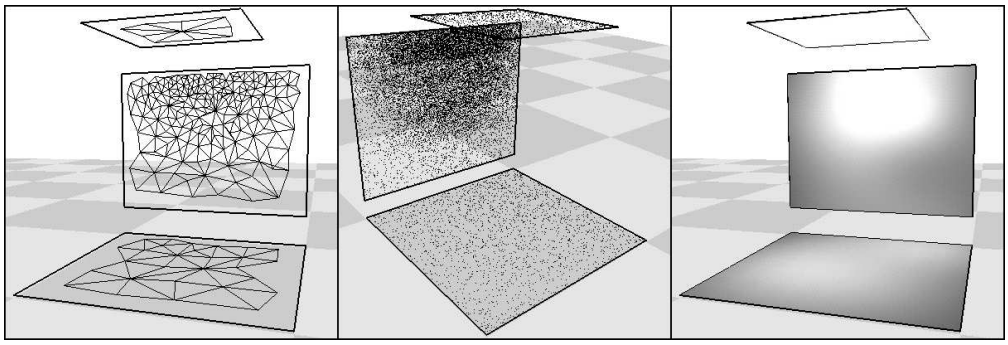


Fig. 8. A simple example scene. From left to right, the created surface network, the sample distribution, and the final radiosity result are shown.

the created shading network is shown which is used as the FE meshing. In the middle, the generated sample set is exposed. It can be observed that the sample distribution accounts for the final color shading on the surfaces. The latter is displayed on the right hand side of figure 8.

In figure 9 two blockers are inserted in the light flow from the ceiling, which generate sharp shadow edges. On the left hand side, the initial scene geometry is exposed, in the middle, the shading network which is created during the training phase, and on the right hand side, the final intensity distribution generated by the ceiling light source and the two light blockers.

The diagrams in figure 10 show error plots of the kernel approximation accuracy of the presented approach, *growing cells radiosity* (GCR), for the geometries of figures 8 and 9. The error is compared to a classical method — hierarchical radiosity. The horizontal axes display the granularity of the two approximation models — they expose the number of function base components and it can be observed that the error is significantly lower than that of the classical HR method.

Figure 11 compares the two kernel approximation models. On the left hand side (pictures (a) and (b)), results of the hierarchical radiosity approach, on

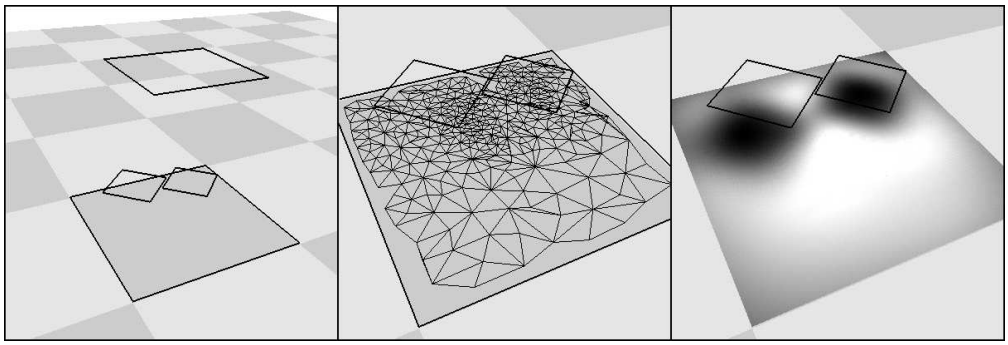


Fig. 9. A simple example scene with two light blockers creating sharp shadows. From left to right, the scene geometry, the network, and the computed result.

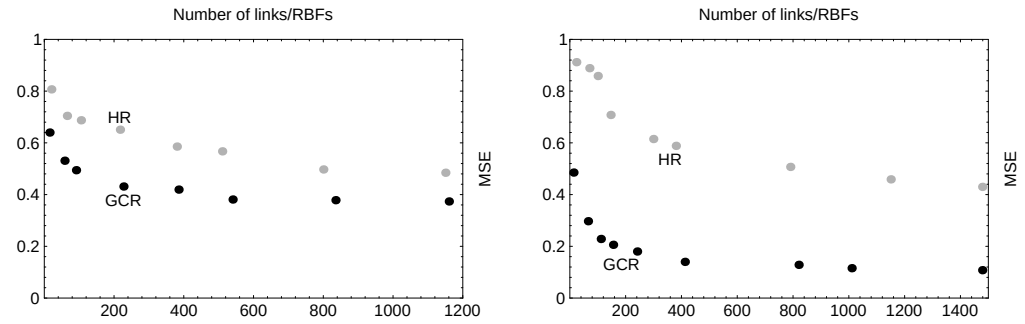


Fig. 10. Approximation accuracy of the kernel model of the two geometries from figures 8 and 9 — this work (GCR) if compared to the hierarchical radiosity approach (HR). The horizontal axes show the granularity of the approximation models, i.e., the number of base function components.

the right hand side (pictures (c) and (d)), results of the GCR approach of this work is exposed. Pictures (a) and (c) show the different discretizations, pictures (b) and (d) the kernel approximation characteristics.

4 Conclusion

We present an approach for calculating the flow of light in virtual three-dimensional geometries by using a scheme of two self-organizing neural networks. It outperforms classical methods in several ways and presents a bunch of novel, uncommon ideas outlined as follows.

- A neural network is interpreted as a virtual, geometric scene description. Vice versa, a finite element method is realized using the internal representations of two neural networks as an FE mesh.
- The light flow is analyzed directly, which has not been accomplished in

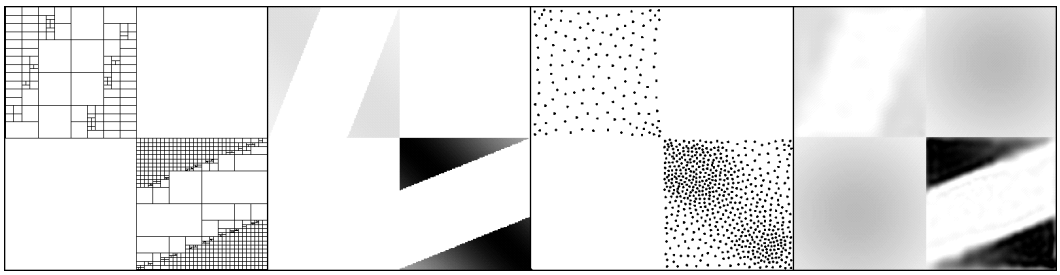


Fig. 11. Images (a) and (b) show the distribution of base function components and the kernel approximation for the classical HR radiosity approach for an example Flatland geometry. Images (c) and (d) expose the same approximation task by a ISGCS — the points in picture (c) are centers of the Gaussian radial base functions.

earlier radiosity approaches and which leads to an outstanding efficient representation of the approximation model.
- The presented work is completely independent from the description of the scene geometry — the finite element mesh is not constrained by the polygon geometry which is a vital disadvantage of classical radiosity approaches.

In the author's opinion, the main attraction of the presented method is its self-organizing nature — by iteratively adjusting a set of neural networks, an FE mesh is generated capable of being used to solve this work's particular FE task. Meshing which commonly is the most difficult task in similar approaches has completely been left to neural networks, and thus, the approach is able to profit from the typical outstanding neural network facilities of analyzing arbitrary data distributions. It results in an efficient meshing and in significantly lower error rates if compared to classical methods.

References

[1] Christian-A. Bohn. An incremental unsupervised learning scheme for function approximation. In *Proceedings of the 1997 IEEE International Conference on Neural Networks*, volume 3, pages 1792–1797, Piscataway, NJ, June 1997.

[2] Christian-A. Bohn. *Radiosity on Evolving Networks.* Infix, Sankt Augustin, Germany, April 2000. Dissertationen zur Computergraphik, ISBN 3-89838-603-1.

[3] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.

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

[5] B. S. Duran and P. L. Odell. *Cluster Analysis (A Survey)*, volume 100 of *Lecture Notes in Economics and Mathematical Systems.* Springer, Berlin/New York, 1974.

[6] 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. http://pikas.inf.tu-dresden.de/~fritzke/papers/fritzke.tr93-26.ps.gz.

[7] Bernd Fritzke. Incremental learning of local linear mappings. In *Proceedings of the ICANN-95*, Paris, France, 1995.

[8] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 212–22, July 1984.

[9] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.

[10] 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.

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

[12] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communications*, COM-28(1):84–95, January 1980.

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