



**HAL**  
open science

# Spectral domain decomposition method for natural lighting and medieval glass rendering

G. Gbikpi-Benissan, Remi Cerise, Patrick Callet, F. Magoules

► **To cite this version:**

G. Gbikpi-Benissan, Remi Cerise, Patrick Callet, F. Magoules. Spectral domain decomposition method for natural lighting and medieval glass rendering. 16th International Conference on High Performance and Communications (HPCC 2014), Aug 2014, Paris, France. 10.1109/hpcc.2014.17 . hal-01273946

**HAL Id: hal-01273946**

**<https://hal.science/hal-01273946v1>**

Submitted on 20 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Spectral Domain Decomposition Method for Natural Lighting and Medieval Glass Rendering

Guillaume Gbikpi-Benissan\* Rémi Cerise\* Patrick Callet\* Frédéric Magoules\*

## Abstract

In this paper, we use an original ray-tracing domain decomposition method to address image rendering of naturally lighted scenes. This new method allows to particularly analyze rendering problems on parallel architectures, in the case of interactions between light-rays and glass material. Numerical experiments, for medieval glass rendering within the church of the Royaumont abbey, illustrate the performance of the proposed ray-tracing domain decomposition method (DDM) on multi-cores and multi-processors architectures. On one hand, applying domain decomposition techniques increases speedups obtained by parallelizing the computation. On the other hand, for a fixed number of parallel processes, we notice that speedups increase as the number of sub-domains do.

**Keywords:** Domain decomposition; image computing; physically-based rendering; optical constants; cultural heritage

## 1 Introduction

This work is about a study of ancient stained glass windows as they might be used in the Cistercian art in France in the middle of the XIIIth century. Using a model of the abbey church of Royaumont, optical simulations are done with Virtuelium, a physically-based rendering open source software developed at Ecole Centrale Paris (France). More specifically, our interest is to produce consistent visualization based on some hypotheses about the visual appearance of Cistercian stained glass windows as they could have been seen in the church around 1250 AD. We use several samples of medieval glasses for this study, many of them coming from the abbey of Maubuisson, which is very close to Royaumont in terms of geographic location and time. As well, the methodology required more recent glasses, although manufactured in a very traditional way, like the complete set of Saint-Just glasses (Saint-Gobain group) we also used.

Interactions between natural lighting and medieval glass materials are hard to render in their whole complexity because of their heterogeneity in terms of optical behavior. For instance, measurements on glass tiles revealed a non uniform thickness and the presence of micro particles of air inside the tiles. That is responsible of an irregular volume scattering of light. Traditional physically-based rendering engines [40, 21, 43] use extrinsic properties of materials, such as spectral reflectances and transmittances. These approaches are not complete enough to efficiently address our problem. An originality of the Virtuelium approach consists in coupling both extrinsic and intrinsic properties (optical constants).

The specific methodology of acquisition on glass samples is thus described in the first section. Section 2 presents some rendering algorithms including those used in Virtuelium, then we give details about how these algorithms are computed in parallel. Finally, a presentation of resulted images and speedups will close our demonstration.

---

\*Ecole Centrale Paris, France (correspondence, frederic.magoules@hotmail.com).

## 2 Data acquisition

Extrinsic and intrinsic properties of glass materials can be measured with two distinct methods. Spectrophotometry [2] is commonly used to acquire spectral responses in reflection or transmission. This approach mainly consists in lighting the surface to be measured with known spectral (the emission spectrum) and geometrical (the incident angle) conditions. By analyzing the energetic quantities measured along a wavelength range, we can deduce the spectral functions for reflectance or transmittance. Then, the material is fully characterized in one point of its surface by repeated single measurements for regular sampling of both incident and view angles. In the case of glass materials, the obtained function is called Bidirectional Transmittance Distribution Function (BTDF) and it belongs to a larger family of spectral distribution functions.

Another method based on spectroscopic ellipsometry can be used to determine a BTDF. This second method offers a way to measure the complex index of refraction

$$\tilde{n}(\lambda) = n(\lambda) + ik(\lambda) = n(\lambda)(1 + i\kappa(\lambda)) \quad (1)$$

which is called "optical constants" [39, 5]. The quantities  $n(\lambda)$  denotes the optical index,  $k(\lambda)$  stands for the index of absorption and  $\lambda$  represents the wavelength. At the opposite of extrinsic properties, optical constants really define the electronic behavior of dielectric materials, and not just a spectral response. Besides, they are often used to simulate the visual appearance of metallic surfaces [3, 48, 15] as these materials fully satisfy the Fresnel conditions (non-scattering and homogeneous). But that is not true for glass materials and particularly medieval glass materials.

In order to deal with the heterogeneity of medieval glass materials, input data are measured from a piece of a height-hundred-years-old stained glass window coming from the Maubuisson abbey. On another hand, a library of modern samples (from Saint Just Corporation) with correct Fresnel conditions are also used. Then, by respecting a rigorous colorimetric protocol, we select appropriate modern glass materials which are visually close to the ancient one. More details can be found in [7].

## 3 Rendering algorithms

To simplify next discussions, we will only consider equations for the Bidirectional Reflectance Distribution Function (BRDF). The reasoning for the BTDF is the same except that we do not only consider a dome for incident lights but the entire sphere since the studied surface is non-opaque. The global rendering equation is deduced from the Radiative Transfer Equation (RTE) and described by Kajiya [18] as follow:

$$L_r(\vec{\omega}_o) = \int_{\Omega} F_r(\vec{\omega}_i, \vec{\omega}_o) L_i(\vec{\omega}_i) \vec{n} \cdot \vec{\omega}_i d\omega_i \quad (2)$$

where  $\Omega$  is the dome of incident lights. Literally, a radiative balance is calculated. The remitted light  $L_r$  in a direction  $\vec{\omega}_o$  depends on every incident light  $L_i$  emitted by the dome. It is then sufficient to know the BRDF,  $F_r$ , for computing the remitted light from a given incident light. In an applicative way, the equation (2) can be simplified as follow if we consider only point or directional light-sources:

$$L_r(\vec{\omega}_o) = \sum_{s1}^N F_r(\vec{\omega}_s, \vec{\omega}_o) L_s(\vec{\omega}_s) \vec{n} \cdot \vec{\omega}_s \quad (3)$$

where  $N$  is the number of light-sources,  $L_s$ , the emission spectrum of the current light-source  $s$  and  $\vec{\omega}_s$  the incident direction.

At the scope of a whole 3D scene, image rendering engines observe two major kinds of interaction between lights and geometric objects. We talk about local illumination when the

only lights hitting an object are those directly coming from the light sources without any other interaction. At the opposite, the mutual contributions of all the objects present in the 3D scene is then responsible of global illumination. As a result, rendering algorithms can be sorted between those which compute global illumination and those which ignore it. For now, Virtuelium provides one algorithm of each kind: the "Scanline rendering" technique for local illumination and the "Photon Mapping" for global illumination.

### 3.1 Without global illumination

The "Scanline rendering" [49] is based on the idea of inverse ray tracing algorithm [1]. Indeed it is easier not to traverse light paths in the logic direction, but from the camera to light-sources. As the image to be computed can be viewed as a matrix of pixels, a light-ray is emitted from each pixel, orthogonally to the image plane. Once emitted, each light-ray evolves independently. When a ray intersects with the closest object on his road, two actions can be executed. In a first time, we have to evaluate the received luminance at the given viewed direction. In order to achieve this goal, new rays are shot from the hit point to each light-source, thus determining all the needed incident directions. Then, secondary rays are thrown regarding to reflection and/or refraction laws and the process is repeated. The algorithm stops when the energetic value attached to the ray goes bellow a threshold or after the ray has bounced a predetermined number of times. Additional information retrieved from a hit point also allows more complex computations. By example, this can be texture coordinates for spatial distribution maps.

The complexity of ray tracers are mostly contained in the complexity of intersectors. That is why the main difference between algorithms of this family resides in the way polygons of objects are sorted. Sort in "Scanline rendering" is achieved by projecting every polygons onto the image plane. Then, the image is computed line by line, from top to bottom, determining the color of each pixel by considering the closest polygons around. Other very common algorithm also exists (see the "Z-buffer technique [6] which is nowadays implemented by default on graphic cards). The main advantage of the "Scanline rendering" is that each pixel is only evaluated once. In return, the memory cost is high because all the polygons of the scene must be loaded at the same time, leading to bad performances for scenes with complex geometries.

### 3.2 With global illumination

Including global illumination (GI) in image rendering processes is a major progress in the quest of photo-realism, but great differences exist between GI techniques. For example, "Radiosity" methods [47, 44] transform the phenomenon of global illumination into a system of linear equations, with different ways of resolutions: direct resolution [4] (very effective but with a high complexity) or iterative algorithms [9]. In another direction, the use of the stochastic algorithm of "Monte-Carlo" [20] is sometimes used despite its slower convergence. "Path Tracing" methods [38] launch random rays from pixels of the image plane until one hits an object. It can be bi-directional (shots rays from camera and sources simultaneously). "Metropolis Light Transport" algorithm (MTL) [14] optimizes "Path Tracing" by replacing the random shooting with heuristics. Modern versions of theses algorithms are progressive. This means they are not limited by a maximum number of bounces but instead continue to converge while the user allows it (or until a quality condition is reached).

The version of "Photon Mapping" algorithm implemented in Virtuelium is not progressive yet but is simpler to implement. It was first defined by Jensen in 1996 and is improved since this date [16, 17]. It decomposes the rendering process into two steps which are executed sequentially. In the pre-rendering step, the position of photons (light-rays launched from light-source) hitting objects are stored in several appropriate structures (photon maps). At least two photon maps are needed, one for the global illumination itself and one for caustics. During the next step, four different contributions are evaluated based on the fact that

$L_r(\vec{\omega}_o)$  can be decomposed into a sum of different integrals. First, the direct and specular contributions are computed the same way than in "Scanline Rendering". Then, the caustic and indirect diffuse contributions are deduced from the two photon maps. Most recent versions of Photon Mapping are nevertheless progressive [14, 13].

## 4 Standard parallelization

The most common way of parallelizing an image rendering algorithm consists in decomposing the image grid. Indeed, as each pixel can always be treated separately without any interaction, it is simple to distribute the computations of several of them over multiple CPUs for a full asynchronous execution. However, some area of the image are probably longer to be rendered than others because of the heterogeneous spacing of objects, materials and lights-sources in the scene. Thus, the first optimization to bring is a dynamic job-balancing mechanism allowing faster threads to work more and ensuring there is no inactivity period for any of them. Something similar can be applied to the global illumination pre-rendering step, replacing the image grid by the list of light-sources. Besides, because many photons are shot in several directions from the sources, rays can also be distributed dynamically in order to maintain a constant activity on every threads.

The main problem with this fully distributed solution is linked to the fact that the whole scene geometry must be known by each computational node. Predetermining the whole light path of a ray is obviously nearly impossible. Furthermore, each polygons in the scene can be hit several times by different rays. For these reasons, every polygon has to be copied onto the memory of each thread, what is not optimal. Hybrid computing (distributed + shared memory) can be used to assure that only a single copy exists on a computational node (and not one per thread), but the problem remains on multiple-node architectures. We know that copying modern complex scenes with several millions of polygons can become a limitation factor, depending on the network bandwidth.

When dealing with full spectral rendering engines like Virtuelium, another idea which is simple to implement consists in decomposing the spectral data themselves. Whereas RGB or RGBA values have only 3 or 4 components, size spectral values used in Virtuelium are often fixed to an array of 81 scalar values encapsulating the visible part of the wavelength range (380 to 780 nm with sampling step of 5 nm). But previous rendering techniques always consider wavelength separately. That is why it is possible to cut the spectral array by group of  $n$  sub-values (for instance,  $n$  can be equal to 3 or 4 to return to something very close to the RGB world). Only fluorescent phenomena bring interactions between wavelengths but this problem is solved if we gather wavelengths craftily to take into account this physical law. Nevertheless, the same problems remains and the scene geometry always has to be copied on every computational node.

The solution we proposed is thus to apply the ray-tracing domain decomposition method introduces in [27].

## 5 Domain Decomposition Method

By splitting a global domain into several small sub-domains, domain decomposition methods [45], [41], [46], [19] allow to load input data and gather results in a parallel way, as each sub-domain can be associated to a unique processor. This splitting can be done only once for multiple execution of the processing algorithm. That is an important advantage particularly when considering very large models besides the fact that in such case, problems of memory allocation are avoided if the sub-domains are small enough.

A basic approach consists in simply splitting the set of pixels into multiple sets, one per sub-domain. In each sub-domain the light rays are shot in the whole geometry. A great part of the work is duplicated, especially the intersections detection and the loading/preprocessing

of the model. Since there is absolutely no communication between the processes, this is a good candidate for largely distributed systems.

However, this method raises some load balancing issues since the processing time of each sub-domain can vary a lot. Thus there is another idea consisting in loading sub-domains on demand but it requires to compute a hierarchical acceleration structure in order to only load the first levels at the start. During the traversal of this hierarchical structure the data corresponding to a node are loaded when the node is reached. If it is an interior node, these data are the child nodes information, if it is a leaf, the data is the corresponding mesh and materials data. This system is pretty complicated to implement and needs a specific pre-computed data structure and it would be necessary to cover the latency of the node loading. Such a method is useful in a context where light rays launched by a process don't spread out of a specific bounded part of the model. Otherwise the process would tend to load the complete geometry.

The method described in [27] takes more advantage of some efficient domain decomposition techniques [31], [23], [32]. Besides the splitting of the global geometry itself, information along interfaces is shared between computational units which are processing neighboring sub-domains. A continuous approach [10], [12], [8], [11], [26], [24], [22] can be used to design efficient interface conditions. Similarly a discrete approach [33], [42], [37], [34], [36] can be used, which may increase significantly the performance of the algorithm. The link between the continuous and discrete interface condition can be established like in [35].

In this work, we split the geometry of the model into multiple sub-domains and base our method on the domain decomposition method [25], [29], [30] where the interface conditions assure the continuity of the light ray properties (such as direction, amplitude, angle, etc.) from one sub-domain to another one, as detailed in [27]. We only analyze the rays passing through interfaces between sub-domains. From a processor point of view, one could replace all neighboring sub-domains model by a simplified version in order to be more efficient.

Yet, unlike classical domain decomposition methods, a computational unit does not process only one sub-domain. Our concern here is related to the load balancing which could be very bad, for instance if there were only one light-source. In such case, most of the workload would be held by the unit processing the sub-domain containing the light-source. This is why we use a less static load-balancing scheme. From a processor point of view, the idea is to start by loading a certain number of sub-domains according to memory limitation. When it remains few light rays to be handled in a sub-domain, this sub-domain is unloaded if there is still other currently not handled sub-domains with a lot of rays not processed. Then the processor starts loading one or more of these sub-domains while handling another sub-domain already available in the memory. Unloading sub-domains allows doing most of the results gathering during the processing of other rays. This overlapping gathering and processing is efficient since gathering mainly uses the communication system. One could find a more complete description of an efficient implementation in [27].

## 6 Results and discussions

An image rendered with Virtuelium is presented in figure 1. Although it is not fully viewable, the whole church model has been rendered using our new parallelization method. The model we used for representing stained glass windows is quite particular. Indeed, instead of directly applying a texture, we used a distribution map of optical constants. According to medieval Cistercian art, nearly clear glass materials were used rather than highly colored tiles. Nevertheless, with our distribution map, we can create diversity between tiles, what is particularly visible on the right windows (where reflections of architecture and other windows are visible). The yellow edgings are also created only with the map. Distribution maps can also be used to distribute other information along the object geometry. This is the case here for glass thicknesses. Only refractive properties of glass materials are simulated for now. One of our future objectives is to develop the material model in order to extend

	16 threads	32 threads	64 threads	128 threads
1 sub-domain	10.6	16.7	25.4	20.2
2 sub-domains	11.9	22.1	34.3	45.4
4 sub-domains	10.4	22.3	35.1	50.7
8 sub-domains	11.2	24.2	39.8	66.9

Table 1: Speedup of the Virtuelium DDM program (Ethernet) with respect to the number of threads and sub-domains.

	16 threads	32 threads	64 threads	128 threads
1 sub-domain	14.5	22.8	27.4	21.0
4 sub-domains	14.6	26.6	46.1	49.2
8 sub-domains	14.7	26.4	47.9	75.9

Table 2: Speed-up of the acoustic DDM program (Ethernet) with respect to the number of threads and sub-domains.

our simulations to a larger part of the physical phenomenon (by example, light scattering is needed). Again, new distribution maps could be used to accurately represent the material complexity.

Speedups of Virtuelium execution are shown in table 1. They are very closed to what we obtained with the acoustic simulation software presented in table 2 [28]. Simulations were run on a hybrid, both distributed and shared memory, computational platform consisting of 4 nodes containing a quad core processor (a total of 16 cores). Each node were provided with 8 Gigabytes RAM (Random Access Memory). As we were expecting, DDM techniques significantly improved the performance of the parallelization. Although the speedups from acoustic simulation are quite better, we can notice that in both cases, from 16 to 128 threads, 8 sub-domains decomposition allowed to multiply the speedup by nearly 6, while classical parallelization only reach a factor less than 2. On another hand, for a fixed number of threads, the speedup keeps increasing as the number of sub-domains do.

## 7 Conclusion

In this paper, we proposed an original ray-tracing domain decomposition method for image rendering with natural lighting. According to domain decomposition methods principle, light rays characteristics have been matched as interface constraints between neighboring sub-domains. We presented a test case on a model of the church of the Royaumont abbey where

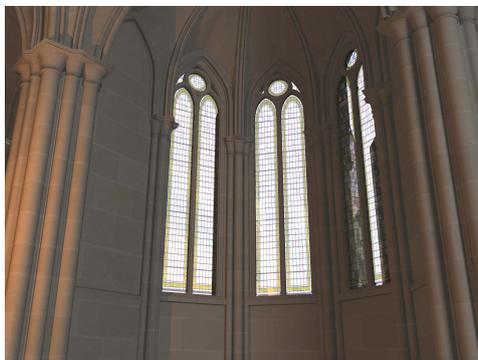


Figure 1: Illustration of the image rendering in the church of Royaumont abbey (interior view)

we particularly deal with medieval glass material properties. It outlined the performance and efficiency of our method, relatively to multi-core architectures.

## Acknowledgements

The authors acknowledge the Foundation Royaumont for its help and in particular Jerome Johnson and Nathalie Le Gonidec for the helpful discussions and comments.

## References

- [1] J. Arvo. Backward ray tracing. In *ACM SIGGRAPH 86 Course Notes - Developments in Ray Tracing*, pages 259–263, 1986.
- [2] M. Bass. *Devices, measurements, and properties*. Number 2 in Handbook of Optics. McGraw-Hill and Optical Society of America, 1995.
- [3] K. Berger, A. Wilkie, A. Weidlich, and M. Magnor. Modeling and verifying the polarizing reflectance of real-world metallic surfaces. *Computer Graphics and Applications*, 32(2):24–33, Mar. 2012.
- [4] J. Bu and E. F. Deprettere. A vlsi system architecture for high-speed radiative transfer 3d image synthesis. *The Visual Computer*, 5(3):121–133, 1989.
- [5] P. Callet. *Couleur-lumière, couleur-matière: interaction lumière-matière et synthèse d'images*. Sciences en actes. Diderot éditeur, Arts et sciences, 1998.
- [6] E. E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, 1974. AAI7504786.
- [7] R. Cerise, F. da Graça, F. Magoulès, and P. Callet. Natural lighting and medieval glass - scientific data acquisition, methodology and physically based rendering. In *Proceedings of the 4th International Conference on Progress in Cultural Heritage Preservation, EuroMed'12*, pages 636–643, Berlin, Heidelberg, 2012. Springer-Verlag.
- [8] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.
- [9] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '88*, pages 75–84, New York, NY, USA, 1988. ACM.
- [10] B. Després. Domain decomposition method and the Helmholtz problem.II. In *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*, pages 197–206, Philadelphia, PA, 1993. SIAM.
- [11] M. Gander, L. Halpern, and F. Magoulès. An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *International Journal for Numerical Methods in Fluids*, 55(2):163–175, 2007.
- [12] S. Ghanemi. A domain decomposition method for Helmholtz scattering problems. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods*, pages 105–112. ddm.org, 1997.
- [13] T. Hachisuka and H. W. Jensen. Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph.*, 30(5):114:1–114:11, 2011.

- [14] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. *ACM Trans. Graph.*, 27(5):130:1–130:8, Dec. 2008.
- [15] J. Jellison, G.E. and F. Modine. Optical constants for silicon at 300 and 10 k determined from 1.64 to 4.73 eV by ellipsometry. *Journal of Applied Physics*, 53(5):3745–3753, May 1982.
- [16] H. W. Jensen. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pages 21–30, London, UK, 1996. Springer-Verlag.
- [17] H. W. Jensen. A practical guide to global illumination using ray tracing and photon mapping. In *ACM SIGGRAPH 2004 Course Notes*, SIGGRAPH '04, New York, NY, USA, 2004. ACM.
- [18] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, Aug. 1986.
- [19] J. Kruis. *Domain Decomposition Methods for Distributed Computing*. Saxe-Coburg Publications, 2007.
- [20] E. P. Lafortune. *Mathematical models and Monte Carlo algorithms for physically based rendering*. PhD thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, Feb. 1995.
- [21] LuxRender, 2014. Available online at [http://www.luxrender.net/en\\_GB/index](http://www.luxrender.net/en_GB/index).
- [22] Y. Maday and F. Magoulès. Non-overlapping additive Schwarz methods tuned to highly heterogeneous media. *Comptes Rendus à l'Académie des Sciences*, 341(11):701–705, 2005.
- [23] Y. Maday and F. Magoulès. Absorbing interface conditions for domain decomposition methods: a general presentation. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3880–3900, 2006.
- [24] Y. Maday and F. Magoulès. Improved ad hoc interface conditions for Schwarz solution procedure tuned to highly heterogeneous media. *Applied Mathematical Modelling*, 30(8):731–743, 2006.
- [25] Y. Maday and F. Magoulès. Optimal convergence properties of the FETI domain decomposition method. *International Journal for Numerical Methods in Fluids*, 55(1):1–14, 2007.
- [26] Y. Maday and F. Magoulès. Optimized Schwarz methods without overlap for highly heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1541–1553, 2007.
- [27] F. Magoulès. Décomposition de domaines pour le lancer de rayons. France Patent no. 1157329. 12 August 2011.
- [28] F. Magoulès, R. Cerise, and P. Callet. A beam-tracing domain decomposition method for sound holography in church acoustics. In *Distributed Computing and Applications to Business, Engineering Science (DCABES), 2013 12th International Symposium on*, pages 61–65, Sept 2013.
- [29] F. Magoulès, P. Iványi, and B. Topping. Convergence analysis of Schwarz methods without overlap for the Helmholtz equation. *Computers and Structures*, 82(22):1835–1847, 2004.

- [30] F. Magoulès, P. Iványi, and B. Topping. Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 193(45–47):4797–4818, 2004.
- [31] F. Magoulès, K. Meerbergen, and J.-P. Coyette. Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry. *Journal of Computational Acoustics*, 8(3):503–521, 2000.
- [32] F. Magoulès and F.-X. Roux. Lagrangian formulation of domain decomposition methods: a unified theory. *Applied Mathematical Modelling*, 30(7):593–615, 2006.
- [33] F. Magoulès, F.-X. Roux, and S. Salmon. Optimal discrete transmission conditions for a non-overlapping domain decomposition method for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 25(5):1497–1515, 2004.
- [34] F. Magoulès, F.-X. Roux, and L. Series. Algebraic way to derive absorbing boundary conditions for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):433–454, 2005.
- [35] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3742–3759, 2006.
- [36] F. Magoulès, F.-X. Roux, and L. Series. Algebraic Dirichlet-to-Neumann mapping for linear elasticity problems with extreme contrasts in the coefficients. *Applied Mathematical Modelling*, 30(8):702–713, 2006.
- [37] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approach to absorbing boundary conditions for the Helmholtz equation. *International Journal of Computer Mathematics*, 84(2):231–240, 2007.
- [38] A. Pajot, L. Barthe, M. Paulin, and P. Poulin. Combinatorial bidirectional path-tracing for efficient hybrid cpu/gpu rendering. *Computer Graphics Forum*, 30(2):315–324, 2011.
- [39] E. Palik. *Handbook of Optical Constants of Solids*. Number 1. Elsevier Science, 1985.
- [40] M. Pharr and G. Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.
- [41] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, UK, 1999.
- [42] F.-X. Roux, F. Magoulès, L. Series, and Y. Boubendir. Approximation of optimal interface boundary conditions for two-Lagrange multiplier FETI method. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Proceedings of the 15th International Conference on Domain Decomposition Methods, Berlin, Germany, July 21-15, 2003*, Lecture Notes in Computational Science and Engineering. Springer-Verlag, Haidelberg, 2005.
- [43] P. Shirley, R. K. Morley, P.-P. Sloan, and C. Wyman. Basics of physically-based rendering. In *SIGGRAPH Asia 2012 Courses*, SA '12, pages 2:1–2:11, New York, NY, USA, 2012. ACM.
- [44] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '89, pages 335–344, New York, NY, USA, 1989. ACM.

- [45] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, UK, 1996.
- [46] A. Toselli and O. Widlund. *Domain Decomposition methods: Algorithms and Theory*. Springer, 2005.
- [47] J. R. Wallace, M. F. Cohen, and D. P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *SIGGRAPH Comput. Graph.*, 21(4):311–320, Aug. 1987.
- [48] J. A. Woollam, W. McGaham, and B. Johs. Spectroscopic ellipsometry studies of indium tin oxide and other flat panel display multilayer materials. *Thin Solid Films*, 241(1-2):44 – 46, 1994. Papers presented at the European Materials Research Society 1993 Spring Conference, Symposium C: Ion Beam, Plasma, Laser and Thermally-Stimulated Deposition Processes, Strasbourg, France, May 4-7, 1993.
- [49] C. Wylie, G. Romney, D. Evans, and A. Erdahl. Half-tone perspective drawings by computer. In *Proceedings of the November 14-16, 1967, Fall Joint Computer Conference*, AFIPS '67 (Fall), pages 49–58, New York, NY, USA, 1967. ACM.