HAL
open science

# Autonomous Lighting Agents in Photon Mapping

Adrien Herubel, Venceslas Biri, Stéphane Deverly

## To cite this version:

Adrien Herubel, Venceslas Biri, Stéphane Deverly. Autonomous Lighting Agents in Photon Mapping. ISVC 2009, Nov 2013, Las Vegas, United States. pp.919-928, 10.1007/978-3-642-10520-3_88 . hal-00681777

**HAL Id: hal-00681777**

**https://hal.science/hal-00681777**

Submitted on 22 Mar 2012

**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.

# Autonomous Lighting Agents in Photon Mapping

A. Herubel, V. Biri and S. Deverly

Laboratoire d'Informatique Gaspard Monge
Duran Duboi

**Abstract.** In computer graphics, global illumination algorithms such as photon mapping require to gather large volumes of data which can be heavily redundant. We propose both a new characterization of useful data and a new optimization method for the photon mapping algorithm using structures borrowed from Artificial Intelligence such as autonomous agents. Our autonomous lighting agents efficiently gather large amounts of useful data and are used to make decisions during rendering. It induces less photons being cast and shorter rendering times in both photon casting and rendering phase of the photon mapping algorithm which leads to an important decrease of memory occupation and slightly shorter rendering times for equal image quality.

## 1 Introduction

Physically-based global illumination algorithms such as photon mapping [1] have a linear progression between complexity and quality. To a given quality, rendering time scales linearly with computer performances. With Moore's law call in question and increasing demand in quality, those algorithms need more and more

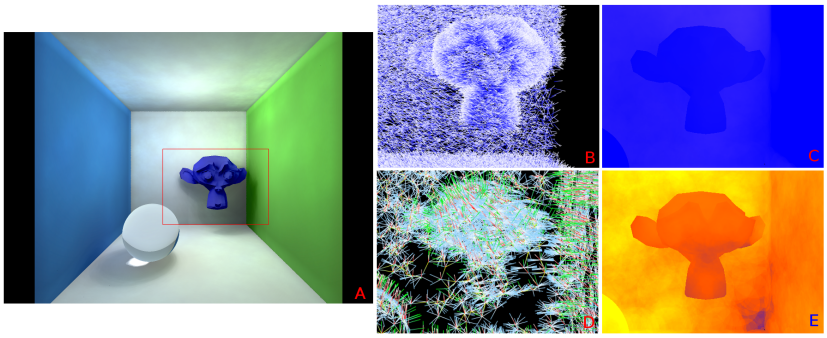


**Fig. 1.** A. Our test scene with classical indirect photon map (B. middle top) and ALA map (D. middle bottom) and corresponding local useful data density (respectively C. top right for classical photon map and D. bottom right for ALA map); In false color, red means high useful data when blue means low

optimisations. Classical optimisations such as irradiance caching [2] or shadow photons [3] are themselves linear in performance gain. They usually consist of adding knowledge about the scene and using it to interpolate previously computed values. Data is gathered in large volumes although due to heavy redundancy we observe a low density of useful data (see Fig 1.B and C).

We propose a brand new optimisation method for the photon mapping global illumination algorithm which uses Artificial Intelligence structures and concepts to resolve the sparse data problem in global illumination presented above. More precisely, we create our own autonomous agent structure called Autonomous Lighting Agents (ALA) that performs an agent-based scene discovery. They can efficiently gather and store large amounts of useful data (see Fig 1.D and E) thanks to various sensors. Moreover ALA exist within multiple graph representations that depict relations between agents such as neighbourhood or light paths. The resulting structures are networks of agents storing data non-uniformly with increased density in area of interest. Then any ALA and the associated networks can be queried during rendering to make decisions regarding, for example, irradiance caching or shadow casting. Therefore, fewer photons are cast reducing computing time during both the photon casting and the rendering phases of the algorithm.

In the next section, we present the photon mapping algorithm and overview the main optimisations on that method. Then we present our new criterion to evaluate usefulness of local data density. To match closely this particular data density, we detail our method showing how ALA discover the scene and how they are queried for final rendering. Finally, we present our first results showing an important decrease of memory occupation and slightly shorter rendering times, compared to optimized photon mapping.

## 2 Global illumination and photon mapping

We call global illumination the simulation of all light scattering phenomena in a virtual scene. Photon mapping presented in [4, 5] is currently one of the most efficient physically based algorithm capable of global illumination rendering. Compared to other global illumination methods like radiosity [6, 7] or Metropolis Light Transport [8, 9], photon mapping is a robust and consistent two-pass algorithm [10], is able to handle many light effects including caustics, and is modular since it separates illumination in several layers [4]. The first pass traces photons from all lights through the scene and stores them, at hit points, in so called photon map. The second pass uses ray tracing to render the image. In the ray tracing pass the photon map is used to estimate the radiance at different locations within the scene. This is done by locating the nearest photons and performing a nearest neighbor density estimation (see Fig. 2). A detailed presentation of this method can be found in [1].

Despite its numerous qualities, photon mapping is still a slow and complex algorithm. Classical optimisations include shadow photons, direct computing and importance sampling of the indirect illumination evaluation. Shadow photons,

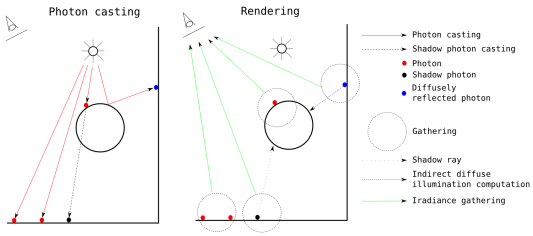


**Fig. 2.** The two passes of photon mapping

as shown in [3], are stored in a separate photon map which is used to speed-up ray traced shadow computing. The photon map can also quickly compute indirect illumination since it has a low variance. To achieve this, diffusely reflected photons must be stored in a separate photon map that will be queried directly to compute indirect illumination at a given point. It is advised to use precomputed irradiance described in [11] to speed up the photons gathering. Finally, for the complete indirect illumination evaluation, the photon map can be used to accelerate computing. This is done thanks to importance sampling [4] by gathering the diffusely reflected photons stored near the hit point and retrieving their incoming direction. Ward et al [2] proposed also irradiance caching, used to accelerate the indirect diffuse illumination. The key idea is to interpolate, if possible, irradiance value between several points which are relatively close and share the same orientation. Unfortunately, this optimization is itself costly and not adapted to multithreaded implementations of path tracing.

Advanced optimization address the drawback of the density of the photon map. Indeed, the photon distribution is driven by lights, and for large models where only a small part is visible from the viewer, like the example scene presented in [8], it may be better to cast photon only where they are needed. Techniques like [12, 13] focus on the camera to influence the direction of photons casted by lights. However this techniques, like Importons, may induce high variance in the photon map by generating high-power photons as explained in [11] (pp 146). Density control, introduced in [14], consists of limiting the presence of photons in bright area allowing more photons in darkest areas. Recent algorithms [15, 10], inverting the two passes of photon mapping, allow to minimize the bias, but they are very dependent on the view position and still use a traditional photon mapping inducing the same drawbacks in photon density. Nevertheless, the locality of photon maps in [10] involves a limited density for each iteration of the progressive photon mapping.

## 3 Local useful data density

In photon mapping, lights are responsible for the distribution of photons and an inadequate repartition will keep the photon map from playing its role in rendering. The first pass sends an important amount of photons with only slight

variations in their own data. Unfortunately, they are required to achieve sufficient visual quality, or noise can appear as shown in [1] (pp 150).

Classical optimization, as density control or inversion techniques, presented in 2, address partially this issue by restricting the number of photons using local density and visual importance. But they are very view dependant, still send a lot of photons or assume strong variance in lighting in order to be efficient, especially for the density issue.

Therefore, our first goal is to found a mean of characterizing the usefulness of data in each area of the scene. Indeed, if we observe Fig. 1.B. we find a high density photons which are locally very similar. It leads us to introduce the concept of local useful data density (LUDD) in the photon map to represent the local density of useful data for any criterion. First, for a particular criterion $k$, we define the local density of useful $k$-data, denoted by $\delta_k$. It is defined as the standard deviation of $k$ computed for $N$ included samples $\{x_{kj}\}$ on a local sphere.

$$\delta_k = \frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(x_{ki} - \frac{1}{N}\sum_{j=1}^{N} x_{kj}\right)^2}}{\frac{1}{N}\sum_{j=1}^{N} x_{kj}} \tag{1}$$

The whole local density of useful data $\delta$ for $K$ criteria is simply :

$$\delta = \frac{\sum_{k=1}^{K} \delta_k}{N} \tag{2}$$

Criteria used to perform this measure include surface normal, light power or direction. This measure can be expanded for every additional data carried by the photon map such as visual importance. The $\delta$ value computed Fig. 1.C. is particularly low which implies that added value of each photon is very small. However in Fig. 1.E., $\delta$ is higher which means our agent map has locally a strong useful data density. Our observations show that local useful data density is not evenly distributed in the photon map and that this measure is strictly correlated to local conditions in the scene.

## 4 Our method using Autonomous Lighting Agents

### 4.1 An autonomous agent approach

To maximize $\delta$ measure, we need to create a new structure dedicated to scene discovery instead of carrying light flux with an evenly strong $\delta$. We want it to be independent of human interaction, of any tessellation, and of visual importance. The ability to discover local phenomena as geometry gradient and light variations leads us to the autonomous agent software paradigm [16]. In consequence our structure is not a monolithic entity but a swarm of agents able to perceive their environment through sensors [17] and to make behavioural decisions based on the given criteria. We define our autonomous lighting agents (ALA) as entities with a temporal persistence coexisting in a scene with other ALA and interacting with them [18].

We use formal description as described in litterature [19] to represent mathematically our model. We give each agent an immutable and unique identity $i$. An ALA has three abilities, it can be cast by an arbitrary light source, it can discover the area around its location and it can answer simple questions related to this area. The model also specifies attributes or sensors owned by agents and denoted by $a$. Therefore, our new ALA improved photon mapping is organized as follows :

1. Replace the classical photon casting pass with ALA casting pass (see 4.2)
2. Make ALA discover their environment (see 4.3)
3. Do the rendering pass using ALA decision making for different layers (see 4.4)

### 4.2 ALA casting

We choose to cast ALA from the lights, uniformly on a sphere. When hitting a surface an ALA acts like both a photon and a shadow photon in the photon mapping algorithm, as it is stored and duplicated. Like shadow photon, the obtained ALA clone passes through the surface. In consequence ALA are present in occluded zones. Agents are stored using a kd-tree in the same way than photon maps. A visualization of the ALA network is shown in Fig. 1.D. After this stage, agents are evenly distributed in the scene at location $\{x_i\}$.

### 4.3 ALA deployment

**Local discovery with sensors** Each agent can deploy sensors to gather local data about the scene and other agents. The photon map only provides data at each discrete position of each photon whereas our ALA deploy sensors on a local parametric discovery area. A sensor is identified with an identity $j$ and is attached to an agent $i$. Each sensor corresponds also to an attribute $a$, so we define a value discovered by a sensor as $a_{ij}$. Each agent can deploy more than one sensor for each kind of attribute $a$. Sensor position are denoted by $x^a_{ij}$. Finally, for each sensor $a$, we define a coefficient of interest $\sigma_a$, computed by the difference of the value perceived by the sensor with an arbitrary chosen median value $\sigma^m_a$. Sensor used are light sensor, penumbra sensor, geometry gradient sensor and proximity sensor. Table 1 shows the different measures and median values used for each sensor.

**Gold miner decision algorithm** As we aim to achieve an optimal LUDD repartition across the scene, ALA are entitled with the decision of choosing the quantity of data they carry. In the classical photon map, LUDD is mainly related to local photon density. ALA strongly differ on this point, as data is not related by the presence of the agent itself. A single agent can deploy dozens of sensors on large distances as well as many agents can choose not to use any. Each agent will choose which types of sensors are deployed, how many of them and how far they can go. We use a decision algorithm making the agent acting like a gold miner (see Alg. 1).

| **Sensor** | **Measurement** | **Median Value $\sigma_a^m$** |
|---|---|---|
| Incomming irradiance $i$ | Integrated incomming irradiance on an hemisphere above the $i$ sensor | Mean distance between objects. |
| Penumbra $p$ | Number of intersections with surfaces along the ray between $p$ sensor location and light source divided by distance to source light | 1.0 |
| Geometry gradient $g$ | $\frac{1}{3}\sum_{k=1}^{3}\frac{1}{d_k}$, $d_k$ being the distance to the closest intersection from sensor to any surface in a arbitrary direction | $\frac{2}{3}$ |
| Proximity $n$ | ALA density around the sensor | $\frac{ALA\ cast}{nb\ light} \times \frac{1}{100}$ |

**Table 1.** Different sensors used with their measurement and median value used

A sensor reports its coeficient of interest $\sigma$, an interesting result will increase the stimulation factor $\gamma$ of the agent. On the contrary, a non-pertinent result will decrease this factor. Stimulation factors include : little number of sensors, strong irradiance gradient, high geometric gradient, high or low number of neighbours. Discouragement factors include great number of sensors or planarity of surface. Factors are of different natures and some of them implies computing the gradient of two given criteria $a_{ij}$. Between each deployment we test if an excitement or unexciting factor is triggered. The agent will continue to deploy sensor and gather data until the stimulation factor drops below a certain level.

**Computing the proximity network** When all ALA have been cast and all sensors are deployed, we compute the proximity network on the whole agent set. The network is a graph in which each agent is a node related to its closest neighbours in the scene space. The graph is computed quickly using the kd-tree to find the neighbours for each agent.

**Algorithm 1** Gold miner algorithm

```
γ = 0.0, Σσ = 0.0;
for all sensor_types do
  while γ > 0 do
    direction = random()
    distance += ε
    σ = deploy_sensor(direction,distance)
    γ += σ - (Σσ/10)
  end while
end for
```

### 4.4 Rendering pass using ALA

The ALA structure allows an efficient agent based local discovery of the scene, data is then used during the rendering pass. The ALA network aim to replace the photon map for all kind of decision making optimisations therefore avoiding the costly generation of direct photons, indirect photons and shadow photons.

We define our model as an agent based decision maker called the oracle algorithm. Decision making optimisations can be seen as "Should I?" questions such as "Should I trace shadow ray?" or "Should I recompute incoming irradiance?". We formalize thoose questions in an Artificial Intelligence procedure.

**Avoiding shadow rays** The oracle algorithm, answers the question : "Should I trace shadow rays ?". This is used each time we want to trace a shadow ray from a particular position. Practically, the oracle will locate the closest ALA using the ALA kd-tree and compute the difference between the sum of its penumbra sensors $p_{ij}$ (as defined in Table. 1) and the median penumbra sensor value.

$$\left(\sum p_{ij} - \sigma_p^m\right) > 0 \rightarrow \{YES, NO\}$$

To achieve more robustness it is possible to use the median response of neighbours agents of $i$ using the proximity network.

**Caching Irradiance** Our implementation allows to compute an approximation of diffuse indirect illumination at a position $x_{ij}$. We implement a procedure which compute the mean distance to closests irradiance sensors $g_{ij}$ at positions $x_{ij}^g$ and compare it to the median value of this sensors.

$$\left(\sum x_{ij}^g - x_{ij} - \sigma_i^m\right) > 0 \rightarrow \{YES, NO\}$$

If the procedure answers negatively, indirect illumination is computed by interpolating irradiance values of closest sensors. In application due to homogeneous repartion of ALA few indirect diffuse illumination values need to be computed during the rendering phase.

## 5 Results and discussion

In this section we present results based on our implementation of the ALA structure and algorithms. For comparison we implemented a standard photon mapping algorithm with shadow photons, importance sampling and irradiance caching. Images have been rendered on a 2.67GHz Intel Core i7 920 using four cores with a resolution of 1600×1200. The test scene shown in Fig. 4 is a common Cornell box with a mesh and a glass ball. The scene is illuminated by a spherical light source on the ceiling.

Fig. 4 shows a rendering of our test scene including only direct lighting, caustics and shadows. We can see that difference is subtle and due uniquely to

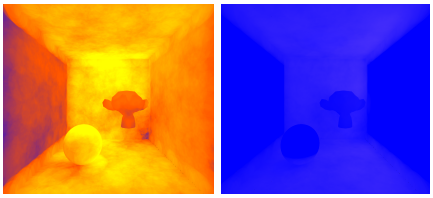

**Fig. 3.** Local useful data density for the indirect on ALA (left) and on classical the photon map (right)

some aliasing near the silhouette. In Fig. 5 1M photon are cast leading to a 4.9Mb shadow photon map. We observe that the ALA method, while achieving faster rendering, avoids the usage of shadow photons and produces nearly identical shadows with only 50000 ALA. The indirect diffuse illumination is computed using importance sampling on the indirect photon map and irradiance caching using the direct and indirect photon map. The ALA allow to avoid computation of the two photon maps while providing an accurate but perfectible estimation of the indirect illumination in slightly shorter rendering times as shown in Fig. 3.

We observe in Fig. 3 that the local useful data density considering the irradiance criteria is uniformly low in the photon maps. This is explained by the local redundancy of the irradiance data. Our structure show a considerably better density, as the number of sensors perfectly fits with the redundancy of data. This difference leads to an ALA casting phase shorter than the photon casting phase as well as a significant reduction of memory usage as show in 2.

## 6 Conclusions and future work

In this paper, we stated that photon density is not necessarily related to useful data for some light estimation. We therefore propose a new estimation of the local density of useful data we call LUDD for local useful data density. We then presented a new agent-based model to discover efficiently a scene, in the sense of matching correctly LUDD. To implement the method we introduced a new structure : Autonomous Lighting Agents (ALA), which are an efficient container to handle data about the scene. We show how the ALA network can be used as

| **Map** | **Total casting times (s)** | **Memory occupation (mb)** |
|---|---|---|
| Direct PhotonMap | 46 | 18 |
| Indirect PhotonMap | | 9.4 |
| Caustic PhotonMap | | 0.004 |
| Shadow PhotonMap | | 4.9 |
| ALA Network | 27 | 5.2 |

**Table 2.** Comparative results with ALA in memory occupation and casting times

| Image | Number of photons/agents | Rendering times (s) |
|---|---|---|
| Photon mapping | 1M | 21570 |
| ALA | 50000 | 16557 |

**Table 3.** Comparative results with ALA in rendering times

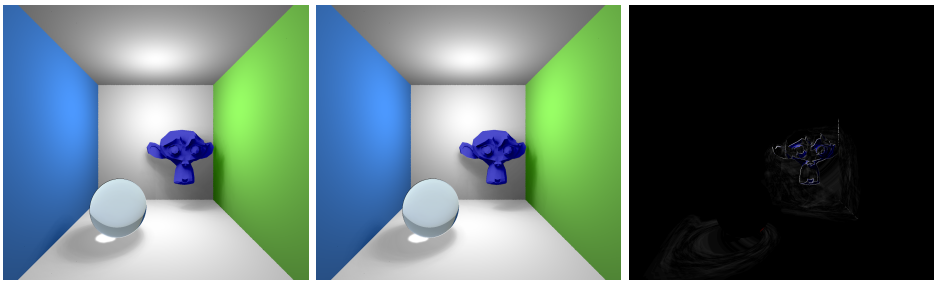

**Fig. 4.** Rendering test scene without indirect diffuse lighting with standard photon mapping (left), ALA photon mapping (center) and difference multiplyed by 5 between two rendered image (right)

an oracle or estimation to lower rendering times and reduce memory overhead in the photon mapping algorithm.

ALA concept and networks could be extended to answer other critical question related to rendering, like halting or not path tracing recursion or handle importance sampling in each intersection related to the whole light distribution in the scene. ALA network could also force photon casting in importance direction, leading light path. We plan also to improve greatly irradiance estimation with ALA networks to correctly replace irradiance caching. Finally, this approach of concentrating efforts on local area, where useful data can be found, could be tested on other kind of vizualisation of 3D data or on other global illumination algorithm like Metropolis Light Transport.

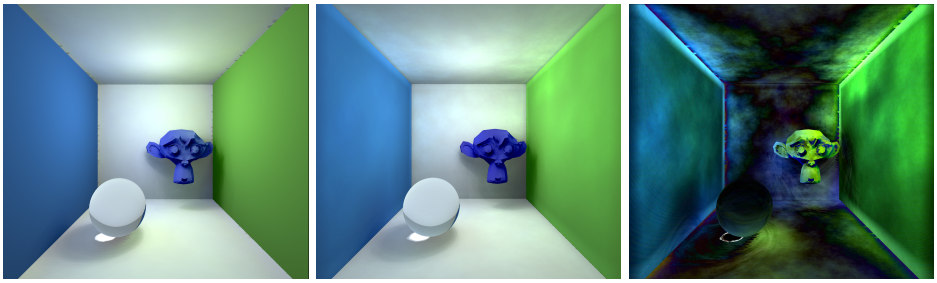

**Fig. 5.** Rendering test scene with standard photon mapping (left), ALA photon mapping (center) and difference multiplyed by 5 between two rendered image (right)

## Acknowledgements

We would like to thank the DuranDuboi R&D team for their support and for providing the rendering infrastructure.

## References

1. Jensen, H.W.: Realistic Image Synthesis Using Photon Mapping. A K Peters (2001)
2. Ward, G., Rubinstein, F., Clear, R.: A ray tracing solution for diffuse interreflection. Proceedings of the 15th annual conference on Computer graphics and interactive techniques (1988) 85–92
3. Jensen, H., Christensen, N.: Efficiently Rendering Shadows using the Photon Map. Compugraphics'95 (1995) 285–291
4. Jensen, H.: Importance driven path tracing using the photon map. Rendering Techniques **95** (1995) 326–335
5. Jensen, H.: Global Illumination using Photon Maps. Rendering Techniques '96 (1996) 21–30
6. Goral, C., Torrance, K., Greenberg, D., Battaile, B.: Modeling the interaction of light between diffuse surfaces. In: proceedings of SIGGRAPH'84, Computer Graphics. Volume 18(3). (1984) 213–222
7. Keller, A.: Instant radiosity. In: SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. (1997) 49–56
8. Veach, E., Guibas, L.J.: Metropolis light transport. In: SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. (1997) 65–76
9. Segovia, B., Iehl, J.C., Péroche, B.: Metropolis Instant Radiosity. Computer Graphics Forum **26** (2007) 425–434
10. Hachisuka, T., Ogaki, S., Jensen, H.W.: Progressive photon mapping. In: SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers, ACM (2008) 1–8
11. Christensen, P.: Faster photon map global illumination. Graphics Tools: The Jgt Editors' Choice (2005) 241
12. Pietrek, I.: Importance driven construction of photon maps. In: Rendering Techniques' 98: Proceedings of the Eurographics Workshop in Vienna, Austria, June 29-July 1, 1998, Springer Verlag Wien (1998) 269
13. Fan, S., Chenney, S., chi Lai, Y.: Metropolis photon sampling with optional user guidance. In: Rendering Techniques '05 (Proceedings of the 16th Eurographics Symposium on Rendering), Eurographics Association (2005) 127–138
14. Suykens, F., Willems, Y.: Density control for photon maps. Rendering techniques **2000** (2000) 23–34
15. Havran, V., Herzog, R., Seidel, H.P.: Fast final gathering via reverse photon mapping. Computer Graphics Forum (Proceedings of Eurographics 2005) **24** (2005) 323–333
16. Maes, P.: Designing autonomous agents: theory and practice from biology to engineering and back. MIT press (1990)
17. Maes, P.: Modeling adaptive autonomous agents. Artificial Life **1** (1994) 135–162
18. Treuil, J.P., Drogoul, A., Zucker, J.D.: Modélisation et simulation à base d'agents. Dunod (2008)
19. Lerman, K., Galstyan, A., Martinoli, A., Ijspeert, A.: A macroscopic analytical model of collaboration in distributed robotic systems. Artificial Life **7** (2001) 375–393