Eurographics Workshop on Urban Data Modelling and Visualisation (2013)
V. Tourre and G. Besuievsky (Editors)

NightLighting: a nocturnal urban illumination approach

I. Muñoz-Pandiella[1], C. Andújar[1], G. Patow[2]

[1]ViRVIG Institute, BarcelonaTech - Universitat Politècnica de Catalunya, Barcelona, Spain
[2]ViRVIG Institute, Universitat de Girona, Girona, Spain

Figure 1: Our algorithm computes an initial approximate solution for the illumination of a city (suitable for far views) which is then refined locally on-demand using photon mapping.

Abstract

Real time rendering of cities with realistic global illumination is still an open problem. In this paper we propose a two-step algorithm to simulate the nocturnal illumination of a city. The first step computes an approximate aerial solution using simple textured quads for each street light. The second step uses photon mapping to locally compute the global illumination coming from light sources close to the viewer. Then, we transfer the local, high-quality solution to the low resolution buffers used for aerial views, refining it with accurate information from the local simulation. Our approach achieves real time frame rates in commodity hardware.

1. Introduction

Urban rendering has become an important area in many computer graphics applications ranging from urban planning to entertainment [CBG*07, DBCG*09, ABCN10, AAP12], which often need to provide not only realistic and detailed urban space renderings, but also realtime frame-rates and improved image quality with global illumination effects. However, urban environments still pose a serious problem when the inter reflections of light come into play. Taking into account global illumination effects has done only for the case of day-lighting, where a single high-intensity source (i.e., the sun) emits light [AAP12]. However, nocturnal urban environments represent the opposite situation, where thousands of low-intensity sources (e.g., street lamps, windows, car headlights) illuminate highly localized areas. Because of this, real-time nocturnal urban rendering with realistic illumination is still a challenging problem.

In this paper we address, in a efficient way, the nocturnal illumination of a city. Our approach is based on a key observation: when viewers are flying over the city, they will be able to view all light emitters but with little detail. For this situation we propose a solution based on direct *OpenGL* rendering. On the other hand, when observers are on the streets, a small number of lights with a high level of detail would be observed. For such views we use an accurate photon mapping simulation taking into account only the most influential light sources. The algorithm we propose works at these two levels, trying to optimize resources to provide an interactive, global illumination solution for cities at night.

DOI: 10.2312/UDMV/UDMV13/013-016

2. Previous work

In the last years a number of algorithms for real-time rendering of urban environments have been proposed [CBG*07, DBCG*09, ABCN10]. These approaches achieve interactive frame rates by adapting traditional acceleration strategies (level-of-detail, image-based rendering and visibility culling) to the particular properties of city models: 2.5D overall shape, plane-dominant geometry, regular structure, dense occlusion, and large texture datasets.

The poor performance of traditional mesh simplification algorithms on urban models has motivated the development of algorithms specifically designed for buildings [HW10, DB05] and groups of buildings [YZM*11], most of them generating discrete LoD representations.

Several image-based representations have been proposed to accelerate the rendering of distant buildings, including impostors [MS95], textured depth meshes [SDB97], and point-based impostors [WWS01]. Some recent approaches represent the geometry of 2.5D cities as hierarchies of displacement maps which are rendered using relief mapping techniques [CBG*07, DBCG*09, ABCN10].

Despite all the above techniques, current approaches for large-scale urban rendering only support direct illumination plus shadow mapping. Indirect illumination effects in urban rendering are limited to environments maps and precomputed ambient occlusion [DBCG*09], thus limiting photorealistic appearance under changing lighting conditions [VAW*10]. Argudo et al. [AAP12] proposed a photon map-based approach, but its practicality is limited to daylight situations, where a single, high intensity source illuminates the whole scene.

3. Urban Illumination at Night

3.1. Model preprocessing

We assume that the input urban model consists of a collection of textured-mapped buildings together with an orthophoto capturing the appearance of floor and ceilings. This kind of representation is common for models of existing cities [ABCN10]. The orthophotos that can be obtained for any city are, in general, taken during the day. This fact precludes their usage for night illumination, as the result would be unnatural due to the daylight shading. Instead of attempting to extract the surface reflectance properties from the orthophoto, we address this problem through a simple *Histogram matching* [PCR11] technique to change the photo's color towards the typical orange color of high pressure sodium light illumination. Given a source and a target image, the purpose of this technique is to modify the source image so that its histogram matches the target image histogram. We use the RGB channels of the orthophoto as source images, and a user-provided nocturnal photo as target image. Since the target image is expected to be dark, better results can be achieved by ignoring the pixels whose luminance is below some threshold when computing the histogram of the target image. This results in a brighter, more realistic result, see Figure 2.

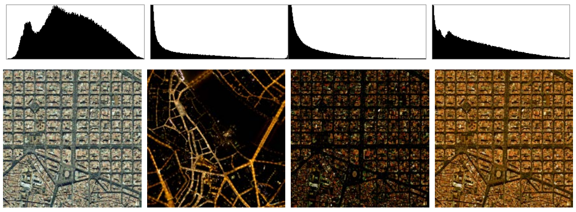
(a) (b) (c) (d)

Figure 2: Histogram matching example: (a) original orthophoto, (b) target image, (c) result of histogram matching, (d) result of histogram matching ignoring the darkest pixels in the target image.

3.2. Aerial approach

Aerial views of the city show the effect of hundreds or thousands of light sources but with very little detail. We thus propose to compute a light map $\mathcal{L}$ encoding the illumination on the floor, and reuse it also to illuminate the facades.

First, we approximate the illumination at a point P due to a single street lamp at Q using a Phong-like formula depending on the distance $dist(P,Q)$, allowing a smooth transition between the most illuminated parts and the dimmest ones. In our implementation, we defined a maximum influence radius of 30 meters, enough to allow interaction between the street lights. This results in an illumination pattern that is encoded in a texture.

Then we generate the light map $\mathcal{L}$ (a single-channel texture) by drawing a quad of size $2r_{max}$ centered at every street light position using a zenithal camera. Each quad is textured with the illumination pattern above. Either the accumulation buffer or an additive alpha blending function can be used to gather the contribution of multiple lamps at each individual pixel.

The light map $\mathcal{L}$ is generated once and used at runtime to compute the illumination of the scene whenever the observer is far away from the city. Computing the illumination at a point P on the floor is straightforward as it requires a simple query of its corresponding texel in $\mathcal{L}$. For the facades, the algorithm works in a more approximate way. The illumination at a point P on a facade is computed as the illumination of its corresponding floor texel in $\mathcal{L}$ dimmed by a factor that depends on the height of the P and the height of the street lights. The left image in Figure 1 shows an aerial view of a city rendered using our approach.

3.3. Close-view approach

The approach described above is suitable for aerial views but lacks accuracy and detail for close-up views. For such close-up views, we aim to simulate the nocturnal illumination as fast and as physically realistic as possible, using a GPU-based photon mapping approach [ML09].

Instead of tracing the photons from hundreds or thousands of lamps simultaneously, we adopt an on-demand approach where the photon maps $\mathcal{P}_i$ of individual lamps are computed on-the-fly. These photon maps are stored as textures instead of alternative structures such as kd-trees or screen-space maps to guarantee their persistence and re-usability accross multiple frames.

Due to the structure of our scene, we decided to use separate collections of photon maps for the floor and the facades. Photons hitting the floor are encoded in per-lamp floor photon maps. Each floor photon map covers a squared region centered on the street light position. This square region matches the size of the quad used for the aerial approach, which provides an efficient usage of space. In our test dataset, this size constraint resulted in discarding only around 2% of the traced photons.

Photons hitting a building facade are encoded in per-block facade photon maps. We use two photon maps for each city block, one for exterior facades and one for interior ones. The mapping from object space facade positions to texture space coordinates is achieved through the implicit double cylindrical parametrization proposed by Argudo et al. [AAP12]. The interior and exterior facade photon maps are arranged into a small collection of texture atlases. See Figure 3.

All photon maps are generated on-demand, taking into account only the most influencial subset of light sources. City lights are distributed in a regular grid. In our experiments, we used 40 × 40 m cells, resulting in about 5 lights per cell.

At runtime, whenever the camera's height is below a user defined threshold, we query the grid to determine the subset of street lights to be computed through photon mapping.

The contribution of each street light to the photon maps proceeds as follows. We first emit photons in random directions and trace them through the scene (using NVIDIA Optix's engine). The initial energy should represent the characteristics of the light. For this reason, the initial energy of each photon p is set in accordance to its direction so that its energy matches that used in the aerial approach. The behavior of the method is different according to the material of the surface hit x. When the photon hits a diffuse surface, we store the hit in an intermediate hit buffer. When the photon hits a specular surface, the ray is reflected using the normal of the surface, and tracing continues. The radiance energy is attenuated using a Fresnel coefficient.

After the photons have been traced through the scene, we transfer the information in the auxiliary hit buffer to the corresponding floor and facade photon maps through splatting. In both cases (floor and facades) we keep constant the size of the photon influence region.

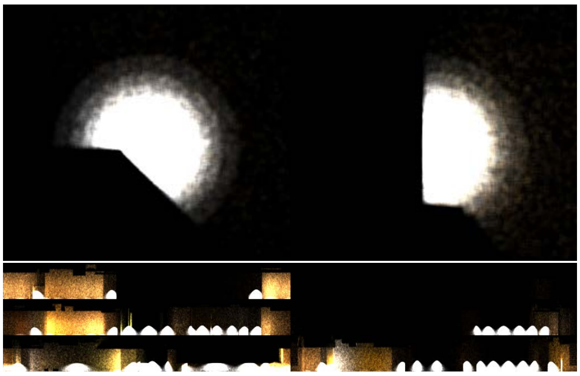

Figure 3: Photon map examples. Floor photon maps (top), and facade photon maps (bottom).

The last step is to use the photon map to compute the illumination. This is done by checking, for every pixel, if it is in the photon map, and then use the information stored (energy density) to compute its illumination.

It is important to notice that these high-resolution photon maps are computed for each light source. As it is impossible to predict what street lights may be computed at each moment, we build atlases with the respective photon maps $\mathcal{C}_x$, and use them as a cache with a *Last Recently Used* (LRU) policy. When a light is chosen from the grid, it is only computed if it is not in the floor photon map cache.

3.4. Push-pull integration

The last part of our proposed method is the integration between the aerial and close-view approaches so that each method takes advantage of the information gathered by the other one. Transferring information from the aerial approach to photon mapping consists of fixing the initial energy of the photons before being traced. To transfer information to the aerial approach, we propose new structures to keep the old information (aerial) separated from the new one (photon maps). These structures are two new light maps $\mathcal{I}_x$: one for the floor and one for the facades. They store a reduced-resolution version of the photon maps $\mathcal{P}_i$ covering the whole city. The improved floor light map is generated drawing quads textured with the floor cache information, downsampled through mipmapping. The improved facade light map consists of a texture atlas of buildings, which is textured with the facade cache information each time an element is generated or updated, again downsampled with mipmapping. The information of each light is removed from the original lightmap $\mathcal{L}$ because it is not longer useful when the improved version is computed.

After the information exchange between aerial and close-view approaches, the final solution uses both and their improved versions to compute the final image. For each point, its illumination might be represented up to three times: in the lightmap $\mathcal{L}$, in the improved light map $\mathcal{I}_x$ and in the photon map cache $\mathcal{C}_x$. So, we downsample the cache again and subtract this energy from the improved light map. Thus, the final evaluation of the illumination of a point can be described as: $E_{total}(p) = \sum_{i\in\mathcal{L}} l_i^{\mathcal{L}}(p) + \sum_{i\in\mathcal{I}_x, i\notin\mathcal{C}_x} l_i^{\mathcal{I}_x}(p) + \sum_{i\in\mathcal{C}_x} l_i^{\mathcal{C}_x}(p)$ where $l_i(p)$ represents the evaluation for one street light in the point p. The cache $\mathcal{C}_x$ contains the more recently computed street lights, so we use the accurate photon map energy for them. For all the other street lights, we use the information of the light map $\mathcal{L}$ or the improved light map $\mathcal{I}$ according to whether they have been computed or not. Therefore the irradiance of each pixel will be computed as the sum of the irradiance stored in the light map, the one stored in the photon map and the one in the improved light map.

4. Results and Discussion

We tested our approach on a city model from Tele Atlas with 166 blocks and 141K triangles. Figure 4 shows the performance during an extensive nagigation with an Intel Core i5-2400 CPU with 4 GB memory and an NVIDIA GeForce GTX 460 GPU with 1 GB memory. For the close-view approach we traced 75K photons to obtain high quality images with low computational costs. For aerial views we get over 200 fps, falling to 15-20 fps when multiple cache updates are done.

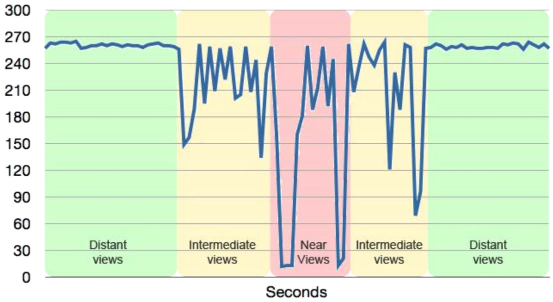

Figure 4: Runtime performance.

In conclusion, our technique computes the nocturnal illumination of a city in real time, which involves the computation of a high number of street lights. We have divided the problem in two cases, offering an approximate solution for far views, and a physically-based realistic solution for near views. Furthermore, their correct integration provides a unified solution with a smooth transition. The technique is not free from limitations, though, like being the cache update a bottleneck when there are too many misses.

Acknowledgements

The authors would like to thank Oscar Argudo for his valuable support. This work was partially funded by grants TIN2010-20590-C01-01 and TIN2010-20590-C02-02 from *Ministerio de Ciencia e Innovación*, Spain.

References

[AAP12] ARGUDO O., ANDÚJAR C., PATOW G.: Interactive rendering of urban models with global illumination. In *Computer Graphics International* (Bournemouth University, United Kingdom, June 2012). 1, 2, 3

[ABCN10] ANDUJAR C., BRUNET P., CHICA A., NAVAZO I.: Visualization of large-scale urban models through multi-level relief impostors. *Computer Graphics Forum 29*, 8 (2010), 2456–2468. 1, 2

[CBG*07] CIGNONI P., BENEDETTO M. D., GANOVELLI F., GOBBETTI E., MARTON F., SCOPIGNO R.: Ray-casted blockmaps for large urban models visualization. *Computer Graphics Forum 26*, 3 (2007), 405–413. 1, 2

[DB05] DÖLLNER J., BUCHHOLZ H.: Continuous level-of-detail modeling of buildings in 3d city models. In *Proc. of the ACM international workshop on geographic information systems* (2005), GIS'05, pp. 173–181. 2

[DBCG*09] DI BENEDETTO M., CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., SCOPIGNO R.: Interactive remote exploration of massive cityscapes. In *Proc. of the International Symposium on Virtual Reality, Archaeology and Cultural Heritage* (2009). 1, 2

[HW10] HAUNERT J.-H., WOLFF A.: Optimal and topologically safe simplification of building footprints. In *Proc. of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2010), GIS'10, pp. 192–201. 2

[ML09] MCGUIRE M., LUEBKE D.: Hardware-accelerated global illumination by image space photon mapping. In *Proceedings of the Conference on High Performance Graphics 2009* (New York, NY, USA, 2009), HPG '09, ACM, pp. 77–89. 3

[MS95] MACIEL P. W. C., SHIRLEY P.: Visual navigation of large environments using textured clusters. In *Proc. of the 1995 symposium on Interactive 3D graphics* (1995), I3D'95, pp. 95–ff. 2

[PCR11] POULI T., CUNNINGHAM D. W., REINHARD E.: A survey of image statistics relevant to computer graphics. *Comput. Graph. Forum* (2011), 1761–1788. 2

[SDB97] SILLION F. X., DRETTAKIS G., BODELET B.: Efficient impostor manipulation for real-time visualization of urban scenery. *Computer Graphics Forum 16*, 3 (1997), 207–218. 2

[VAW*10] VANEGAS C. A., ALIAGA D. G., WONKA P., MÃĲLLER P., WADDELL P., WATSON B.: Modelling the appearance and behaviour of urban spaces. *Computer Graphics Forum 29*, 1 (2010), 25–42. 2

[WWS01] WIMMER M., WONKA P., SILLION F.: Point-based impostors for real-time visualization. In *Proc. of the Eurographics Workshop on Rendering 2001: London, United Kingdom, June 25-27, 2001* (2001), Springer Verlag Wien, pp. 163–176. 2

[YZM*11] YANG L., ZHANG L., MA J., XIE J., LIU L.: Interactive visualization of multi-resolution urban building models considering spatial cognition. *International Journal of Geographical Information Science 25* (February 2011), 5–24. 2