

LJ-1

# 天空のアダプティブサンプリングによる 天空光を考慮した屋外景観画像の高速生成

A Fast Method for Rendering Outdoor Scenes  
with Skylight Illuminance by Adaptive Sampling of The Sky

高田 信太郎\* 土橋 宜典\* 山本 強\*

Shintaro Takada Yoshinori Dobashi Tsuyoshi Yamamoto

## 1 はじめに

近年、コンピュータグラフィックス (CG) を用いた屋外景観画像が作成されている。CG を用いることによって、天候や太陽位置に応じた建築物の見え方を事前に評価することができる。この際、リアルな屋外景観画像を作成するためには、太陽光だけでなく天空光によって照射された物体の輝度計算をする必要がある。しかし、天空光による照度計算は非常に多くの計算時間を必要とする点が問題となっている。

本稿では、天空光による照度を高速に計算する手法を提案する。提案手法はグラフィックスハードウェアを利用して画像生成を行う。グラフィックスハードウェアは著しく高性能化しており大幅な計算時間の短縮が実現できる。また、提案手法では天空をアダプティブにサンプリングすることによってサンプル数を抑え、かつ LOD (Level of Detail) の考え方を利用することで計算コストの削減を図る。提案手法を用いることによって、都市景観画像のような大容量のポリゴンデータを扱う際にも高速に画像を作成することができる。

## 2 天空光による照度計算のモデル

天空光は地上を覆う極めて大きな半径を持った半球状の光源と考えることができる [1]。ただし、半球状光源の輝度分布は一樣ではない。図 1 に示すように、半球を  $(\theta, \phi)$  方向の小さな領域 (天空要素と呼ぶ) に分割し、この微少な天空要素を用いて天空輝度分布  $L$  を半球にわたって積分することにより、計算点  $P$  の照度  $I$  は次式のように求められる。

$$I = \int_{\theta} \int_{\phi} H(\theta, \phi) L(\theta, \phi) \cos \theta \sin \theta d\theta d\phi \quad (1)$$

ここで、 $H$  は影の有無を表す関数であり、計算点  $P$  から天空要素が可視であれば 1 を返し、不可視であれば 0 を与える関数である。

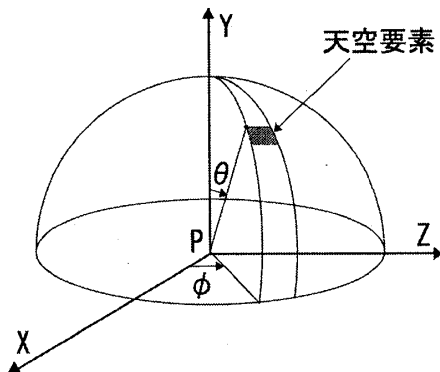


図 1 天空要素の位置関係

## 3 グラフィックスハードウェアを利用した画像生成

### 3.1 基本的な考え方

式 (1) で表される照度  $I$  を解析的に計算するのは困難であるため、数値積分によって計算する。すなわち、計算点  $P$  の照度  $E$  は次式で計算される。

$$I = \sum_i \sum_j H(\theta_i, \phi_j) L(\theta_i, \phi_j) \times \cos \theta_i \sin \theta_i \Delta \theta \Delta \phi \quad (2)$$

ここで  $\Delta \theta$ 、 $\Delta \phi$  はサンプリング間隔を表す。天空要素は微少であるため、天空要素内では輝度分布一定と仮定し、天空要素を一つの点光源で近似する。こうすることでグラフィックスハードウェアを利用して物体の輝度計算を高速に行うことが可能となる。また、 $H$  の計算についてもグラフィックスハードウェアを用いたシャドウマッピングが利用可能であり、任意の点での  $H$  の値を高速に求めることができる [2]。天空輝度分布  $L$  については経験式を用いる [1]。

サンプリング間隔については、サンプリング間隔が大きい (天空要素への分割数が少ない) 場合は物体の影を十分にサンプリングできずエイリアシング問題が生じる。これを解決する最も単純な方法はサンプリング間隔を小さくすることであるが、単純に分割数を増加すると天空要素の数に比例して計算時間が著しく増加してしまう。

太陽の周辺は輝度が急激に変化するため、細かくサンプリングしなければ十分な精度が得られない。一方、輝度の小さい領域では少ないサンプル数で十分である。そこで、精度良くサンプリングするために、輝度分布に応じてアダプティブにサンプル点を配置する。すなわち、ある天空要素  $l$  からの照度を  $I_l$  とすると、 $I_l$  がある閾値  $\epsilon$  よりも小さくなるようにアダプティブにサンプリングをする (図 2 参照)。こうすることで、照度が高い太陽の周辺がより細かくサンプリングされ、単純に分割数を増加するよりも少ないサンプル数に抑えることが可能となり、計算コストを削減することができる。

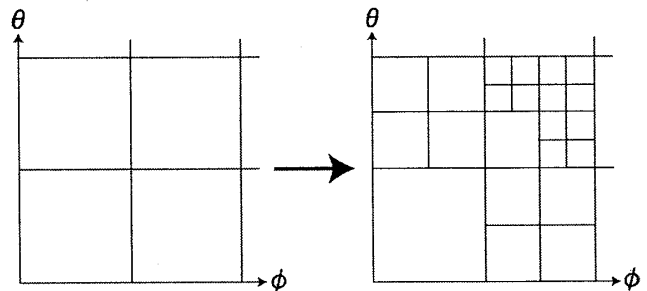


図 2 アダプティブサンプリング

\*北海道大学大学院工学研究科

天空要素への分割が完了した後に、それぞれの天空要素に点光源を配置して画像を描画し、グラフィックスハードウェアの機能であるアルファブレンディング (本稿では単純に二つの色を足し合わせる機能を利用する) によりフレームバッファにその色を足し込むことで、画像を生成することができる。

### 3.2 LOD の考え方を利用した画像生成

屋外景観のような大量のポリゴンにより構成されるデータから画像生成をする場合は、天空光による照度計算は非常に計算コストがかかる。そこで、LOD の考え方を利用してさらなる計算コストの削減を図る。

透視投影で画像を作成する場合、視点から近くにある物体は大きく見え、遠方にある物体は小さく見える。そのため、基本的に遠方の物体ほどスクリーンへ投影される面積は小さくなり視覚的な影響は小さくなっていく。これを考慮して、視覚的な影響が小さい物体を描画する際は、多少の照度計算の誤差を許容しサンプル点を少なくすることで計算コストを減らすことを考える。

前節の方法により、天空は  $n$  個の天空要素に分割されているとする。また、天空要素は照度の値が大きい順に並べられているとする。いま、ある物体  $i$  を考える。  $k$  番目の天空要素により計算した照度を  $e_k$  とすると、  $k$  番目までの天空要素から計算した照度  $E_k$  は以下のような式で表せる。

$$E_k = \sum_{p=1}^k e_p \quad (3)$$

すべての天空要素から計算した照度を  $E_{all}$  とし、  $k$  番目までの天空要素で照度計算をした場合の計算誤差を  $R_i(k)$  と表すと、  $R_i(k)$  は以下のような式で表せる。

$$R_i(k) = E_{all} - E_k \quad (4)$$

このとき、  $R_i(k) > R_i(k+1)$  を満たす。また、スクリーン上のオブジェクトの面積は、視点とオブジェクトの距離に反比例し、オブジェクトの大きさに比例する。よってオブジェクト毎に次式でスクリーン上での見かけの誤差を評価する。

$$R_{si}(k) = \alpha R_i(k) \cdot \frac{M_i}{d_i} \quad (k = 1, 2, \dots, n) \quad (5)$$

ここで、  $R_{si}(k)$  はスクリーン上での見かけの誤差、  $\alpha$  は比例係数である。  $d_i$  は物体  $i$  と視点との距離、  $M_i$  は物体  $i$  のスクリーン上での大きさである。  $d_i$  および  $M_i$  は以下のように近似的に求める。それぞれのオブジェクト毎に bounding box を求めその中心と視点の距離を  $d_i$  とし、出力画像にはオブジェクトの表の面が描画されると仮定して bounding box の表面積の  $1/2$  を  $M_i$  とする。提案手法では、閾値  $\mu$  を設定し  $R_{si}(k) < \mu$  を満たすサンプル数  $k$  を決定する。すなわち、物体  $i$  に対して照度の計算に必要な天空要素の数  $k_i$  を決定する。そして、このサンプル数  $k_i$  で各物体を描画することで画像を生成する。

レンダリングは前節で述べたハードウェアを利用した方法を利用し、各天空要素で画像を描画しそれらを足し込むことで画像を生成する。そのため、前節で述べたように物体毎のループで描画するのではなく、天空要素のループで 1 番目の天空要素から順に処理を行う必要がある。そこで、  $j$  番目の天空要素に対する画像を生成する際、  $j < k_i$  を満たす物体のみを描画する。これにより、各天空要素に対してすべての物体を描画するのではなく一部の物体を選んで描画するため、計算コストを削減することができる。

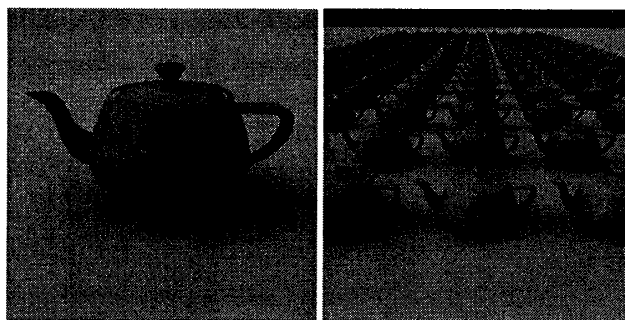


図3 作成された画像

## 4 適用例

簡単な例を用いて、提案手法の有効性の検討を行った。まず、天空を等間隔にサンプリングして作成した画像と、アダプティブにサンプリングして作成した画像の比較実験を行った。前者では、アダプティブにサンプリングした際の最も細かいサンプリング間隔で等間隔にサンプリングした。図3の左の画像はアダプティブにサンプリングして作成した画像である。計算時間は、前者では24.3秒であるのに対し、後者では5.5秒であった。差分画像を求め、その平均輝度値を求めたところ4.6であった。このことから、アダプティブにサンプリングすることによって、ほぼ同程度の画質の画像を約  $1/4$  の速度で作成できた。

次に、図3の右の画像のように大量のポリゴンにより構成されるデータに対して、アダプティブサンプリングで作成した画像と、それに加えてLODの考え方を利用した方法で作成した画像の比較実験を行った。計算時間は、前者では344秒であるのに対し、後者では127秒であった。差分画像を求め、その平均輝度値を求めたところ7.2であった。このことから、LODの考え方を利用することによって、ほぼ同程度の画質の画像を約  $1/3$  の速度で作成できた。これらのことから、提案手法は速度の面で非常に有用である。

計算機はCPUがAthlonXP1700+であり、グラフィックスハードウェアとしてNVIDIA社のGeForce3を搭載したPCを用いた。画像サイズは  $512 \times 512$  である。

## 5 まとめ

高速に屋外景観画像を生成する手法を提案した。提案手法では、天空をアダプティブにサンプリングすることによって、単純に等間隔でサンプリングするよりも計算コストを抑えて高速化を実現した。また、LODの考え方を利用することによって、視覚的に影響の少ない物体に対して天空のサンプル数を削減することでさらに計算コストの削減を図った。

今後の課題として、さらなる高速化、鏡面反射成分を考慮した高速画像生成方法の開発が挙げられる。

## 参考文献

- [1] T.Nishita, E.Nakamae, "Continuous Tone Representation of Three-Dimensional Objects Illuminated by Sky Light", Computer Graphics, 20, No.4, pp.125-132 (1986)
- [2] M.Segal, C.korobkin, R.V.Widenfelt, J.Foran, P.E.Haeberli, "Fast Shadows and Lighting Effects Using Texture Mapping", Computer Graphics, 26, No.2, pp.249-252 (1992)