

OpenGL 環境下における全方向環境マッピング

7P-6

青野雅樹、大淵竜太郎、清水和哉
日本アイ・ビー・エム (株) 東京基礎研究所

1 はじめに

OpenGL [3] は、三次元グラフィックス API としてワークステーションから PC まで幅広く用いられており、レンダリングとくにテクスチャマッピングの機能は豊富である。本報告では、このうち環境マッピングに焦点をあて、現状での問題点を指摘し、OpenGL というグラフィックス API の範囲内でその解決案を提示するとともに、テスト結果を述べる。

2 環境マッピングとその問題点

環境マッピングは、テクスチャマッピングの一手法で、物体の頂点に付随する静的なテクスチャ座標でテクスチャをアクセスするかわりに、視点の動きとともに頂点ごとに動的に決まる反射ベクトルでテクスチャをアクセスすることで、グローバルな反射効果をシミュレーションしようとするものである [2]。

環境マッピングに使用するテクスチャは、「環境マップ」と呼ばれるテクスチャで、通常空間内のある一点から見た周囲の「環境」を記述したものである。OpenGL で採用されている「環境マップ」は、魚眼レンズで撮影した画像に相当する「球面マップ」と呼ばれる形態の画像である。球面マップは、ほぼ半球領域に対する環境をカバーする。この他に、QuickTimeVR [1] で用いられている「円柱マップ」、RenderMan [4] で用いられている「キューブマップ」や「メルカトルマップ」などが知られている。

環境マップはその簡便さと高速性のため、レイトレーシングの代用としてよく用いられる。しかし、環境マップは空間のある点の回りで作られたものなのでその点から離れるに従って歪みが大きくなり、グローバルな隠蔽関係も不正確になる、という本質的な問題がある。

特に OpenGL のもとでの（球面によるテクスチャ座標が自動生成される）環境マッピングには次のような別の問題がある。

- 全方向（立体角度 720° ）をカバーできない。
- 球の赤道付近で歪みが大きい。
- 反射ベクトルが視点座標系でしか生成されない。

3 全方向環境マッピング

OpenGL 環境下という制限のもとでのわれわれの全方向マッピングは、以下の二つのステップから成る。

第一ステップでは、全立体角方向 (720°) に渡ってカバーするような環境マップを作る。われわれがインプリメントしたのは、六枚のキューブマップと六枚の球面マップを使うものである。（4節以降では、前者の手法のみ述べる。）

第二ステップでは、OpenGL のテクスチャ座標自動生成機能に頼らずに、環境マップのテクスチャがマップされる図形の各頂点のテクスチャ座標を、視点の移動に合わせて動的に生成していく。全立体角をカバーする環境マップがある限り、欲するテクスチャ座標はどこかにあるはずである。問題は、二枚以上の環境マップをマージした形のテクスチャを用いるため、最小図形要素（通常は三角形）の全頂点のテクスチャ座標が同一の環境マップに含まれず、複数の環境マップにまたがる場合があることである。

4 テクスチャ座標の動的生成

簡単のため、最小図形要素を三角形とし、六枚の環境マップをマージした形態のデータ（キューブマップ）を用いて、動的なテクスチャ座標生成による環境マップの方法を説明する。キューブマップを用いた理由は、その他の全方向環境マップに比べて歪みが小さいからである。原理は下図に示す通りである。

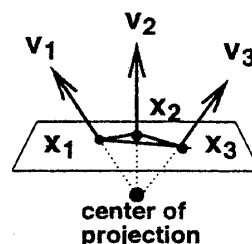


図 1: キューブマップによるテクスチャ座標の決定

三角形の各頂点での反射ベクトルをそれぞれ $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ とする。これらが、単位キューブの上（たとえば y 軸方向の面）と $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ で交わったとする。このとき、これらの点の対応するキューブマップ上のテクスチャ座標が求める値である。ここで反射ベクトル $\mathbf{v}_i (i = 1, 2, 3)$ は次式で与えられる。

$$\mathbf{v}_i = 2(\mathbf{n}_i \cdot \mathbf{I}_i)\mathbf{n}_i - \mathbf{I}_i$$

ただし $\mathbf{n}_i$ は点 $\mathbf{x}_i$ での単位法線ベクトルで、 $\mathbf{I}_i$ は

$$\mathbf{I}_i = (\mathbf{e} - \mathbf{x}_i) / |\mathbf{e} - \mathbf{x}_i|$$

で与えられ、 $\mathbf{e}$ は視点を表す。

キューブマップにおける問題は、図2のような場合である。

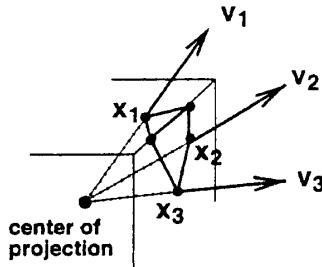


図 2: キューブマップ上でのクリッピング

この場合、点 $\mathbf{x}_1$ が含まれる面と残りの二点が含まれる面が同一でない。このように複数画像にまたがる場合、六枚の環境マップがマージされたキューブマップ内で隣接した画像になっている保証はない。このため、三角形の各頂点の反射ベクトルを単位キューブでクリッピングして、それぞれの環境マップ内に収まる多角形に分割する必要がある。ここにあげた例では、図2に示すように二つのポリゴンに分割する必要がある。各環境マップの境界線上のテクスチャ座標値は、環境マップの投影中心と二つの点（たとえば図1の点 $\mathbf{x}_1$ と $\mathbf{x}_2$ ）を通る平面を求め、これと単位キューブ上の対応する辺との交点計算で求めればよい。三つの反射ベクトルが全部異なる単位キューブ面と交わる場合も同様な処理をして三つのポリゴンに分割する。

5 テスト結果

六面に異なるテクスチャの張られた四角い部屋のデータを作成し、中央に視点を置いてキューブマップを生成した。図3が六枚の環境マップを作成したときの座標系を示し、図4がこれら六枚からなるキューブマップを表す。

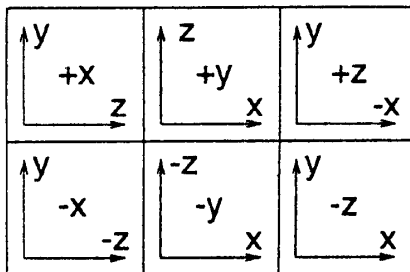


図 3: キューブマップの座標系

この部屋に、視点を中心とする鏡面球を置き、本手法の環境マップを適用した。図5(a)が本手法による環境マップを OpenGL 環境下で適用したもので、(b)が同じ視点でレイトレーシングを適用したものである。

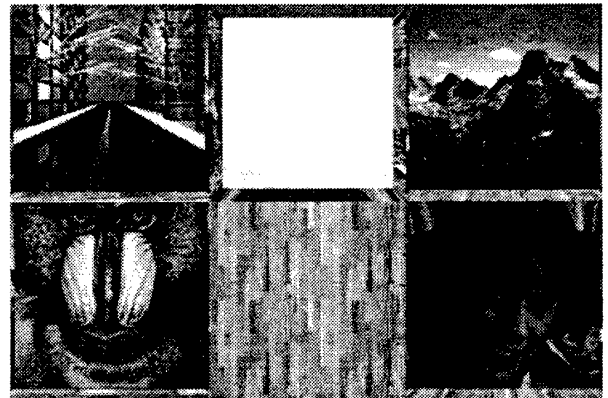
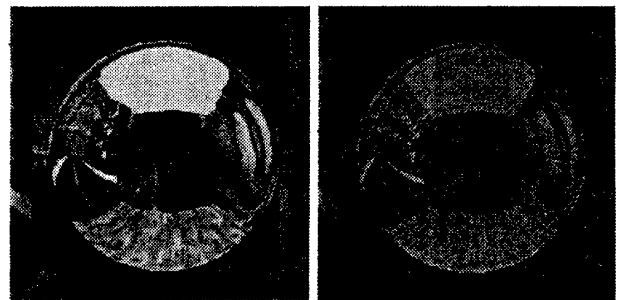


図 4: キューブマップの例



(a) 本手法の適用例

(b) レイトレーシング例

図 5: 環境マッピングのテスト

6 おわりに

全立体角方向の環境マップを一枚の画像（キューブマップ）で保持し、マップされる物体の各頂点での動的なテクスチャ座標を生成することで、OpenGL のもとでも視点の動きとともに、映り込んだ画像も動くようにできた。今後改良すべきものとして、レイトレーシングに匹敵する高品位化が望まれる。

参考文献

- [1] S. E. Chen, "QuickTime™ VR - An Image-Based Approach to Virtual Environment Navigation," *Proc. SIGGRAPH '95*, 29-38, 1995.
- [2] N. Greene, "Environment Mapping and Other Applications of World Projections," *IEEE Computer Graphics and Applications*, Vol. 6, No. 11, 21-29, November 1986.
- [3] J. Neider, et al., *OpenGL™ Programming Guide*, Addison-Wesley Publishing Co., 1993.
- [4] S. Upstill, *The RenderMan™ Companion*, Addison-Wesley Publishing Co., 1992.