

LJ-4

3D グラフィックアクセラレータを利用した
リアルタイムシームレスモザイク

A Study on Real-time Seamless Mosaicking Using 3D Graphics Accelerator

土居意弘*

山本強*

DOI Munehiro YAMAMOTO Tsuyoshi

1 はじめに

デジタル映像が本格的に普及し始めるに伴い、さまざまな新しい映像表現技法が研究されているが、その一つに全周動画画像システムがある。

我々は、新しい全周動画画像システムの開発を行っている [1],[2]。4つの広角ビデオカメラで全周を撮影して全周映像を合成するイメージモザイク型のシステムであるが、変換処理をクライアント側で行うため伝送までの前処理が不要で、さらに4分割装置を用いて4つの映像信号を1つのNTSC信号として送信することで、ライブ送信が容易にしたという特徴がある。

我々のシステムにおいては、再生クライアントでのモザイク処理を高速に行う技術が特に重要である。ビデオレートでのモザイク処理およびレンダリングを行うため、モザイク処理の大半をテクスチャマッピングに置き換えることで高速に行う方法を提案する。テクスチャマッピングは、多くのPCが標準的に備える3Dグラフィックスアクセラレータによって非常に高速に行われるため、滑らかな全周動画画像の再生が可能である。

2 リアルタイムシームレスモザイク

2.1 モザイク処理

パノラマ映像の撮影に関しては従来から非常に多くの研究がある。特に近年は、通常のデジタルカメラで撮影した多数の画像をつなぎ合わせてパノラマ画像を得る「イメージモザイク」といわれる手法が、イメージベースレンダリングの進展とともに多く用いられるようになってきた [3],[4]。この手法は、双曲面ミラーなどの特殊な光学系を用いる手法に比べて機器が安価で入手性が良く、また合成処理も一般のPCで行うことができるため手軽に行えるというメリットがある。

モザイク処理は、継ぎ目位置の判定、パノラマ座標系への変換と継ぎ目処理の3つの部分に分けられる。このうち継ぎ目位置の判定は、我々のシステムではカメラが固定されているために、再生中に変化しないと考えるため予め計算しておくものとする。また広角カメラを使用しているために生ずる歪みも補正する必要があるが、パラメータは固定しているものとする。

以下、座標変換と継ぎ目処理と、それらをテクスチャマッピングを用いて実行する方法について述べる。

2.2 座標変換

4つのカメラは90°ごとに回転した視野方向の画角左右55°程度を撮影しており、各カメラの映像を入力フレームに4分割配置する。例えば $\phi = 0^\circ \pm 55^\circ$ の映像は左上に、 $\phi = 90^\circ \pm 55^\circ$ は右下、以下同様に左下、右上の順に配置されている(図1(a))。

これに視点の真上からの水平面へ下ろした天頂角 θ を組み合わせて、再生システムに入力された4分割画像フレームと、レンダリング先の投影空間とのマッピングを行う。すなわち、マッピング関数は、投影空間座標 $\mathbf{w} = (\theta, \phi)$ に対し、フレーム座標 $\mathbf{p} = (u, v)$ を与えるものとなる。

しかし、これでは極座標とフレーム座標が一対一の関係にならない。例えば $\phi = 50^\circ$ 付近の映像は左上の画像と右下の画像に含まれる。そこで、カメラ番号 n を加え、投影空間の極座標 \mathbf{w} 、入力フレーム座標 \mathbf{p} に対して、 $\mathbf{p} = C(n, \mathbf{w})$ とすることで一意になるようにする。さらに広角レンズによる歪み関数 [5] $\mathbf{p}' = D(\mathbf{p})$ (\mathbf{p} はフレーム内座標、 \mathbf{p}' は歪んだフレーム内座標)を掛け合わせる。マッピング関数は最終的に次の形となる。

$$\begin{aligned} \mathbf{p}' &= D(C(n, \mathbf{w})) \\ &= M(n, \mathbf{w}). \end{aligned} \quad (1)$$

これをフレームに配置したのが図1(a)である。

2.3 継ぎ目の除去

投影空間にレンダリングする際には、隣接するカメラと重複する部分を違和感なくつなぎ合わせる必要がある。このためには、継ぎ目の位置を正しく合わせるだけでは不十分で、各カメラごとの自動露出補正に伴う輝度・色の違いを調整しなければならない。高いコストを掛けずに効果的な継ぎ目除去を行うため、クロスフェードを利用する。

隣接するカメラ n_l, n_r の重なる部分は、分割画像それぞれの左右両側の端にのみ存在すると考えてよく、 ϕ だけで決められる。フレーム位置 \mathbf{p}' のRGB値を $\mathbf{c} = I(\mathbf{p}') = I(n, \phi)$ とすると、重なり範囲 $\phi_1 \leq \phi \leq \phi_2$ に関して、 $\alpha(\phi) = (\phi_2 - \phi) / (\phi_2 - \phi_1)$ なる関数を用いると、最終的な出力となるRGB値 $I'(\phi)$ は次のようになる。

$$I'(\phi) = \alpha(\phi)I(n_l, \phi) + (1 - \alpha(\phi))I(n_r, \phi). \quad (2)$$

2.4 テクスチャマッピング

(1)式で表される座標変換と、(2)式で表されるクロスフェードはテクスチャマッピングを利用して行う。

投影空間のポリゴン(図1(b))に対し、フレーム画像(図1(a))をテクスチャマッピングする。関数 M は一般に非線形な連続関数であるのに対しマッピングは離散的になるため、幾何学的変換はポリゴン単位のアフィン変換になってしまうが、ポリゴンの「網」を細かくすればより正確な近似となる。重なりのない部分ではフレーム画素をそのまま、重なりのある部分は2つのフレーム画素を(2)式に従ってアルファブレンディングを行うが、マルチテクスチャを用いることで重なりのあるなしを区別することなくまとめて実行できる。アフィン変換やブレンディ

*北海道大学大学院工学研究科, 札幌市

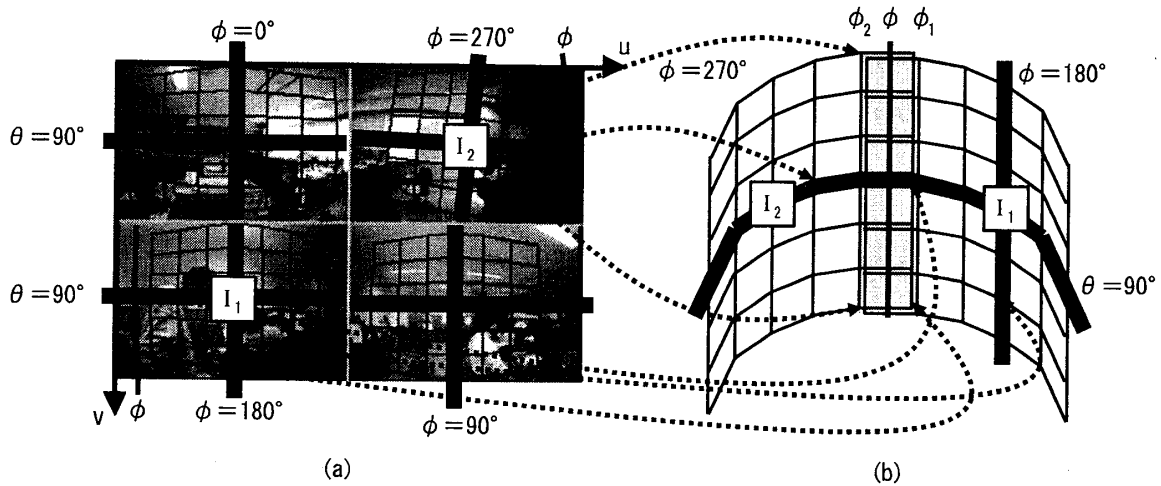


図 1 (a) 座標変換メッシュを配置した入力フレーム画像と (b) 投影空間へのマッピング

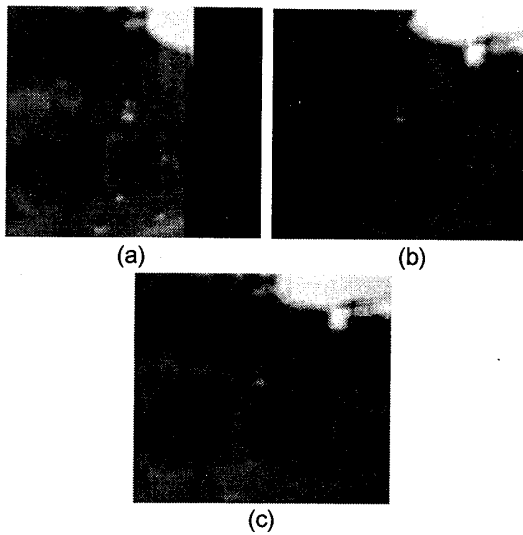


図 2 継ぎ目合わせの例. (a) 左画像, (b) 右画像, (c) 合成画像

ングは、テクスチャマッピングに伴う線形補間が自動的に計算するため、すべての画素に対して α 値やアフィン変換を計算することはまったく不要である。また、入力画像は 4 分割画像された一枚のフレームであるため、投影座標系全体のマッピングを一度のパスで実行することができる。これらの作業は、3D アクセラレータも利用することで非常に高速に行うことができる。結果を図 2 に示す。

3 実験と考察

本手法を、Windows2000 上で Direct3D8API を用いて実装した。実験には、720 × 480 ピクセルの 30fps の動画を DV・MPEG 形式の 2 通りで圧縮したファイルを用いた。いくつかの PC 上で再生した結果を表 1 に示す。

PC によっては、コマ落ちが生じたり継ぎ目の除去が行われなかったりすることがあった。コマ落ちは CPU が遅いほど、また MPEG のようにエンコード手法が複雑なほど増えるため、主な原因はデコード処理の時間であると考えられる。一方、継

CPU	3D アクセラレータ (発売年)	平均 fps	継ぎ目
Pen4 2.0GHz	Radion8500('01)	30.0/30.0	良
Pen4 1.7GHz	GeForce256('99)	30.0/30.0	良
Pen3 930MHz	GeForce2MX('00)	30.0/15.9	良
Duron 1.0GHz	VoodooBanshee('98)	30.0/16.6	不良

※ Pen=Pentium, 平均 fps は DV/MPEG の値。

※ 3D 性能はほぼ発売年ごとに向上していると考えてよい。

表 1 再生環境と性能

ぎ目の除去の良・不良は、3D アクセラレータのアルファブレンド機能への対応の可否が原因と考えられる。

4 まとめ

本稿では、3D アクセラレータを利用した、モザイク処理の高速処理手法を提案した。

今後は、モザイクパラメータの推定を精度よく求める手法を含めた高速化手法についてさらに検討するとともに、画質の向上を目指したい。

参考文献

- [1] M.Doï and T.Yamamoto, "PanoVi: A Scalably Extensible Panoramic Movie System for a PC-Centered Environment," NICOGRAPH International 2002, pp.143-148, 2002
- [2] T.Yamamoto, M.Doï, "PanoVi: Panoramic Movie System for Real-Time Network Transmission," 2001 IEEE Fourth Workshop on MSP, pp.389-394, 2001
- [3] S.E.Chen, "QuickTime®VR— An Image-Based Approach to Virtual Environment Navigation," Computer Graphics Proc., pp.29-38, 1995.
- [4] R. Szeliski and H.Y. Shum, "Creating full view panoramic image mosaics and environment map," Computer Graphics Proc., pp.251-258, 1997.
- [5] H. S. Sawhney and R. Kumar, "True Multi-Image Alignment and its Application to Mosaicing and Lens Distortion Correction," CVPR, pp.450-456, 1997.