

## 視点依存を考慮した全方位画像列の圧縮手法 Compression of View Dependent Texture for Omni-directional Image

尾池 治郎†  
Jiro Oike

thanda oo†  
thanda oo

川崎 洋†  
Hiroshi Kawasaki

大沢 裕†  
Yutaka Ohsawa

### 1. はじめに

近年、都市空間を計算機内に仮想的に再現する研究が盛んに行われている。これを実現する方法として、大きく Model Based Rendering(MBR) と Image Based Rendering(IBR) の二つが知られている。

MBRは、ポリゴン等の3次元データとその表面のテクスチャ情報を用いることで、あらゆる見えを容易に生成することができる。しかし、都市空間において、街路樹や電柱など細かい形状やテクスチャ等の情報を全てモデル化することは極めて難しい。

一方IBRでは、撮影した画像を直接利用するため、モデル化の難しい複雑な物体であっても容易に描画することが可能であるが、再現する空間すべての画像データを保持せねばならず、データの容量的な問題がある。また、IBRで用いる画像データは近傍カメラにおいてほとんど変化がなく、一般に冗長である。そこで本稿では、この冗長性を考慮し、効率よく全方位画像を圧縮する手法を提案する。

### 2. 全方位画像列

#### 2.1 車載全方位画像

圧縮の対象とする全方位画像は、GPS、ジャイロセンサなどを搭載した車の屋根にカメラを載せ、都市空間を走ることによって取得した画像とする。また車は等速直線運動を仮定している。こうすることで3.1節で説明するEPIを生成することができ、このEPIを用いることで画像の統合、パターンへのトラッキングなどを効率良く行うことができる。ところが、実際の都市空間において道路は必ずしも直線ではなく、緩やかに歪曲していたり交差点では90°に曲がっている。そこで、このような場合には道路を直線セグメントに分けることで、本手法を適用するものとする。

#### 2.2 カメラ

都市空間のデータの撮影には、鏡面型全方位カメラを用いる方法と、複数カメラを用いる方法がある。

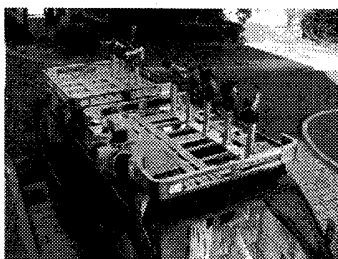


図1: 車載カメラ

†埼玉大学工学部, Saitama University

鏡面型全方位カメラは、一度に水平方向360°の画像を撮影することができる反面、解像度が低くなるため、現在のCCD密度では現実感の高い都市のモデリングに十分な解像度を得ることが難しい。

一方複数カメラを用いれば、解像度の問題は解決できるが、画像を統合することによる歪みが問題となる。しかし、この歪みについては多くの解決方法が提案されている[1]。特にカメラを図1のように一直線上に配置すればカメラの光学中心を一致させることができ、画像を統合する際に都合がよい[2]。そこで、本稿においてもデータの撮影にはこのカメラを用いた。

### 3. 画像の圧縮手法

画像列の中で同じ建物は図2(a)のように複数の連続したフレームに存在しており、冗長性が高い。そこで、以下に述べる手法で圧縮を行う。

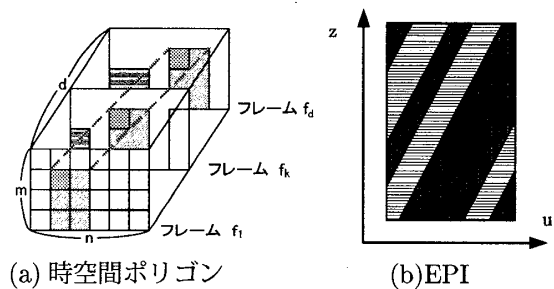


図2: 車載カメラの時空間画像

#### 3.1 EPI

等速直線運動する移動体に、カメラを進行方向に対して鉛直横向きに設置して撮影した連続画像を、時間軸方向に積層することで、時空間ポリゴンを作ることが出来る(図2(a))。この、時空間ポリゴンをエピポーラ線に平行に切断したものがEPI(Epipolar Plane Image)である。

図2(b)はEPIの例であり、 $u$ 軸は画像の水平方向(epipolar方向)を、 $z$ 軸は時間方向を表している。

EPI上での特徴点の動きは直線状の軌跡を描く。また、カメラに対するオブジェクトの奥行き値が大きく変化せず、表面のテクスチャが一様な場合には、オブジェクトはEPI上で『帯』状の軌跡を描く。さらに、複数のオブジェクトが、カメラの進行方向に対して平行に存在しているとすると、オブジェクトの奥行きに大きな変化がないため、図2(b)のように、それぞれのオブジェクトが作る帯は重なること無くほぼ平行な軌跡を描くことになる。

#### 3.2 ハフ変換を用いたトラッキング

EPIにおいて直線の傾きはオブジェクトの推移を表しているため、このパターンを抽出できればオブジェクトのトラッキングが可能となる。そこでEPIの直線の傾き

をハフ変換を用いて検出する。  $\rho$  を座標原点から直線へおろした垂線の長さ、  $\theta$  を垂線と  $x$  軸とのなす角とすると、EPI上の直線は

$$\rho = x \cos \theta + z \sin \theta \quad (1)$$

の式で表される。EPIのエッジ画像にハフ変換を適用し  $\theta, \rho$  のパラメータを求める。EPIにおいて  $z$  軸は時間方向を表すため、  $z$  軸方向に1ピクセル動くことは1フレーム変化することを意味する。よって、式(1)より

$$\frac{\delta x}{\delta z} = -\tan \theta \quad (2)$$

となり、1フレームでのオブジェクトの  $x$  方向の移動量は  $-\tan \theta$  となる。これを用いてオブジェクトのトラッキングが可能となる。

### 3.3 KL展開による圧縮

まず画像を図2のように  $m \times n$  の小さなブロック単位に分割する。このそれぞれのブロックに関して、  $d$  フレームまで3.2節のトラッキングを適用し、類似ブロックの集合を得る。図2(a)においては、点線で示してある。

こうして生成したブロックの集合をKL展開し、  $k$  次元の固有ベクトルに圧縮する。これを  $m \times n$  全てのブロックに対して行うことで画像列を圧縮することができる。この時、ブロックの集合は建物の同じ場所を視点を変えながら撮影したものであるため類似度が高く、高い圧縮率が期待出来る。

## 4. 実験

### 4.1 ハフ変換を用いたトラッキング

実際のEPIに対してハフ変換を用いて直線を検出した。図3にその結果を示す。赤い直線が検出した直線であり、傾きは  $1.10[\text{rad}]$  となった。これを用いたところ1フレームの移動量は  $1.96[\text{pixel}]$  となった。これを用いてトラッキングした結果を図5(a)~(c)に示す。少しずつずれているものの、正しくトラッキング出来ていることが分かる。

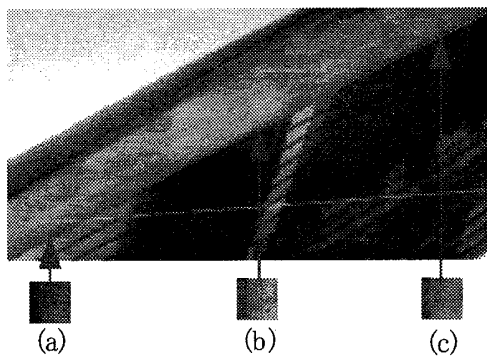


図3: ハフ変換による直線検出

### 4.2 圧縮結果

画像から切り取った  $96 \times 112[\text{pixel}]$  の画像列65枚を対象に、一つのブロックを  $16 \times 16[\text{pixel}]$  として圧縮を

行った。図4の(a)と(b)は、それぞれ最初と最後のフレームの圧縮前の画像である。図4の(c-1)及び(c-2)は、それぞれ50%と90%の寄与率を達成するために必要とした固有画像の枚数をグレースケールで表現したものである。暗い程枚数が少なく、明るい程多いことを示しており、(d)は実際に使用した枚数と色との対応関係を示している。具体的には(c-1)の平均は1.0枚であり、(c-2)の平均は15.1枚となった。それぞれの寄与率で画像を復元したものが(a-1)~(b-2)である。

寄与率50%ではブロックの集合を1つの固有ベクトルで復元しているため、木や建物の間がぼやけている。寄与率90%では元の画像とほぼ同じ画質で復元することが出来ているが、デプスを一つに仮定しているため、視点に依存して見え方が大きく変化するビル谷間や画像下の木の部分は圧縮率が悪いことが図(c-2)から分かる。

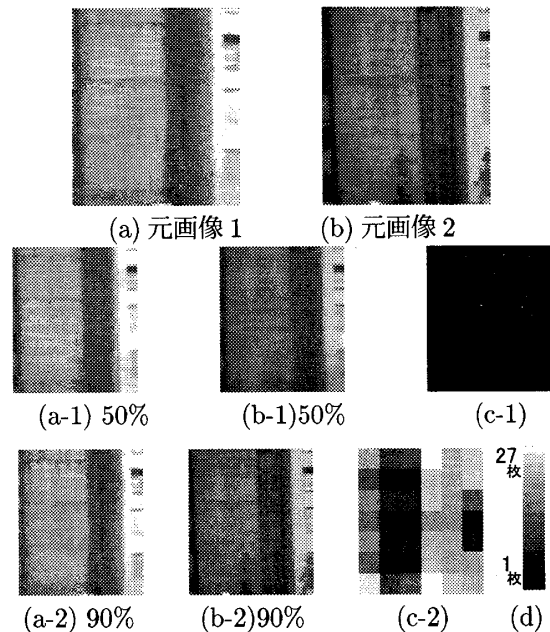


図4: 復元結果の画像

## 5. まとめ

本稿では視点依存を考慮した全方位画像列の圧縮手法について提案した。フレームのトラッキングは、EPIにハフ変換を適用し得られる傾きを用いた。トラッキングを用いて得られたブロックの集合は、同じ建物を視点を変えながら撮影したものであるため類似度が高く、効率良く圧縮することが出来た。

### 参考文献

- [1] 宮本充徳, 丸山達生, 尾池治郎, 川崎洋, 大澤裕. 複数カメラ画像合成のための画像平行化手法. *Forum on Information Technology(FIT2003)*, Vol. 3, pp. 5-6, 2003.
- [2] 小野晋太郎, 川崎洋, 池内克史, 坂内正夫. EPI解析による複数ビデオカメラの画像統合. *Computer Vision and Image Media 2003*, Jan. 2003.