自由に撮影した画像からの全周疑似ビデオ表示

坂本 雅俊　　金谷 健一

岡山大学大学院自然科学研究科

全方向を自由に撮影した少数枚の画像から，あたかも視点の 360° 周りをビデオカメラで撮影したかのような動画像を表示する手法を提案する．この表示手法を用いれば，計算機内部に全周画像や動画系列を生成・内蔵する必要はない．これは「有向射影幾何学」に基づいて，視点を移動するごとに個別の入力画像から直接にリアルタイムで画素を選んでディスプレイ上に写像するものであり，カメラや撮影状況の知識（焦点距離，撮影方向など）を必要としない．提案手法を用いれば，ユーザーが任意の視線移動を指定し，その方向に見えるシーンをインタラクティブに表示することができる．

Pseudo-Video Display of Surrounding Scenes from Freely Taken Images

Masatoshi Sakamoto and Kenichi Kanatani

Department of Computer Science, Okayama University, Okayama 700-8530 Japan

A new technique is presented for displaying an image sequence of the surrounding 360° scene taken as if by a video camera, using only a small number of freely taken images. We need not generate any 360° panoramic image or a motion sequences in advance or store them in the computer. Each time the viewing direction is specified, the pixel values to be displayed are directly chosen from input images using "oriented projective geometry" and mapped to the display real time. No knowledge is required about the camera or the shooting conditions such as the focal length and camera orientation. Our technique enables one to view the surrounding scene interactively by arbitrarily specifying the viewing direction.

1. まえがき

本論文では，ユーザーが観光地などで，手持ちのディジタルカメラで全方向を自由に撮影し，その個別の画像から，あたかも視点の 360° 周りをビデオカメラで撮影したかのような動画像を表示する手法を考える．もちろん初めからビデオカメラを携帯して，それで撮影すれば問題ないが，現在でもまだディジタルカメラのほうが安価，コンパクトで携帯に適している．

基本原理は非常に広い視野の画像（「パノラマ (画像)」と呼ばれる）を作成することであり，これには全方位レンズやカメラ回転装置のような専用の光学系や撮影機構を用いる方法 [17, 23, 24] と個別の画像を張り合わせて作る方法がある．後者は「(画像) モザイク生成」と呼ばれ，さまざまな手法が存在する．本論文では何らの特殊装置を用いないことを前提にするので，後者を用いる．

モザイク生成では隣接する画像が張り合わせ部分でよく合致するように，一方の画像に射影変換を施して他方の画像に張り合わせる．そして，そのための射影変換の計算法がいろいろ研究されている．しかし，単に張り合わせたのでは，視野が中心からが ±90° 方向に発散して，360° を表示することができない．

前報 [19] ではユーザが手持ちのデジタルカメラで周囲を撮影した画像を，視点の周りの仮想的な円筒面上に張り合わせる方法を示した（「円筒モザイク生成」）．そして,「有向射影幾何学」[21] を利用すれば，カメラや撮影方法に関する何らの情報も用いないでこれが実現されることを示した．さらに，張り合わせの誤差から，視線を 360° 回転すると最初の位置に一致しないという問題を解決するために，隣接する画像間の射影変換すべての合成が恒等変換になるという制約のもとで張り合わせを最適化する数学的な手法を提案した．

しかし，そのようにしてできた連続した仮想的な円筒面をそのまま展開して表示したり，それを移動しながら表示しても，ビデオカメラで撮影したかのような画像にはならない．ビデオカメラで撮影したような画像にするためには，視線を移動する度にその方向に見える円筒面上の画像を切り出し，それを

*700-8530 岡山市津島中 3–1–1, (086)251-8173
{samoto,kanatani}@suri.it.okayama-u.ac.jp

透視投影画像に変換して表示しなければならない．

したがって，個別の画像から全周の疑似ビデオ表示を行なうには，原理的には次の 2 段階を踏めばよい．

1. 個別の画像を仮想的な円筒面に張り付けて展開した円筒モザイク画像を作成する．
2. 視線移動を指定する度に，その視線方向に見える部分を円筒モザイク画像から切り出して透視投影画像に変換して表示する．

よく知られた QuickTime VR ™ [4] はこれを実現するものである．ただし，第 1 段階では機械的な回転制御装置によって画像を撮影して，それを円筒上に写像したものを内部に保持している．

しかし，よく考えると，前報 [19] の円筒モザイク生成は “計算” によって実現されていた．すなわち，360° の円筒モザイク画像上の各画素値は入力画像からコピーしたものであり，どの入力画像のどの画素をコピーするかを有向射影幾何学によって計算した．

一方，円筒面から透視投影画像を生成するのも “計算” である．すなわち，ディスプレイ上の各画素値を円筒モザイク画像からコピーするもので，どの画素をコピーするかを透視投影の幾何学によって計算すればよい．

とすれば，この 2 段階を合成した計算を行えば，円筒モザイク画像をあらかじめ作成する必要はなく，またそれを計算機内部に内蔵しておく必要もない．ディスプレイ上の各画素値を入力画像から直接にコピーすればよく，どの入力画像のどの画素をコピーするかを有向射影幾何学によってリアルタイムで計算しては表示すればよい．本論文ではこれを実現する方法を述べる．

さらに，前報 [19] の方法を補完する次の処理を行ない，より自然な表示を実現する．

- 最終画像と最初の画像間の不連を前報 [19] の方法で最適化すると，その不連続がすべての隣接画像間に分配される．これによって生じる不連続を補正する．
- 各画像を個別に撮影すると，シーン中を移動する物体（人や車）が隣接する画像の一方に写り，他方には写っていない場合がある．これを張り合わせると，継ぎ目でそのような物体が切断されることがある．これを補正する．

そして，提案手法の有効性を実画像例によって示す．

2. 全周のインタラクティブな表示

2.1 射影変換

画像フレームの中心を原点 O とし，x 軸を上方に，y 軸を右方にとる．そして，シーン中のある点が一つの画像上では点 (x, y) に，他の画像上では点 (x', y') に写っているとき，これらを次のベクトルで表す．

$$\boldsymbol{x} = \begin{pmatrix} x/f \\ y/f \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}' = \begin{pmatrix} x'/f \\ y'/f \\ 1 \end{pmatrix} \tag{1}$$

ただし，f は基準となる画像（以下,「基準画像」と呼ぶ．通常は画像系列の第 1 画像をとる）の撮影時における焦点距離であり，これは前報 [19] に示した射影変換の最適化の過程から計算できる（焦点距離を正しく推定しなければ全射影変換の合成が恒等変換にはならない）．シーンが遠方にある，あるいはカメラがレンズ中心の周りに回転すると[1]，2 画像は射影変換で結ばれ，次の関係が成り立つ．

$$\boldsymbol{x}' = Z[\boldsymbol{H}\boldsymbol{x}] \tag{2}$$

ただし，$Z[\cdot]$ は z 座標を 1 とするように定数倍する正規化であり，$\boldsymbol{H}$ は 2 画像間の射影変換の行列である．$\boldsymbol{H}$ には符号の不定性があるので，その (33) 要素が正[2]になるように符号を選ぶ[3]．そして，基準画像を 1 とし，隣接する順に 2, 3, 4, ... と番号をつけ，全周が閉じるように最適化した第 k 画像を第 $k+1$ 画像に写像する射影変換を $\boldsymbol{H}_{k(k+1)}$ とする．

2.2 表示領域の画素値の計算

ディスプレイ上に表示されるのは，ある仮想的な視点位置の回りに焦点距離 f を固定した仮想的なカメラを回転させて撮影したものであると解釈する（焦点距離を変えるには，単に表示された画像をそれに応じて拡大・縮小すればよい）．仮想カメラの現在の向きは基準画像の撮影位置から回転行列 $\boldsymbol{R}$ だけ回転したものであるとし，ディスプレイ上に表示領域をその仮想カメラの投影面と同一視する.「有向射影幾何学」[21] を用いれば，各画素 p の値は次ように計算できる．

[1] 実際にはその両方がほぼ成立している．それからの微小なずれは後に示すように，画像処理によって補正する．

[2] 射影変換 $\boldsymbol{H}$ の (33) 要素が 0 であれば原点が無限遠点に写像されるが，重なりのある隣接 2 画像間ではそのようなことは生じない．

[3] これが「有向射影幾何学」[21] の特徴である．それに対して，普通の射影幾何学では射影変換行列の符号は任意である [11].

1. 画素 p に対応する視線方向（仮想カメラの視点から見て，仮想カメラの画像面上の点 p の見える方向）を表すベクトルを $\vec{Op}$ とする．
2. ベクトル $\boldsymbol{u} = \boldsymbol{R}\vec{Op}$ を計算し[4]，仮想カメラの画像面を第 1 画像と同一視したとき，$\boldsymbol{u}$ が**前方**（画像面のある側）を指し，かつその指す点が第 1 画像のフレーム内にあればその画素値を画素 p の値とする[5]．
3. そうでなければベクトル $\boldsymbol{u}' = \boldsymbol{H}_{12}\boldsymbol{u}$ を計算し，仮想カメラの画像面を第 2 画像と同一視したとき，ベクトル $\boldsymbol{u}'$ が**前方** を指し，かつその指す点が第 2 画像のフレーム内にあればその画素値を画素 p の値とする．
4. そうでなければベクトル $\boldsymbol{u}'' = \boldsymbol{H}_{23}\boldsymbol{u}'$ を計算し，仮想カメラの画像面を第 3 画像と同一視したとき，ベクトル $\boldsymbol{u}''$ が**前方** を指し，かつその指す点が第 3 画像のフレーム内にあればその画素値を画素 p の値とする．
5. 以下同様に第 4 画像, 第 5 画像, ... と進み，すべての画像が尽くされたら終了する（どの画像からの画素値が得られなければ点 p の画素値は未定義）．

この計算では視線方向 $\boldsymbol{u}$ が前方を指すか後方を指すかで対応する画像面上の点を異なるものと解釈している．これが「有向射影幾何学」[21] の特徴である[6]．

上記の計算では，表示領域に表示すべき部分を入力画像の中で番号の小さいほうから探している．したがって，重なり部分では番号の小さい画像が上に，小さい画像が下に重なっているように表示される．しかし，最終画像と第 1 画像との関係が変則的になる．そこで最終画像が第 1 画像の上になるように修正する．

2.3 視線移動の指定

上述の計算には，仮想カメラの現在の向きを表す回転 $\boldsymbol{R}$ が必要となる．そのためには初期に $\boldsymbol{I}$ から出発し，ユーザが表示領域内のある点 P （以下,「注視点」と呼ぶ）を指定する度に[7]，その点が表示領域の中心に来るように仮想カメラを回転させ，回転 $\boldsymbol{R}$ を次々と更新する．

[4]現在の仮想カメラの位置はその初期から $\boldsymbol{R}$ だけ回転しているので，ベクトル $\vec{Op}$ を初期方向の仮想カメラから見ると $\boldsymbol{u} = \boldsymbol{R}\vec{Op}$ となる．

[5]実数座標の点の画素値は周囲の整数座標の画素値から双線形補間する．以下の処理でも同様である．

[6]それに対して，普通の射影幾何学では射影空間の点は原点を通る向きのない視線に対応する [11].

[7]注視点の入力は，マウスによる指定，キーボードの矢印（→, ←, ↑, ↓）キーによる移動ジョイステックによる移動，あるいは HMD（ヘッドマウントディスプレイ）を装着して，その向きの移動に連動させてもよい．

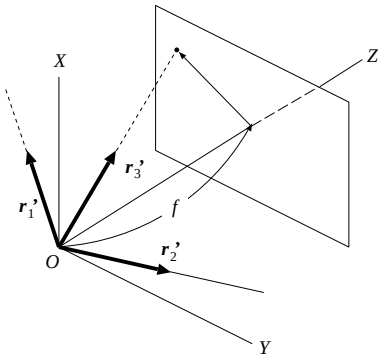


図 1: 注視点に対応する仮想カメラの回転.

仮想カメラの視点から見た注視点 P の方向を表す単位ベクトルを $\boldsymbol{n}$ とする．現在の仮想カメラが初期状態から $\boldsymbol{R}$ だけ回転した向きであるとすると，ベクトル $\boldsymbol{n}$ は初期フレームから見ると $\boldsymbol{Rn}$ である．そこで仮想カメラの光軸を次の向きに移動する．

$$\boldsymbol{r}'_3 = N[\boldsymbol{Rn}] \tag{3}$$

ただし，$N[\cdot]$ はノルムを 1 とする正規化を表す．仮想カメラの Y 軸（水平）方向は $\boldsymbol{r}'_3$ と鉛直方向 $\boldsymbol{i} = (1,0,0)^\top$ に直交するから，次のようになる．

$$\boldsymbol{r}'_2 = N[\boldsymbol{r}'_3 \times \boldsymbol{i}] \tag{4}$$

残る X 軸方向は Y 軸，Z 軸に直交するから，次のようになる．

$$\boldsymbol{r}'_1 = N[\boldsymbol{r}'_2 \times \boldsymbol{r}'_3] \tag{5}$$

したがって，視線の移動後の仮想カメラの向きは次の回転行列 $\boldsymbol{R}'$ となる（図 1）．

$$\boldsymbol{R}' = \begin{pmatrix} \boldsymbol{r}'_1 & \boldsymbol{r}'_2 & \boldsymbol{r}'_3 \end{pmatrix} \tag{6}$$

このようにすれば，どのように視線を移動しても表示される画像中の水平方向は常に水平に保たれる．もちろん，オプションによって視線回りの回転（ロール）やズーム変化を行うことも可能である．

2.4 画像例

図 2 はディスプレイ上の表示領域に描かれた画像をその中に示した × 印を注視点とするように次々の視線を移動して全周囲を眺めた画像列である．黒い部分は原画像に撮影されていない部分である．

3. 継ぎ目の補正

隣接する画像間に対応点を指定して，射影変換によって連続的に張り合わせていくと，全方位を一周

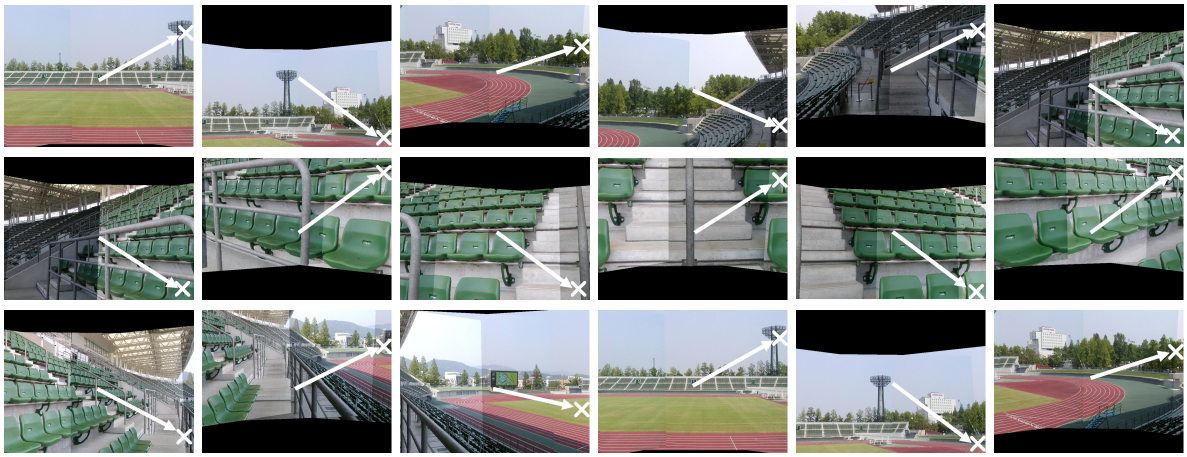


図 2: 視点を連続的に移動させながら全周囲を眺めた表示．画像の中心が×印の位置に移動する．

した画像間に不連続が生じる．これを前報 [19] の方法で，一周する射影変換の合成が恒等変換になるという制約のもとで個々の射影変換を最適化する．その結果，不連続が隣接する画像間に分配され，各画像間に微少な食い違いが生じる．

画像間の食い違いの補正は従来からさまざまな方法が提案されている．これらは，わずかな位置合わせの誤差を補正したり，輝度を連続化したりするものから [2, 16, 7, 9, 15, 20]，食い違いが最も少ないパスを動的計画法 [13] やグラフカット [8] で探索すること [10, 14]，視差が原因で位置が大きさが異なる物体を張り合わせるために，対応位置を探索して，それらを無理やりに合わせ，生じた大きな歪みを画像内部に連続的に波及させること [10]，多数の重なり画像から最適なものを選択すること [5]，インタラクティブな編集を行うこと [1, 18]，などが行われている．さらにはテクスチャが異なる領域を連続的に混合する手法も開発されている [3, 6].

しかし，本論文で考える応用では食い違いの原因が全く異なっているので，従来手法は適用できない．全周を張り合わせて不連続が生じるのは，特徴点の指定誤差や射影変換の計算誤差の蓄積のためもあるが，より大きな要因はレンズ歪みによって視野の周辺が内部にやや収縮するためと思われる．これは，2 画像間の境界が合うように接続すると，全周の最後の画像と最初の画像の間に常にギャップが生じ，逆の場合の余計な重なりが生じることがなかったからである．

これが原因であるなら，あらかじめ参照板を用いてレンズ歪みの幾何学的補正を行えば問題が解決する．しかし，本論文では画像のみが与えられ，カメラや撮影状況に関するデータは存在しないと仮定している．そこで以下のように画像処理によって対処する．前節と同様に第 1 画像（左側）が第 2 画像（右側）の上に重なっているとする（図 3(a)）．

3.1 輝度の補正

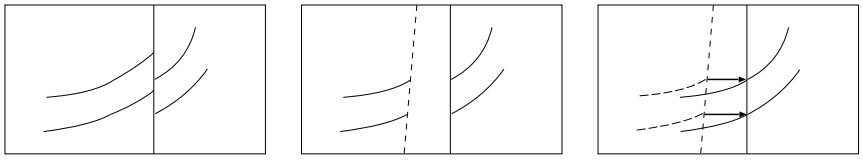
前処理として 2 画像の重なる画素の輝度をおおまかにそろえる．これは射影変換を最適化する前の段階（食い違いが生じない状態）で第 1 画像と第 2 画像の輝度の変化が線形変換であると仮定して，比例定数（ゲイン）と定数項（バイアス）を R, G, B ごとに最小二乗法で求める（付録 A）．そして第 1 画像の継ぎ目の輝度を第 2 画像に一致するように補正し，重なりの反対側の境界では変化させず，その間を線形に比例配分する．第 2 画像では反対向きに比例配分し，重なりで両画像の輝度をほぼ一致させる．

3.2 継目のずれの解消

このように補正した後，第 1 画像の内部に（一般に傾いた）直線を引き，その左側の領域の画素値を消去して空白部分を作る（図 3 (b)）．そして，その直線上の画素を画像の左端に移動させて空白を埋め，その画素の移動を画像の内部に線形に波及させる（図 3 (b)）．その直線の位置と向きは，第 2 画像の継ぎ目の左側部分をテンプレートとして第 1 画像上の最もよくマッチする位置を探索して定める．

3.3 継目のずれと輝度の微調節

次に，画像に継ぎ目において第 1 画像の各画素を数画素の範囲で連続的に上下させて第 2 画像と最もよく一致するように補正する．これには動的計画法 [13] を用いる（付録 B）．その後，再び第 1 画像の輝度が継ぎ目でそれに重なる第 2 画像の画素と最もよく一致する線形変換を最小二乗法で求めて補正する



(a) (b) (c)

図 3: (a) 継ぎ目のずれ．(b) 画素値を消去する．(c) 空白を埋める．

(a) (b)

図 4: (a) 最適な射影変換による張り合わせ．(b) 継ぎ目の補正．

（付録 A）．そして，第 1 画像の左端では原画像のままとし，画像全体に渡って線形に補正する．これを隣接する画像ごとに順に行い，これが全周に渡って輝度が連続であるように数ラウンド反復する．

3.4 重なり部分の画素値の混合

最後に継ぎ目の内部の数画素の幅において第 1 画像の画素値をその下の第 2 画像の画素値との重み付き平均値に置き換える．その重みは継ぎ目では第 1 画像が 0, 幅の内部では第 1 画像が 1 となるように比例配分する．以上の処理により，継ぎ目の不連続がほぼ完全に消失する．

3.5 画像例

図 4(a) は最適化した射影変換によって張り合わせた 2 画像の継ぎ目部分である．ずれが波及して食い違いが生じている．図 4(b) は上述の手順で第 1 画像を補正したものである．継ぎ目の不連続が解消していることがわかる．

4. 移動物体の除去

簡単のために，隣接する画像間で一方に写り，他方には写っていない部分を「移動物体」と呼ぶ．これがたまたま画像の継ぎ目部分に現れると画像間に不連続が生じる．

説明上，考えている 2 画像の左側を「第 1 画像」，右側を「第 2 画像」と呼び,「第 1 画像」が上になるように張り合わせるとする．問題が生じるのは次の二通りの場合である．

- 第 1 画像の右端に右の一部が欠けた移動物体が写っている．
- 第 2 画像中の移動物体上に第 1 画像の右端が張り合わされる．

本論文では，常に第 1 画像（上側）を補正する．具体的には，第 1 の場合はその移動物体部分を除去し，第 2 画像の対応する部分をコピーする．第 2 の場合も第 2 画像上の移動物体上に重なる第 1 画像部分を除去し，第 2 画像をコピーする．

こうすると，何が移動物体であるかという判定を行わず，単に 2 画像間で一致しない領域を第 1 画像から除去し，対応する第 2 画像をコピーすればよい．したがって，この処理は不一致の領域を検出することに帰着する．このような処理は従来から多く行われ，差分によって不一致の領域を検出し，動的計画法 [13] やグラフカット [8] などによって計算した最適なパスにそって切り出したり，多数の重なり画像から最適なものを選択する手法が開発されている [1, 2, 5, 10, 14, 22].

しかし，実際に行ってみると，本論文が想定するような 2 枚のみの画像の重なりから差分によって不一致領域を検出することは非常に困難であることが判明した．これは次のような本論文の問題設定に由来するところが大きい．

- カメラは露出が自動化されているので，日照や空の割合が異なると，画像間で輝度が変化する．検出される差分の一部はこれによる．
- 画像が完全に重ならず，物体の境界部分に差が検出されやすい．
- 移動物体（人や車）が背景（道路など）と色や明るさが似ていると，差分では移動物体の一部が検出されなかったり，内部に移動物体と判定されない “穴” が生じる．

第 1 点への対処はこれまでもよく行われ，それほど問題ではない．第 2 点は対応点の指定誤差や撮影時の視差も影響するが，より大きい原因はレンズの収差による画像自体の歪みであると思われる（後述）．しかし，カメラや撮影状況の情報がないと仮定しているのでカメラ校正はできない．第 3 点は移動物体が何かという知識がなければ困難である．

本研究ではこれを解決するためにさまざま方法を試みたが，移動物体を完全に除去することはできなかった．しかし，ある程度の除去は行なえたので，その方法を述べる．

4.1 輝度の補正

第１点に対処するために，差分が小さい領域50%を指定して，その領域間で２画像の輝度をなるべく一致するように補正する．具体的には，第２画像をコピーした作業領域（「仮の第２画像」と呼ぶ）を用意し，差分の小さい重なり部分で第１画像の画素値となるべく一致するように仮の第２画像の R, G, B 値を線形変換する．比例係数（ゲイン）と定数項（バイアス）は最小二乗法で定める（付録 A）．

4.2 位置の微調節

第２点に対処するために，第１画像の右端の移動物体が生じやすいと考えられる領域を指定し，(輝度を補正した）仮の第２画像を上下左右に微小にサブ画素精度で平行移動し，この領域で差分が最も少なくなる位置を探索する．

4.3 移動物体の検出

第３点に対処するには画像処理のみでは不十分で，何らかの事前知識を用いざるを得ない．そこで含まれているかも知れない移動物体の大きさを想定し，その大きさの長方形領域を第１画像の右端に沿って上下にスライドさせ,(輝度を補正し，位置を補正した）仮の第２画像との差分が最も大きくなる位置を探索する．その後，その内部を２値化し，収縮・膨張フィルタを施して値１の孤立点やエッジを除去し，隣接する差分領域を統合する．そして，これに連結領域のラベル付けを行い，第１画像の右端を含む連結領域のみを残して残りを除去する．さらにこれを左から走査して，途中で値 0 となる部分（“穴”）を値１で埋める．

4.4 移動物体の除去

このようにして検出した差分領域を最後にやや膨張させ，安全のために移動物体の周辺を余分に取り出す．そして，その領域に（輝度値を補正し，位置を補正した）仮の第２画像を上書きする．ただし，領域の境界での不連続を防ぐために，境界の内部と外部の数画素では二つの画像を連続的に線形補間する．

4.5 画像例

図５は隣接する２画像である．右画像には左画像にない人物が写っている．これらを共通する特徴点

図 5: 移動物体のある隣接 2 画像.

(a) (b)

図 6: (a) 直接の張り合わせ．(b) 移動物体除去後の張り合わせ.

を指定して射影変換によって張り合わせると図 6(a) のようになり，人物の一部が切り離される．これに上述の処理を行い，第２画像の人物部分の輝度や位置を補正して，第１画像上に上書きしたものが図 6(b) である．これにより，その人物が始めから第１画像に写っていたような自然な画像が得られる（両画像の継ぎ目の不連続は，この後で 3 節の処理を行なって補正する）．

5. 全周の疑似ビデオ表示の例

図７は用いた 12 枚の入力画像である．図８はそれを用いて全周表示し，視線方向を連続的に変化させながら疑似的のビデオ動画像の表示の例である（その一部を適当な間隔でならべている）．

6. まとめ

本論文では全方向を自由に撮影した少数枚の画像から，あたかも視点の 360° 周りをビデオカメラで撮影したかのような動画像を表示する手法を提案した．この表示手法を用いれば，計算機内部に全周画像や動画系列を生成・内蔵する必要はない．これは「有向射影幾何学」に基づいて，視点を移動するごとに個別の入力画像から直接にリアルタイムで画素を選んでディスプレイ上に写像するものであり，カメラや撮影状況の知識（焦点距離，撮影方向など）を必要としない．

また，隣接する画像の継目に生じるずれや輝度の不連続を解消したり，撮影中に移動する物体の像が

図 7: 入力画像 (12 枚).

図 8: 疑似ビデオ画像列 (一部).

画像の継ぎ目で切断されることを防ぐ方法を示した.

そして，実画像を用いて，ユーザーが任意の視線移動を指定してその方向に見えるシーンをインタラクティブに表示したり，あたかもビデオカメラを連続的に動かして 360° を撮影したかのような臨場感が実現することができることを示した.

参考文献

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, Interactive digital photomontage, *ACM Trans. Graphics*, Vol. 23, No. 3, (2005-6), 294–302.

[2] M. Brown and D. G. Lowe, Automatic panoramic image stiching using invariant features, *Int. J. Comput. Vis.*, **74**-1 (2007-8), 59–73.

[3] P. J. Burt and E. H. Adelson, A multiresolution spline with applications to image mosaics, *ACM Tran. Comput. Graphics*, **4**-2 (1983-10), 217-236.

[4] S. E. Chen, QuickTime VR — An image-based approach to virtual environment navigation, *Proc. SIGGRAPH'95*, August 1995, Los Angeles, U.S.A. (ACM Press, New York, NY, U.S.A.), pp. 29-38.

[5] J. Davis, Mosaics of scenes with moving objects, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 1998, Santa Barbara, CA, U.S.A., pp. 354–360.

[6] A. Efros and W. T. Freeman, Image quilting for texture systhesis and transfer, *Proc. SIGGRAPH*, August 2001, Los Angeles, CA, U.S.A., pp. 341–346.

[7] D. Hastler and S. Süsstrunk, Color handling in panoramic photography, *Proc. SPIE, Vol. 4309: Videometrics and Optical Methods for 3D Shape Measurement* (Eds. S. F. El-Hakim and A. Gruen), San Jose, CA, U.S.A, January 2001, pp. 62–72.

[8] 石川博, グラフカット, 情報処理学会研究報告 2007-CVIM-158-26 (2007-3), 193–204.

[9] J. Jia and C.-K. Tang, Image registration with global and local luminance alignment, *Proc. 10th Int. Conf. Comput. Vis.*, Vol. 1, October 2003, Nice, France, pp. 156–163.

[10] J. Jia and C.-K. Tang, Eliminating structure and intensity misalingment in image stiching, *Proc. 10th Int. Conf. Comput. Vis.*, Vol. 2, October 2005, Beijing, China, pp. 1651-1658.

[11] 金谷健一, 「形状ＣＡＤと図形の数学」, 共立出版, 1998.

[12] 金谷健一, 「これなら分かる応用数学教室—最上二乗法からウェーブレットまで—」, 共立出版, 2003.

[13] 金谷健一, 「これなら分かる最適化数学—基礎原理から計算手法まで—」, 共立出版, 2005.

[14] V. Kwatra, A. Schödl, I. Essa, G. Turk and A. Bobick, Graphcut textures: Image and video synthesis using graph cuts¡/A¿, *AMC Trans. Graphics*, Vol. 22, No. 3 (2003-6), 277–286.

[15] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, Seamless image stiching in the gradient domain, *Proc. 8th Euro. Conf. Comput. Vis.*, May 2004, Prague, Czech, Vol. 4, pp. 377–389.

[16] P. F. McLauchlan and A. Jaenicke, Image mosaicing using sequential bundle adjustment, *Image Vis. Comput.*, **20**-9/10 (2002-8), 715–759.

[17] 長原 一, 八木康史, 高精細全方位パノラマ画像の生成, 情報処理学会研究報告 2005-CVIM-147-21 (2005-1), 163–170.

[18] P. Pérez, M. Gangnet, and A. Blake, Poisson image editing, *AMC Trans. Graphics*, Vol. 22, No. 3 (2003-6) 313–318.

[19] 坂本雅俊, 酒井祐貴, 金谷健一, 菅谷保之, 自由に撮影した画像による全周パノラマ生成のための射影変換の最適化, 情報処理学会研究報告 2006-CVIM-154 (2006-9), 219–226.

[20] H.-Y. Shum and R. Szeliski, Construction of panoramic image mosaics with global and local alignment, *Int. J. Comput. Vis.*, **36**-2 (2000-2), 101–130.
[21] J. Stolfi, *Oriented Projective Geometry: A Framework for Geometric Computation*, Academic Press, San Diego, CA, U.S.A., 1991.
[22] M. Uyttendaele, A. Eden, and R. Szeliski, Eliminating ghosting and exposure artifacts in image mosaics, *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* Vol. 2, December 2001, Kauai, HI, U.S.A., pp. 509–516.
[23] 八木康史, 横矢直和, 全方位ビジョン：センサ開発と応用の最新動向, 情報処理学会論文誌：コンピュータビジョンとイメージメディア, **42**-SIG 13 (2001-12), 1–18.
[24] 横矢直和, 山澤一誠, 竹村治雄, 全方位ビデオカメラを用いた視覚情報メディア, 情報処理学会論文誌：コンピュータビジョンとイメージメディア, **42**-SIG 13 (2001-12), 59–70.

付録 A：最小二乗法による輝度補正

第 1 画像 I の (i,j) 画素が第 2 画像の I' の (i',j') 画素に重なっているとする（i, j は整数とは限らない）．この関係を $i'=i'(i,j)$, $j'=j'(i,j)$ と書く．第 1 画像 I が重なりにおいて第 2 画像 I' となるべく同じ輝度を持つように I を $\hat{I}=aI+b$ の形に補正する比例係数（ゲイン）a と定数（バイアス）b を次の最小二乗法で定める [12].

$$J=\frac{1}{2}\sum_{(i,j)\in\mathcal{R}}\Big(I'(i'(i,j),j'(i,j))-aI(i,j)-b\Big)^2\to\min \tag{7}$$

ただし，$\mathcal{R}$ は第 2 画像の輝度に近づけたい第 1 画像上の領域である．式 (7) を a, b で微分すると次のようになる．

$$\begin{aligned}\frac{\partial J}{\partial a}&=\sum_{(i,j)\in\mathcal{R}}\Big(I'(i'(i,j),j'(i,j))-aI(i,j)-b\Big)(-I(i,j))\\ \frac{\partial J}{\partial b}&=\sum_{(i,j)\in\mathcal{R}}\Big(I'(i'(i,j),j'(i,j))-aI(i,j)-b\Big)(-1)\end{aligned} \tag{8}$$

これらを 0 と置くと次の正規方程式を得る．

$$\begin{pmatrix}A & B\\ B & N^*\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix}=\begin{pmatrix}p\\ q\end{pmatrix} \tag{9}$$

ただし，N^* は領域 $\mathcal{R}$ の画素数であり，A, B, p, q を次のように定義する．

$$A=\sum_{(i,j)\in\mathcal{R}}I(i,j)^2,\quad B=\sum_{(i,j)\in\mathcal{R}}I(i,j) \tag{10}$$

$$\begin{aligned}p&=\sum_{(i,j)\in\mathcal{R}}I'(i'(i,j),j'(i,j))I(i,j)\\ q&=\sum_{(i,j)\in\mathcal{R}}I'(i'(i,j),j'(i,j))\end{aligned} \tag{11}$$

正規方程式 (9) をクラメルの公式 [12] を用いて解けば a, b が次のように求まる．

$$a=\frac{N^*p-Bq}{AN^*-B^2},\qquad b=\frac{Aq-Bp}{AN^*-B^2} \tag{12}$$

付録 B：動的計画法による端の補正

画像サイズを $N\times M$ とし，第 1 画像の右境界 $j=M-1$ 上の点 $(i,M-1)$ が第 2 画像に重なっているとする．重なっている第 2 画像を d 画素だけ上にずらしたときの第 1 画素の画素 $(i,M-1)$ の近傍（上下に ±2 画素，左に 4 画素をとった）における差分（R, G, B 値の差の絶対値の和）を $J(i,d)$ とする．ただし，最初の状態の食い違いの平均値（$J(i,0)$, $i=0$, ..., $N-1$ の平均）を $\bar{J}$ とし，$J(i,d)$ が $\bar{J}$ 以下であれば $\bar{J}$ とし，平均以上の食い違いを削減する．

まず，補正のための画素の上下移動の上限 $d_{\max}$ を定め（$d_{\max}=5$ 画素とした），$(2d_{\max}+1)\times N$ の配列 $D(d,i)$，$P(d,i)$, $|d|\le d_{\max}$, $0\le i\le N-1$ と，N 次元配列 $d^*(i)$, $0\le i\le N-1$ を用意し，次のように初期化する．

$$P(d,0)=0,\qquad D(d,0)=\begin{cases}0 & d=0\\ \infty & d\neq 0\end{cases} \tag{13}$$

そして次の計算を $i=1$, ..., $N-1$ の順に行う．

$$\begin{aligned}D(d,i)&=\min_{d'}\Big[(1+|d-d'|)J(i,d)+D(d',i-1)\Big]\\ P(d,i)&=(\text{その最小値をとる } d')\end{aligned} \tag{14}$$

ただし，d' は $-d_{\max}\le d'\le\min(d+1,d_{\max})$ の範囲を調べ，これをすべての $|d|\le d_{\max}$ について計算する．最後に，$d^*(N-1)=0$ と置いて，次の計算を $i=N-2, N-3$, ..., 0 の順に行う．

$$d^*(i)=P(d^*(i+1),i+1) \tag{15}$$

これが定まれば，画像の端の食い違いの補正を画像内部に波及させる幅を H とし（$H=20$ 画素とした），次のように補正画像 $\tilde{I}(i,j)$ を作成する．

$$\tilde{I}(i,j)=\begin{cases}I(i,j) & 0\le j\le M-H-1\\ I(i+v(j)d^*(i),j) & M-H\le j\le M-1\end{cases} \tag{16}$$

ただし，関数 $v(j)$ を次のように定義する．

$$v(j)=\frac{j+H-M+1}{H} \tag{17}$$