

映像からのパノラマ画像の実時間生成と可視化

Real-time image mosaicing and its visualization from a video sequence

興梠 正克[†]

蔵田 武志[‡]

星野 准一^{‡*}

村岡 洋一[†]

Masakatsu Kourogi[†]

Takeshi Kurata[‡]

Jun'ichi Hoshino^{‡*}

Yoichi Muraoka[†]

早稲田大学工学部[†]

電子技術総合研究所[‡]

セコム IS 研究所^{‡*}

Abstract 本研究では、動き物体を含むカメラ映像からパノラマ画像を実時間生成/可視化する手法を提案する。提案手法は、輝度の空間勾配が十分に大きな画素上で、勾配方程式に基づいて動きベクトルを近似計算して、画素単位マッチングによりテストする。マッチングに成功する動きベクトルから、フレーム間の位置合わせのアフィン動きパラメータを推定する。この推定パラメータに基づいて、フレーム間の動きを補償する。動きが補償されたフレーム画像に対して、補償誤差が十分小さくなるか、一定の反復回数を超過するまで同様のパラメータ推定を繰り返す。動き物体の存在を考慮して、パラメータ推定には M-推定を用いる。次に、フレーム間の位置合わせに失敗する画素を領域併合して、動き物体の分離を行う。提案手法を汎用的な PC のソフトウェアとして実装、評価した。その結果、毎秒 10-11 フレームの処理速度 (2 台の PC による並列処理) でカメラ映像からパノラマ画像の生成/可視化に成功した。また、映像中に動き物体が含まれていても、背景のパノラマ画像と動き物体を重ね合わせたパノラマ動画を実時間生成できた。

1 はじめに

複数枚の画像を貼り合わせることで、広視野・高解像度のパノラマ画像を生成する研究が盛んである [3][5]。特にビデオ映像から直接パノラマ画像を生成 (ビデオモザイク) することは、多くの応用分野にとって有用な要素技術である。

本研究では、視点非固定のカメラ映像からの実時間パノラマ画像生成 (実時間ビデオモザイク) を目的とする。この実現により、映像の各フレームを蓄積することなくパノラマ画像をオンライン生成できる。また、生成されたパノラマ画像中にライブ映像をはめこんで表示することが可能である。本研究では、具体的な応用例として、パノラマ画像+はめ込みライブ映像を実時間生成、表示、保存することで、周囲の環境や状況をより分かりやすく可視化、提示するシステムを PC のソフトウェア処理により実現する。この利用方法として、例えばノート PC とカメラを用いて、旅行先のパノラマ映像をオンライン生成して、ビデオメールとして送ることで、現地の様子を分かりやすく伝えることができる。

映像からのパノラマ画像生成において、最も計算コストを要する処理は、フレーム間の位置合わせである。複数の動き領域の存在にロバストな位置合わせを行う多くの従来手法 [2][6][7] が提案、検討され

ているが、そのほとんどは処理の計算コストが高く、PC のソフトウェアによる実時間処理は難しい。処理の実時間性を考慮した従来研究としては、輝度投影相関を用いた手法 [12] が提案されているが、カメラの回転や映像中の動き物体について考慮がなされておらず、その適用可能な映像は限定される問題点がある。

一方、我々は、通常のカメラワーク (パン・チルト・ズーム) に加えて回転を扱えるアフィン変換を動きモデルに取る実時間処理可能な動き推定手法を提案している [1]。本研究では提案手法を拡張して、動き物体の分離とパノラマ画像上へのフレーム画像の位置合わせを実現する。まず、フレーム間の位置合わせに失敗する画素を検出して、その画素集合を領域併合することで、動き物体の領域分離を実現する。次に、パノラマ画像上へのフレーム画像の位置合わせについては、フレーム間の大きな動きを推定する必要があるため、パノラマ画像の大きさに応じて、動き推定の初期値を複数個持たせる。それぞれの初期値から推定された結果のうち、最も良いフレーム間マッチングと低い動き補償誤差を与えるものを推定結果とする。

本論文の構成は以下の通りである。2-5 では、動き物体を含む映像からパノラマ画像を生成する手法に

ついて述べる。6では、提案手法をPCのソフトウェアとして実装する方法について述べ、その計算コストを詳細に見積もる。7では、実装された提案手法を実画像に適用することで、その有効性と処理の実時間性について評価する。

2 パノラマ画像の生成の概要

本研究では、以下の1-3.の処理を入力動画の各フレーム画像に対して行うことで、画像中の動き物体を分離しつつパノラマ画像を順次生成する。

1. フレーム画像の位置合わせ：前フレーム画像との位置合わせを行う。すでに生成されたパノラマ画像と重なる場合、パノラマ画像との位置合わせを行い、動きパラメータを得る。
2. 動き領域の分離：処理1.で得た動きパラメータに基づいて、フレーム間の位置合わせを行う。この位置合わせに失敗する画素の領域を併合して、動き領域を分離する。
3. パノラマ画像へのマージ：処理1.で得た動きパラメータに基づいて、フレーム画像を変換してパノラマ画像にマージする。その際に、処理2.で分離された動き領域は除去して空白にする。

また、この手順により生成されたパノラマ画像と入力映像のフレーム画像との位置合わせを行うことで、映像をパノラマ画像上にはめこみ表示することができる。

以下、3では処理1.について説明して、4では処理2.を、5では処理3.を説明する。

3 フレーム画像の位置合わせ

3.1 フレーム画像間の位置合わせ

視差の影響が小さい状況下では、カメラワークや手ぶれによる動きはアフィン動きとして近似できる。そこで、映像中の連続する2枚のフレーム間のグローバルなアフィン動きパラメータを推定することで、これらのフレーム間の位置合わせを行うことができる。動き推定の方法として、我々が提案したフレーム間のグローバルなアフィン動きパラメータの推定手法[1]を用いる。この手法は動き物体の存在にロバストであり、PCのソフトウェアにより実時間処理できるので、パノラマ画像生成のための位置合わせに適用できる。

3.2 パノラマ画像上への位置合わせ

パノラマ画像上へのフレーム画像の位置合わせにおいては、非常に大きな動きパラメータを推定する必要がある。しかしながら、前述の位置合わせ手法は、動きベクトルと輝度勾配の拘束方程式に基づいているため、フレーム間の大きな動きパラメータを推定することは難しい。

そこで、推定すべきフレーム間の動きパラメータが十分小さくなるように、パノラマ画像の大きさに応じて、動きパラメータの初期値を複数個用意する。それぞれの初期値から前述の手法に基づいて推定された動きパラメータのうち、最も良いマッチング結果を与えるものを出力結果とする。

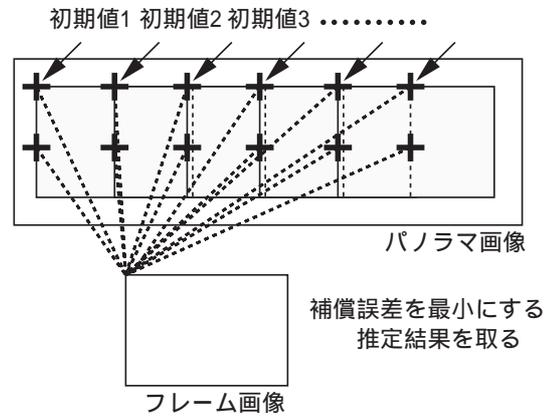


図1: パノラマ画像との位置合わせ

4 動き領域の分離

4.1 フレーム間での動き領域の検出

3で述べた位置合わせ手法により得られたアフィン動きパラメータ (a_1, a_2, \dots, a_6) を以下の式(1)(2)に代入して、点 (x, y) の各画素上での予測動きベクトル (u, v) を求める。

$$u = a_1x + a_2y + a_3 \quad (1)$$

$$v = a_4x + a_5y + a_6 \quad (2)$$

この動きベクトルが現フレーム $I_c(x, y)$ と参照フレーム $I_r(x, y)$ 間の画素単位マッチングに失敗する場合、その画素を動き領域の候補とみなす。この際に、有意な情報を得るため、輝度勾配の絶対値が十分に大きな画素だけを対象とする。画素単位マッチングは、

輝度勾配の絶対値が大きいほど動きベクトルの誤差の影響を受けやすい．そこで，勾配による影響を正規化するため，以下の式 (3) が定義する正規化誤差 e を画素単位マッチングの成功判定に用いる．

$$e = \frac{|I_c(x+u, y+v) - I_r(x, y)|}{|\frac{\partial I}{\partial x}| + |\frac{\partial I}{\partial y}|} \quad (3)$$

この正規化誤差 e がしきい値 T_e を越えるとき，画素単位マッチングに失敗したとみなす．

画素単位マッチングに失敗する画素から成る集合を S_e とおく．点 $(x, y) \in S_e$ からの距離が T_d 以下にある画素を動き領域の候補範囲 R とする．すなわち $R = \{(p, q) | d((p, q), (x, y)) < T_d\}$ である．なお， $d(P, Q)$ は 2 点 P, Q 間の距離関数である．ここでは処理の簡便性から $P_i = (x_i, y_i), P_j = (x_j, y_j)$ のとき，距離関数を $d(P_i, P_j) = \max(|x_i - x_j|, |y_i - y_j|)$ と定義する．点 P から距離 T_d 以内の領域は P を中心として一辺が $2T_d$ の正方形領域となる．

2 点 $P_i, P_j \in S_e$ の距離 $d(P_i, P_j)$ が一定のしきい値 T_d 以下であるとき，これらの画素がつくる候補領域 R_i と R_j を図 2 に示すように併合する．

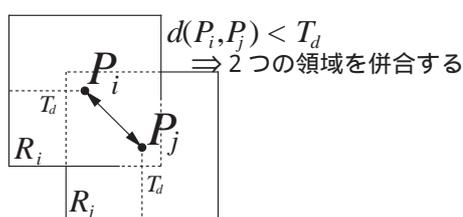


図 2: 動き領域の併合条件

以上の領域併合処理を行うと，いくつかのお互いに共有部分のない領域が得られる．これらの各領域のうち，領域内に含まれている点 $(x, y) \in S_e$ の画素数が十分に大きい領域を動き領域として取り出す．これにより，ノイズ等の影響で画素単位マッチングに失敗した画素がつくる領域を取り除く．

4.2 処理の計算コスト

前述の動き領域分離の手法は次の二つの処理から成る．(a) 画素ごとに式 (3) をテストして動き領域の候補となる画素を選び出す．(b) 候補となる画素がつくる領域を併合する．処理 (a) は式 (3) が定義する正規化誤差を算出するために，輝度勾配の逆数を求める演算と画素格子内に落ちたサブピクセルの輝度

を求める演算 (ここでは線形補間を用いる) が必要である．

動き分離で対象とする画素数の上限を N 個，一つの動き領域に併合される可能性がある画素の個数の上限を r 個とすると，本手法が必要とする演算処理とその実行回数の上限は以下の表 1 に示す通りである．

演算処理	実行回数 [回]
輝度勾配の逆数計算	N
線形補間 [マッチング]	N
領域の併合	$N \times r$

表 1: 動き分離の計算コスト

5 パノラマ画像の合成

前述の方法により得られた位置合わせパラメータと動き領域の分離結果を用いて，以下に述べる処理によってパノラマ画像の生成と，パノラマ画像上への映像のはめ込み表示を実現する．

5.1 パノラマ画像生成

各フレーム画像をフレーム間の位置合わせパラメータにしたがい変換することで，図 3 に示すように，パノラマ画像に順次マージすることができる．この際に，動き領域についてはマージを行わず，空白にしておく．後からマージされるフレーム画像の同一領域が動き領域でない (すなわち，動き物体がその領域から移動した) 場合，その空白領域を埋める．このようなマージ処理によって，動き領域が取り除かれた背景だけのパノラマ画像を生成することができる．

5.2 パノラマ画像上への映像のはめ込み表示

すでにパノラマ画像が生成されている場合は，3.2 節で述べた方法に基づいて，映像のフレーム画像とパノラマ画像の位置合わせを行う．その位置合わせ結果にしたがい映像の各フレームを変換してパノラマ画像上にはり合わせることで，パノラマ画像上への映像のはめ込み表示を実現できる．

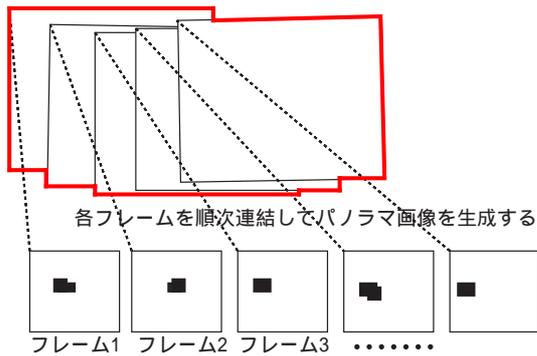


図 3: パノラマ画像の順次生成

6 PC のソフトウェアへの実装

6.1 実装する計算機環境

提案手法を，汎用的な PC 2 台 (OS: Linux-2.2.4 SMP 対応, CPU: PentiumII-450MHz [2 CPU 構成]) を 100M イーサネットによりネットワーク結合したクラスタ上に PVM 通信ライブラリを用いてソフトウェア実装した．画像取り込みハードウェアには，BT848 チップを搭載した画像キャプチャカードを用いて，Linux システムによる取り込み処理は，Video for Linux によるシステム API を使って実装した．画像表示には X11 ライブラリ (共有メモリ拡張あり) 関数を用いた．取り込まれるフレーム画像は 320×240 24 ビット RGB カラーである．

6.2 実現方法

提案手法によるカメラ映像からパノラマ画像の生成と可視化を，以下の 5 つの処理に分けて，それぞれを独立した UNIX プロセスとして実装する．(1) 画像の取り込み，(2) フレーム間の位置合わせ，(3) 動き領域の分離，(4) パノラマ画像へのマージ，(5) 生成されたパノラマ画像の表示．プロセスで実装する利点は，マルチプロセッサ PC 上で本ソフトウェアを実行する場合，OS が対応していれば，プロセッサの台数効果による高速化を期待できることである．これら 5 つのプロセスのうち，プロセス (1)(2)(3) を 1 台目の PC 上で，プロセス (4)(5) を 2 台目の PC 上で実行させる．同一 PC 上のプロセス間の通信には共有メモリを用いて，異なる PC 上のプロセス間の通信には PVM 通信ライブラリを用いる．

6.3 計算コストの見積り

6.3.1 位置合わせの計算コスト

上記の計算機環境では，2 枚のフレーム間の位置合わせ手法 [1] に要する処理時間は 30 ミリ秒以下であることが確認されている．パノラマ画像との位置合わせでは，複数個の初期値から位置合わせパラメータ推定を行う．その初期値の個数を P とすれば， $(P \times 30)$ ミリ秒の処理時間を要する．

6.3.2 動き領域分離の計算コスト

安定した動き領域の分離を実現するには，フレーム画像を水平，垂直方向に 4 ピクセルごとの画素上で位置合わせの可否を調べれば十分であることが実験的に分かっている．すなわち，画素数 $N = \frac{320 \times 240}{4 \times 4} = 4800$ 個について処理を行う計算コストを要する．動き領域分離の計算コストのうち，動き領域併合の演算処理がその大半を占めている．以下，この処理に要する計算コストを見積もる．

領域併合の判定を行う画素間の距離のしきい値 T_d は， $T_d = 8$ ピクセルが最適であると実験的に分かっている．4 ピクセルごとに画素がサンプルされるので，領域併合される可能性がある画素の個数 $r = 5 \times 5 - 1 = 24$ 個である．したがって， $4800 \times 24 = 115200$ 回の領域併合の処理が必要である．上記の計算機環境では，領域併合の処理 150 万回に要する処理時間は実験的に 1 秒であることが確かめられている．すなわち，提案手法による動き領域の分離には 80 ミリ秒の処理時間が必要である．

6.4 処理の高速化のための工夫

6.2 節に示した 5 つのプロセスの流れ処理を単純に実装すると，最も処理速度の遅いプロセスに合わせて，全体のスループットが律速される．実際に，前述の計算機環境での本実装では，動き領域分離のプロセス (3) とパノラマ画像へのマージするプロセス (4) は，位置合わせのプロセス (1) の約 3 倍の処理時間を要する．このため，位置合わせのプロセスは本来処理可能なフレーム枚数の $1/3$ しか処理できず，したがって，時間的に離れたフレーム間の位置合わせを行うことになり，これが精度を低下させる原因になる．

そこで，次のような改良を加える．位置合わせのプロセスが終了した時点で，動き領域分離のプロセ

ス(3)がまだ終了していない場合、新しいフレーム画像を取り込んで次のフレーム間の位置合わせを行う。プロセス(3)が終了していたら、今までに推定した位置合わせパラメータの累積結果と最初と最後のフレーム画像を渡して、それらに基づいて動き領域の分離を行わせる。このようにして位置合わせパラメータ推定の処理速度を低下させずに全体の流れ処理を実現できる。

7 実験

7.1 パノラマ画像の実時間生成と可視化

カメラ映像を入力として、動き領域を分離しつつパノラマ画像を生成・可視化する実験を行なった。実験条件は次の通りである。動き領域の分離では、抽出する領域の個数の上限を $P = 2$ 個として、画素単位マッチングの正規化誤差のしきい値 $T_e = 10.0$ 、領域併合判定の画素間の距離のしきい値 $T_d = 8$ とした。

図4には、動き物体(歩行人物)を追跡するようにカメラをパニングして撮影した入力映像の初期フレームと最終フレームを示し、図5にはその動き領域の分離結果を示し、図6にはその動き領域を分離しつつオンライン生成したパノラマ画像の一例を示した。



図 4: 動き物体を含む映像のフレーム画像



図 5: 動き物体を含む映像(左)とその分離結果(右)

このときの1フレームあたりのプロセスごとの処理時間と処理の流れのスループットを以下の表2に示す。マルチプロセッサPC(CPU×2)2台をパイ



図 6: 動き物体を除去後のパノラマ画像

プライン並列動作させることでスループットを毎秒10-11フレームまで向上できることが確認できた。

プロセス	処理時間
フレーム間の位置合わせ	30 msec
動き領域の分離	80 msec
パノラマ画像へのマージ	80 msec
パノラマ画像の表示	50 msec
合計	240 msec
スループット	80-100 msec

表 2: 本手法の処理時間とスループット

7.2 パノラマ画像+はめ込み映像による可視化

あらかじめ生成しておいたパノラマ画像上に入力映像をはめ込み表示する実験を行なった。図7にパノラマ画像(サイズ 800×220)と入力映像のフレーム画像(サイズ 320×240)を示し、図8にはめ込み表示された映像を示した。初期値パラメータは原点位置から水平、垂直方向に80ピクセルおきに取り、計20個を用意した。入力映像の初期フレーム画像の位置合わせに30ミリ秒 \times 20 = 600ミリ秒の処理時間を要した。それ以降のフレーム画像については、前回の推定結果を初期値として位置合わせを行うので、フレームあたり30ミリ秒で処理できた。

実験で用いた映像には、風景中を人物が移動しながら建物を紹介する様子が撮影されている。パノラマ画像+はめ込み映像を用いた可視化で、周囲の状況をよりわかりやすく伝えられることが分かる。



図 7: パノラマ画像 (上段) と入力映像 (下段)

8 おわりに

本論文では、動き物体を含むカメラ映像からパノラマ画像を実時間生成・可視化するために必要な位置合わせと動き領域の分離の手法について述べた。本手法を PC 上のソフトウェアとして実装して、その有効性と処理の実時間性について良好な結果を得た。また、パノラマ画像+はめこみ映像による可視化の有効性と処理の実時間性を検証した。

今後の課題としては、動き領域の分離の判別基準に位置合わせの成功の可否以外の基準を導入することで、輝度勾配が平坦な動き領域を安定に分離できるように、提案手法を拡張することが挙げられる。

参考文献

- [1] 興梠, 村岡, “グローバルなアフィン動きパラメータの実時間推定手法”, 信学論 (D-II), vol.J82-D-II, No.8, 1999. (掲載予定)
- [2] L.G.Brown, “A survey of image registration techniques”, ACM Comput. Surv. 24, No.4, pp. 325-376, 1992.
- [3] 谷口, 阿久津, 外村, “PanoramaExcerpts: パノラマ画像の自動生成・レイアウトによる映像一覧”, 信学論 (D-II), vol. J82-D-II, No.3, pp. 390-398, 1999.
- [4] 和田, 浮田, 松山, “視点固定型パンチルトズームカメラとその応用”, 信学論 (D-II), vol.J81-D-II, No.6, pp.1182-1193, 1998.
- [5] L. Teodosio, M. Mills, “Salient video still: Content and context preserved”, Proc. ACM Multi-

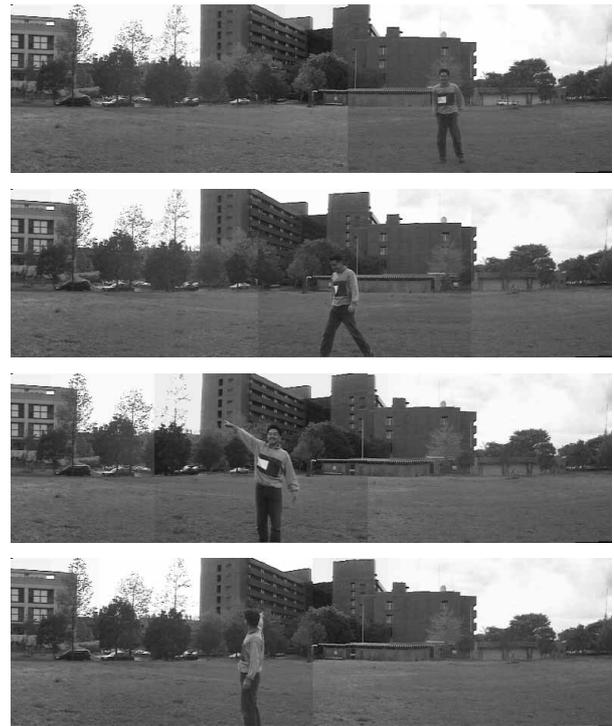


図 8: パノラマ画像+はめこみ映像による可視化例

media '93, pp. 39-46, 1993.

- [6] M. Irani, P. Anandan, and S. Hsu, “Mosaic based representation of video sequences and their applications”, ICCV95, pp.605-611, 1995.
- [7] H. S. Sawhney, S. Ayer, “Compact representation of videos through dominant and multiple motion estimation”, IEEE PAMI, vol.18, No.8, pp. 814-830, 1996.
- [8] H. S. Sawhney, R. Kumar, “True multi-image alignment and its application to mosaicing and lens distortion correction”, IEEE PAMI, vol.21, No.3, pp. 235-243, 1999.
- [9] R. Szeliski, “Video mosaics for virtual environments”, IEEE Computer Graphics and Applications, 16(2):22-30, 1996.
- [10] R. Szeliski, H.Y.Shum, “Creating full view panoramic image mosaic and environment maps”, Proc. SIGGRAPH '97, 1997.
- [11] H. S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon and V. Paragano, “Video-

Brush : Experiences with Consumer Video Mosaicing”, Proc. of Fourth IEEE Workshop on Applications of Computer Vision, pp. 56-62, 1999.

[12] 長坂, 宮武, “輝度投影相関を用いた実時間カメラワーク検出とその応用”, 第4回知能情報メディアシンポジウム 98, pp. 105-110, 1998.

[13] Demin Wang and Limin Wang, “Fast and robust algorithm for global motion estimation”, SPIE pp.1144-1151, 1997.