

人物動作を対象とした非同期撮影された多視点映像の疑似同期の一手法

利田 貴人[†] 東海 彰吾[†] 長谷 博行[†]

[†] 福井大学

〒 910-8507 福井市文京 3-9-1

E-mail: [†]rita@monju.fuis.u-fukui.ac.jp, ^{††}{tokai,hase}@u-fukui.ac.jp

あらまし 同期信号を用いない非同期なカメラ群で撮影した映像を利用する際、同一時刻のフレームを対応付ける必要がある。本研究では、共有する視野に含まれる同一の被写体の動きとその変化を、異なる視点間で対応付けることで、カメラ間のフレームずれ量を算出する方法を提案する。具体的には、各カメラで検出されるオプティカルフローの平均長の時間推移を求め、それらの重ね合わせを評価するものである。さらに、複数台のカメラを用いて多視点撮影実験を行い、スポーツ状況においての本手法の有効性を検証する。

キーワード 多視点映像, オプティカルフロー, フレーム同期

1. はじめに

近年、多視点映像の利用が進んでいる。多視点で撮影された映像群を処理する場合、それぞれの映像のフレーム同士の時間的な対応付け、すなわち、フレーム同期が必要である。ステレオ視など多視点映像情報を統合して利用する場合は、処理を行うフレームの時刻が揃っていないければ意味のある結果を得ることは難しい。

放送用のビデオカメラなどでは、同期信号を用いたハードウェアによるフレーム同期撮影が行われており、得られる多視点映像のフレーム同期が保証される。しかし、我々が普段利用するビデオカメラは、単一視点撮影を前提としていることもあり、通常は同期信号を扱うことはできない。本研究ではこのような非同期のカメラの使用を前提とする。

非同期なカメラ群を利用する場合、カメラの録画処理の開始・停止をリモート制御することで、ある程度の同期撮影が可能である。しかし、カメラ毎の内部処理でおこる時間差の影響などで、実際には数フレーム程度のずれが生じる。そこでここでは、撮影された映像群に対する後処理として、ソフトウェアによる多視点フレームの対応付けを行うことを考える。

基本的には、多視点撮影で共有する視野の中で観測される事象の同時観測性を前提とするものである。例えば、撮影開始後にフラッシュを点灯するなどし、そのフラッシュを検出したフレームを抽出して、対応付けの基準とする方法が考えられる。しかし、フラッシュを利用する場合、広域な状況や明るい屋外での使用が難しくなるなど、適用可能な状況の制約が少なくない。

そこで、本研究では、共有する視野の中で観測される被写体の動きやその変化の同時観測性に着目し、オプティカルフローを利用した方法を提案する。

以下、2 では提案手法の処理の流れを説明し、3 では実際に行った 3 視点のカメラ映像に対する実験結果を示

し、本手法の有効性を検証する。さらに 4 で本報告をまとめる。

2. 擬似的なフレーム同期処理

2.1 基本的な考え方

本研究では、被写体の動きとその変化の同時観測性に着目し、画面内の見かけの動きを抽出するオプティカルフローを利用した方法を提案する。まず、各視点の映像毎にオプティカルフローを抽出し、その平均ベクトルの大きさの時間変化を算出する。次に、得られた時間変化を異なる視点間で対応付ける。これは動きを伴う同一の対象物を観測している場合、その動きの量やその変化のタイミングは異なる視点でも同様に観測されていると考えられるためである。

そこで、対応付けにおいては、一方の視点の映像から得られた時間変化の波形に、別視点の映像から得られた時間変化波形を、その相関が高くなるように時間軸方向にずらしながら重ねて評価し、最も重なるずれ量、すなわち、視点間のフレームの時間的なずれ量を推定するものである。

フレーム同士の対応付けだけでは、完全なフレーム同期ではなく、擬似的なフレーム同期となるが、近年普及している高速度撮影可能なカメラを利用する際、疑似フレーム同期でも十分な利用価値がある。

また、オプティカルフロー抽出を安定に行い、かつ、必要な対象物のフローのみを抽出するため、いくつかの前処理と、不正なフローの除去処理、また、時間変化波形を比較しやすくするための波形の整形処理を組み合わせ処理している。

具体的には、(1) 映像の明度補正処理、(2) オプティカルフローの抽出、(3) フローの時間変化の抽出と整形、(4) 相関を用いたフレームずれ量の算出、の 4 段階の処理で構成される。以下、それぞれの処理を具体的に述べる。

2.2 映像の明度補正処理

本研究では300FPSの高速度撮影映像を利用する。後の処理でオプティカルフロー抽出を行うが、その場合、明度の安定性は正確なフロー抽出のための必要条件である。しかし、高速度撮影映像では、蛍光灯のフリッカーなどの影響による明度変動を無視できない。そこでここでは、前処理として、明度変動を抑える処理を行う。

具体的には、画面の大部分が背景にあたるという前提で、映像全体から作成した各画素の平均明度からなる平均画像の第 i, j 画素の画素値 $y(i, j)$ に対して、第 k フレームの第 i, j 画素の画素値 $x_k(i, j)$ を、

$$y(i, j) \approx a_k x_k(i, j) + b_k \quad (1)$$

とモデル化できるような、 a_k, b_k を、その第 k フレームの全体の画素で成り立つように、最小二乗法で求めることとする。実際には、以下の正規方程式を満たす a_k, b_k を求める。 $\sum \sum$ は i, j を画像から求めた画像の全画素での総和を表す。

$$\begin{pmatrix} \sum \sum x(i, j)^2 & \sum \sum x(i, j) \\ \sum \sum x(i, j) & \sum \sum 1 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} \sum \sum x(i, j)y(i, j) \\ \sum \sum y(i, j) \end{pmatrix} \quad (2)$$

得られた a_k, b_k をそれぞれのフレームに適用して、補正後のフレームを生成し以下の処理で用いる。

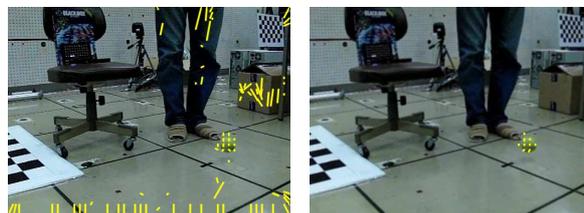
2.3 オプティカルフローの抽出

明度補正後の連続する2フレームから、画面内での対象の見かけの動きを表すオプティカルフローを抽出する。抽出には、ブロックマッチング法を利用する。高速度撮影映像からのオプティカルフロー抽出では、物体の動き自体も比較的小さく、そのようなフローを正確に抽出することが求められる。一方で、シーンの見え方やノイズの影響から不正なマッチングを行ってしまい、誤抽出となる場合も多い。

そこで、ここでは以下の方針で、ノイズの抑制と有効なフローの選択を行う。

- 処理フレームの平滑化：ノイズを抑制するためにここでは 5×5 のガウシアンフィルタを施す。
- フロー長による判定：正確なフロー選択のためにブロックサイズ以上の長さのフローは無視する。
- 近傍フローによる判定：誤抽出防止のために、注目フローの近傍 5×5 ブロックに有効なフローが複数抽出されなければそのフローは無視する。
- 事前知識による判定：対象となるシーン状況がある程度既知と考えて、2次元的なフローの縦横成分の大小など、前提とする対象の動きの事前知識を利用して絞り込む。以下の実験では、縦方向の成分が横方向の成分より大きいことを抽出の条件としている。

具体的な処理結果の違いを図1に示す。図1はボールの落下運動の状況を撮影した連続2フレームよりオプティカルフローを抽出した画像である。画像右下にボールがあるのだが、フロー選択なしでは、本来なら動きのない所にフローが誤検出されている、それ比べてフロー選択ありでは、ボール部分にのみ有効なフローが検出されていることが分かる。



(a) フロー選択なし (b) フロー選択あり

図1 有効なオプティカルフローの抽出

2.4 動きの時間変化の抽出と整形

各フレームに対して得られた有効なオプティカルフローから、それらの平均ベクトルを求めて長さを算出し、時間進行に対して、どのような動きの履歴が観測されたかを抽出する。ボールのバウンドを観測した状況で、具体的に得られた波形を図2に示す。ボールのバウンドに対応した時間変化が観測できていることがわかる。ただし、ボールの位置の軌跡でないことには注意が必要である。

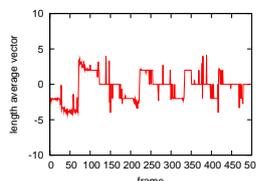


図2 平均ベクトル変動

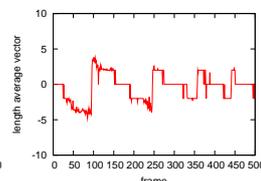


図3 別視点波形

同様にして別の視点から得られた時間変化波形は図3のようになる。両者を見比べると、絶対的な振幅には違いがあるものの、その変化の形状が非常に類似していることが分かる。さらに、両者の間には、時間軸方向のずれが生じていることが分かる。

そこで、まず、ノイズ的な変動成分を除去するため、波形を移動平均処理によって平滑化する(図4左)。さらに、最大振幅が等しくなるように正規化する。結果得られる波形(図4右)を後段の処理に用いる。

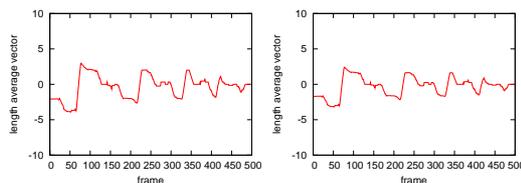


図4 整形の処理結果

2.5 相関を用いたフレームずれ量の算出

2つの波形をフレームずれを考慮して重ね合わせ、両者が最もよく重なるずらし量を算出し、2つの映像のフレームずれとする。このフレームずれがフレームの対応付けの基準となる。具体的には、基準となる波形 $F_0(t)$ に対して、 τ フレームずらした波形 $F_1(t + \tau)$ を考え、両者の差の2乗の総和を最小とするフレームずれ量 τ を求めるものである。

$$\min_{\tau} \sum_t (F_0(t) - F_1(t + \tau))^2 \quad (3)$$

結果として得られる τ フレーム分だけずらした番号のフレームを取り出せば、異なる視点の映像において、時刻が対応するフレームを抽出できる。

3. 実験

前述の方法をスポーツ状況を撮影した映像に適用した実験について述べる。チアリーディングの演技はジャンプやステップ等の急激な動きの変化が多くみられるので、提案する手法を適用する事ができるのではないかと考え実験を行った。実験には、全16台のEX-F1 (CASIO製)を使用し、チアリーダーのダンスを撮影した。各カメラはハイスピードモードでの撮影とし、得られる映像は、512×384画素で300FPSの画像である。この内の3視点の映像を利用する。実験状況を図5に、得られる動画映像の同じフレーム番号の画像を図6に示す。上からカメラ1、カメラ2、カメラ3の画像である。縦方向で画像を見比べると、異なる時刻の画像であることが分かる。

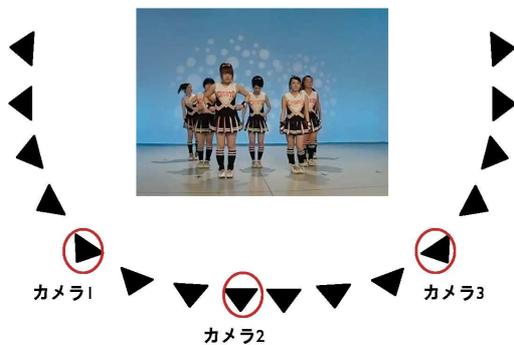


図5 簡単な撮影状況

今回の撮影環境ではフリッカーレスの蛍光灯を使用しているため、本実験では、2.2の映像の明度補正処理は行わない。よって本実験は(2)オプティカルフローの抽出、(3)フローの時間変化の抽出と整形、(4)相関を用いたフレームずれ量の算出の三段階の処理を行う。まず、オプティカルフローの抽出を行う。フロー選択なしと、フロー選択ありの画像を図7に示す。

オプティカルフロー抽出処理によって得られる、2000フレームの平均ベクトル長の時間変化のグラフを図8に示す。上段から、元データ、移動平均処理後、正規化処

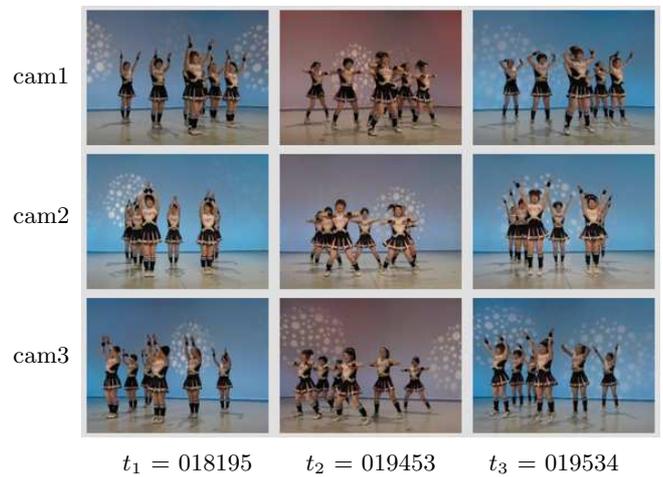
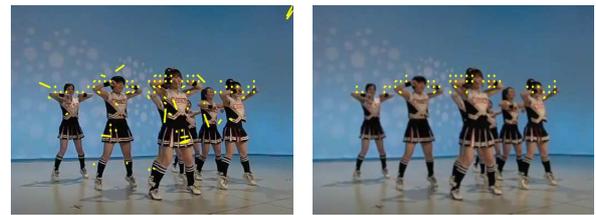


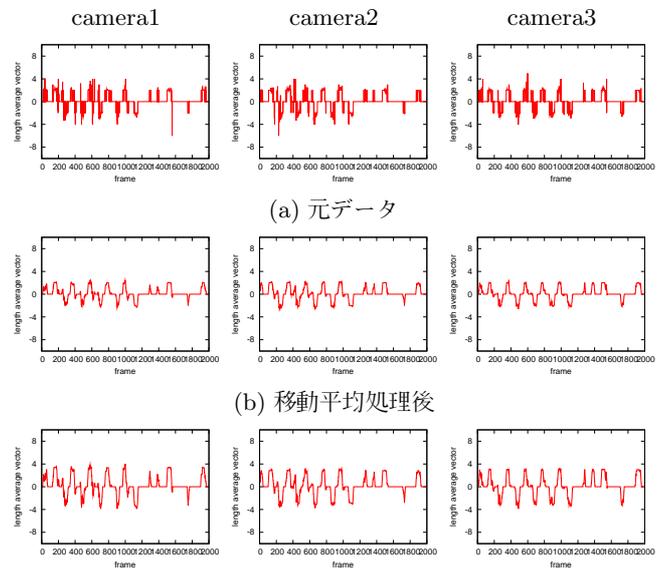
図6 取得した各動画のフレーム



(a) フロー選択なし (b) フロー選択あり

図7 有効なオプティカルフローの抽出

理後を表す。それぞれを見比べると、ほぼ同様の変化を表す波形が得られていることが分かる。



(c) 正規化処理後

図8 波形データ

このカメラ2を基準として、カメラ1とカメラ3について波形の重なりを式(3)で評価した結果が図9である。それぞれの評価値は、二次関数となっておりあるずらし量で最小値をとっていることが分かる。

目視で確認したずれの真値と比較した結果を表1に示す。本実験ではカメラ2とカメラ3の評価結果は真値と

表 1 フレームずれ量

カメラ対	算出したずれ	真値	誤差
camera2-camera1	29	30	-1
camera2-camera3	15	15	0

同じ結果が得られ、カメラ2とカメラ1の評価結果は真値との誤差が-1frameという結果が得られた。真値と同じ値にならなかった原因は、オプティカルフローの抽出の際の情報量が少なかったのではないかと考えられる。しかし、この誤差は300fpsの映像における誤差である。この誤差は一般的なビデオレートである30fpsで考えれば、ずれが生じているか認識できない誤差である。

この値を利用して、フレームを抽出した結果を図10に示す。画像から見てとれるようにほぼ同一時刻のフレームが抽出できていることが分かる。

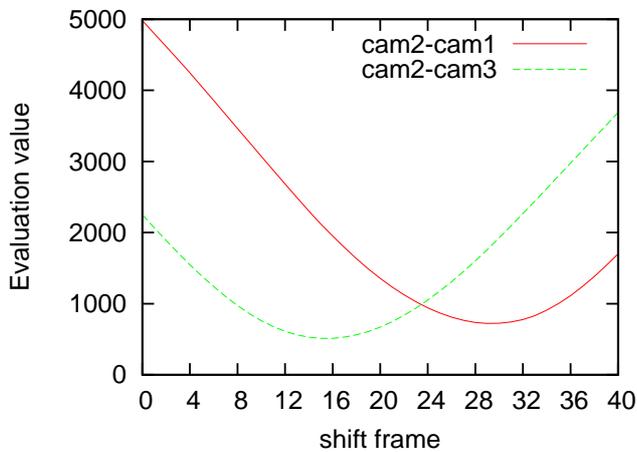


図 9 フレームずらし量に対する評価値の変化



$t_1 = 018195$ $t_2 = 019453$ $t_3 = 019534$

図 10 実験の値を利用して抽出したフレーム

4. ま と め

本文では、非同期なカメラ群による多視点映像撮影状況において、撮影後の映像に写った被写体の動きとその変化の同時観測性に着目し、オプティカルフロー抽出に基づく動き量の時間変化波形を異なる視点間で重ね合わせることによって、時刻ずれを算出する疑似フレーム同期の手法を提案した。

実験では、被写体が複数人であるにもかかわらず精度の高い同期を実現できた。また、本手法は疑似フレーム同期であるものの、ハイスピードカメラなど時間軸方向の分解能が高い映像では、十分に耐える同期が実現できると考えられる。

将来的には手持ちカメラ群への展開も検討しており、カメラ操作に伴う画面内の動きと、有効な対象物体の動きを分離する方法の開発が必要である。

謝辞

本研究の一部は(独)情報通信研究機構の委託研究として行われた(No.143イ105)。

文 献

- [1] 利田貴人, 東海彰吾, 長谷博行, “非同期撮影された多視点映像の疑似同期に関する一手法”, 映像学技報, Vol.34, No.44, ME2010-157, pp.69-72 (2010)
- [2] 奈良先端大 OpenCV プログラミングブック製作チーム, “OpenCV プログラミングブック”, 毎日コミュニケーション出版 (2007)