手持ちカメラを用いた走者ストロボ画像の自動生成*

長谷川 邦洋** 斎藤 英雄**

Automatic Generation of Runner's Stroboscopic Image Using a Handheld Camera

Kunihiro HASEGAWA and Hideo SAITO

This paper discusses the method for the automatically generating stroboscopic image from a video image using a handheld camera comprehensively. The stroboscopic image can be used in order to measure stride length and speed of the runner based on the authors' previously proposed method, in which it is shown that measuring stride length and speed of the runner from the stroboscopic image is theoretically possible. The proposed automatic stroboscopic image generating method has two steps. The first step is generating a stitched background image from input video image. In the second step, runner images extracted from arbitrary frames are overlapped onto the stitched background image. Thus, this method can generate a stroboscopic image including the runners in arbitrary frames without overwriting them in some frames. These processes can be executed automatically, and these are noteworthy features. In the experiments under various environment to show the effectiveness of this method, it was confirmed that could generate a stroboscopic images well in respectively environment. It is also demonstrated that stride length and speed of the runner captured into the generated stroboscopic image can be measured with satisfactory accuracy by manually selected landing points in the stroboscopic image.

Key words: sports vision, stroboscopic image, background image, runner, overlap, handheld camera

1. 序 論

様々なスポーツにおいて、競技に対する多様な科学的解析が行われている。その結果は競技力の向上や、テレビ中継における視聴者への提供情報のひとつとして役立っている。また、解析手法のひとつとして画像を入力としたコンピュータビジョンを用いた方法の研究も盛んである。例えば Hamid ら¹⁾は、サッカーにおいて複数人物のトラッキングを行うことによりオフサイドラインの可視化を可能にしている。次に Lu ら²⁾や Perše ら³⁾はバスケットボール対象として、複数人物の識別とトラッキング、あるいは動きの解析等を行う手法を提案している。また、Bialkowski ら⁴⁾は、フィールドホッケーを題材にチームスポーツにおけるイベントの認識を行っている。さらに、Atmosukarto ら⁵⁾による、アメリカンフットボールにおけるフォーメーションの認識方法の研究を挙げることできる。他にも、Beetz ら⁶⁾は各種スポーツに用いることを想定した、複数プレーヤーやボールをトラッキングし解析モデルを作成するシステムの提案を行っている。

以上のような解析の対象の一つとして陸上競技を挙げることができる。陸上競技において解析する項目は多々存在しており、その解析に対する研究も様々なものが存在している。例えば画像を用いたものとして Yang ら⁷⁾は、足の動きの違いを認識し、走行中、歩行中といった走者の走行状態を認識する方法を提案している。また、様々なセンサを用いた研究も盛んである。例えば、Strohmman ら⁸⁾はランニング中の疲労が走者の動作にどのような変化を与えるかの解析を行っている。これは、走者の全身に 12 個のウェアラブル方位センサを装着することによりランニング中の各関節や腰の位置・角度等の変位を

測定し、その蓄積データを比較することによって実現している。次に Eskofier らは⁹⁾は、加速度センサを用いて走行時の蹴り足の動作の分類を行っている。この分類は靴のフィッティング等に用いることができる。また Oliveira ら¹⁰⁾は、心拍センサを用いて走行中の走者の心拍数の変化を可視化している。

このような陸上競技における解析対象として、上記のもの他にも走者の一定区間毎の速度や、走者の歩幅であるストライド、歩数といったものを挙げることができる。これらのデータは全国大会や世界大会などの大きな大会の際に収集され、選手の技術向上に役立てられている¹³⁾。また近年ではランニングへの関心の高まりにより、いわゆる市民ランナーと呼ばれる人々の間でもこのような科学的解析に対する関心が高まっている。従って、市民ランナーでも容易に扱える解析手法および解析対象が必要になってきている。以上のような状況に対し、速度とストライドは結果のわかりやすさや可視化の簡単さといった利点を持っている。この速度とストライドの解析に対する既存の方法としては、レーザを用いた測定器¹¹⁾による方法や、先述の研究のようにセンサを、例えば GPS 等のセンサを身体に取り付ける方法¹²⁾等を挙げることができる。しかし、これらは日常使用されないものを用いる、もしくはセンサを身に着ける手間がかかるという課題を持っている。また、先述の研究、例えば Beetz ら⁶⁾が提案したシステム等も存在するが、ストライド計測等の陸上競技に関する解析への応用には向かず、また市民ランナーが利用するために必要と考えられる手軽さもない。この課題に対し筆者らは DIA2009 において、手持ちのビデオカメラ 1 台で走者の速度とストライドを測定する手法を提案した¹⁴⁾。この際、解析のために十分な計測精度を得られることを示したが、接地点の入力は手動で行っていた。つまりその分手間がかかってしまうという課題が残っていた。そこで我々は、手持ちカメラで撮影した動画から接地タイミングの判定、及び接地位置取得の自動化を最終ゴールと考えた。本論

* 原稿受付 平成 26 年 5 月 12 日

掲載決定 平成 26 年 9 月 30 日

** 慶應義塾大学 (神奈川県横浜市港北区日吉 3-14-1)

文では、これらを実現するために最初に解決すべき課題として以下の画像を生成する手法を提案する。

1. 背景のみを抽出したスティッチ画像 (図 1(a))
2. 各フレームの走者を投影したスティッチ画像 (図 1(b))

ここで生成対象となるスティッチ画像とは、共通した領域を持つ複数の画像の貼り合わせを行い、貼り合わせた結果として一つの画像を生成したものである。市販のソフトやカメラ等ではパノラマ画像とも呼ばれており、またその生成機能をスティッチングと呼ぶこともある。また、本提案によって生成されたスティッチ画像は、例えば図 1(b) に示すような、任意のフレームの人物がすべて表示されたものになる。このような画像はストロボスコープを使用して撮影した画像と同様のものであり、ストロボ画像と呼ばれている。従って、本提案手法によりストロボ画像を最終的に生成すると言い換えることができる。なお、図 1(b) では見やすいように数フレームの走者のみを表示しているが、実際の生成結果では全フレームの走者を表示することを想定している。これらは走者の接地タイミング判定や、接地位置計測の自動化に必要なものである。例えば、この 2 つの画像間での差分を取ることで、走者のみを抽出した画像を作成することができる。そして、その画像を解析することにより、先述の接地タイミング判定の課題を解決できると考えている。ここで、ストロボ画像を生成するには様々な方法が考えられる。ストロボスコープのような機器を利用するのもひとつの手段であるが、わざわざカメラ以外の機器を用意しなければならないため、容易に扱える手法を作るという観点にはそぐわない。従って、カメラとその画像に対する処理のみでストロボ画像を生成することが望ましい。画像に対して何らかの処理を行い、ストロボ画像を生成する手法は、例えばグラフカットと勾配空間を利用した方法¹⁵⁾や時空間画像を利用して作成する方法¹⁶⁾が存在しているが、我々は先述のとおりスティッチ画像を利用する方法を選択した。これは、フリーハンドのカメラで撮影された動画にも比較的容易に適用できること、既提案の手法¹⁴⁾において、スティッチ画像の生成に必要なフレーム間の Homography を算出する処理を既に導入していたことから、既提案手法の無理のない発展のさせ方として最良であると判断したためである。

本論文では、2 章において筆者らが構想しているシステムの概要と、DIA2009 で筆者らが提案した手法¹⁴⁾についての概略を述べる。そして、3 章では本論文で提案する背景のみを抽出したスティッチ画像の生成法と、その上に走者の画像を重畳する手法について説明する。また 4 章で各手法の実験結果を示し、それに対する考察を 5 章で議論、6 章において本論文の成果をまとめる。

Fig.1 Generation result images

2. 速度・ストライド測定システム

2.1 全体像

ここでは、我々が完成を目指しているシステムの全体像および今回提案する手法が担う部分を明確にする。1 章で述べたとおり、我々の最終目標は手持ちカメラで撮影した動画から接地タイミングの判定、及び接地位置取得の自動化である。この目標を達成するために開発を行っているシステム全体の処理の流れを示したものが図 2 である。この図で示すとおり、本システムは走者のストロボ画像生成、接地タイミングの判定、接地位置の計測の 3 要素からなっている。このうち、接地位置の計測に関しては先述の通り、筆者らが以前提案した手法を適用することが出来る。一方、今回提案するのは前処理部分にあたる走者のストロボ画像生成である。先述の通り、ここで生成されるストロボ画像を利用して残る接地タイミングの判定を行う。

計測のための撮影は、図 3 のように走るコースを一定間隔に一直線に置いたマーカ 2 列で示し、その間をランナーが一方向に走るという形で行った。このようにマーカをおくことは陸上競技の練習において日常的に行われていることであるため、陸上競技選手の速度、ストライドを測定するという目的に対して特段追加の手間がかかるわけではなく、無理のない状況設定となっている。ただし、マーカを置く位置が大きすぎると当然計測精度に悪影響を及ぼすため、設置の際にはメジャー等距離を正確に測れるものを用いるべきである。本提案および DIA2009 での提案¹⁴⁾の際の撮影では、市販のメジャーを用いて 1cm 単位の精度でマーカの設置を行った。なお、今回は競技場やグラウンド等で練習を行っているシーンを対象として想定しているため、背景には他の人物がいない、少なくとも人間であると明確に認識できる大きさ、形では存在していないと仮定している。さらに、このシーンを撮影する撮影者は 1 箇所にとどまり、手持ちのカメラを自由に動かして走者をフレームアウトしないように、できるだけ画面の中央に映るように追跡して撮影を行っている。使用したカメラは市販のデジタルビデオカメラもしくはデジタルスチルカメラである。画像サイズは 720 × 480 ピクセルもしくは 640 × 480 ピクセルであり、フレームレートは 29.97fps である。この撮影の際には、走者を後述の HOG ディスクリプタで抽出できる程度の大きさで映されている必要がある。今回はおよそ 100 × 150 ピクセルから 150 × 200 ピクセル程度の大きさで走者が捉えられている動画画像を使用した。このサイズから多少外れても問題はないと思われるが、極端に大きく、もしくは小さく捉えられてしまうと、HOG ディスクリプタで抽出できなくなったり、スティッチ画像を生成するための特徴点が取れなくなってしまう。この撮影法については、課題設定から考えると撮影者が走者を自転車等で追走して撮影することも考えられる。しかし撮影場所によっては、例えば競技場ではそのような撮影を行うことが現実的でないため、先述の方法を採用している。撮影時に撮影者は、走行シーン全体において走者を先述の大きさで映すことができるよう、走路の中央から数十 m 離れた位置に立ち、ズームを使わずに撮影を行った。また、走者の服装や撮影時の背景については、実利用シーンを考え、背景と服装の色が完全に同色にならないこと以外の制約はつけていない。

以下、本章の残りで接地位置の計測について、次章においてストロボ画像の生成について述べる。

Fig.2 Flowchart of overall the system

Fig.3 Image of the shooting environment

2.2 接地位置計測手法

計測はまず、このマーカを位置が既知な特徴点として画像座標系—世界座標系間の Homography を算出する。そしてフレームを進めていく中で走者が接地しているシーンが現れた際には、手動で接地位置を指定し、この Homography を用いて世界座標系へと投影することによって実現している。また、このマーカを自動追跡するために第 1 フレームにおいて手動で特徴点の位置を入力する。そして次フレームからは、隣接するフレーム間で Homography を算出して前フレームの特徴点の位置を現フレームへと投影することにより追跡を行い、フレームアウトしたら追跡を中止する。フレーム間 Homography の算出には、地面上の SIFT 特徴量¹⁷⁾を使用し、RANSAC¹⁸⁾により誤対応点を除去している。この提案手法の接地位置の計測精度を評価するため、この方法により得られた接地位置と、市販のメジャーを用い手動で 1cm 単位の計測を行った結果を比較したところ、平均誤差が 0.22~0.16m となった。また、この接地位置から 5m ごとの平均ストライドを算出したところ、その誤差は約 0.04m となった。なお、この 5m ごとの平均ストライドを算出しているのは文献 13 等で行われている解析方法に従ったためである。さらに、走りの解析を行う際には算出されたデータの変化傾向が重要となってくる。そのために、算出されたストライドの変化をグラフにしたものが図 4 である。このグラフから、算出データと正解データとの間に多少の誤差はあるものの、変化傾向としてはほとんど一致している。従って、走者の解析のために十分な計測精度が得ることができたと考えている。

本論文では、1 章でも触れたとおりこの手法の自動化を実現するための第一段階として、走者の走行全範囲に渡って接地位置の検出に利用可能な画像の生成を行った。次章において走者のストロボ画像を生成する手法を提案する。

Fig.4 Change of a runner's stride length

3. 提案手法

3.1 手法概要

本論文で提案する手法では走者のストロボ画像を生成することが最終目標である。そのためには、任意フレームの走者が生成されたスティッチ画像に表示されている必要がある。しかし、既存のスティッチ画像生成方法¹⁹⁾では図 5 に示す通り、図 5(a) に示す入力フレームの走者の一部が、図 5(b) に示す生成画像で消えてしまうという問題が発生する。これは、画像を順次重ねているため、前のフレームで走者が存在している部分に後から入力されたフレームが重なっていると、その部分は上書きされてしまうためである。そこで、本論文ではその問題点の解決を試みた。

図 6 に、提案する走者のストロボ画像生成法全体のフローチャートを示す。この図で示すとおり、まず背景のみのスティッチ画像を生成し、次にその画像上に走者を投影するという方法によって、先述の問題をクリアしている。前者に関しては Cho らが提案している Mean-Shift を用いた背景画像生成方法²⁰⁾を利用した。また後者に対しては、HOG ディスクリプタを利用した人物検出²¹⁾によって各フレームから走者を抽出した後、スティッチ画像に投影することによってそれぞれ実現している。なお、処理方法としては先に人物検出を行い、その位置をスティッチ画像上にマスクとして投影した上で背景画像を生成するという方法も考えられる。しかしこの場合には、人物を正しく検出できなかった場合に背景画像をきれいに生成できなくなってしまうため、本提案手法の順番で処理を行った。今回の提案のスコップ外にはなるが、本提案手法において人物の検出を正しく行えなかった場合には、1,2 フレーム程度であれば前後のフレームからその位置を推測することが可能であるため、その後の処理も問題なく行えると考えている。

3.2 背景抽出スティッチ画像生成

本手法の第一段階として背景のみを抽出したスティッチ画像を生成する。ここでは先ほども述べたとおり、Cho らが提案した手法²⁰⁾を用いている。この手法のフローチャートを示したものが図 7 である。まず入力された動画の各フレームから特徴点群を抽出する。そしてその特徴点群を基に各フレーム間の Homography を求め、スティッチ画像を生成する。特徴点には実用上背景が複雑な屋外での撮影に対応できる必要があるため、そのようなシーンでも使え、かつ処理速度も速い SURF²²⁾を用いている。また、スティッチ画像の生成には OpenCV の Stitch 関数を用いている。

次に背景のみを抽出したスティッチ画像を生成する。ここでは、スティッチ画像の各画素に対し、その画素に投影される元画像群の画素値をサンプル点として Mean-Shift²³⁾²⁴⁾による背景の推定を行うことによって背景の抽出を実現している。こ

(a) Input images

(b) Generated image

Fig.5 Result of the generating stitched image by the existing method

Fig.6 Flowchart of a runner's stroboscopic image generation

これは、動体がある一定の場所に存在している時間は、その場所に存在していない時間よりも短いという仮定に基づくものである。つまり、各フレームからは動体による画素値よりも背景による画素値のほうが多くスティッチ画像に投影されるため、Mean-Shift を用いれば背景が抽出されるということである。これを模式的に表した図が図 8 である。この図上の現実世界、各フレーム、スティッチ画像のそれぞれに描かれている点は同じ位置を示している。この位置には、大半のフレームでは背景が映っているが一部のフレーム、この図ではフレーム 2 において走者が映っている。従って、そのままスティッチ画像を生成すると走者の画素値がブレンドされてしまい、背景が描かれなくなってしまう。そこでこの位置が映っているすべてのフレームから対応する画素値を取得し、それらに対して Mean-Shift 推定を行う。すると背景を示す画素値の方が多いため、結果として背景を示す画素値を得ることができる。もちろん先述の仮定が成立しない場合、例えば動体がある一定の場所にとどまっている場合はこの手法を用いることは出来ない。しかし今回は走者が対象であるため、この仮定は基本的に成立すると考えて問題ない。

Mean-Shift による背景画素決定処理の詳細は以下の通りである。まず、初期値として投影される元画像の画素値を代入する。次に以下の式 (1) によって画素値の更新を行う。

$$y_{t+1} = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{y_t - x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{y_t - x_i}{h}\right\|^2\right)} \quad (1)$$

ここで x_i はフレーム i における画素値、 $g(x)$ はカーネル関数、 h はバンド幅である。カーネル関数は Cho ら²⁰⁾の実験と同様 Epanechnikov カーネルを用いている。この計算を、更新幅が一定値以下になるまで繰り返す。さらに、上記計算を対象画素に投影される元画像の画素値それぞれを初期値にして行うことにより、複数の背景画素の候補値が算出されることになる。これらに対し重み付けを行い、一番重みが大いものを背景画素として採用する。なお重み付けについてはガウス分布を利用し以下の式 (2) で算出している。

$$w_j = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(m_j - \mu)^2}{2\sigma^2}\right) \quad (2)$$

なお、 w_j は候補値 m_j に対する重み、 σ は候補値の標準偏差、 μ は候補値の平均である。

Fig.7 Flowchart of a generation of stitched image extracted only a background

Fig.8 Image of background extraction processing

3.3 走者重畳

次に各フレームの走者をスティッチ画像に重畳する方法について説明する。まずこの処理のフローチャートを図9に示す。最初に、入力動画の各フレームから人物領域を抽出する。人物を抽出する方法は、例えば色情報を使うもの²⁵⁾や幾何学情報を使うもの²⁶⁾が存在するが、ここでは走行中という動きが比較的激しいシーンが対象であることや、服や背景に特に制限を設けていないことから、このようなシーンでも安定に人物抽出可能な手法として広く知られているHOGディスクリプタ²¹⁾を用いた。

次にこの領域を3.2節で生成した背景抽出スティッチ画像に投影する。ここで投影のための計算には、3.2節で算出した各フレームからスティッチ画像を生成する際のHomographyを用いている。これは、Mean-Shiftによって背景のみを抽出した画像を作成しても、元画像とスティッチ画像間の投影位置の対応関係は崩れないことを利用している。従って、上記の方法で人物領域をスティッチ画像上の正しい位置に投影することが可能である。これを模式的に示した図が図10である。この図では現実世界を撮影した n フレーム分の画像からスティッチ画像を生成する例を示している。既存の手法では図10の左下に示すように一部のフレームの走者が部分的に、あるいは全部消えてしまう。そこで上述の方法により、まず枠で囲んだ人物領域を抽出する。そしてこの領域をスティッチ画像に投影することで、人物領域がスティッチ画像上に書き込まれる。これにより、図10の右下に示すように、任意のフレームの人物全体を表示することが可能になる。

Fig.9 Flowchart of runner's overlapping processing

Fig.10 Image of runner's overlapping processing

4. 実験

4.1 実験概要

本章では、前章で触れた2つの課題解決法に関する実験、および文献14で提案したシステムと組み合わせて接地位置の測定を行う実験について説明する。

まず、前章の内容に対する実験について説明する。ここでは、それぞれの動作を示すために背景抽出スティッチ画像生成、走者重畳を別々に処理した。双方とも入力としては市販のビデオカメラ、もしくはデジタルカメラで人物が走っている場面を撮影した動画から、数フレームを抽出したものをを用いた。最初に述べたとおり、撮影時のカメラは三脚等で固定することなく手で保持し、走者を追うようにカメラを動かすのみである。また、ある特定の環境下でなくとも利用できることを示すために、地面がコンクリート、土、草である3シーンで撮影を行った。この入力画像群の一部を図11に示す。また、地面がコンクリートの際の入力画像群の一部として図5(a)を利用している。フレーム数はそれぞれ20フレーム分を用いた。

次に、接地位置の測定実験について説明する。これは、今回提案した手法で生成されたストロボ画像を入力として接地位置の測定を行うものである。つまり、図2で示した測定システムのフローのうち、走者のストロボ画像生成部分と接地位置の計測部分を組みあわせて正しい動作をするか確認するものである。そのため、開発前の接地タイミング判定部分は手動で行っている。また、入力画像群の一部を図12に示す。この動画像に関しては、インタレースを解除したものを使用していたため、フレームレートは14.985fpsとなっている。ここでは、走者が30m走ったシーンを撮影した約7秒程の動画像1本を入力とした。この際、走者はスタート時を含め18回接地していたので、測定データは18個得られたことになる。なお、実装時のIDEとしてMicrosoft Visual Studio 2010、言語としてC++、画像処理用ライブラリとしてOpenCV 2.4.6をそれぞれ用いた。また、実験に用いたPCのスペックはCPU:2.40GHz/4コア、メモリ:8.00GBであった。

4.2 背景抽出スティッチ画像生成

まず、背景のみを抽出したスティッチ画像の生成実験について述べる。前節でも示した図11および図5(a)を入力画像の一

Fig.11 Example of input images

Fig.12 Example of input images for the measurement experiment

部として使用した実験結果を示したものが図 13 である。それぞれ図 13(a),13(c),13(e) がコンクリート, 土, 芝の地面のシーンで OpenCV の Stitch 関数のみを使用してにスティッチ画像を生成したもの, 図 13(b),13(d),13(f) が同じシーンで提案手法を用い, 背景を抽出したスティッチ画像を生成したものである。投影時の計算等の影響で多少ノイズが乗ってしまっている部分もあるが, 動体である走者に関してはどの画像もほぼ消えていることを確認できる。なお, 本実験では 4.1 節でも述べたとおり, 特定の環境下でなくとも, つまり背景が異なるテクスチャでも提案手法が適切に機能することを確認するために, 複数の入力画像を用いて有効性を確認した。図 13(f) のみ最初のフレームの走者の靴と思われる部分が赤丸の位置に残っているが, これは今回使用した動画画像が走っている途中からのものであったためであると考えられる。つまり, その部分がすぐに映らなくなってしまい, 十分な数の背景を示す画素値が得られず, 走者の一部が残ってしまったということである。この点に関しては撮影時に撮り方次第で回避できると考えられる。

4.3 走者重畳

次に, 走者を重畳した画像の生成実験について述べる。図 14 が実験結果を示したものである。なお重畳する走者のいるフレームは前節と同様に図 11 と図 5(a) に示したものをを用いた。また, 比較対象である既存手法でスティッチ画像を生成した結果は 4.2 節と同様であるため省略する。図 14(a),14(b),14(c) がコンクリート, 土, 芝の地面のシーンで提案手法を用い, 各フレームの走者を重畳したものである。既存手法によるスティッチ画像では一部のフレームの走者のみ表示されているのに対し, 提案手法で生成した画像にはどれも全走者が欠けることなく表示されていることがわかる。従って, 4.2 節と同じく様々な環境下での提案手法の有効性を確認できた。これにより, 走者のストロボ画像を生成することができたといえる。

4.4 接地位置計測

最後に, ストロボ画像を入力とした接地位置の計測実験について述べる。この実験により計測された接地位置を可視化した, つまり走者の足跡を表したものが図 15 である。これは, 文献 14 の方法により計測された接地位置を画像のスケールに変換し, その位置を青丸で表現したものである。この図によって走りの変化を直感的につかむことができる。また, この手法による, 18 回の接地位置の計測結果の誤差は平均 0.14m, 最大 0.61m, 最小 0.01m, 標準偏差 0.14m となった。また, 5m 毎の平均ストライドの算出結果の誤差は平均 0.07m となった。これは文献 14 の実験とほぼ同じ精度である。このように, 走者のストロボ画像生成部分と接地位置の計測部分を組み合わせることにより適切な結果を得られることを確認できた。

5. 考 察

前章の実験結果より, 提案手法が 3 章で提示した課題を解決できることを確認した。これにより, 図 2 に示した走者の速度とストライドを測定するシステムを構成する要素のうち, 走

(a) Existing method - Concrete ground

(b) Proposed method - Concrete ground

(c) Existing method - Dirt ground

(d) Proposed method - Dirt ground

(e) Existing method - Lawn grass ground

(f) Proposed method - Lawn grass ground

Fig.13 Experiment results of a generation of stitched image extracted only a background

(a) Proposal method - Concrete ground

(b) Proposal method - Dirt ground

(c) Proposal method - Lawn grass ground

Fig.14 Experiment results of runner's overlapping

Fig.15 Visualization of a runner's grounding position

者のストロボ画像生成が完成したといえる。このストロボ画像と背景のみを抽出した画像を用いれば、各フレームの走者を同一座標系に重畳した画像を生成することが出来る。そしてこの画像を解析することによって接地タイミングの自動判定を行えると考えている。ただし、本提案手法は基本的に 2.1 節で述べた条件下・方法で撮影を行うことを前提としている。従って、この条件・方法から外れてしまうとうまく処理が行われないので、その点には留意する必要がある。またその他にも、カメラを速く動かしすぎた場合も考えられる。これはカメラを速く動かしすぎたために画像にブレが生じ特徴点を取れない、もしくはスティッチ画像が作れるほどの共通部分がフレーム間で現れ

ないということである。ただし、本手法はあくまで人間が走る場面を止まっている人物が撮影している場合をユースケースとして想定している。従ってこの問題が生じるほどの速度でカメラを動かす場面はまず起こらないと考えてよい。

本手法を適用するにはストロボ画像を生成する基となるビデオ画像のフレームレートも重要な要素になると考えられる。なぜなら、フレーム間隔が大きすぎると解析のための十分な計測精度を得られないためである。この点については、筆者らが文献 14 で提案した手法における実験がひとつの参考になる。この際実験に用いた動画は、フレームレートが 14.985fps であった。解析はこの動画から得られたデータに対して行い、速度やストライドの変化の傾向をつかむことができた。従って、本手法においても 14.985fps 以上のフレームレートで撮影された動画あれば問題ないと考えられる。

フレームレートに関しては、背景を抽出する部分でも検討したい。背景を抽出するためには、3.2 節で述べたとおり、ある画素に対し走者の映っているフレーム数 < 背景の映っているフレーム数が成立していればよい。従って、最小では走者が 1 フレーム、背景が 2 フレーム映っていればよいことになる。この条件を満たせるフレームレートについて 2.1 節で述べた条件を元に考えてみる。今回使用した動画を調べてみると、ある一箇所がフレームインしてからアウトするまでの最小のフレーム数は 36 フレームであった。動画は 29.97fps で撮影されていたため、このフレーム数を時間に換算すると約 1.2 秒ということになる。従って、約 1.2 秒の間に 3 回映されていれば上記条件を満たせる。このことから約 1.6fps で撮影できれば理論上最低条件をクリアして背景画像を生成できるということになる。また、基本的に今回の撮影条件に従っている限りは、走行スピードが速くなるほどフレームレートも速くする必要があり、遅くなるほど遅くても問題ないと考えられる。これは、人物をフレーム中央に捉えるように撮影している限りは、スティッチ画像内の特定の画素に対し背景⇒走者⇒背景といった形で表示されるため、走者の映っているフレーム数 < 背景の映っているフレーム数が成立するためである。ただし、一箇所に滞留していたり途中で逆方向に動き出すなどしてしまうとこの限りではない。

また、本手法の計算時間については、現状最も時間がかかる場合で 2 時間程度を要している。その大半は背景抽出部分である。これは、1 ピクセル毎に投影される全フレームの画素を用いて Mean-Shift 推定を行っていることが主な要因であると思われる。従ってこの処理方法を改善することにより、処理速度を速くすることができると考えられる。具体的には、処理対象を人物がいる部分のみにする、入力フレームを間引いて背景抽出を行うといったことを挙げることができる。

最後に、本章の冒頭でも触れた、考案中のシステムの残りの部分である、今回得られた結果を用いた接地タイミングの取得方法の理論について説明する。まず、背景抽出スティッチ画像に人物領域を重畳した画像と、背景抽出スティッチ画像との差分を取る。これにより、重畳された各フレームの人物のみが丁度走った軌跡として残ることになる。無論、人物輪郭の抽出手法は様々なものが提案されているが、この方法を取ることにによりより正確かつ簡単に人物を抽出することができる。そして、この人物軌跡を解析することにより接地タイミングを取得できる。解析方法としては、人物の上下動を見る、前後のフレームでの走者の足の重なり具合などを見るといったことを考えてい

る。例えば前者の場合には、描かれた軌跡の中から人物の上下動の変化が降下から上昇に変わるタイミングを探し出し、それらを接地タイミングとする。また後者の場合には、接地時に走者の足元はほとんど動かないことを利用し、フレーム間で走者の足が重なる面積が大きいところを接地タイミングとする。これらの方法は、上述のストロボ画像をうまく作れないパターンでの撮影にしなければ基本的に適用可能であると考えている。また、詳細な方法に関しては今後検討する予定である。

6. ま と め

本論文では、手持ちのビデオカメラ1台で走者の速度とストライドを測定するシステムの実現のために必要な、走行フォームのストロボ画像自動生成手法の提案を行った。手法の実現のためには、Mean-Shiftを利用した背景抽出画像生成と、その画像上にHOGディスクリプタを使用して抽出した走者領域の重畳を行った。また、提案手法を用いた実験を行った結果、本手法の有用性を確認することが出来た。今後は本手法によって生成されたストロボ画像を利用した、接地タイミングの自動判定手法を開発する予定である。

参 考 文 献

- 1) R. Hamid, R. K. Kumar, M. Grundmann, K. Kim, I. Essa and J. Hodgins: Player localization using multiple static cameras for sports visualization, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2010), 731.
- 2) Wei-Lwun Lu, Jo-Anne Ting, J. J. Little, and K.P. Murphy: Learning to Track and Identify Players from Broadcast Sports Videos, IEEE Transactions on Pattern Analysis and Machine Intelligence, **35**, 7, (2013), 1704.
- 3) M. Perše, M. Kristan, S. Kovačič, G. Vučković and J. Perš: A trajectory-based analysis of coordinated team activity in a basketball game, Computer Vision and Image Understanding (CVIU), **113**, 5, (2009), 612.
- 4) A. Bialkowski, P. Lucey, P. Carr, S. Denman, I. Matthews and S. Sridharan: Recognising Team Activities from Noisy Data, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2013), 984.
- 5) I. Atmosukarto, B. Ghanem, S. Ahuja, K. Muthuswamy and N. Ahuja, : Automatic Recognition of Offensive Team Formation in American Football Plays, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2013), 991.
- 6) M. Beetz, N. v. Hoyningen-Huene, B. Kirchlechner, S. Gedikli, F. Siles, M. Durus and M. Lames: ASpoGAMo: Automated Sports Game Analysis Models, International Journal of Computer Science in Sport, **8**, 1, (2009), 1.
- 7) Yang Jinyan, Xu Guanlei and Li Yu: Running state recognition in videos via frames' frequency and positions of two feet, Fourth International Conference on Intelligent Control and Information Processing (ICICIP), (2013), 310.
- 8) C. Strohmman, H. Harms, C. Kappeler-Setz and G. Troster: Monitoring Kinematic Changes With Fatigue in Running Using Body-Worn Sensors, IEEE Transactions on Information Technology in Biomedicine, **16**, 5, (2012), 983.
- 9) B. M. Eskofier, E. Musho and H. Schlarb: Pattern classification of foot strike type using body worn accelerometers, IEEE International Conference on Body Sensor Networks (BSN), (2013), 6.
- 10) G. Oliveira, J. Comba, R. Torchelsen, M. Padilha and C. Silva: Visualizing Running Races through the Multivariate Time-Series of Multiple Runners, 26th SIBGRAPI - Conference on Graphics, Patterns and Images, (2013), 5.
- 11) LDM300C-Sport: <http://www.4assist.co.jp/lm300c/lm300c-001.html> 2014.4.12 アクセス
- 12) 長嶋拓哉, 高橋隆行: GPS 搬送波位相測位を用いた陸上競技者ピッチ・ストライド計測システムの開発とそれを用いた陸上競技者計測(動作の計測および定量化), 日本機械学会スポーツ・アンド・ヒューマン・ダイナミクス講演論文集, (2011), 84.
- 13) Korean Society of Sport Biomechanics: Biomechanics Research Project in the IAAF World Championships Daegu 2011, <http://www.iaaf.org/news/news/biomechanics-research-iaaf-world-championsh> 2014.4.12 アクセス
- 14) 長谷川邦洋, 斎藤英雄: ハンドヘルドカメラを用いた陸上競技選手の接地位置計測に基づく速度・ストライド測定, 動的画像処理実利用ワークショップ(DIA), (2009), 215.
- 15) A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin and M. Cohen: Interactive Digital Photomontage, ACM Transactions on Graphics (TOG), **23**, 3, (2004), 294.
- 16) C. W. Ngo, T. C. Pong, and H. J. Zhang: Motion analysis and segmentation through spatio-temporal slices processing, IEEE Transactions on Image Processing, **12**, 3, (2003), 341.
- 17) D. G. Lowe: Object recognition from local scale-invariant features, IEEE International Conference on Computer Vision (ICCV), (1999), 1150.
- 18) M. A. Fischler and R. C. Bolles: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communication of the ACM, **24**, 6, (1981), 381.
- 19) Microsoft Corporation: Microsoft Research Image Composite Editor (ICE), <http://research.microsoft.com/en-us/um/redmond/groups/ivm/ICE/> 2014.4.17 アクセス
- 20) S. H. Cho, H. B. Kang: Panoramic background generation using mean-shift in moving camera environment, Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV), (2011), 829.
- 21) N. Dalal and B. Triggs: Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), (2005), 886.
- 22) H. Bay, A. Ess, T. Tuytelaars and L. V. Gool: SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), **110**, 3, (2008), 346.
- 23) K. Fukunaga and L. Hostetler: The estimation of the gradient of a density function, IEEE Transactions on Information Theory, **21**, 1, (1975), 32.
- 24) D. Comaniciu and P. Meer: Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, **24**, 5, (2002), 603.
- 25) P. Dollár, Z. Tu, P. Perona and S. Belongie: Integral Channel Features, British Machine Vision Conference, **2**, 3, (2009), 5.
- 26) D. Hoiem, A. A. Efros, and M. Hebert: Putting objects in perspective, International Journal of Computer Vision, **80**, 1, (2008), 3.