

記録系競技スポーツ向けの汎用タイムスタンプ分析システム開発

林 勇 樹
追手門学院大学

Development of a general-purpose time stamp analysis system for record-based competitive sports

HAYASHI Yuki
Otemon Gakuin University

要旨

競技スポーツ，特に記録競技においては競技結果が勝敗を決めるが，レースの振り返りやトレーニングの際には，途中の区間タイムも重視される．国際大会・全国大会では運営側が区間タイムを提供する場合もあるが，地方大会など多くの競技会においては，そのような情報は提供されない．本研究では，①誰でも②手元の環境を問わずに③撮影した映像から時間分析可能なツールの開発を通して，選手が賢く強くなる下支えをすることを目的とする．また，競泳競技における本システムの応用事例についての概要を報告する．開発されたシステムを用いた一般レベルの競技者からの利用報告も得ており，本システムの有効性が示された．

Abstract

In competitive sports, especially in record sports, the race results determine the winner. However, when looking back on the race and training, section-time is also important. In international or national competitions, the organizers may provide such information, but it is not provided to the competitors in most domestic local competitions. This research aims to support athletes to become smarter and stronger by supporting the analysis of section-time, which is essential for performance improvement. This research will be conducted through the development of a tool that enables (1) anyone (2) from any environment (3) to analyze the time from the captured video. In addition, report a summary of the application of this system in swimming local competitions. By reports of the usage by general level competitors, it indicates the effectiveness of this system.

キーワード：タイムスタンプ分析，時系列データ，区間タイム，Web アプリ，HTML5，Canvas API

Keywords：Timestamp analysis, Time-series data, Section, Web application, HTML5, Canvas API

1. 背景

競技スポーツの中でも、競泳や陸上、スピードスケートなどは「記録系競技」と呼ばれ、順位は審判員による採点などによらず、純粋に速くゴールした選手が勝利するものである。ゆえに勝敗こそわかりやすいが、1分1秒を縮めるためには様々な要因が考えられ、どこを短縮できるかを総合的に判断しながら選手・指導者は日夜トレーニングに励んでいる。この判断の元になるのが、競技結果より細かくレース内容を分析した「区間タイム」と呼ばれるデータである。陸上競技や競泳競技では10m毎に分析されること [1] [2] が多く、各区間における加減速やストライド（1歩で進む距離）、ストローク長（1かきで進む距離）などが分析されている。スピードスケートでは、施設設備が整っている場合にはさらに詳細な分析がなされること [3] もあり、競技結果だけでなくその過程における内容も重視されていることがわかる。このようなデータは、競技会場に置かれた固定設置ビデオカメラからの映像を元に分析をしたり、観客席などから撮影者が手でビデオカメラを把持してパニングしながら撮影した映像を分析したりして求められる。この分析には従来市販のソフトウェア [4] や、Microsoft Excel マクロ機能を用いた自作アプリケーション [5]、OS に標準で付属の映像再生ソフトウェア [6] などが利用されてきたが、映像をコマ送りに適したファイルフォーマット（AVI形式）に変換する時間的な問題に加えて、費用面での問題や、使用するPCのOSバージョンアップへの非対応などの環境変化に対する弱さが問題であった。このことは、現場でソフトウェアを使用するスタッフの作業負担を高めるだけでなく、分析者の育成にも大きなハードルとなってきた。本研究では、ここで挙げた問題点を解決するべく、①誰でも②手元の環境を問わずに③時間分析が可能なツールを開発し、十分な分析が得られない環境でトレーニングをする一般レベルの競技者であってもハードルなく時間分析に触れられ、競技力向上への足がかりとなるシステムの構築を目的とする。

2. 方法

2.1 システム構成

本研究で開発するシステムは、Windows や macOS などの端末に依存しないことを目的とし、Microsoft

Edge や Google chrome, Apple Safari, Mozilla Firefox などに代表される Web ブラウザを用いるものとした。理由として、Web ブラウザの基礎になっている HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets) などの言語は W3C (World Wide Web Consortium) によって標準化されており、異なる Web ブラウザを用いてアクセスした Web ページが、同じレイアウトで表示され、同じように動作することが期待される。このことは、従来問題であった環境依存を解決するだけでなく、OS 側のアップデートへの対応が不要で、システム本体のメンテナンスのみで済む点が大きな利点である。

2.2 HTML5

HTML5 は、前述の W3C が策定している規格であり、現在リリースされている Web ブラウザのすべてがこの仕様に沿って開発されている。一つ前のバージョンである HTML4 から現行の HTML5 へのアップデートあたり、従来重視されていなかった動的な要素（動画、音声データなど）に対する操作機能が標準搭載され、Web ブラウザによってもバラバラであったメディア対応が行われたことが特筆すべき点である。

2.3 JavaScript

プログラミング言語の一種で、HTML を始めとするマークアップ言語の要素を操作することができる特徴を持つ。HTML や CSS だけでは表現できなかった動的な操作を提供し、クリックやドラッグなどといった GUI 操作に欠かせない動作をトリガーに、様々な操作を可能にするものである。

2.4 HTML5 における Video 要素

HTML5 では、Video 要素を JavaScript から操作するための機能が用意され、読み込み・再生・一時停止・早送りなどの操作が標準で可能になった。本研究では、この Video 要素をシステムのベースに置き、読み込んだ映像ファイルのパラメータ（ファイル名、再生時間、再生速度など）を JavaScript を用いて操作することで、Web ブラウザ上で動作するアプリケーションとして開発を行った。

2.5 表計算ライブラリ Js spreadsheet

Js spreadsheet (旧名 jExcel) は、OSS (Open Source Software) プロジェクトで開発されている JavaScript のライブラリであり、Js spreadsheet のプログラムを読み込むことで、Web ブラウザ上で表計算ソフトと同様に選択・コピー&ペースト・カット・計算・集計などの機能を提供するものである。本研究では、前述の通り HTML5 と JavaScript の機能を用いたシステム開発を行うが、取得された再生時間をもとに汎用のタイムスタンプ分析ツールへ仕上げるためには、Web ブラウザ上に Excel などの表計算ソフトウェアを模擬したシステムを構築する必要があったため、本ライブラリを選定し、システムに組み込んだ。図1は本ライブラリを元にシステムに組み込まれた画面である。表計算ソフトウェアと同等の機能が備わっていることがわかる。

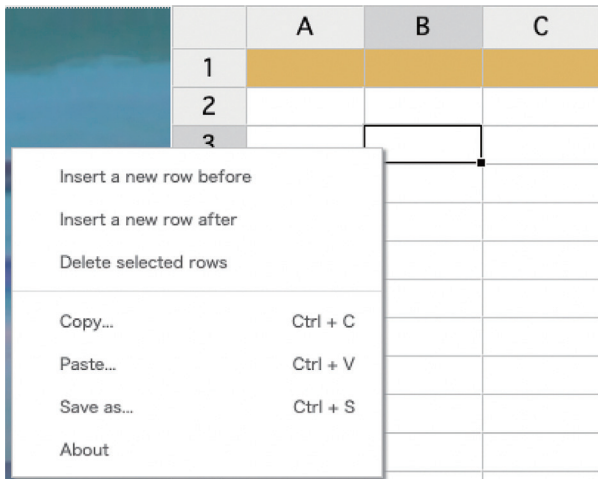


図1 システム上の表計算ライブラリ

2.6 Canvas API

本研究では、さらに Web ブラウザの機能を最大限に活かすため、Canvas API を用いた画面描画システムを組み込むことにした。Canvas API は、アニメーション、データ可視化、写真加工、動画処理などを Web ブラウザ上で実現するためのシステムで、今回のようにユーザが画面上に任意の絵を描画することも可能な技術である。2.4 で述べた video 要素で読み込まれた映像の上に、層構造となるよう重ねた透明のキャンバスを用意することで、ユーザは映像を見て必要な場所に書き込みながら分析作業をすすめることができる。記録競技の区間分析をする上で、この線描画機能は極めて重要で、特にパン

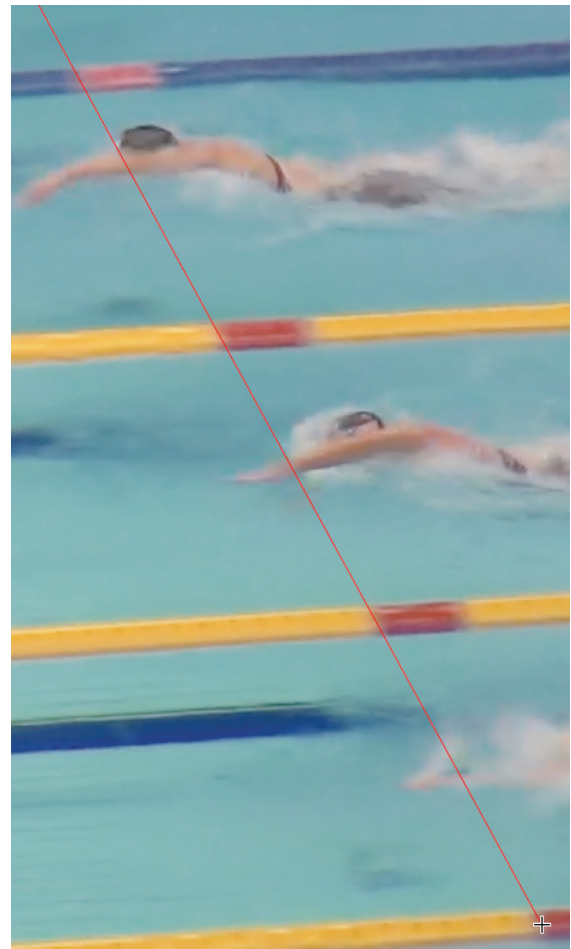


図2 Canvas API による画面描画

ニング撮影された映像を分析する際には、1 フレームごとに異なる画角の中からできる限り正確なフレームを探し出す上で効果を発揮する。図2は競泳競技の分析において、レーンロープの色の境目に線を書き込むことで、正確な通過時間を計測する様子を示したものである。

3. 結果

3.1 開発されたシステムの全容

図3に開発された汎用のタイムスタンプ分析ツールの画面を示す。読み込んだ映像は画面全体に表示され、表1に示すショートカットキーの操作が可能である。

表1に記載のショートカットキーの操作だけで、タイムスタンプ分析に必要なすべての機能を利用できる。

3.2 汎用タイムスタンプ分析

記録競技において、スターターの持つスタート信号発生装置は大抵の場合信号発生時に LED 部分が強く発

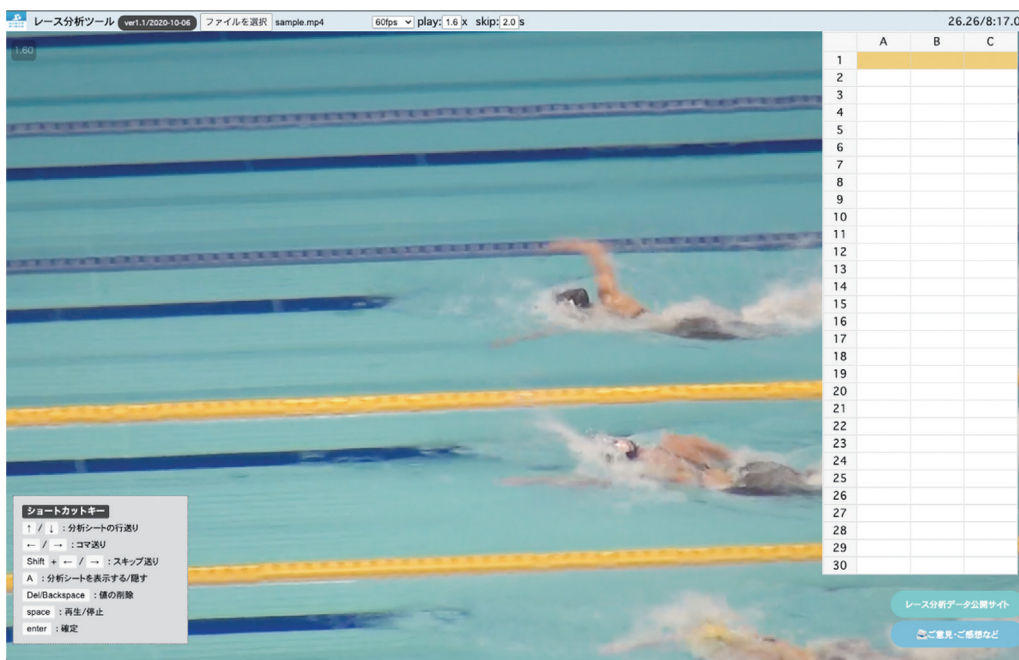


図3 開発システムの画面デザイン

表1 ショートカットキー一覧

キー操作	機能
↑ / ↓	表計算シートの行送り
← / →	映像のコマ送り
Shift + ← / →	映像のスキップ
A	操作パネルの表示 / 非表示
Delete	タイムスタンプの削除
Space	映像の再生 / 一時停止
Enter	タイムスタンプの取得

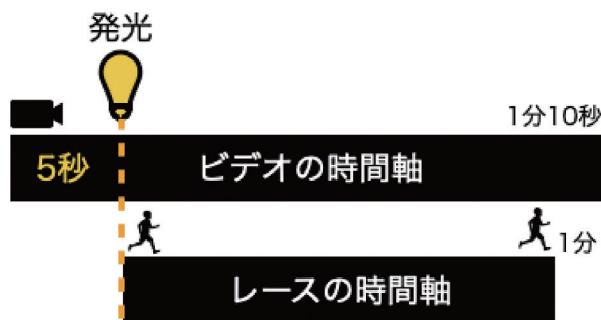


図4 シグナル同期の原理

光する。この発光をビデオカメラで撮影しておくことで、ビデオカメラで撮影された映像の時間軸と、レースの時間軸を一致させることができる。図4のように1分ちょうどのレースが行われたとして、映像が全体で1分10秒、スタート信号が録画開始から5秒ちょうどで発光したとすると、本システムの分析欄にスタート信号発生時間として5秒をマークし、以降の区間分析タイムスタンプの値から適宜5秒を引き算することで、レース内での正しい区間タイムを算出できる。また、表計算プラグインであるため、表内に数式を導入することも可能である。競泳においては通過時間の区間分析に加えてストローク情報を分析する際に3かきごとの平均値を用い

る。たとえば【C3= (B3-B2) /3】(B3セルの値からB2セルの値を引いて3で割った値をC3セルに表示させる操作。B2セルにはストロークの開始タイムスタンプを、B3セルにはストロークの終了タイムスタンプが分析されているものとする。)などの数式を設定したフォーマットを用意しておくことで、分析者のニーズによって様々な活用が考えられる。

3.3 利用者からのフィードバック

本システムは日本水泳連盟科学委員会のサーバ上で稼働[7]させ、営利目的の利用を禁止しつつ、広く一般の利用者(競泳での利用にとどまらず)への展開を想定

して運用を行っている。競泳競技に加えて、陸上競技の分析に利用しているという使用例も報告されており、ニーズに合ったシステム開発が行えたことを示唆している。

3.4 本システムを用いた技術の応用例

本研究で開発されたシステムを基礎にさらに応用して、競泳日本代表選手のサポートにおいては、国内・国外、試合の大小を問わずにスタッフが帯同し、レース後すぐに分析を実施するサポートが行われている。本研究の範疇を超える内容ではあるが、分析データがデータベースに直接送信される機能が追加で開発され、撮影エリアで分析されたデータがすぐさま選手・指導者のもとへ届くシステムの重要な役割を担っている。

4. 考察

4.1 研究目的の達成度合い

本研究では、前述の通り3つの課題解決を目指した。「①誰でも」については、分析ソフトウェアのWebサイト上での一般公開として実現された。「②手元の環境を問わずに」は、Webブラウザベースの開発方針に則り、映像の読み込みから分析データの出力までを一貫して行えるシステムを構築したことで実現された。「③時間分析が可能なツールを開発」については、Webブラウザ上に表計算ソフトを模擬したライブラリを導入することで、決まったフォーマットによらず、分析者自身が分析項目・計算式を用意して利用できるシステムとして実現された。

4.2 開発全体を通して

本研究では、地方競技団体や学校体育、スポーツクラブなどの一般的なレベルの利用者でも簡易にタイムスタンプ分析を利用できる環境を提供することを目的に開発を行った。利用者から得られたコメントからも一定の成果を得られたと考えられる。一方で表計算システム上には、何もプリセットが表示されておらず、利用者が分析に使用したい区間や計算式などの情報を予め設定しておく必要がある。汎用性を求めるために、このような構成にしたが、分析をこれから始めようとするチーム・スタッフにはそれぞれがハードルの高いものになっている可能性がある。こと競泳競技に関しては筆者がカスタマイズした別バージョンを使用することでより分析者が高

速に分析を実施できる環境が整っているが、陸上やスピードスケートなどの別競技に関する知見はまだない。今後は各競技の種目毎の分析プリセットを収集、テンプレート化し、分析前に選択できるような機能を盛り込む形での利便性追求を目指したい。

4.3 OS, ブラウザ間の微妙な差について

本研究の2つ目の目的であった「手元の環境によらずに」は実現されたものの、操作感（動画がスムーズに再生できるかなど）は端末のスペックによる部分がある。現状最も操作が適しているのはmacOS (Apple Silicon M1) のSafariであり、Windows版ではMicrosoft Edgeが比較的良好に動作する。(Chromiumベースの新Edgeでは、一部動作が不安定な場合が報告されている)これらの差異は、Webブラウザのメモリ確保量やキャッシュの扱いによるものだと考えられ、現状映像ファイルサイズは2GB程度が動作限界である認識を持っている。本原稿執筆時点(2022年1月)でも、12月までと1月まででSafariの操作感が向上するなどWebブラウザ自体の性能もアップデートにより変化することがあり、一概にこの構成が望ましいと言えない点は今後の課題だと考える。

5. 結論

HTML5やJavaScriptなどの標準化された規格を用いたWebブラウザをソフトウェアの基盤に用いることで、分析者の利便性と開発側のメンテナンス性をバランスさせることができた。機能面は充実しているが、応用的な利用を提案することが今後の課題である。

6. 謝辞

本研究は公益財団法人日本水泳連盟科学委員会のレース分析プロジェクトを地方展開するための取り組みとして実施され、システム開発においては科学委員長で追手門学院大学の松井健氏をはじめ、レース分析プロジェクトコアメンバーで奈良教育大学の立正伸氏、追手門学院大学の植松梓氏、東京女子体育大学の酒井紳氏、鈴鹿工業高等専門学校の宝来毅氏、朝日大学の高橋篤史氏からのフィードバックを大いに活用させていただきました。この場を借りて深く御礼申し上げます。また、第14回世界選手権(25m)において、HTML5によるWebブラウザベースのシステム開発をすすめる方向性を示して

くれた AWS Japan の今井元太氏（日本水泳連盟科学委員会協力スタッフ）なしでは、このシステム開発は成り立ちませんでした。改めて御礼申し上げます。

参考文献

- [1] 大沼勇人, 小林海, 松林武生, 高橋恭平, 山中亮, 渡辺圭佑, 綿谷貴志, 広川龍太, 2019年度主要競技会における男子100mのレース分析, 公益財団法人日本陸上競技連盟陸上競技研究紀要, 第15巻, p.131-137, 2019
- [2] 野村照夫, 若吉浩二, 奥野景介, 生田泰志, 高木英樹, 後藤慎二, 競泳のレース分析における局面距離変更に伴う問題の検討, 水泳水中運動科学, 第3巻, p.42-46, 2000
- [3] 横澤俊治, 加藤恭章, 紅楳英信, 熊川大介, スピードスケート国際競技会の中長距離レースにおける滑走軌跡と速度の分析, Sports Science in Elite Athlete Support, 第3巻, p.27-38, 2018
- [4] Sérgio Jesus, Mário J. Costa, Daniel A. Marinho, Nuno D. Garrido, António J. Silva, Tiago M. Barbosa, 13th FINA WORLD CHAMPIONSHIP FINALS: STROKE KINEMATICS AND RACE TIMES ACCORDING TO PERFORMANCE, GENDER AND EVENT, International Conference on Biomechanics in Sports, 第29巻, p.275-278, 2011
- [5] 植松梓, ジュニア競泳選手のレース特性の解明～男子100m自由形を対象として～, 上月財団スポーツ研究助成事業(第12回)報告書, 2014
- [6] 津野天兵, 井上智博, 萬久博敏, 和田智仁, 大学水泳部における映像・分析サポートの実践, スポーツパフォーマンス研究, 第8巻, p.216-228, 2016
- [7] レース分析ツール(日本水泳連盟) <https://science.japan-swimming.jp/app/>

7. 付録

Canvas API を用いた画面描画に関するコード

- ・ Canvas 要素は、Video 要素の上に置く (z-index)
- ・ ボタンや表計算部分を Canvas 要素が覆うと操作できなくなるため、範囲を決める
- ・ ドラッグ中のみ描画するコードのため、mousedown 時のカーソル座標値をドラッグ中保持して、現在のカーソル座標値とドラッグ開始時点の座標値との間で直線を描画する
- ・ カーソルを動かすごとに線が描画されるため、カーソル座標値の変化と同時に Canvas 要素を適宜 Clear する必要がある

```

var mousedown = false;
var downstrPos = [0,0];
var downendPos = [0,0];
var currentPos = [0,0];

function clearCanvas(){
  $('#canvas').clearCanvas();
}

$("#canvas").mousedown(function(e){
  if(drawLineVal == 1){
    mousedown = true;
    downstrPos = [e.offsetX,e.offsetY];
  }
});

$("#canvas").mousemove(function(e){
  tmpPointX = e.clientX;
  tmpPointY = e.clientY;
  if(drawLineVal == 1){

    if(mousedown){
      currentPos = [e.offsetX,e.offsetY];
      centerPos = [0,0];
      $('#canvas_tmp').clearCanvas();
      $('#canvas_tmp').drawLine({
        strokeStyle: '#FF0000',
        strokeWidth: 1,
        x1: downstrPos[0], y1: downstrPos[1],
        x2: currentPos[0], y2: currentPos[1]
      });
      centerPos[0] = (downstrPos[0] + currentPos[0])/2;
      centerPos[1] = (downstrPos[1] + currentPos[1])/2;
    }
  }
});

$("#canvas").mouseup(function(e){
  if(drawLineVal == 1){

    if(mousedown){
      downendPos = [e.offsetX,e.offsetY];
      var centerPos = [];
      $('#canvas').drawLine({
        strokeStyle: '#FFF',
        strokeWidth: 1,
        x1: downstrPos[0], y1: downstrPos[1],
        x2: downendPos[0], y2: downendPos[1]
      });
      centerPos[0] = (downstrPos[0] + downendPos[0])/2;
      centerPos[1] = (downstrPos[1] + downendPos[1])/2;
    }
    mousedown = false;
  }
});

```

Video 要素のコントロールに必要なコード

- ・ video.currentTime で取得できる現在時間をコントロールすることで、映像を操作できる
- ・ 時間表記 (0:00.00) に変更するなどの細かい内容は本コードに含んでいない

```

function playVideo(){
  if(video.paused){
    video.play();
    updatect = setInterval(function(){
      var timeline = document.getElementById("t");
    }, 100);
  }else{
    video.pause();
    clearInterval(updatect);
  }
}

function setTimeFormat(time){
  if(time*10%10 == 0){
    time = time+".00";
  }
  var tmptime = time.toString().split(".");
  tmptime[1] = tmptime[1]+"00";
  return tmptime[0]+"."+tmptime[1].slice(0,2);
}

function restart() {
  video.currentTime = 0;
}

function jumpto(sec) {
  video.currentTime = sec;
}

function skip(sec){
  jumpto(video.currentTime+parseFloat(sec));
}

function prev(sec){
  jumpto(video.currentTime-parseFloat(sec));
}

```

