Estimation of Runners' Number of Steps, Stride Length and Speed Transition from Video of a 100-Meter Race

Kentaro Yagi
Keio University
Yokohama, Japan
yagi@hvrl.ics.keio.ac.jp

Kunihiro Hasegawa
Keio University
Yokohama, Japan
hiro@hvrl.ics.keio.ac.jp

Yuta Sugiura
Keio University
Yokohama, Japan
sugiura@keio.jp

Hideo Saito
Keio University
Yokohama, Japan
hs@keio.jp

ABSTRACT

The purpose of this study is sensing movements of 100-m runners from video that is publicly available, for example, Internet broadcasts. Normally, information that can be obtained from a video is limited to the number of steps and average stride length. However, our proposed method makes it possible to measure not only this information, but also time-scale information like every stride length and speed transition from the same input. Our proposed method can be divided into three steps. First, we generate a panoramic image of the 100-m track. By doing this, we can estimate where the runners are running in a frame at the 100-meter scale. Second, we detect whether the runner steps in the frame. For this process, we utilize the detected track lines and leg joint positions of runners. Finally, we project every steps to the overview image of the 100-m track to estimate the stride length at the 100-m scale. In the experiment part, we apply our method to various race videos. We evaluate the accuracy of our method via comparison with the data measured using typical methods. In addition, we evaluate the accuracy of estimation of the number of steps and show visualized runners' steps and speed transitions.

CCS CONCEPTS

• **Computing methodologies → Computer vision**;

KEYWORDS

track and field; a 100-m race; image stitching; analysis of runners

ACM Reference Format:

Kentaro Yagi, Kunihiro Hasegawa, Yuta Sugiura, and Hideo Saito. 2018. Estimation of Runners' Number of Steps, Stride Length and Speed Transition from Video of a 100-Meter Race. In *1st International Workshop on Multimedia Content Analysis in Sports (MMSports'18), October 26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3265845.3265850

MMSports'18, October 26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5981-8/18/10…$15.00
https://doi.org/10.1145/3265845.3265850

1 INTRODUCTION

In recent years, sensing of forms, conditions, or movements of athletes has been actively conducted in various sports. For example, Strohrmann et al. [23] conducted research to measure the athletes' fatigue by putting 12 wearable sensors on the body. Ghasemzadeh et al. [9] made it possible to evaluate golf players' swing by sensing the movement using a sensor attached on the golf club. Beetz et al. [4] proposed a model to estimate player skills, action-selection criteria and player and team strengths and weaknesses using multiple sensors in soccer games. To collect data of moving players of sports, various devices, such as GPS, acceleration sensors, or infrared sensors, are used.

Measurements using images and computer vision techniques are also actively researched. For example, Hamid et al. [11] proposed a method of visualizing offside lines by tracking multiple players for a soccer games. Atmoskaruto et al. [3] proposed a method of recognizing the formation in American football games from an image sequence. Gedas et al. [5] predicted basketball players' movements using first person-view videos using deep learning. Cioppa et al. [8] proposed a method for automatic interpretation of soccer games using image sequences from the main camera, without camera calibration, field homography, player tracking, or ball position estimation.

There are some methods used as broadcasting contents. Owens et al. [21] proposed a method of judging whether the ball is over a line. Currently, this technique is used in real games of tennis, soccer, football, and so on [17]. In addition, there is a system for superimposing the 10 yard line in an American football game [22]. To make the video natural, 10 yard line is superimposed behind players by detecting the field area and players. For alpine skiing, there is a system of superimposing virtual athlete from different races on the video, so that audiences can visually compare the player with virtual player [15]. This system uses the technique to repeat the same trajectory of the camera.

In this paper, we focus on runners' movements in a 100-m race. For sensing track and field athletes, the following methods are mainly used; The laser range finder [2] is the most common measurement method for speed transition. By pointing the laser at the back of the target runner, the user acquires the speed transition.

However, there is a problem in that the user needs to be skilled to measure it accurately. Infrared sensors such as OptiTrack [20], are also used to measure athletes movements. This method obtains accurate three-dimensional positions of the small markers. However, users needs to attach small infrared reflective markers to their entire body; therefore, the effect of attaching these sensors on the performance must be considered. Furthermore, the sensors must be captured from multiple viewpoints. Thus, it is difficult to use this method in a real race. Hobara et al. frame[13, 14] measured runners' movements by counting runners' steps frame by frame. This method only requires race videos. However, the information that can be obtained is limited, since it is impossible to know where runners are running on a 100-m scale from an image sequence.

Data measured by those methods are used for many studies. Hobara et al. [13, 14] analyzed differences in performances of amputee sprinters according to the cutting sites of the legs and ethnicity . They stated that it takes time to obtain sufficient data from video. Therefore, shortening the human labor of measurement is one of the motivation of our research. In addition, the International Association of Athletics Federation (IAAF) and the Japan Association of Athletics Federation (JAAF) measured athletes' movement by combining laser range finder and multiple cameras to obtain runners' stride length and speed transitions for evaluating their performances. As described above, there are various measurement methods used for research or evaluation of athletes' performances.

In addition to the measurements described above, there are some methods that use computer vision techniques. Li et al. [16] proposed the method for detecting and analyzing complex athletes' actions in moving background automatically. Yang et al. [24] detected runners' foot positions from the transition of images frame by frame. Stefano et al. [18] proposed a method of identifying and localizing athletes' positions in video using printed numbers and texts on athletes' uniforms. Hasegawa et al. [12] visualized movement of sports players by generating strobe images from continuous image sequence. However, this method cannot detect the steps of runner automatically. In addition, there is a problem whereby the approach does not work in a textureless environment, such as an athletics stadium.

In this paper, we propose a method for measuring runners' movements using a publicly available video. Normally, information obtained from a race video is limited to the average stride length and number of steps. However, our method makes it possible to obtain not only the number of steps and average stride length but also every stride length and speed transition of athletes with only a video. In addition, the processes of our method are conducted semi-automatically. Therefore, compared with counting the runner's steps manually, we can reduce the measurement time by 75% for one race.

The remainder of the paper comprises four sections. Section 2 explains the details of the algorithms used in our method. In Section 3, we show some results of our method. We apply our method to real race videos such as from the Olympics and world championships. In Section 4, we discuss the results obtained in Section 3 and consider future research. In Section 5, we summarize this paper.

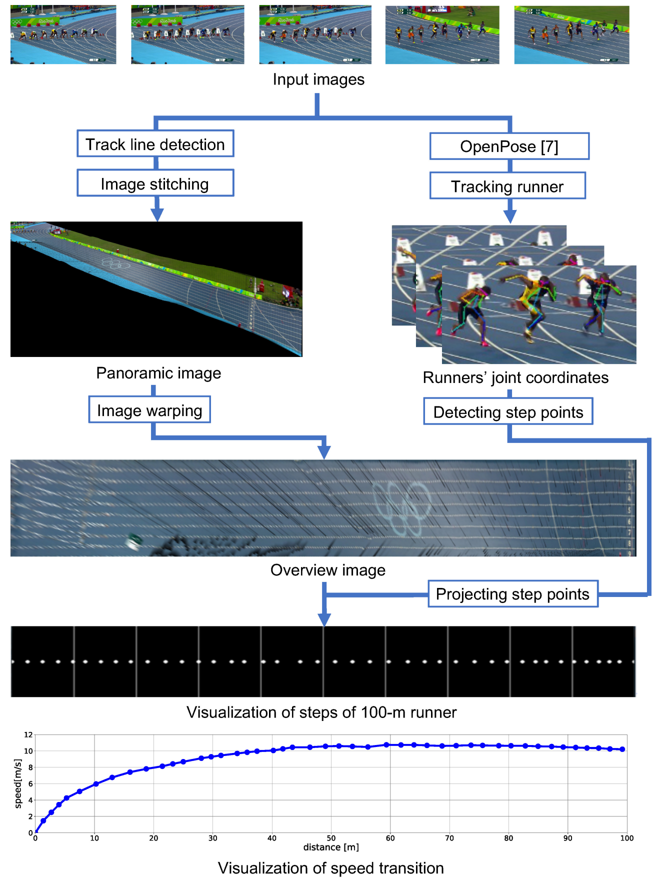

Figure 1: Flow chart of our proposed method.

2 METHODS

Our research objective is estimating the stride length, number of steps, and speed transition of 100-m race runners from video that is publicly available via Internet broadcast. Our method can be divided into three main steps. The first step is image stitching. We generate a panoramic image of the 100-m track. This is because it is impossible to determine which frame is taken at how many meters away from the start line from an image. By generating a panoramic image of the 100-m track, we can determine which frame corresponds to which part of the track. We propose an image stitching method that utilizes the consistency of the track lines for image matching. The second part is detecting the runner's steps. We use the information of the joint parts' position of people in an image obtained by OpenPose [7], which is a real-time, multi-person system for detecting the human body's joints from single images. The version of OpenPose [7] which we use is 1.0.2. From the moving of the leg joints, we judge whether each runner steps in the frame. Finally, we estimate the homography matrix from the panoramic image to the overview image to obtain the runner's stride length at the 100-m race scale. In addition, we output the visualized steps and speed transition as the results. The flow of our method is shown in Figure 1.

2.1 Track line detection

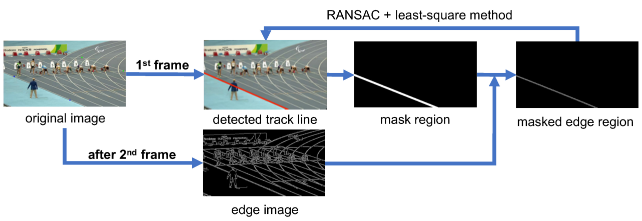

Figure 2: Flow of track line detection process.

To generate the panoramic image, we utilize the track lines of the 100-m track lane. Typically, matching methods using local feature points, such as SIFT [19], AKAZE [1], and so on are used for image stitching [6]. However, those methods do not work for our case. The detected feature points have similar feature values because the image sequence of the 100-m race, which is the input for our method, has simple textures. Wrong matching occurs between two frames, and the generated panoramic image exhibits large distortion. Therefore, in our method, we propose an image stitching approach that utilizes the consistency of the track lines.

First, we detect the track lines of a 100-m track that goes the same direction as runner runs. In addition, we track each detected line frame by frame to asses correspondences of the lines between two frames for image matching.

Here, we explain how we detect and track each track line. In the first frame, each track line is detected by clicking two points on the line. From the second frame, the line detection and tracking process can be divided into three steps. First, we apply a vertical differential filter to obtain the horizontal edges of the track. We let the pixels of which value is over 40 be edges.

In the video of the 100-m race, the track lines are supposed to be contained in these edges, since the camera moves in parallel with the track lines. Second, to detect each track line, we mask the peripheral region of each track line of the previous frame. Since we suppose that the camera does not move a lot in continuous frames, we can expect that each track line can be found in this masked region. Finally, each line is obtained by applying RANSAC and the least-squared method to the white pixel area in each masked edge image. Since our objective is not only detect track lines but also track each line between frames, the line detection process shown in Figure 2 is done for each line. The detected lines from this process are shown in red in Figure 3.

2.2 Estimation of the homography matrix

Next, we conduct image matching using the detected track lines. Our purpose is obtaining a homography matrix from each frame to the first frame. In the proposed method, first, we obtain a homography matrix between adjacent frames, and then we can obtain a homography matrix from each frame to the first frame by multiplying each homography matrix to generate a panoramic image. When we execute image matching between two images, at least eight corresponding points are required. However, the homography matrix that the detected track lines overlap in continuous frames

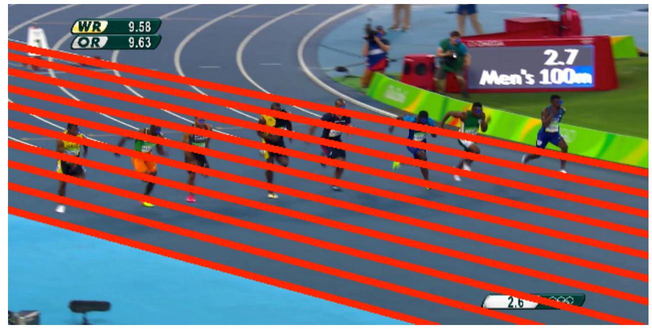

Figure 3: Detected track lines.

is countless; therefore, we uniquely determine the homography matrix by supposing that camera movement in continuous frames is approximated as simple translation.

First, we obtain a translation between two images by applying template matching. We use sum of squared difference (SSD) as the cost of template matching. The translation in which the SSD is the minimum is regarded as a rough translation between two frames. We use all pixels for this process. While calculating the SSD, to obtain the camera movement to a track plane, we mask texts and images which are superimposed on a screen and human detected regions of OpenPose [7] from template images. To do this, we select superimposed regions by clicking beforehand. In addition, when we calculate the SSD, we use the value image in the HSV scale to prevent the shadows of runners from affecting the SSD value.

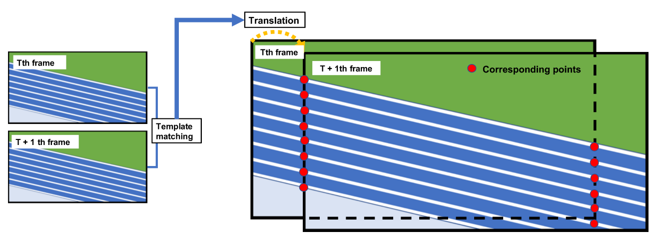

Figure 4: Flow chart of the matching process.

After obtaining the translation, we choose corresponding points between 2 frames to calculate homography matrix. Here, we give a description of the method of choosing the corresponding points. This flow is shown in Figure 4. First, let w be the image width and let $T(t_x, t_y)$ be the translation from the $t+1$ th frame to the t th frame. As I mentioned in Section 2.1, we have already detected track lines. The point $p_t(p_x, p_y)$ on the I th detected line from the inner lane is shown as $p_y = a_i p_x + b_i$. Then, the point $p(t_x, a_i t_x + b_i)$ in the t th frame and the point $p(0, b_i)$ in the $t+1$ th frame are regarded as corresponding points. In addition, the point $p(w, a_i w + b_i)$ in the t th frame and the point $p(w - t_x, a_i(w - t_x) + b_i)$ in the $t+1$ th frame are regarded as corresponding points.

Then, we calculate the homography matrix H_t from the $t+1$ th frame to the t th frame using these corresponding points. Following this, we calculate the homography matrix H'_n from each frame to the first frame. H'_n can be obtained by multiplying H_t up to the first frame. Thus, H'_n can be expressed by the following equation.

$$H'_n = \prod_{t=0}^{n} H_t \tag{1}$$

Thereafter, a panoramic image of the entire track is generated using H'_n. The panoramic image generated is shown in Figure 5.

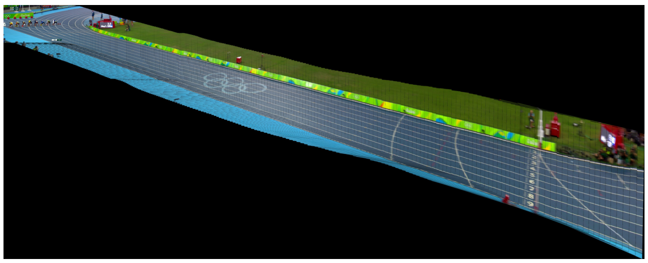

Figure 5: Generated panoramic image of the 100-m track.

2.3 Tracking a runner

The next step is tracking a target runner. For this process, we use the human joint positions, which are obtained by OpenPose [7]. For tracking the runner, we calculate the distance of the detected joint parts between two adjacent frames. OpenPose [7] can acquire 18 joint coordinates of a person (nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle, right eye, left eye, right ear, left ear) in an image. When we calculate the distances to detect the target runner in the next frame, we use the total distances of the positions of the neck, nose, right eye and left eye, as those joints are considered to have a relatively small movement between continuous images compared with other joints, such as the wrists or ankles.

2.4 Detecting steps

To determine whether the runner steps in the frame, we also use the information from OpenPose [7]. We employ the distance D between the leg joints of the target runner and the outside line of the track line, as shown in Figure 6. Since the target runner's leg is on the track plane when the runner steps, we can suppose that D becomes smaller when the runner steps. Since, in the 100-m race, each runner runs in a specific lane, separated by the detected line, a leg must be on the track lane that the runner is running in when the runner steps, as shown in Figure 7. Thus, when the following conditions are met, we judge the target runner's steps in the frame as follows:

- D is smallest in seven continuous frames; and
- The leg joint is in the lane where the runner is running.

Now, the steps of the right leg and left leg are detected separately. Although the runner steps alternatingly with both legs, sometimes, it is judged that both legs step in the same frame or continuous frames. This occurs when OpenPose [7] wrongly detects the leg positions. This error mainly occurs when the occlusion of a leg

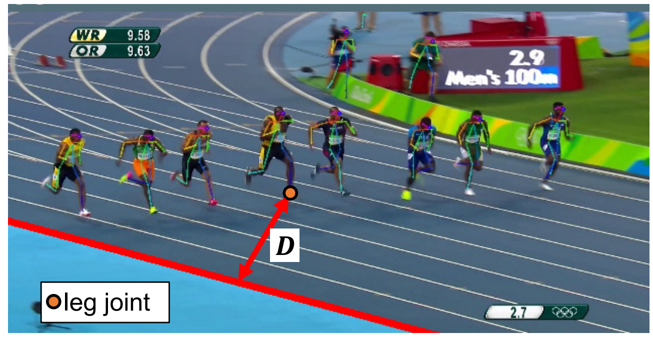

Figure 6: Distance between the outer track line and leg joint.

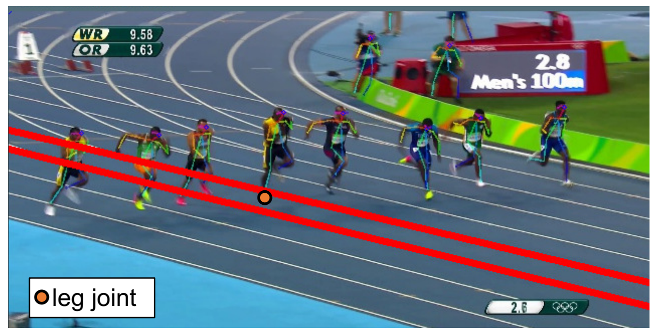

Figure 7: Visualization of the track lane and runner's leg joint.

happens. When the runner steps and the feet crosses, the other leg is occluded, as shown in Figure 8. In this case, it is difficult for OpenPose [7] to differentiate between the right and left legs. As a result, our method method detects runner steps with both legs in the same frame or continuous frames. Therefore, if both legs are judged to step in two consecutive frames, we delete one of them as a wrongly detected step. In this case, we regard the step where D is small as the correct step point. This is because, when the runner steps, D becomes small as the leg is on the track.

2.5 Completion and eliminating steps

Next, we complement and eliminate the estimated step points. This is because estimated step positions are largely affected by the OpenPose [7] result. In addition, the accuracy of OpenPose [7] is not perfect especially in the scene shown in Figure 8.

We use the average stride length for judging if the detected steps are correct. To obtain the stride length at the 100-m scale, we first obtain the homography matrix $H_{overview}$, which projects panoramic image to overview image. For obtaining four corresponding points to calculate $H_{overview}$, we click four corner points of the 100-m track in the panoramic image. The overview image generated in this step is shown in Figure 9. By projecting the detected step points using $H_{overview}$, we can obtain the target runner's steps at the 100m scale.

Next, we show the algorithm of completion steps for which no steps are detected and eliminating wrongly detected steps. Now, we have all the steps' distances from the starting line as vector $X = x_1, x_2, x_3, ..., x_n$; then, the I th step's stride length S_i can be obtained by $x_i - x_{i-1}$. While the runner is accelerating, the stride length is shorter than the average stride length; therefore, we propose a different process depending on the value of x_i. We set the threshold th, which represents the distance where the runner's stride length reaches the average stride length. In this paper, we set th as $20m$. This parameter is empirically determined. The algorithm for eliminating wrongly detected steps is shown in Algorithm 1.

For completing the steps process, we propose a different process depending on the value of x_i. We also use same th mentioned in the previous paragraph. The algorithm for completing steps is shown in Algorithm 2. In Algorithm 2, x_c indicates completed point.

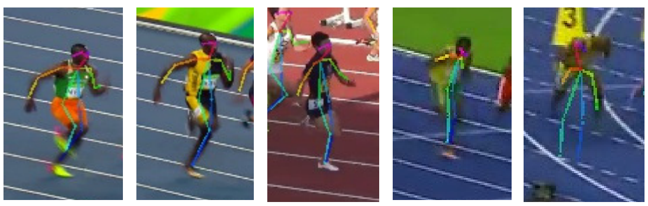

Figure 8: Scenes in which the leg joints are wrongly detected.

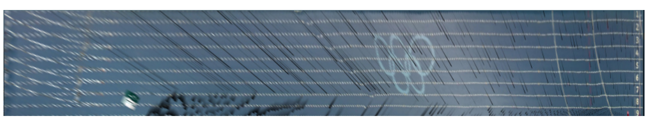

Figure 9: Overview image generated from the panoramic image.

Algorithm 1 Deleting wrongly detected steps

```
for i ⇐ 1 do
  if x_i < th then
    if S_i < 0.8S_average then
      delete x_i
    end if
  else if x_i > th then
    if S_i < S_average then
      delete x_i
    end if
  end if
end for
```

3 EXPERIMENT

We applied our method to some videos of a 100-m race that are accessible on Youtube. Each of those videos have different background and intrinsic camera parameters. The resolution of each video is 640×320. We show images of input videos in Figure 10. Excluding runners who did not finish the race, we obtained data for a total 29 runners. In Section3.1, we compare the results of our

Algorithm 2 Completing steps

```
for i ⇐ 1 do
  if x_i < th then
    if S_i > S_average then
      insert x_c ⇐ (x_{i-1} + S_average) between x_{i-1} and x_i
    end if
  else if x_i > th then
    if S_i > 1.3S_average then
      insert x_c ⇐ (x_{i-1} + S_average) between x_{i-1} and x_i
    end if
  end if
end for
```

method with the data observed with laser range finders and multiple cameras [10]. In Section 3.2, we evaluate how our method can correctly count the number of runners' steps. We set the number of steps counted manually by watching the video frame by frame as ground truth. In Section 3.3, we show the visualized data of the runners' steps and speed transition in the 2016 Rio Olympics Men's Final for the 100-m race.

Figure 10: Inputs of the experiments.

3.1 Comparison with laser range finder and multiple cameras

In this section, we compare our method with ground truth which is measured by laser range finder and multiple cameras. We show the results of our method for the IAAF World Championship Track and Field 2009, in which the current world record for the men's 100-m race was set by Usain Bolt. We compare his speed transition, step length, and step frequency estimated by our method with the ground truth data provided by IAAF [10]. The 20-m interval time, average stride length, and stride frequency of Usain Bolt from the ground truth and data estimated by our method are shown in Figures 11, 12, and 13.

3.2 The accuracy of the step estimation

In this section, we show how our method can estimate runners' steps correctly. We set the result of counting the number of steps frame by frame manually as the ground truth data. Table 1 shows the results of the evaluation. Error in table 1 signifies differences in the number of steps between the ground truth and our method.

3.3 Visualizing steps and speed transition

In this section, we provide visualized data of the steps and speed transition of runners using our method. Figure 14 shows the visualized data from the 2016 Rio Olympics Men's Final for the 100-m race. The left side of the Figure 14 signifies the result of the visualization of the steps. The horizontal axis represents the distance at the meter scale, and a vertical white line is shown every 10 m. In addition, each white circle shows runners' steps. The right side of Figure 14 shows the visualization of the speed transition. The horizontal axis represents the distance from the starting line to the goal line, and the vertical axis shows the speed.

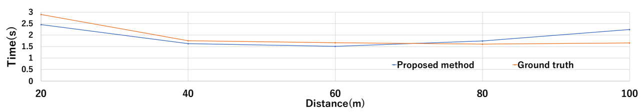

Figure 11: Usain Bolt's 20-m interval time.

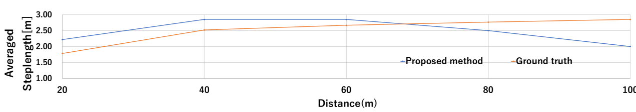

Figure 12: Usain Bolt's 20-m interval stride length.

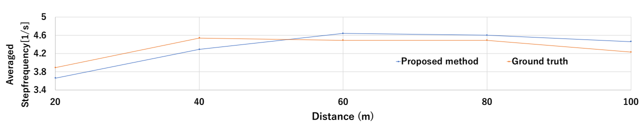

Figure 13: Usain Bolt's 20-m interval stride frequency.

Table 1: Evaluation of estimating steps.

error(times)	0	1	2	3	over 4
percentage(%)	21	21	7	24	27

4 DISCUSSION

4.1 Comparison with laser range finder and multiple cameras

According to Figure11,12, and 13, the first and last 20 m had more noise than the other terms did. For the first 20 m, one of the possible causes of the noise is the runner's posture. The accuracy of OpenPose [7] depends on the posture of people in the image. If all the body's joints are captured in an image, the accuracy becomes higher. However, in this case, in the first part of the acceleration section, which was the first 10-20 m, the runner bent the upper body. Therefore, the accuracy is reduced, since every joint is not shown in an image.

In the last 20 m, we assume that the accumulated noise of the homography matrix affects the result. During the image stitching, the homography matrix from each frame to the first frame is obtained by multiplying the homography matrices until the first frame. Therefore, the latter frame accumulates more noise. We assume this is why the estimated step length is too low in the last 20 m. Currently, we match two images by regarding the camera movement between the two frames as a translation. This may be the reason for the noise; therefore, we must consider alternative methods for image stitching.

4.2 The accuracy of the step estimation

According to Table 1, we can correctly estimate the number of steps with no error in 20% cases, and we can estimate the number of steps within a three-step error in 70% cases. Error in table 1 signifies differences in the number of steps between the ground truth and our method. However, in the rest of the cases, the estimation results were totally wrong.

We assume one of the causes of failure is the tracking algorithm. With our tracking algorithm shown in Section 2.3, it is difficult to track the runner robustly when runners overlap. In the 100-m race video, overlapping of runners often occurs, even in the early part of the race. In this case, the target runner is still accelerating when the tracking is lost. As shown in Section 2.5, once tracking is lost, the rest of the steps are completed based on the assumption that the runner finishes the race with an average stride length. Therefore, completion of steps is performed with a shorter stride, so that the number of steps can be larger.

In a case where the estimated number of steps is totally different from the ground truth, we suppose that this is because of a failure of joint position detection with OpenPose [7]. One of the reasons of noise in OpenPose [7] is the differences from the learning dataset and input images. Therefore, it is a future task to use dataset of track and field runners for the learning process of CNN. By doing this, we can expect better accuracy of joint part detection.

Regarding the tracking algorithm, it is thought that a more robust algorithm can be provided using the line information detected in Section 2.1. As we mentioned in this section, it is possible to judge which lane the runner is currently running in from the positional relationship between the runner's step position and the track line. This is also a future task to implement a more robust tracking algorithm using line information.

4.3 Visualizing steps and speed transition

As shown in the visualized result of the speed transitions in Figure 14, the runners gradually accelerate from the start, and they reach the maximum speed at the early middle part of the race. This is a typical way for 100-m runners to run. Thus, it seems that our method can obtain the correct result.

As shown in the visualized steps in Figure 14, the stride length gradually increases from 0 to 20 m; then, the runner maintains almost the same stride length. This also means that the runner

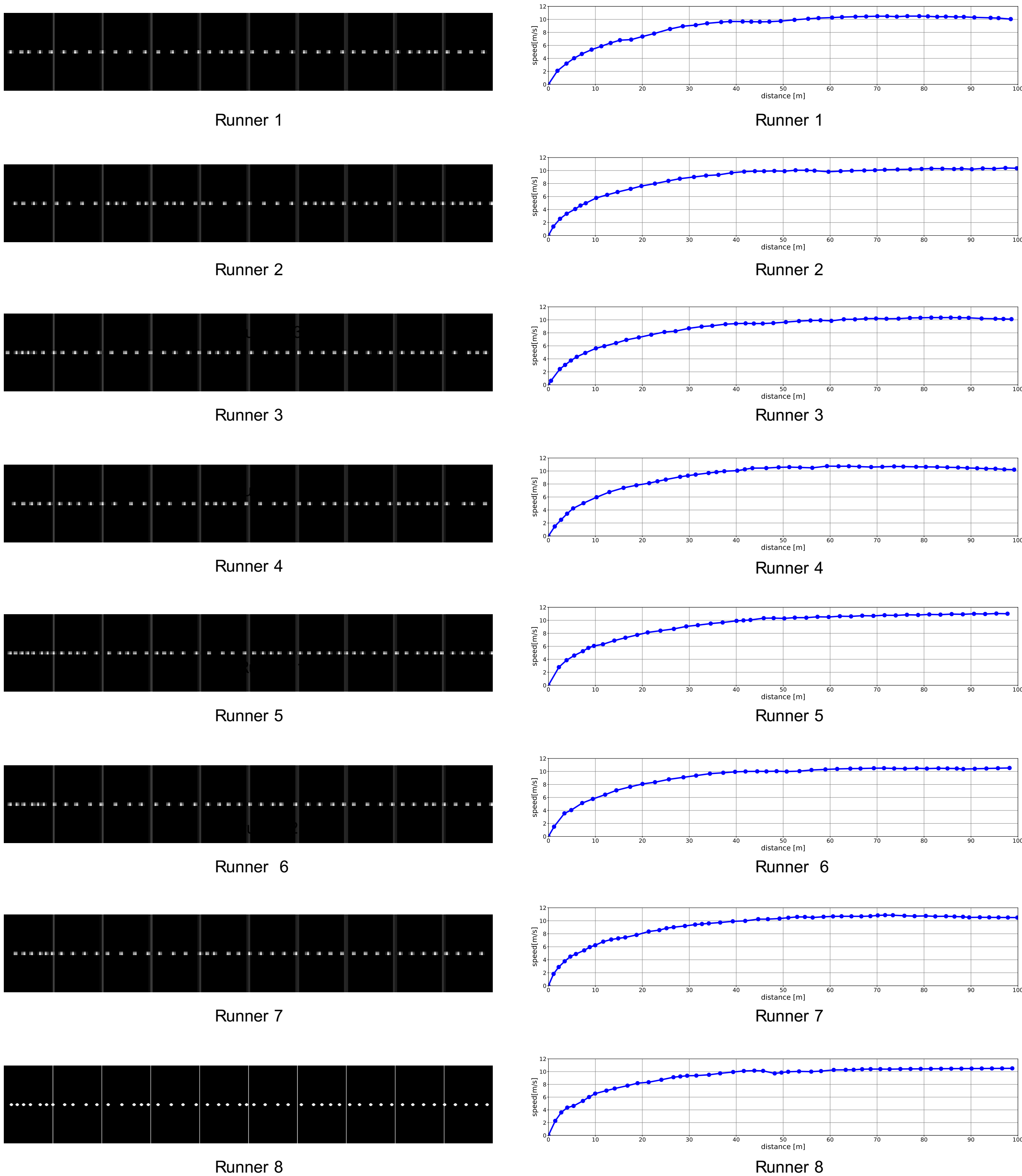

Figure 14: Visualization of race of runners of the 2016 Rio Olympics Men's Final for the 100-m race: (left) visualization of steps, (right) visualization of speed transition.

gradually accelerates and finishes the race with maximum stride length.

In the cases of Runner 7 and Runner 8 on the left side of Figure 14, the stride length does not seem to change after around 60 m, which is the point at which the tracking is lost. As mentioned in Section 2.5, we completed the steps based on the assumption that the runner finishes the race with the average stride length. This is why the stride length looks same after around 60 m. Since it is considered that the completed strides are different from the real strides, we must formulate a better way to complete runners' steps.

As shown in the left part of Figure 14, only the x axis information is visualized. This is because we integrated right and left steps, as mentioned in Section 2.4. However, if we can make the accuracy of joint part detection better, it will be possible to obtain and visualize y axis information. As mentioned in Section 4.2, preparing datasets for OpenPose [7], which is optimized for joint detection of runners, is required for this.

Concerning the application of our method, we think that our method is useful as broadcasting content. Not only the visualized data shown in Figure 14, but also other features can be realized. For example, we can expect that it is possible to create a virtual competition system by superimposing steps in the different race videos. Although virtual competition system [15] needs to move the camera with same trajectory from the same viewpoint, our method can compare multiple runners of different races, with footage taken with different camera. In addition, we can generate strobo images of runners. In Hasegawa et al.'s method [12], camera position has to be static. However, in our method, we can generate strobo images from a video taken by moving camera, since we have a homography matrix from each frame for the panoramic image.

Figure 15: Scenes in which athletes overlap.

5 CONCLUSION

In this study, we proposed a method of measuring the movements of 100-m race runners using a publicly available race video. Basically, the information that we can obtain from a race video is limited to the number of steps and average stride length. However, our proposed method made it possible to obtain not only this information but also every stride length and speed transition. Our method can be divided into three sections. First, we generate a panoramic image of a 100-m track to estimate where every runner is running on a 100 m scale. We proposed an image stitching method that utilizes the consistency of the track line. Second, we judge the frames in which the runner steps using the movement of the leg joints. Finally, we obtain a homography matrix from the panoramic image to an overview image so that we can obtain the stride length at the 100 m scale. In the experimental part, we applied our method to various videos of 100-m races. First, we evaluated our method by comparing it with the measuring method using laser range finder and multiple cameras, which are typically used. Second, we evaluated the estimation accuracy concerning the number of steps. We determined that we can correctly estimate the number of steps with no error in about 20% cases. Finally, we visualized the steps and speed transition of the runners and showed the possibility of using the solution for broadcasting content. One future work is using images of runners to train CNN model in joint part detection for estimating the data more accurately.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Hiroaki Hobara and Dr. Yoshinobu Kobayashi of the National Institute of Advanced Industrial Science and Technology (AIST) for suggesting this paper's topic and data.

REFERENCES

[1] Pablo F. Alcantarilla and T. Solutions. 2011. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* 34, 7 (2011), 1281–1298.
[2] Four Assist. 2018. LDM301S. Retrieved July 6, 2018 from https://www.4assist.co.jp/
[3] Indriyati Atmosukarto, Bernard Ghanem, Shaunak Ahuja, Karthik Muthuswamy, and Narendra Ahuja. 2013. Automatic recognition of offensive team formation in american football plays. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 991–998.
[4] Michael Beetz, Bernhard Kirchlechner, and Martin Lames. 2005. Computerized real-time analysis of football games. *IEEE Pervasive Computing* 4, 3 (2005), 33–39.
[5] Gedas Bertasius, Aaron Chan, and Jianbo Shi. 2018. Egocentric Basketball Motion Planning from a Single First-Person Image. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5889–5898.
[6] Matthew Brown and David G. Lowe. 2007. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision* 74, 1 (2007), 59–73.
[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2016. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050* (2016).
[8] Anthony Cioppa, Adrien Deliège, and Marc Van Droogenbroeck. 2018. A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.
[9] Hassan Ghasemzadeh, Vitali Loseu, Eric Guenterberg, and Roozbeh Jafari. 2009. Sport training using body sensor networks: A statistical approach to measure wrist rotation for golf swing. In *Fourth International Conference on Body Area Networks*. ICST, 2.
[10] Rolf Graubner and Eberhard Nixdorf. 2011. Biomechanical analysis of the sprint and hurdles events at the 2009 IAAF World Championships in Athletics. *New Studies in Athletics* 26, 1/2 (2011), 19–53.
[11] Raffay Hamid, Ram Krishan Kumar, Matthias Grundmann, Kihwan Kim, Irfan Essa, and Jessica Hodgins. 2010. Player localization using multiple static cameras for sports visualization. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 731–738.
[12] Kunihiro Hasegawa and Hideo Saito. 2016. Synthesis of a stroboscopic image from a hand-held camera sequence for a sports analysis. *Computational Visual Media* 2, 3 (2016), 277–289.
[13] Hiroaki Hobara, Satoru Hashizume, Yoshiyuki Kobayashi, Yuko Usami, and Masaaki Mochimaru. 2016. Ethnicity and spatiotemporal parameters of bilateral and unilateral transtibial amputees in a 100-m sprint. *SpringerPlus* 5, 1 (2016), 343.
[14] Hiroaki Hobara, Wolfgang Potthast, Ralf Müller, Yoshiyuki Kobayashi, Thijs A Heldoorn, and Masaaki Mochimaru. 2016. Normative spatiotemporal parameters during 100-m sprints in amputee sprinters using running-specific prostheses. *Journal of Applied Biomechanics* 32, 1 (2016), 93–96.
[15] NHK Science & Technology Research Laboratories. 2012. virtual competition system using 4K camera. Retrieved July 7, 2018 from http://www.nhk.or.jp/strl/open2012/html/tenji/028_e.html
[16] Haojie Li, Jinhui Tang, Si Wu, Yongdong Zhang, and Shouxun Lin. 2010. Automatic detection and analysis of player action in moving background sports video sequences. *IEEE Transactions on Circuits and Systems for Video technology* 20, 3 (2010), 351–364.
[17] Hawk-Eye Innovations Ltd. 2015. HAWK-EYE. Retrieved July 7, 2018 from https://www.hawkeyeinnovations.com/

[18] Stefano Messelodi and Carla Maria Modena. 2013. Scene text recognition and tracking to identify athletes in sport videos. *Multimedia Tools and Applications* 63, 2 (2013), 521–545.

[19] Pauline C. Ng and Steven Henikoff. 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Research* 31, 13 (2003), 3812–3814.

[20] Ltd. OptiTrack Japan. 2018. OptiTrack. Retrieved July 6, 2018 from https://www.optitrack.co.jp/

[21] NEIL Owens, C. Harris, and C. Stennett. 2003. Hawk-eye tennis system. In *Visual Information Engineering*. IET, 182–185.

[22] SportsMEDIA Technology SMT. 2018. SMT. Retrieved July 7, 2018 from http://smt.com/football/

[23] Christina Strohrmann, Holger Harms, Cornelia Kappeler-Setz, and Gerhard Troster. 2012. Monitoring kinematic changes with fatigue in running using body-worn sensors. *IEEE Transactions on Information Technology in Biomedicine* 16, 5 (2012), 983–990.

[24] Jinyan Yang, Guanlei Xu, and Yu Li. 2013. Running state recognition in videos via frames' frequency and positions of two feet. *Fourth International Conference on Intelligent Control and Information Processing* (2013), 310–313.