Stroboscopic Image Synthesis of Sports Player from Hand-Held Camera Sequence

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

Hideo Saito
Keio University
saito@hvrl.ics.keio.jp

Abstract

A method for synthesizing the stroboscopic image of a moving sports player from a hand-held camera sequence is proposed. In this method, a player is extracted from each frame of an input video sequence using a HOG-based people detector. After removing the bounding box of the extracted player from each frame, all frames are stitched together to generate a background image without the player. The player's area is then overlaid onto the stitched background image and a stroboscopic image of the player is synthesized. By using the bounding box of the player detected by HOG and by subtracting the images, computational speed and accuracy can be improved compared with the conventional method that segments the player's region. In addition to the stroboscopic image synthesis, we also remove the player's shadow to improve the appearance of the resultant stroboscopic image. Experimental results demonstrate the effectiveness of the proposed method.

1. Introduction and motivation

Analyzing the performance of players using various sensing technologies has recently become popular in a variety of sports. Athletics is one of the targets of such analysis. For example, the fatigue of athletic runners was evaluated using 12 wearable sensors to measure the displacement of the positions, angles, etc. of each joint and of the waist [1]. A changes in ventricular rates when running have been visualized with heartbeat sensors [2]. Eskofier et al. classified the motion of leg kicks during running by using a acceleration sensor [3]. This classification could be used for fitting shoes, among other applications.

For the analysis of athletic running, stride length and speed in each interval provide significant information for improving running performance. However, most conventional sensing technologies require special equipment such as multiple cameras and/or motion sensors attached to the runners, which is expensive. Moreover, the interest of amateur runners in such analyses has been growing, and for these users, easy and on-site analysis of running performance is needed.

Using a video camera is one solution for easy analysis. Various other sports such as soccer [4], basketball [5][6], and American football [7] are already using computer vision technologies. However, these methods only extract the positions of players, whereas we aim to measure the landing position of each step in athletics. For sensing such detailed player movements from a video sequence, the player should be captured in higher resolution. Therefore, conventional methods for team sports analysis [4][5][6][7] cannot be applied because they mostly rely on fixed cameras that capture groups of players distributed over the wide area of a playing field.

In our previous work, we proposed measuring stride length and speed with a video camera by manually moving a hand-held camera so that it captures the entire running performance (e.g. completing a 100-m sprint) with sufficient resolution [8]. In that method, we measured the runner's landing position by using the homography between an image coordinate system and the real world coordinate system to estimate stride length and speed. We also tried to use a stroboscopic image that captured changes of a runner's poses in motion for automated measurement of speed and stride length. In our synthesis of a stroboscopic image, an image including only the background is synthesized first and then the runner is extracted from each frame and overlaid on the background image. The image in which only the runner is extracted can be synthesized by taking the difference between the background image and the image in which the runner was previously overlaid. We felt that the problem of landing timing determination could be solved by analyzing the stroboscopic image, but the background image synthesis we used was based on the mean-shift of each pixel, which was computationally expensive.

In the present work, we propose a method for synthesizing the stroboscopic image of a sports player from a hand-held camera sequence based on a computationally efficient algorithm that generates a stitched background image by the

bounding box of the player detected by HOG and subtracting images. We also analyzed various other sports using video camera images and found that it should be possible to use the stroboscopic image synthesized by our method for sports other than running. Therefore, we consider our use of the stroboscopic image for runners to be a representative example.

2. Environment and conditions of capturing video

In this paper, we assume that video footage is captured using a hand-held camera operated by a person who controls the camera to capture the subject with a sufficient resolution in each frame, as shown in Figure 1. We assume the operator takes a video in a scene where the player is practicing in a running track, on an athletic ground, etc. It is assumed that no other people are in the background under this condition. The person controlling the video camera is standing still and catches the player at the center of the screen as much as possible. The video camera is held in the operator's own hand and is moved freely. The operator stands a few tens of meters away from the center of the track to capture the player in the entire moving scene by using zoom. There are no restrictions on the player's clothes and background except that the color of the background and clothes should not be exactly the same.

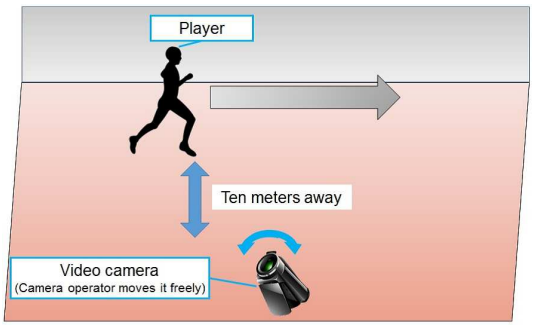


Figure 1. Image of capturing environment.

3. Proposed method

The goal of our previous method [8] was to synthesize a stroboscopic image for the automatic measurement of speed and stride length. In that method, a stitched image without a player has been synthesized and then the player has been overlaid on the stitched background image. As noted above, synthesizing the background image has been time-consuming because Mean-Shift estimation [9][10] has been executed on every pixel of the image. This has been the main drawback of the method [8]. Therefore, we need a way of synthesizing the image in a short amount of time.

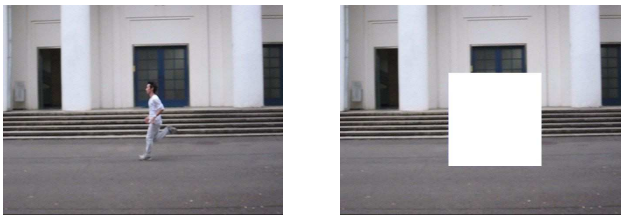
In the method we propose in the present work, in order to reduce computation time, we mask only a region in which the player exists in the image of each frame used for the stitched image synthesis. As a result, any images containing only background are synthesized because the region where the player exists is excluded from the stitched image synthesis. We improved other parts of our previous method [8], as well. Specifically, we added methods for overlaying the player and removing the player's shadow. Figure 2 shows a flowchart including these improvement points, with the differences between our previous and present methods highlighted. Red frames indicate that a step for processing content is different and green frames indicate an added or interchanged step. In the proposed method, we use the bounding box of a player to synthesize a background image. We elaborate upon these points in 3.1, 3.2, and 3.3.

3.1. Synthesis of background image

Here, we discuss the "Synthesize a background image" process in Figure 2. A background image is synthesized by stitching. As previously noted, we mask the area of the player as shown in Figure 3 to reduce the computation time. The homography for this stitching uses feature points in each frame image. In order to extract feature points in complex environments such as the outdoors, SURF [11] is used as feature points and the stitch function of OpenCV is used for the synthesis, the same as in our previous method [8].

The area of this mask is the bounding box of a player detected by HOG-SVM [12]. The mask area of each frame is excluded in synthesizing a background image so that only a background pixel can be used for synthesizing the background image. The HOG-SVM sometimes fails to detect the player in a motion sequence. A Kalman filter [13] is applied for keeping track of the bounding box.

Our previous method [8] calculated each value of HSV for every pixel using Mean-Shift. In the proposed method, instead of Mean-Shift, we execute a stitching, as described above, which speeds up the synthesis of a background image.



(a) Original image (b) Masked image

Figure 3. Images for background image synthesis.

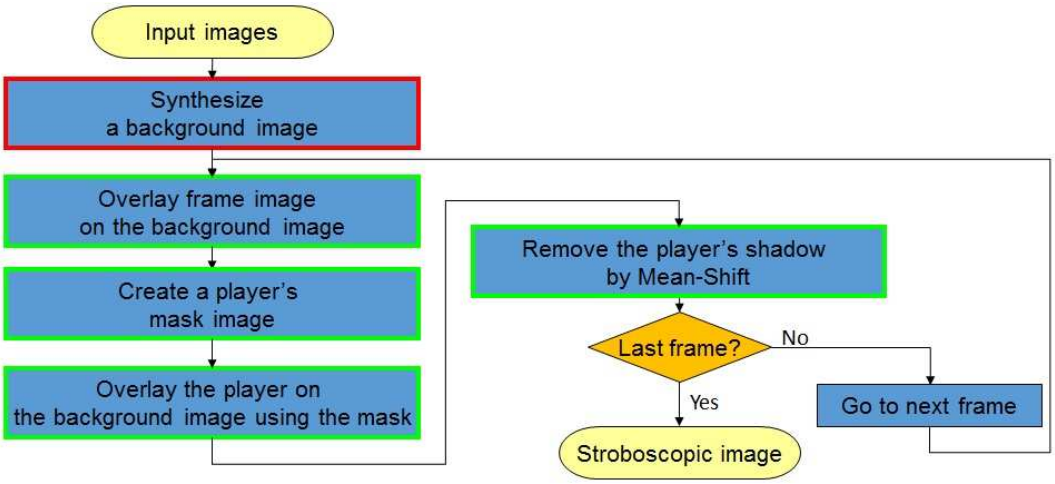


Figure 2. Flowchart of proposed method.

3.2. Synthesis of stroboscopic image

Here, we describe the "Overlay the frame image on the background image", "Create a player's mask image", and "Overlay the player on the background image using the mask" processes in Figure 2. In this step, we use the other mask created by a subtraction of images, an example of which is shown in Figure 4. This mask is the result of the "Create a player's mask image" step and is used for overlaying the player on the background image in the "Overlay the player on the background image using the mask" step. We elaborate upon these processes below.

Figure 5 shows images for each step of this subsection. An input image of each frame, Figure 5(a), is transformed and projected onto a coordinate system of the background image, Figure 5(b). As a result, a projected image, Figure 5(c), is synthesized. The homography for the transformation and projection in this step is the same as in 3.1.

In the "Create a player's mask image" step, the projected image, Figure 5(c), and the background image, Figure 5(b), are utilized again. A player's mask image (Figure 4) is obtained by the subtraction and binarization of these two images. A part of the player area is occasionally lacking in the mask image if the color of the player's clothes is similar to that of the background, whereas we can compensate for this by a morphology process. Hence it is only a minor deficit.

A transformed input image of each frame as Figure 5(a) is overlaid on the background image by using a player's mask image, Figure 4. These steps are repeated to the last frame, at which point a stroboscopic image is synthesized. In our previous method, blending was needed to synthesize a stroboscopic image due to overlap of the bounding box of a player's area [8], and the difference between a player and background might have become almost 0 in the worst case. In contrast, in our new method only the player is overlaid, so the player cannot disappear if the number of frames becomes enormous.

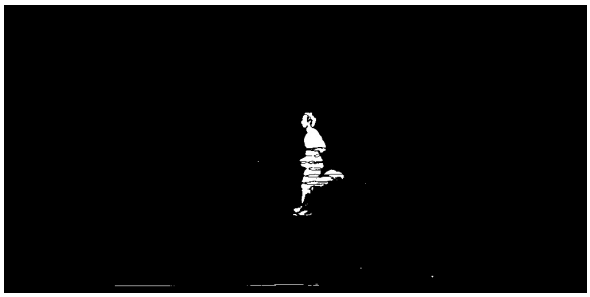


Figure 4. Example of a mask for overlaying a player.

3.3. Removal of player's shadow

Here, we discuss the "Remove the player's shadow by Mean-Shift" process in Figure 2. The stroboscopic image synthesized in 3.2 results in the player's shadows being visible under certain capturing conditions such as a sunny day. An example is shown in Figure 6, where some shadows remain in the red rectangles. In particular, shadows connected to the players expand the players' area. This negatively affects the calculation of the players' landing positions.

In order to remove these shadows, we use a synthesis method utilizing Mean-Shift, as explained in [14]. The number of frames having shadow is smaller than the number of frames having a background in views where a player is moving. Therefore, Mean-Shift can remove shadows. We select this removal area manually. The areas containing shadow are quite small compared to the entire image. Processing time for this small area is a minor deficit

(a) Input image (b) Background image (c) Projected image

Figure 5. Images used at each step in 3.2.

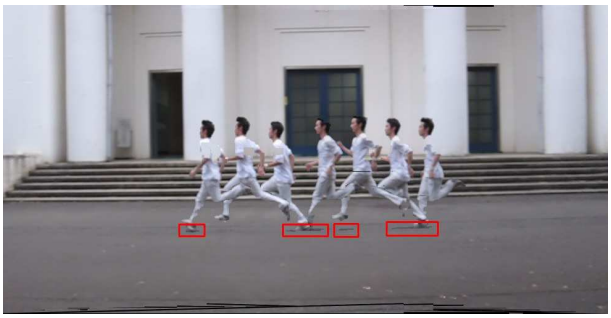


Figure 6. Stroboscopic image with shadow.

4. Experiments

We performed two experiments. In the first, we synthesized a runner's stroboscopic image as a representative example, and in the second, we applied the synthesis method to various scenes to confirm its versatility. We used Microsoft Visual Studio 2010 as the IDE, C++ as the programming language, and OpenCV 2.4.11 as the image processing library for implementation. The PC we used had a 2.40 GHz CPU with 4 cores and 8.00 GB of memory. The HOG-SVM used to create the mask was learned using about 20,000 player images for accurate detection. In some results, we show a subset of frames that have few intervals for visibility. Finally, we compared the results of our method with those of two existing methods: Autostitch [15] and the frame subtraction.

4.1. Synthesis of runner's stroboscopic image

An input video for this experiment was captured by a hand-held camera. As described in 2, the camera is held in the operator's own hand to capture a runner in the center of the camera by manually tracking the runner. Figure 7 shows a part of the input image group used for the experiments. The size of the image was 640 $\times$ 480 pixels.

Figure 7. Example of input images.

4.1.1 Synthesis of background image

Our previous method [8] utilizing Mean-Shift for synthesizing a background image was time-consuming. In our new method, we use the bounding box of a player detected by HOG to reduce the computation time.

A comparison of the processing time results is given in Table 1. We used only the three frames shown in Figure 7 for this experiment. The processing times for the previous and proposed methods were measured 10 times each. As indicated in the table, the proposed method utilizing a mask to reduce processing time is even faster than our previous method [8].

Figure 8 shows the background image synthesized by our previous method [8] and the proposed method. As shown, the two images are almost equivalent. These results indicate that the image synthesized by our proposed method can be applied to the following process.

Table 1. Processing time of synthesizing a background image.

	Previous [8]	Proposed
Average (s)	2.44×10^3	1.72×10^1
Max (s)	2.53×10^3	1.84×10^1
Min (s)	2.35×10^3	1.68×10^1

(a) Previous method

(b) Proposed method

Figure 8. Results of synthesizing background image.

(a) Previous method

(b) Proposed method

Figure 9. Result of synthesizing a runner stroboscopic image.

4.1.2 Synthesis of stroboscopic image

Our previous method [8] needed to blend overlaying runner areas with a background image, which resulted in the difference between the runner and the background becoming almost 0 in the worst case. In the proposed method, we use a mask created by the input and background images to overlay the player on the background.

Figure 9 compares the results of our previous [8] and proposed methods. Runners were blended in the image of our previous method, as shown in Figure 9(a). In this method, although feet in contact with the ground may become more apparent, the bounding boxes are overlapped in practice. In other words, the background is blended, and therefore the foot in the ground becomes less apparent. In contrast, as shown in Figure 9(b), the runners in the stroboscopic image synthesized by the proposed method are not blended with the background.

In the results of the proposed method, the runner's shadow is still not completely removed, as shown in Figure 6. However, the shadow can be removed in the next step, as mentioned above in 3.3, and so we conclude that our proposed method can synthesize a stroboscopic image.

4.1.3 Removal of player's shadow

Under some capturing conditions, a stroboscopic image contains the runner's shadows. To remove these shadows, we use a synthesis method utilizing Mean-Shift, as explained in [14].

The result of removing the runner's shadows from the image in Figure 9(b) is visible in Figure 10. A comparison of Figures 9(b) and 10 clearly shows that our proposed method can remove shadows well. In addition, the processing time of this step is just 6.10 sec on average, which is barely 0.003% of the time it takes to apply Mean-Shift to the entire image. These results demonstrate that our proposed method can remove shadows without consuming a lot of time. They also indicate that our proposed method can synthesize a stroboscopic image to measure the runner's speed and stride length more easily than the previous method.

We also visualized the runner's footprints in this scene by using stroboscopic images. The result of this visualization is shown in Figure 11, where after 28 measurements the average error is 0.13 m, the maximum error is 0.59 m, the minimum error is 0.00 m, and the standard deviation is 0.15 m. These results enable us to understand the change in stride lengths visually.

Figure 10. Result of removing runner's shadow.

Figure 11. Visualization of a runner's footprints.

4.1.4 Comparison with existing methods

We compared the results of our proposed method with those of two existing methods. Figure 12 shows the result of the existing methods. Autostitch does not concern itself with whether an image contains a person or not, therefore Figure 12(a) has people in only a limited number of frames. The method using frame subtraction is originally for a fixed camera. As shown in Figure 12(b), this method does not work on scenes captured by hand-held camera.

Figure 13(a) shows the synthesis results of our proposed method in another scene in order to show that this method can be used in various environments. There are runners in all frames, the same as in Figure 10. The results of the two existing methods are shown in Figures 13(b) and 13(c). They are the same as the results in Figures 12(a) and 12(b). These results clearly demonstrate the superiority of our proposed method.

(a) Autostitch (b) Frame subtraction

Figure 12. Results of existing methods.

4.2. Application to other scenes

As stated previously, we executed our method on other scenes in a second experiment to confirm its versatility. Walking, downhill skiing, figure skating, and speed skating were the targets of this second experiment. The walking scene was captured by us, the same as in the first experiment. The other videos captured by others. The results of this second experiment show that our proposed method can be used for various purposes such as checking player's form. This is because we have assumed that use the image in addition to measuring.

The results are shown in Figures 14, 15, 16, and 17. These results are the same as those in Figures 10 and 12. Our proposed method overlays players onto a background image, therefore players appear in all frames in Figures 14(a), 15(a), 16(a), and 17(a). Autostitch cannot always synthesize a stroboscopic image due to its not being able to extract the player from each frame. Figures 14(b), 15(b), 16(b), and 17(b) contain players in only part of the frames as a result. The method using frame subtraction does not consider that the camera moves freely, so it cannot synthesize a stitched image and a stroboscopic image from a non-fixed camera sequence, as evident in Figures 14(c), 15(c), 16(c), and 17(c).

These results confirm the versatility of our proposed method. If we change the object detection method, we can detect targets other than people, such as cars.

5. Conclusion

In this paper, we have proposed a method of synthesizing a stroboscopic image to enable the measurement of a player's speed and stride length in an automated system. In particular, we have sped up the processing of background image synthesis by utilizing the bounding box of a player detected by HOG. We have also improved the overlaying of a player and the removal of shadows. We have demonstrated the superiority of our proposed method through experiments and confirmed that it can be used in various scenes.

Our proposed method has a few of manual steps. For example, the morphology process in the synthesis stroboscopic image step and the selection of removal area in the removal of a shadow are manual. Our future work will be to improve these steps for the realization of full automation. For example, in the shadow removal step, inputting the area of the shadow is manual, but if this area can be extracted automatically, the problem goes away. Therefore, tracking the shadow area with some sort of tracking method is one of the solutions. We will examine various solution candidates in our future work.

References

[1] C. Strohrmann, H. Harms, C. Kappeler-Setz, and G. Troster. Monitoring Kinematic Changes With Fatigue in Running Using Body-Worn Sensors. IEEE Trans-

(a) Proposed method (b) Autostitch (c) Frame subtraction

Figure 13. Results of stroboscopic image synthesis in another scene.

actions on Information Technology in Biomedicine, 16(5):983–990, 2012. 1

[2] 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, 99–106, 2013. 1

[3] 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), 1–4, 2013. 1

[4] 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), 731–738, 2010. 1

[5] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to Track and Identify Players from Broadcast Sports Video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(7):1704–1716, 2013. 1

[6] M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, and J. Perš. A trajectory-based analysis of coordinated team activity in a basketball game. Computer Vision and Image Understanding (CVIU), 113(5):612–621, 2009. 1

[7] 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), 991–998, 2013. 1

[8] K. Hasegawa and H. Saito. Auto-Generation of Runner's Stroboscopic Image and Measuring Landing Points Using a Handheld Camera. 16th Irish Machine Vision and Image Processing, 169–174, 2014. 1, 2, 3, 4, 5

[9] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function. IEEE Transactions on Information Theory, 21(1):32–40, 1975. 2

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603–619, 2002. 2

[11] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU), 110(3):346–359, 2008. 2

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 886–893, 2005. 2

[13] R. E. Kalman. A new approach to linear filtering and prediction problems. Journal of Fluids Engineering, 82(1):35–45, 1960. 2

[14] S. H. Cho and 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), 829–835, 2011. 3, 5

[15] M. Brown and D. Lowe. Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision, 74(1):59–73, 2007. 4

(a) Proposed method (b) Autostitch (c) Frame subtraction

Figure 14. Example of application to walking.

(a) Proposed method (b) Autostitch (c) Frame subtraction

Figure 15. Example of application to downhill skiing.

(a) Proposed method (b) Autostitch (c) Frame subtraction

Figure 16. Example of application to figure skating.

(a) Proposed method (b) Autostitch (c) Frame subtraction

Figure 17. Example of application to speed skating.