Swimmer’s Stroke Estimation Using CNN and MultiLSTM

Kohei Hakozaki, Naoki Kato, Masamoto Tanabiki, Junko Furuyama, Yuji Sato and Yoshimitu Aoki

Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
E-mail: khakozaki@aoki-medialab.jp, aoki@elec.keio.ac.jp

Abstract

Many methods of sports video analysis have been proposed in the computer vision field. However, the analysis of swimming videos is a challenging task. This is because there is a lot of noise, such as water splashes, making it difficult to see the swimmer’s motion and detect body parts. Thus, it is difficult to automatically estimate a swimmer’s motion, especially the stroke. In this paper, we introduce a novel approach to automatically estimating the stroke in such situations. Firstly, we detect the swimmer from a swimming video using a projective transformation, background subtraction, and a Kalman filter. We next create a model that learns a mapping from a window of frames to a point on a one-dimensional (1D) target signal, which represents a swimmer’s stroke (we call a ‘stroke signal’). We use a convolutional neural network (CNN) and multi long short-term memory (MultiLSTM) which is an expanded model of LSTM. Finally, we estimate swimmer’s stroke from the stroke signal. In a dataset including various environments, the outputs of our system showed higher accuracy than previous ones.

1. Introduction

The analysis of sports videos has been a heavily investigated subject in computer vision. Sports video analysis is extremely helpful in swimming for evaluating the stroke rates and body postures. However, such analysis is still done manually, which is a time-consuming and laborious task. Thus, computer vision may have an important role in analyzing swimmers automatically. To achieve this, there are several methods to estimate and evaluate strokes automatically.

For example, Sha et al. [1] proposed an approach to estimating the stroke rate by computing the angle between the elbow and back. Ries and Lienhart [2] presented an approach which detects swimmers inside a target video and assigns an estimated position to their body parts. These approaches sometimes recognize body parts falsely in highly noisy environments because body parts are not fully visible in such situations.

Victor et al. [3] proposed a method to estimate strokes using a convolutional neural network (CNN). This method can detect strokes without detecting body parts. However, there is still an error when splashing occurs constantly. This is because this method does not take account of temporal sequences.

Thus, we proposed an approach to estimate the stroke using a model which takes account of temporal sequences. Concretely, we made a model using a CNN and MultiLSTM, which is an expanded model of LSTM. Our method applies a window of frames as the model’s input and obtains a point on the swimmer’s stroke signal. The peak of the stroke signal represents the frame where the swimmer’s arm is highest above the water surface (Fig. 1). Finally, we estimate the swimmer’s stroke from the stroke signal.

We obtained datasets from swimmers in many environments and performed experiments. Our approach to estimating strokes is shown to work equally well in various environments.

This work has the following two contributions.

- Improved accuracy of swimmer stroke estimation using a CNN and MultiLSTM.
- Swimmer stroke estimation in various environments.
2. Related Work

2.1 Swimmer detection

It is important to automatically detect a swimmer in an aquatic environment in computer vision. Such techniques are also used to automatically detect people drowning in swimming pool surveillance. Chen et al. [4] first used Gaussian mixture model to represent the background and swimmers. In the next step, the Adaboost [5] algorithm with Haar-like features was adapted to train a swimmer detector. Finally, a swimmer can be accurately separated from the swimming pool. However, when the camera used is handheld and unfixed, background subtraction is unsuitable. Sha et al. [6] solved this problem. By representing swimming videos using six stages (start, end, diving, turning around, under water, normal swimming), this approach can perform skin color segmentation, detect the pixel density in the segmentation result, and locate swimmer.

2.2 Stroke estimation

As a method of stroke analysis related to this work, Zecha et al. [7] presented a method to detect body parts and estimate stroke rate which divides the swimming cycle of each stroke into several intervals and specific object detectors are trained for each pose interval. Then, the stroke rate was computed using the frequency of these pose intervals. The deformable part model (DPM) was used as the body part model, so this approach could capture the positions of moving body parts and analyze strokes for various swimming styles. However, images in the water were necessary in order to capture entire body parts, making it difficult to apply this method to the swimming videos we used.

3. Proposed Method

The proposed framework is shown in Fig.2. Our approach first, detects a swimmer from videos and obtains swimmer images. Next, we input a window of swimmer images into the model of a neural network and obtain a point on a stroke signal. Finally, from the stroke signal, we estimate the swimmer’s stroke.

3.1 Swimmer detection

In this process, we first estimate the pool region using a projective transformation(Fig.3[a]). Next, we apply background subtraction to the pool region, detect the position that most includes foreground mask, and locate the swimmer position(Fig.3[b]). The swimmer’s position may be incorrectly detected from external noise such as water splashes. In our method, we use the Kalman filter to modify the swimmer’s position. The Kalman filter is used to make predictions based on a series of observed measurements which may contain noise and inaccuracies. It is expressed using a state transition matrix $F_t$ and observation matrix $H_t$ (see Eqs.(1) and (2)).

$$x_{t+1} = F_t x_t + G_t w_t$$  \hspace{1cm} (1)

$$y_t = H_t x_t + v_t$$  \hspace{1cm} (2)

In the swimming video we used, the swimmer position is considered as a model that moves on a straight line with constant acceleration. Thus, the state vector and the state transition matrix are given by Eq.(3). $x_t$ represents the location and velocity of the swimmer.

$$x_t = \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix}$$

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$  \hspace{1cm} (3)

3.2 Stroke estimation

3.2.1 Stroke signal

A stroke signal is used as our model’s label (Fig.1). First, we make binary signals. The binary signals show that if a
point is True, the swimmer’s arm is on the water surface and if a point is False, the arm is under the water surface. The center position of sequence which value is one corresponds to the position where a swimmer’s arm is highest above the water surface. Next, we let the position of that frame be the vertex of the cos function. The cos function is fitted so that one period corresponds to frames from a center position to the next center position. Finally, we create a stroke signal by changing the value in the range from 0 to 1.

3.2.2 Network architecture

We used a standard CNN to perform regression (Fig.4). The CNN’s input is a swimmer image, and the one’s label is a point on the stroke signal. The CNN we use is based on VGG-B [3]. In this method, Adam [9] is used for optimization and Relu is used as an activation function. Next, we obtain linear layer’s output from the CNN as a feature vector with 1024 dimensions. Then we use a temporal sequence made of feature vectors as the input of MultiLSTM. Finally, we obtain stroke signal as the output of MultiLSTM.

3.2.3 MultiLSTM

Let \( x \) be an input sequence \((x_1, ..., x_T)\) and \( y \) be an output sequence \((y_1, ..., y_T)\). LSTM maps \( x \) to \( y \) through a series of intermediate representations. MultiLSTM expands the temporal receptive field of both the input and output connections of LSTM.

MultiLSTM does not make any changes to the internal structure of the LSTM unit. However, this process allows a model to make a prediction including more temporal information. Thus, we expect this model to be robust to noise. The estimation by input model and output model of MultiLSTM are shown in Fig.5. The input data is a fixed-length window of feature vectors. The input is a combination of feature vectors and weights whose sum is 1. Concretely, we obtain \( V = (v_1, \cdots, v_T) \) by inputting swimmer images to the CNN. The input is the weighted combination \( x_i = \sum_t a_{it} v_t \), where \( T \) ranges over a fixed-size window of frames previous to \( i \). \( a_{it} \) is the contribution of frame \( v_t \) to input \( x_i \) and is expressed as follows.

\[
a_{it} \propto \exp(w_{ae}^T \tanh(W_{ha} h_{i-1}^h) \odot \tanh(W_{va} v_t))
\]

Here \((w_{ae}^T, W_{ha}, h_{i-1}^h, W_{va} v_t)\) are learned weights, \( \odot \) is element-wise multiplication, and \( a_{it} \) is normalized using the softmax function. Note that a standard LSTM input is a special case of this model, where all attention is focused on the last input window frame.

The MultiLSTM output is the average of the values outputted at time \( t \). A standard LSTM output is a special case where the attention at the current time \( t \) is 1. See [8] for further details.

3.2.4 Stroke estimation

We find stroke peaks from generated stroke signals. We detect the center of a series of frames above the threshold as the peak of the stroke (Fig.6).

4. Experiments

We evaluated our method of stroke estimation by comparing our results and those obtained with a previous method [3]. We also performed experiments in an unknown environment. The dataset consists of four scenes as shown in Fig.7. The environments of the four scenes and the number of data are shown in Table 1.

| Scene | Camera   | Player | Stroke |
|-------|----------|--------|--------|
| Scene1| fixed    | 32     | 2244   |
| Scene2| fixed    | 8      | 520    |
| Scene3| handheld | 3      | 37     |
| Scene4| handheld | 2      | 25     |

We did not use the data of scene 4 to train our model. Thus, the data of scene 4 is data for an unknown environment. The ratio of training data to test data in the dataset is 1:1. In our dataset, the only using a swimming style was freestyle. Detection is not performed in handheld camera, scene 3 and
4. Similarly to the previous method [3], the main evaluation metric used is the F-score. Each predicted stroke peak was considered a true positive if it was within 3 frames of the stroke peak label. The stroke estimation results are shown in Tables 2 and 3.

Table 2: Results of stroke estimation

|                | Scene1 | Scene2 | Scene3 |
|----------------|--------|--------|--------|
| Previous[3]    | 0.970  | 0.791  | 0.770  |
| CNN + LSTM     | 0.991  | 0.939  | 0.955  |
| CNN + MultiLSTM| 0.991  | 0.972  | 1      |

Table 3: Stroke estimation in unknown environment

|                | Precision | Recall | F-Score |
|----------------|-----------|--------|---------|
| Previous[3]    | 0.775     | 0.802  | 0.788   |
| Ours           | 1         | 0.701  | 0.824   |

Table 2 shows that our method can improve stroke estimation in each scene. Table 3 shows that we can correctly estimate more than 80% of strokes in an unknown environment.

5. Conclusions

This paper describes stroke estimation. Firstly, we develop a stroke estimation model using a CNN and MultiLSTM. We show that this model can estimate strokes with higher accuracy than before. Second, we made a dataset from swimming videos including various environments. We show that the model can estimate strokes in an unknown environment.

Acknowledgments

We would like to thank Keio University’s swimming team and Panasonic for providing videos.

References

[1] L. Sha, P. Lucey, S. Sridharan, S. Morgan and D. Pease: Understanding and Analyzing a Large Collection of Archived Swimming Videos, WACV, 2014.

[2] C. X. Ries and R. Lienhart: Automatic pose initialization of swimmers in videos, SPIE, 2010.

[3] B. Victor, Z. He, S. Morgan and D. Miniutti: Continuous Video to Simple Signals for Swimming Stroke Detection with Convolutional Neural Networks, CVsports, 2017.

[4] W. Chen, P. Cho, P. Fan and Y. Yang: A Framework for Vision-Based Swimmer Tracking, URKE, 2011.

[5] P. Viola and M. Jones: Rapid Object Detection Using a Boosted Cascade of Simple Features, CVPR, 2001.

[6] L. Sha, P. Lucey, S. Morgan, D. Pease and S. Sridharan: Swimmer Localization from a Moving Camera, DICTA, 2013.

[7] D. Zecha, T. Greif and R. Lienhart: Swimmer Detection and Pose Estimation for Continuous Stroke Rate Determination, Multimedia on Mobile Devices, 2012.

[8] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori and L. Fei-Fei: Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos, IJCV, 2017.

[9] D. P. Kingma and J. L. Ba: Adam: A method of stochastic optimization, arXiv, 1420.6980, 2014.