5—4

Improvement of Visualization System for Player's Sequential Forms in Sports Broadcasting

Jae Ho Lee, Sang Gil Lee***

Technical Research Institute, Korean Broadcasting System
18, Yoido-dong, Youngdeungpo-gu, Seoul, 150-790, Korea

* *T. : +82-2-781-5967, F. : +82-2-781-5999, email : jaeho@tri.kbs.co.kr,*
** *T. : +82-2-781-5971, F. : +82-2-781-5999, email : sglee@tri.kbs.co.kr*

Abstract

We improved a visualization system for a player's sequential forms called 'Multi-Motion' which has ever been used in NHK live broadcasting. It is to extract only the player's forms from background, and overlap them into one frame sequentially after fast performance for a vault game of gymnastics. It helps TV viewers understand the player's performing forms easily. We increased the performance of Multi-Motion by shortening the distance between the current frame and the reference frame, and the synthesis of the reference frame.

I. Introduction

For sports broadcasting, the interesting scenes are replayed slowly to show TV viewers a player's sequential forms while the sports scenes are lively broadcasted. If a player's sequential forms are displayed together in one background scene after the player has performed with fast speed, it will be helpful for TV viewers to understand the player's performing forms. For example, NHK has ever used 'Multi-Motion' system[1] for broadcasting of gymnastic sports game, which shows a player's all performing forms sequentially in one background scene from starting to final position after extracting only the player at the sequential frames.

There have been several studies in the extraction of moving objects from the static scene[2-4]. However, in order to extract a moving object from the background including undesired moving objects such as audiences, new algorithm is needed. In Multi-Motion system, the motion compensation is used to remove the undesired moving object. It uses the principle that audiences moves slowly while the player moves very fast. In order to extract only the player exactly, the player at the reference frame and the player at the current frame should not be overlapped in the same spatial position. So we should select the reference frame which has no player in the same position as the player's position in the current frame. Therefore the distance between two frames should be long. It deteriorates the performance of the motion compensation which is used in the algorithm of Multi-Motion system.

It is hopeful that the distance between the reference frame and the current frame is as short as possible, and the players in two frames are not overlapped with each other.

we propose a method to shorten and fix the distance between the current frame and the reference frame, removing the player in the reference frame not to be overlapped with the player in the current frame. We can remove the player in the reference frame through the synthesis of the reference frame.

we will explain the proposed algorithm and system environments in section II, and III. And experimental results will be described in section IV, and conclusion will be given.

II Proposed Algorithm

The procedure for the realization of the proposed algorithm is briefly depicted below.

(1) *Decision of an initial processing window* : To reduce the processing time, we process inside the limited processing area which includes the player, rather than the total area of the frame.

(2) *Detection of a starting frame* : It is not necessary to process frames where no player exists. So we detect a starting frame where a player appears for the first time.

(3) *Extraction of the player at the starting frame,* (4) *Synthesis of the reference frame, and* (5) *Extraction of the player at the subsequent frames* : At the starting frame, the reference frame has no player, but the players exist at the subsequent reference frames. So they are synthesized to remove the player in order not to be overlapped with the player at the current frame.

Figure 1 shows the overall block diagram which realizes the extraction of a player's performing forms from the sequential frames. Following the steps from (1) to (5), we will explain the proposed algorithm.

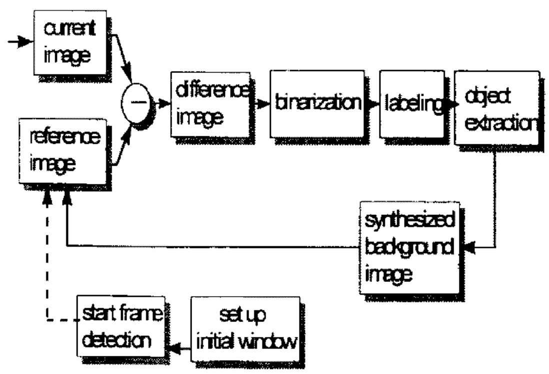

Figure 1. Overall block diagram for the realization of the proposed algorithm.

1. *Decision of an initial processing window*

First, we set a processing window to process the proposed algorithm. If we do not limit the processing area, we should process the algorithm in all area of the corresponding frame. It results in a great amount of time consumption. As shown in Figure 2, we select one among two windows, A and B mode, because a player's moving direction might be changed according to a camera position. The position and size of the window are changed according to the player's movement. It is helpful that no moving object exists inside the initial processing window to detect the starting frame.

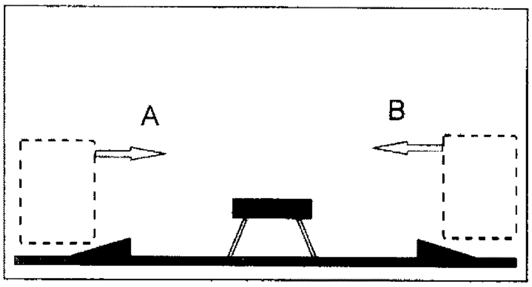

Figure 2. The selection of a initial processing window according to the player's moving direction.

2. *Detection of a starting frame*

In order to extract a player, we should detect the starting frame which will be the first processing frame. It is not necessary to start the processing for frames before a player starts to appear within the frame, because the processing for the frames that has no player results in time consumption only.

In advance, the difference image is obtained between the current frame and the first previous frame, and binarized. In Eq. (1), We count the number of pixels corresponding to the binarized objects. If the number of non-zero pixels becomes larger than an arbitrary value, it means that the player appears inside the processing window.

Otherwise, only noise components are detected because of the movement of background. If we detect the starting frame SF_n, we set the first previous frame SF_{n-1}, as a reference frame RF_n. Therefore the reference frame indicates background which has no player, and is the nearest to the frame in which the player starts to appear.

Eq.(1) is the decision equation whether the player exists or not.

$$D_n > S_{\min} \quad n=0,1,2\ldots, \tag{1}$$

where D_n is the number of non-zero binarized pixels, and $S_{\min}$ is an arbitrary threshold value.

3. *Extraction of the player at the starting frame*

The frame SF_{n-1}, is the first reference frame RF_n, as explained in Section 2, and SF_n becomes the current frame CF_n. We obtain the difference image between this reference frame RF_n, and the first current frame CF_n. At the first current frame, a player starts to appear inside the processing window. The difference images are binarized. If some pixel belongs to the moving objects, it will be '1'. Otherwise, it will be '0'. The player exists among the pixels of which binary values are '1'. In order to extract only the player, labeling processing is applied to those moving objects. Resultantly, only the pixels corresponding to the player might be left with its value being '1' while the other part will be '0'.

4 *Synthesis of the reference frame*

The player should not exist inside the processing window at the reference frame. Otherwise, extracted player at current frame would be overlapped with the player at the reference frame. At the conventional method[1], we select the reference frame so that the distance between the reference frame and the current frame is long enough for the player at the reference frame not to be overlapped with the player at the current frame. However, many useless moving objects might occur. Resultantly we cannot extract the exact player's form at the current frame. We propose an algorithm to remove the player at the reference frame, while making the distance between the reference frame and the current frame short.

Figure 3 shows the block diagram which realizes the proposed algorithm.

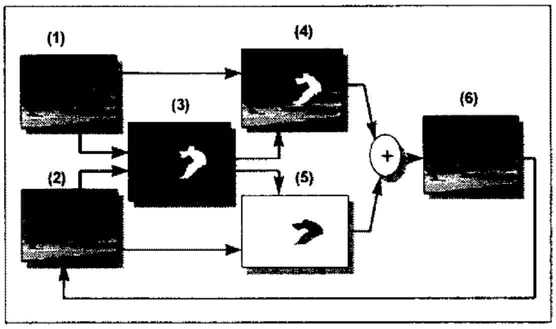

Figure 3. Synthesis of the reference frame. (1) Current frame, (2) Reference frame, (3) Binarized image, (4) The background frame at which the player has been excluded from (1), (5) The background frame at which the player's block in (3) is extracted from (2), (6) The synthesized frame by the frames in (4) and (5).

Assuming that we start processing from the starting frame SF_n becomes the current frame CF_n((1) in Fig. 3), and the first previous frame SF_{n-1}, becomes the first reference frame RF_n((2) in Fig. 3). The player's form which has been extracted through differentiation, binarization and labelling are obtained((3) in Fig. 3).

Before extracting a player at the next frame CF_{n+1}, we should synthesize the reference frame RF_{n+1} instead of using the current frame CF_n, for the reason explained above.

Refering (3) in Figure 3, we remove the area from CF_n((1) in Fig.3) where the player exists ((4) in Fig.3), and take the area at

RF_n((2) in Fig.3) which corresponds to the place where the player exists at CF_n ((5) in Fig.3). And then these two frames are combined into one frame ((6) in Fig.3). This frame becomes the reference frame RF_{n+1}, which will be used for the extraction of the player at the next frame. This frame is smoothed to remove the artifacts resulted from the combination. Using the current frame CF_{n+1}, and the reference frame RF_{n+1}, The player will be extracted as explained in the next section.

5. *Extraction of the player at the subsequent frames*

Using the reference frame, $RF_{n+k}(k \geq 1)$, and the current frame, $CF_{n+k}(k \geq 1)$, we derive the difference image which is considered to be the moving objects. Here the reference frame RF_{n+k} is derived by the method explained in Section 4.

If the luminance component is only used for getting difference image, we cannot often recognize the moving pixels when the luminance values between the current frame and the reference frame are similar even though the color values are different. It might cause the distortion of extraction, such as holes. R, G, and B components are used to reduce the distortion effectively. Firstly we synthesize three reference frames which correspond to R, G, and B, respectively, and derive a binarized difference image by Eq. (2) and Eq. (3).

$$DR_m(x,y) = |\, CFR_m(x,y) - RFR_m(x,y) \,|$$
$$DG_m(x,y) = |\, CFG_m(x,y) - RFG_m(x,y) \,| \quad (2)$$
$$DB_m(x,y) = |\, CFB_m(x,y) - RFB_m(x,y) \,|$$

where $DR_m(DG_m, DB_m)$ is the difference image of $R(G, B)$ component at the m^{th} frame, and $CFR_m(CFG_m, CFB_m)$ and $RFR_m(RFG_m, RFB_m)$ mean the current frame and the reference frame of $R(G, B)$ component, repectively.

$$B_m(x,y) = \begin{cases} 1, & DR_m(x,y) \geq T \text{ or } DG_m(x,y) \geq T \text{ or } DB_m(x,y) \geq T \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where T is the arbitrary threshold value for binarization.

Figure 4 shows a procedure for the extraction of the player by the proposed method.

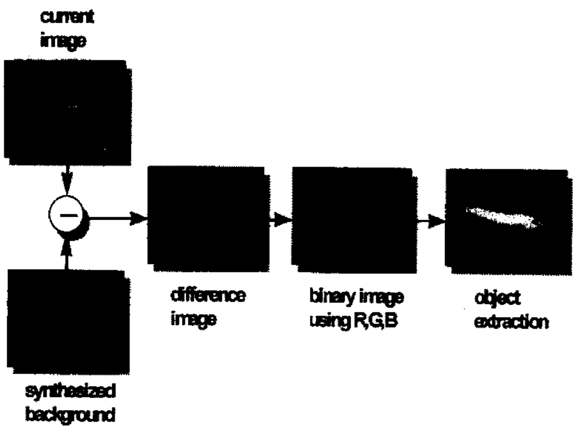

Figure 4. The procedure for the extraction of a player by the proposed method.

Ⅲ. System Environments

The system consists of a real-time hard disk system (HDS), a workstation, A/D & D/A converter, a monitor and a camera as shown in Figure 5.

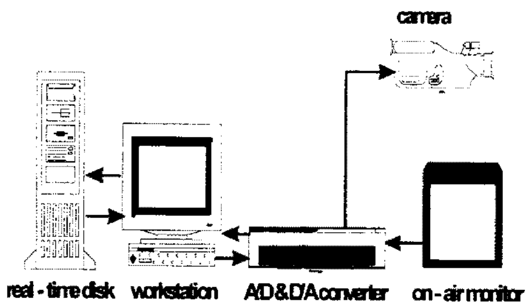

Figure 5. System environments

The HDS transfers video data with the transfer rate of nearly 40MB/sec - more than enough for uncompressed video storage and retrieval at 30 frames/sec. The workstation, 'Indigo2-impact', has a I/O video board, 'Impact Video Board', which can input and output image data with real-time speed.

The frame signals from the camera are inputed to the workstation through A/D converter. And input frames are stored into the memory of the workstation. Using the image data stored into the memory, we extract a player's forms at sequential frames, store the results into real-time HDS. The result are outputed from the workstation through a D/A converter, and displayed for broadcasting.

Ⅳ. Experimental Results

The frame sequences are stored into the memory of the workstation (Indigo Impact2) through the real-time video board(Impact Video Board) from the input source(camera). Instead of storing and processing every frame, we store and process images into the memory every third frame, because it looks adequate and good to display a player's sequential forms with the interval of every three frame in the case of a 'vault game' as shown in Figure 9. We set the size of the initial processing window with 250H×150V(field size : 720H×243V), and value of S_{min} in Eq.(1) with 900. In Eq.(3), T is set to be 10.

Figure 6(b) shows the results obtained by the conventional method[1] after setting the distance by 3 frames between the current frame and the reference frame. The overlapping of the players at the current frame and the reference frame results in the incorrect extraction of the player by overlapping as shown in Figure 6(b). To avoid the overlapping, the distance between the reference frame and the current frame is set with 9 frames, for example. We can find noise even though the overlapping of the players at the reference frame and the current frame disappears as shown in Figure 6(c). Using the proposed algorithm, we obtained the results as shown in Figure 6(d), where not only the noise was reduced, but also the overlapping disappeard. Figure 7 and Figure 8 are the results obtained by the proposed algorithm. Figure 9 and Figure 10 show the final results for a male and a female, respectively, where the extracted players are displayed sequentially in one frame scene.

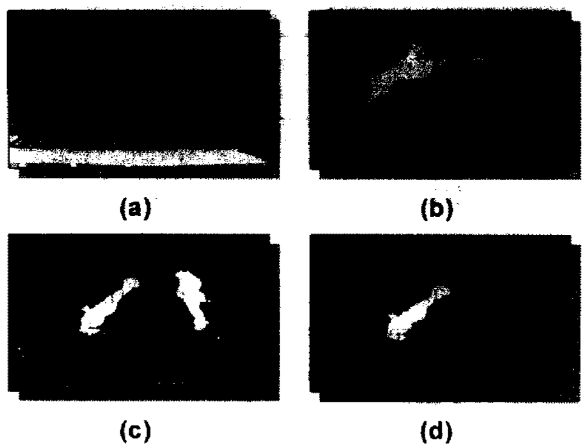

Figure 6. The comparison of the performance by the conventional and the proposed method. (F.I : frame interval) (a) Original image (b) Conventional method(F.I. : 3). The overlapping noise is seen. (c) Conventional method (F.I. : 9). The overlapping noise is not seen, but the noise of moving objects increases (d) Proposed method (F.I. : 3). The overlapping noise is not seen, and the noise of moving objects decreases.

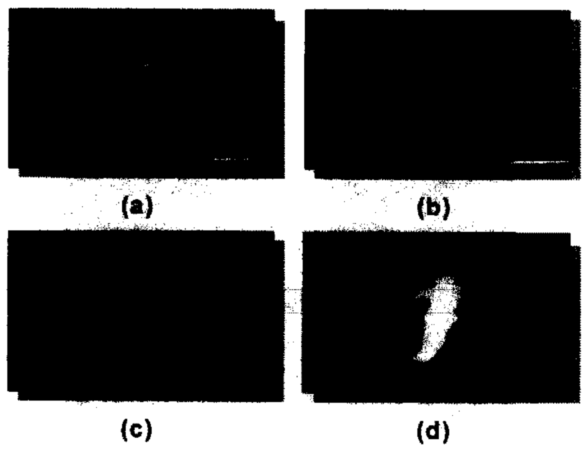

Figure 7. The results by the proposed method. (a) The current frame (b) The synthesized reference frame (c) Binarized image which is combined by three binary images of R, G, and B components (d) Final binarized image obtained by labelling (c).

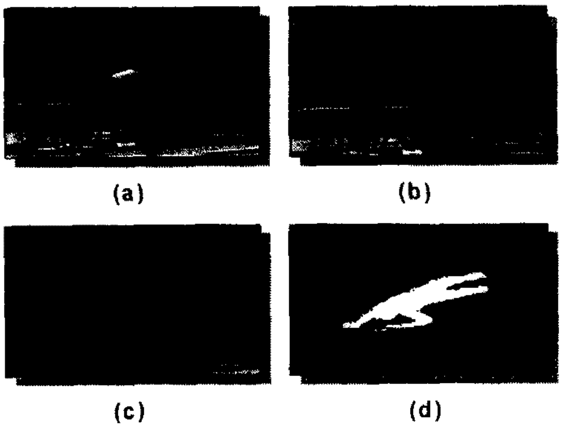

Figure 8. The results by the proposed method. (a) The current frame (b) The synthesized reference frame (c) Binarized image which is combined by three binary images of R, G, and B components (d) Final binarized image obtained by labelling (c).

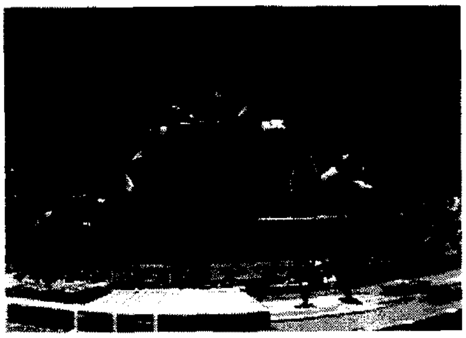

Figure 9. A male player's form displayed sequentially in one scene.

Figure 10. A female player's form displayed sequentially in one scene.

V. Conclusion

In this paper, we proposed the algorithm that accomplishes the improvement of visualization system which can display a player's sequential forms into one frame. In order to do it, we propose the method how we shorten the distance between the reference frame and the current frame, and synthesize the reference frame. We also reduced the processing time by using the memory in the workstation directly while processing.

Our further works include the application to another sports game, automatic adjustment of parameters, and so on.

Acknowledgement

The authors thank Director M. J. Cho and all members in Image Processing Research Division, Technical. Research Institute, KBS.

References

[1] S. G. Lee and N. Yagi, "Visualization System for Player's Sequential Forms at Sports Broadcasting," *Proc. Int'l Workshop on New Video Media Tech*, pp. 88-93, 1996.

[2] S.D. Blostein and T. S. Huang, "Detecting small, moving objects in image sequences using sequential hypothesis testing," *IEEE Trans. on Signal Processing*, vol. 39, no. 7, pp. 1611-1629, Jul. 1991.

[3] M. Bichsel, "Segmenting simply connected moving objects in a static scene," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 16, no. 11, pp. 1138-1142, Nov. 1994.

[4] N. Mine, et al., "Detection of Change Region by Integrating Subtracted Image and Edge Boundary Image," *IEICE D-II*, vol. J77-D-II, no. 3, pp. 631-634, 1994.