SIMPLE AND EFFECTIVE ALGORITHM FOR AUTOMATIC TRACKING OF A SINGLE OBJECT USING A PAN-TILT-ZOOM CAMERA

Yu-Wen Huang, Bing-Yu Hsieh, Shao-Yi Chien, and Liang-Gee Chen

DSP/IC Design Lab
Graduate Institute of Electronics Engineering, National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan
{yuwen, bingyu, shaoyi, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

This paper presents a simple but effective algorithm for a pan-tilt-zoom camera to automatically track a single moving object. The proposed tracking algorithm is suitable for but not limited to stationary background, and it can tolerate reasonable noise and light change. The main idea is to initially capture the background information and to calculate the difference between incoming frame and background buffer. Block-based processing is adopted to reduce computation and to alleviate noise effects. Skin-color detection is combined with spatial and temporal information to let the camera more likely to focus on human faces. Even more, the tracking algorithm can be integrated with region-of-interest video coding, which allocates more bits for human faces to produce better subject views. Many practical situations have been tested and the simulation results show that the proposed tracking algorithm is useful for surveillance systems.

1. INTRODUCTION

Surveillance system assists humans by providing an extended perception and reasoning capability about situations of interest that occur in the monitored environments. However, human operators cannot focus their concentration on many monitors for a long time. Therefore, an intelligent surveillance system that provides the functionality of automatic detection and tracking of intruders may be of help. In this paper, our main contribution is to develop an automatic tracking algorithm that makes the smart camera try to keep the moving object from running out of sight.

The temporal information, change detection, which finds the difference between two consecutive frames, is a very good cue for tracking. However, when the object moves fast, the uncovered background will be detected as changed and thus will influence the tracking result; when the object stops moving for a while, conventional change detection will lose track of it. Background registration [1][2], which was used for video object extraction, can easily solve the uncovered background problem and the still object problem. Instead of finding active regions, background registration collects the stationary background. After the background information is registered, the difference between current frame and background frame is more reliable than the temporal difference between two successive frames. Our tracking algorithm adopts the same scheme. The background information is initially captured, which may take several seconds to several minutes according to the range allowed for the pan-tilt-zoom camera to cover. The tracking algorithm can be very robust with the help of background information.

The rest of this paper is organized as follows. In Section 2, the system overview is described. The tracking algorithm is presented in Section 3. Section 4 shows the experimental results with some discussions. Finally, Section 5 gives a conclusion.

2. SYSTEM OVERVIEW

The tracking algorithm is integrated with video coding, as shown in Fig. 1. The input frames are inputted into both the tracking module and the video coding module. The tracking module feedbacks the "motion vector" to the pan-tilt-zoom camera to decide its next position of image center and to keep the moving object in the middle of the image. At the same time, the tracking module informs the video coding module of the regions, usually human faces, that should be enhanced. The encoded bitstream can either be sent out via network or be stored in hard drives.

An example of the whole view of the monitored space that the camera is allowed to cover is shown Fig. 2(a). Let us denote each pixel in Fig. 2(a) as a position of camera. If the background frames must be stored for all the positions, the frame memories will be too huge. Sprite generation technique [3] can solve this problem. Nevertheless, the reconstructed background frame that is warped from sprite cannot well fit the background in current frame at every position, which is caused by imperfect feature matching, the limitation of global motion model, and errors from interpolation, as shown in Fig. 3. Furthermore, matching and warping is too time-consuming to meet real-time requirements. Fortunately, pixel-accuracy is not needed for our application. The background information of our system is only available on the grid points plotted in Fig. 2(a). That is, when the camera directly faces a grid point in Fig. 2(a), we save the corresponding background frame, as shown in Fig. 2(b). Thus, our system uses larger but reasonable frame memories, provides better robustness, and requires much less computation, compared to the scheme of sprite generation. To sum up, the camera moves between the grid points, and the tracking algorithm is only performed while the positions of camera are on the grid points. At the beginning, the camera is on a certain grid point. If no moving object is detected, it will periodically move to the next grid point with a regular

0-7803-7304-9/02/$17.00 C2002 IEEE

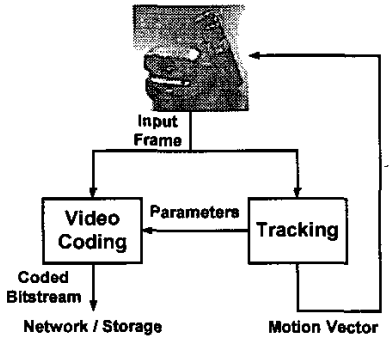
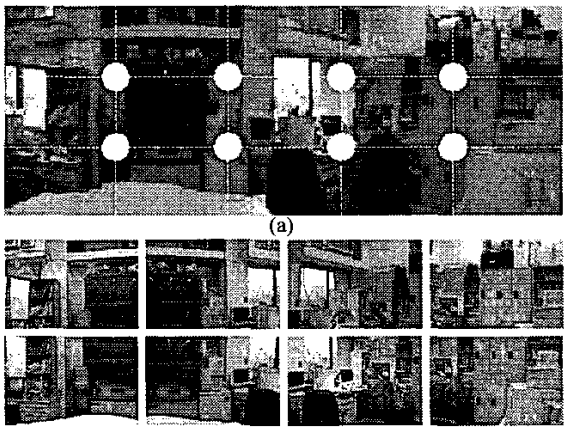


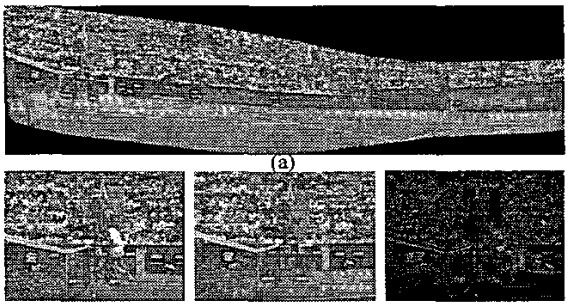
Fig. 1. Integration of tracking and video coding.



(a)

(b)

Fig. 2. (a) Whole view of the monitored space that the camera can cover, (b) corresponding background frames of grid points.



(a)

(b) (c) (d)

Fig. 3. (a) The sprite for sequence *Stefan*, (b) frame #300, (c) corresponding background warped from sprite, (d) absolute difference between frame #300 and its background.

order or remain still in this position. Once a moving object is found, the camera will select the grid point that is closest to the object as the next target position of camera. Before the camera arrives the next grid point, the captured frames with not-on-grid-point camera positions are omitted by the tracking module, and are only inputted to the video coding module.

3. TRACKING ALGORITHM

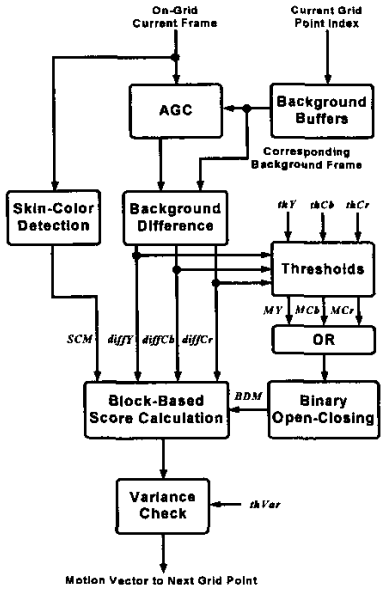
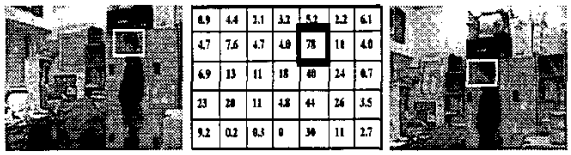


Fig. 4. Flowchart of the tracking algorithm.



(a) (b) (c)

Fig. 5. (a) Current frame, (b) the scaled score of each block, (c) next frame, note that the camera moves the block with the highest score toward the new image center.

The flowchart of our tracking algorithm is shown in Fig. 4. It consists of spatial domain processing, which includes skin-color detection and variance check, and temporal domain processing, which mainly includes background difference. The Block-Based Score Calculation utilizes information in both domains with different weightings and constrains.

3.1 Block-based processing

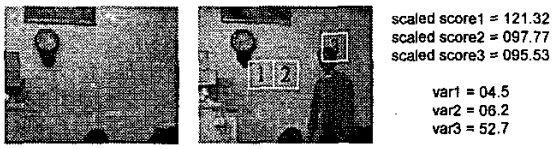
The incoming frame is divided into blocks. Each block is given a score by the tracking algorithm, as shown in Fig. 5. The camera will find the block with the highest score, and then a motion vector for the camera will be mapped. According to the motion vector, the camera will move its center of image to the grid point that leaves the selected block in the middle of the image. That is, the center of the block with the highest score will become the new center of image captured by the camera. The advantages of block-based processing are better immunity against noise and less computation, compared to pixel-based processing.

3.2 Background difference

The absolute difference between on-grid-point current frame compensated by automatic gain control (AGC) and its corresponding background frame is calculated for each of the YCbCr

(a) (b)

Fig. 6. (a) Original frame, (b) the result of skin-color detection.



(a) (b) (c)

Fig. 7. (a) Background frame, (b) current frame under unstable light source and the three blocks with the largest three scores, (c) the scaled scores and variances of the three blocks.

components. Let them be denoted as *diffY*, *diffCb*, and *diffCr*, respectively. Three thresholds, *thY*, *thCb*, and *thCr*, are applied to form three binary masks, *MY*, *MCb*, and *MCr*. For example, if *diffY* is larger than *thY*, *MY* will be 1; otherwise, *MY* will be 0. Next, the three binary masks are ORed, and then performed with binary morphological open-closing operation [4] to eliminate the noise and then to form the binary background difference mask, *BDM*. Note that *BDM* is also 1 or 0.

3.3 Skin-color detection

For surveillance or videoconference systems, human faces are very likely to be of users' interests. Thus, we should give higher scores for the blocks including human faces. In [5], it is found that human faces are almost independent of Y color component but are limited in a small range of Cb and Cr components. The result of skin-color detection is shown in Fig. 6, and a binary skin-color mask, *SCM*, can be generated. The *SCM* is 255 for skin-color; otherwise it is 0. Note that although part of the background similar to skin-color is detected, it will not affect the tracking result, which will be described in the next subsection.

3.4 Score calculation

Let *W*, *H*, *BW*, and *BH* be the image width, image height, block width, and block height, respectively. The score of a block (i, j) can be determined as follows:

$$BlockScore(i,j) = \sum_{n=0}^{BH-1}\sum_{m=0}^{BW-1} PixelScore(i \cdot BW + m, j \cdot BH + n)$$

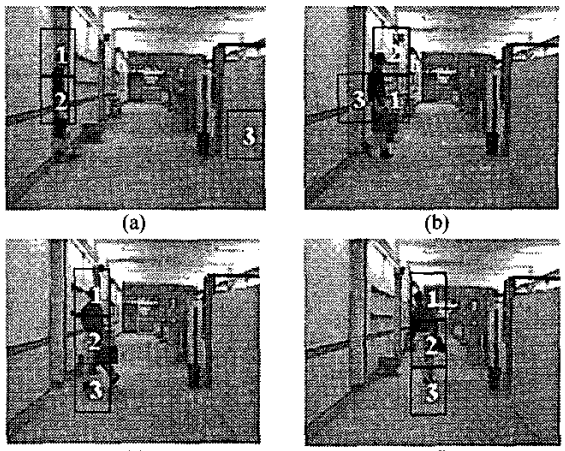
$$PixelScore(x,y) = [(1-\beta)\cdot diff(x,y) + \beta \cdot SCM(x,y)]\cdot BDM(x,y)$$

$$diff(x,y) = diffY(x,y) + diffCb(x,y) + diffCr(x,y)$$

$$0 \le i < W/BW, \quad 0 \le j < H/BH, \quad i, j \in Z$$

$$0 \le x < W, \quad 0 \le y < H, \quad x, y \in Z$$

The *BlockScore* is the sum of *PixelScore* within the block. If the *BDM* is active, the *PixelScore* will be the weighted sum of the temporal information, *diff*, and the spatial information, *SCM*.



(a) (b) (c) (d)

Fig. 8. The detected three blocks with the largest three scores for *Hall Monitor*, (a) frame #21, (b) frame #33, (c) frame #42, (d) frame #74.

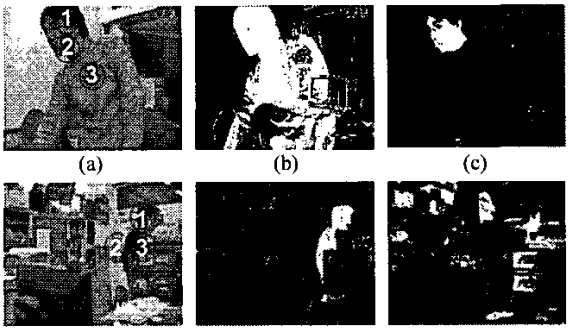
The higher the weighting factor β, the more the skin-color is of camera's interest. On the contrary, the *PixelScore* will be 0 if the *BDM* is inactive. Note that by this definition, the skin-color in the background cannot contribute anything to the *BlockScore* since the *BDM* is inactive in the background region.

3.5 Variance check

If the light source is not stable, some background regions will activate the *BDM* and have large values of *diff*, which will affect the tracking result. Fortunately, it is found by observation that under unstable light source, the intensity variation of flat background tends to be much larger than that of regions with significant texture. Thus, we keep the largest *N BlockScores* and check the variance of these *N* blocks in the original image domain. As illustrated n Fig. 7, the scores of the blocks marked by "1", "2", and "3" are the largest, the second largest, and the third largest, respectively. Note that block 1 and block 2 belong to stationary background, but their scores are high. It is due to the unstable light source, which makes the change of reflection on the white wall much larger. However, the camera will choose block 3 with the help of variance check. The variances of block 1 and block 2 are much smaller than block 3. A proper threshold, *thVar*, can exclude block 1 and block 2. After the target block is found, the camera moves this block toward the centers of the following frames.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The standard sequence *Hall Monitor*, which is filmed by a fixed camera under unstable light source, is tested for our tracking algorithm. We select the first frame, which includes no moving objects, as background frame. The results of the detected blocks with the largest three scores are shown in Fig. 8. Again, the scores of the blocks marked by "1", "2", and "3" are the largest, the second largest, and the third largest, respectively. Except block 3 in Fig. 8(a), the rest blocks correctly fall on the regions including the moving person. Even if the score of block 3 in Fig.



(a) (b) (c)

(d) (e) (f)

Fig. 9. (a)(d) Current frame and the blocks with the three highest scores, (b)(e) background difference mask *BDM*, (c)(f) skin-color mask, *SCM*.

8(a) were the highest, it would not be of camera's interest due to its small variance.

A SONY EVI D30/D31 Pan-Tilt-Zoom camera is placed in our laboratory to test the practical situations. The experimental results are shown in Fig. 9. Fig. 9(a) and (d) are the current frames and the blocks with the three highest scores marked by "1", "2", and "3". Fig. 9(b) and (e) are the background difference masks (*BDM*). Fig. 9(c) and (f) are the results of skin-color detection (*SCM*). Note that although there are many regions similar to skin-color in the background as shown in Fig. 9(f), they hardly contribute anything to the scores since the *BDM* is inactive in these regions, as shown in Fig. 9(e). The "active-*BDM*-and-active-*SCM*" regions tend to have large scores, and thus the camera tracks on the human faces. Our smart camera is also tested for a long period of time. The successive tracking results are shown in Fig. 10. The white rectangles are the target blocks that camera selects to track on. As you can see, once the moving object is detected, the camera keeps the object in the middle of the image no matter where the object moves.

The tracking algorithm is also integrated with video coding. Matching pursuits [6] is chosen as an example. In Fig. 11(a), the dotted rectangle is the block with the highest score, and the solid rectangle is the bounded box of active *BDM* after elimination of too small regions. Atoms are first used in the dotted rectangle. After the residue in the dotted rectangle is coded and if the number of coded atoms does not exceed the budget, the residue in the solid rectangle is coded. Finally the residue in the rest area takes the rest budget if there is still some left. An example of region-of-interest video coding is shown in Fig. 11(b).

If the background is time variant, an updating scheme should be developed for the background buffer. This is a tough task and is left as our future work. In addition to the updating scheme, there exists an easier solution. We can utilize only the temporal difference between two successive frames to replace the *BDM* with change detection mask *CDM*. It has the advantage of much less memory. However, some robustness is sacrificed, since conventional change detection cannot properly deal with uncovered background and still object.

5. CONCLUSION

A simple but effective algorithm for a pan-tilt-zoom camera to automatically track a single object is proposed. Once a moving object is detected, the camera moves with the object to keep it in the middle of the image. Our tracking algorithm utilizes both spatial and temporal information and is in favor of human faces. The entire system also includes region-of-interest video coding.

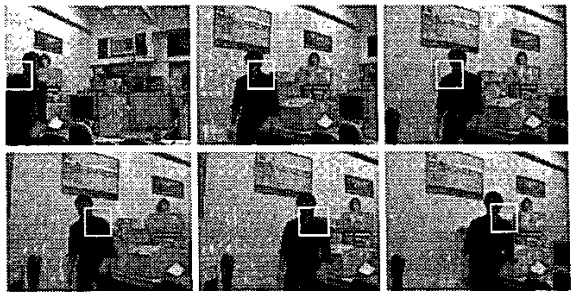
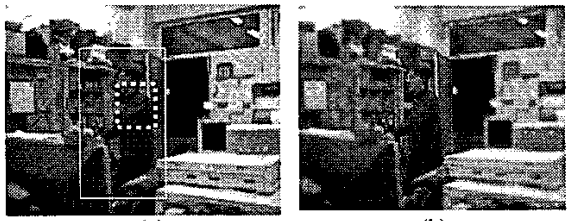


Fig. 10. Successive tracking results.



(a) (b)

Fig. 11. (a) Original frame and the regions that should be enhanced, (b) an example of region-of-interest video coding.

6. ACKNOWLEDGEMENT

The authors would like to thank Mr. Chao-Tsung Huang for the great effort on the implementation of matching pursuits.

7. REFERENCES

[1] S.Y. Chien, S.Y. Ma, and L.G. Chen, "Efficient moving object segmentation algorithm using background registration technique," accepted by *IEEE Trans. on Circuits and Systems for Video Technology*.

[2] Y. W. Huang, S. Y. Chien, B. Y. Hsieh, and L. G. Chen, "Automatic threshold decision of background registration technique for video segmentation," in *Proc. of Visual Communications and Image Processing 2002*, pp. 552-563.

[3] A. Smolic, T. Sikora, and J. -R. Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1227-1242, Dec. 1999.

[4] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1, 1992

[5] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551-564, Jun. 1999.

[6] R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 158-171, Feb. 1997.