International Journal of Smart Home
Vol.10, No.10 (2016), pp.157-168
http://dx.doi.org/10.14257/ijsh.2016.10.10.15

IACTS: A Novel Indoor Active Camera Tracking System

Yahya Imad Mohammed[1] and Jong Myung Rhee[2]

[1]*Dept. of Information and Communication, Myongji University*
116 Myongji-ro, Cheoin-gu, Yongin, Gyeonggi-do, 449-728, Korea
[2]*Dept. of Information and Communication, Myongji University*
116 Myongji-ro, Cheoin-gu, Yongin, Gyeonggi-do, 449-728, Korea
[1]*yahyamohamd@yahoo.com*, [2]*jmr77@mju.ac.kr*

Abstract

An active camera tracking system (ACTS) is comprised of a camera that can rotate automatically toward an object of interest so the object remains within the camera's field of view. To rotate the camera toward an object, the camera needs to detect the object's location. Detecting an object's location can be achieved using image processing techniques. However, conventional image processing techniques require expensive calculations in addition to expensive hardware, which in turn limits the video frame processing speed. In this paper, we propose an effective ACTS to monitor and track objects automatically with high accuracy. The proposed approach automatically initializes the object to be tracked using an absolute difference motion detection technique. To track the object correctly, the minimum output sum of squared error (MOSSE) adaptive correlation tracker is applied to provide an accurate and fast visual tracking. We also implemented an accurate scale estimation for the visual tracking technique that detects the scale changes of the object. Additionally, the proposed ACTS uses a closed loop controller to control the pan-tilt speed of the system. By combining these techniques, the proposed ACTS can provide accurate tracking with smooth rotation.

Keywords*: correlation tracker, active camera, motion detection, pan-tilt controller.*

1. Introduction

Surveillance and monitoring cameras are in wide use today, and most systems use conventional static cameras. A static camera is fixed on a static base to monitor a limited field of view. If the field of view cannot be covered with a single static camera, more than one camera must be installed. On the other hand, an active camera tracking system (ACTS) is comprised of a camera that is attached to a pan-tilt base (PTB) with two motors to rotate the PTB in both horizontal and vertical directions as needed. Consequently, this monitoring system with a single active camera can replace other monitoring systems that use several static cameras. However, the ACTS must be provided with a mechanism that enables it to rotate toward the objects or areas of interest automatically.

To locate the object of interest, an ACTS may use a specific image processing technique or a combination of techniques. In [1], the authors suggested using a motion detection technique. When the system detects a motion, the camera rotates toward the motion location. However, the camera rotation itself produces a motion within the video frames; therefore, they proposed using a motion feedback from the pan-tilt to compensate for the effect of the camera rotation. On the other hand, the adaptive correlation tracker proved an accurate performance and low processing overhead [2]. To track an object, the adaptive correlation tracker must be provided with a template of the object to be tracked. It measures the maximum correlation value between a template image and the video frame, where the highest correlation between them is considered the new object location. The normalized correlation tracker proved robust against the intensity changes because

ISSN: 1975-4094 IJSH

the normalization process remove the intensity effect as in [3]. If the template and the frame intensities are different, the tracker is still able to detect the location of the object of interest. The authors in [4] implemented a modified normalized cross correlation tracker. The authors also proposed using three sizes of dynamic templates—110%, 100%, and 90%—to detect the scale of the object. The window that produces the highest correlation value is considered the tracked object's new scale. Their system shows good performance while running at a speed around 30 frames per second (FPS).

Conventional monitoring systems initialize the object to be tracked manually. In our system, however, we propose to initialize the object to be tracked using an absolute difference motion detection technique [5]. Therefore, our ACTS tracks the initialized object using a tracking algorithm that must overcome a number of challenges, including scale changes, object deformation, intensity changes, object fade, and collision with other objects that might block the view. On the other hand, the visual tracking execution speed is critical for the system. The calculation complexity needs to be low for the system to run at real-time speed. In our proposed system, we implement the adaptive correlation tracker that depends on the minimum output sum of squared error (MOSSE) [6]. It has a low overhead that allows the system to run in real time, and it overcomes the previously mentioned challenges with good accuracy.

We also implemented the accurate scale estimation for visual tracking [7] to detect the object scale changes as the object moves closer or farther from the camera, which also adds a low overhead to the system.

The camera rotation toward the tracked object is important to maintain the tracked object within the camera center. The PTB rotation speed needs to be smooth and varied according to the tracked object speed and its distance from the camera center. We propose a closed loop pan-tilt controller (PTC) that provides the current PTB speed to the system. The PTB speed is used to calculate the tracked object's relative speed. The object distance from the camera center and its speed are used to determine a suitable rotation speed for the PTB so that the tracked object remains within the center of the camera. Our system runs around 140 FPS for a 320 × 240 frame resolution while providing a smooth and accurate rotation toward the tracked object.

The remainder of this paper is organized as follows. In Section 2, we present the proposed ACTS. In Section 3, the experimental results are discussed. Finally, we provide our conclusion and suggestions for future work in Section 4.

2. The Proposed ACTS

We divide our proposed ACTS into three algorithmic models: first, the region of interest initializer (ROII); second, the visual target tracking (VTT); and third, the PTC. The system block diagram is illustrated in Figure 1.

2.1. Region of Interest Initializer

The correlation tracker needs a template image for the object of interest to start the tracking process. The tracker detects the maximum correlation peak value between the template and the video frame to detect the tracked object's location. Conventionally, the tracked object template is provided manually as described in [8]. The template must be provided automatically because we propose to build a fully automatic system.

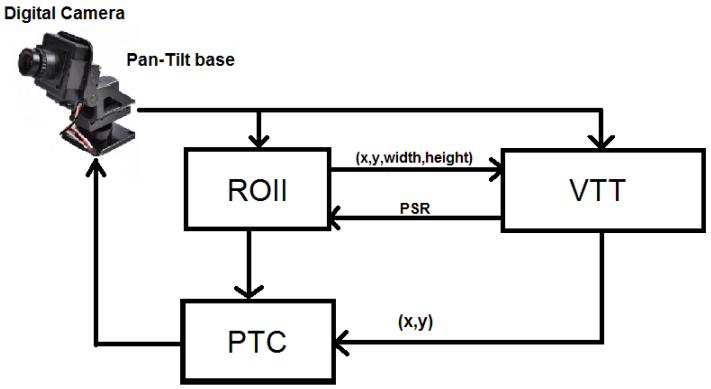


Figure 1. ACTS Block Diagram

The ROII is the part of the process responsible for providing the object template image. In our proposed system, we assumed that the object of interest can move, and the background is static or contains a minor motion. However, as the object moves, the system detects its motion and considers it as the target to be tracked. The ROII detects the produced motion and creates a window around the motion location that captures the moving object image. The window size depends on the size of the motion contour, width, and height. Later, the VTT uses this template to track the object.

To detect the object motion, we need to implement a motion detection technique that suits our system. A number of motion detection techniques have been suggested, such as background subtraction, also known as foreground detection [9]; spatio temporal entropy [10]; temporal difference; and optical flow analysis [11]. Although these methods work with a fixed camera, our camera is active. Therefore, in this paper, we implement the absolute difference motion detection technique, which requires two consecutive video frames to detect motion. The processing flow of the absolute difference technique is shown in Figure 2.

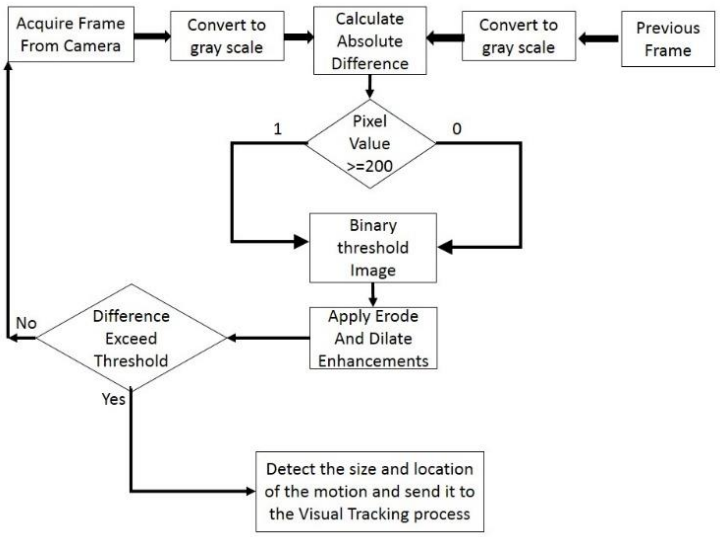


Figure 2. Absolute Difference Motion Detection Process Flow Diagram

The motion detection process works as follows. The ROII acquires two frames from the camera: the current frame and the previous frame. These two frames are in RGB color format, but the absolute difference method works with grayscale images. Thus, the process converts the RGB color format to grayscale, which can be expressed analytically as follows [12].

$$Y = 0.299 * R + 0.587 * G + 0.144 * B \quad (1)$$

R is the red component of the image, G is the green, B is the blue, and Y is the grayscale output of the image. A matrix subtraction provides the difference between the two images as follows.

$$Y_d = |I_{k+1} - I_k| \quad (2)$$

Y_d is the resulted grayscale difference image, I_{k+1} is the current grayscale image, and I_k is the previous grayscale image. The grayscale values range between 0 and 255, where 0 represents absolute black, and 255 represents the maximum white. Now, we have Y_d as the absolute grayscale difference, so we need to convert it to a binary image where 0 is black and 1 is white. Converting Y_d to a binary image can be expressed as follows.

$$Y_b = \sum_0^{w-1} \sum_0^{h-1} \begin{matrix} 1 & Y_{d(h,w)} > 200 \\ 0 & otherwise \end{matrix} \quad (3)$$

Y_b is the binary output image, w is the width of the image, and h is the height of the image. If the Y_d value at any pixel is greater than 200, consider the difference as 1, which means there is a motion pixel. Experimentally, we found that the threshold of 200 was satisfactory to delete the noise and detect the real motion. However, to improve the motion detection, we apply the morphological operations of erode and dilation. Using a small kernel of 3 × 3 pixels, we delete the motion pixels that are smaller than the kernel size and dilate the pixels that are larger than the kernel to enlarge the motion location. Consequently, after the motion location is detected, a rectangular window of the same motion contour size captures the image of the motion location and delivers it as the first object template to the VTT.

The ROII may detect some background noise of moving objects, such as swinging curtains. On the other hand, it may detect an incorrect object size because the tracked object might move only a certain part of its body while the remaining part stays in place. To avoid such effects in our system, we recommend a relatively static background for the system. We also apply a limited motion contour size. If the motion contour is smaller or larger than a certain threshold, neglect the motion; otherwise, capture the object template using the motion contour information. By applying these criteria, a large amount of background noise can be neglected because it is smaller or larger than our object's predefined size limits, and the system will produce a relatively good template that bounds the correct object size.

2.2. Visual Target Tracking

The VTT is the second algorithmic block in our system and is responsible for tracking the object of interest visually, for example, by its dimensions, color, edges, light intensity distribution, and so on. The VTT has many challenges that need to be considered:

- The object scale may change according to its distance from the camera or the camera zoom level.

- It should adapt the object deformation. The tracked object deformation might occur within a small period of time; therefore, the system must run fast enough to adapt the new tracked object appearance.
- Light intensity affects the tracked object appearance by making it darker or brighter.
- The object fade and collision with another object must be considered. If another object blocks the view from the tracked object, the system needs to detect the collision occurrence in order to reinitialize the tracking process using the ROII.
- The processing time is critical for the system. The computational complexity of the VTT needs to be simple enough so that system resources are able to process the data in real time. Because the tracked object might move fast, a slow frame rate is not sufficient to update the object location.

A number of visual tracking techniques have been suggested, such as incremental learning for robust visual tracking [13], which runs at 7.5 FPS using 2.8 GHz CPU, and visual tracking with online multiple instance learning [14], which runs at 25 FPS on a Core 2 Quad processor. Both of these algorithms deliver good accuracy, but the FPS is still not enough to build a fast ACTS. The mean shift tracker [15] influences many computer vision specialists because it is fast and accurate with a speed of 66 FPS on a 3.2 GHz processor. However, the correlation tracker has gained positive attention due to its low calculation complexity and good accuracy. In this paper, we focus on the recent development of the correlation tracker. We implement the adaptive correlation tracker MOSSE, which can run at a high speed that is 20 times faster than conventional trackers. An advantage of MOSSE is that it needs one template image of the object of interest to start the tracking therefore our ROII is designed to provide one template to the VTT. However, MOSSE can be improved by using a robust scale estimation to adapt the scale changes of the target object. If the initial object template was captured with the object close to the camera, and then later the tracked object is far from the camera, it will appear smaller than the original template, and vice versa. Therefore, the background will invade the tracking window, and the tracker might track the background instead of the object of interest. Accurate and efficient scale estimation for visual tracking is based on training a classifier on a scale pyramid. It has low overhead, so the system processing speed will not be affected much. Our system runs MOSSE with the accurate scale estimation algorithm to provide good VTT accuracy and robustness.

2.3. Pan-Tilt Controller

The third algorithmic block in our ACTS is the PTC, which tries to maintain the object of interest at the center of the view. The camera is attached to the PTB, so the camera and the PTB are in synchronous motion. The PTB rotation toward the object of interest needs to be smooth and accurate. The conventional methods did not take into consideration the variable rotation speed. The PTB rotates the camera at a fixed speed while tracking the object as in [16]. If the PTB is slower than the tracked object, the system will lag behind the object. If the PTB is faster than the tracked object, the PTB will oscillate around the tracked object. In [17], the authors proposed a variable pan-tilt speed control. They proposed that the speed is proportional to the difference between the camera center and the tracked object location. However, they did not consider the tracked object speed and the PTB current speed. In [4], they implemented an open loop PTC system to rotate the PTB. Their system produces a variable speed according to the object speed, the distance of the tracked object from the camera center, and the PTB current speed. Their system is an open loop system because they used a stepper motor approach. They considered the current PTB speed as the last speed command sent to the PTB. However, in practice, the motor needs a certain amount of time to reach the study state. Their results motivated us to build a closed loop control system to control our PTB. We use a servo motor potentiometer to acquire the PTB speed; consequently, we can get a precise reading of the

current PTB speed and the tracked object speed in terms of degree/second. In our proposed system, we consider the object speed, its distance from the camera center, and the current pan-tilt speed to provide a suitable speed control. The following expressions describe our closed loop speed control.

$$V_{p(n+1)} = V_{p(n)} - O_{sp} + \alpha * C_{pd}(X - X_o)^2 \quad (4)$$

$$V_{t(n+1)} = V_{t(n)} - O_{st} + \alpha * C_{pd}(Y_o - Y)^2 \quad (5)$$

$V_{p(n)}$ and $V_{p(n+1)}$ are the current and the future velocities of the pan respectively in degree/second. $V_{t(n)}$ and $V_{t(n+1)}$ are the current and future velocities of the tilt respectively in degree per second. O_{sp} and O_{st} are the object speed in pan direction and tilt direction respectively in degree per second. α is the speed gain parameter that controls the amount of the speed added relatively when the object distance from the camera center increases or decreases. Its value ranges from 0 to 1. For our system, 0.1 provides a satisfactory speed gain. C_{pd} is the conversion function that converts pixels per degree, and it needs to have a certain conversion ratio for each image resolution or zoom level. X and Y are the center pixels of the image. X_o and Y_o are the center pixels of the tracked object location.

The velocity of the pan is positive as it moves to the left, and the velocity of the tilt is positive as it moves downward. The velocity of the pan and the velocity of the tilt can be calculated using each servo motor potentiometer feedback as follows.

$$V_{p(n)} = \frac{P_{p(n)} - p_{p(n-1)}}{T} \quad (6)$$

$$V_{t(n)} = \frac{P_{t(n)} - p_{t(n-1)}}{T} \quad (7)$$

The potentiometer returns each motor's current position; therefore, $P_{p(n)}$ and $P_{t(n)}$ are the current position of the pan and the tilt, respectively. $p_{p(n-1)}$ and $p_{t(n-1)}$ are the previous pan and tilt positions, respectively. T is the time between two consecutive frames.

However, the PTB might not be able to rotate the camera to the exact center of the object due to the limitations of the PTB motors. The PTB might be stopped a few pixels away from the object center. The step range of a motor can be larger than the distance between two neighboring pixels, so the motor will not be able to stop at the center of each pixel. If the PTC keeps changing the velocity to stop at the object center, it will oscillate around it and will never stop at the required position. Using a motor with a higher resolution will decrease the PTB error but also increase the price of the system. We propose the following expressions to calculate the maximum error that can be produced due to low resolution motors.

$$P_e = \frac{W_p}{\phi} * M_r \quad (8)$$

$$T_e = \frac{H_p}{\Lambda} * M_r \quad (9)$$

P_e and T_e are the maximum errors produced by the pan and tilt motors in pixels, respectively. W_p and H_p are the width and height resolution of the image, respectively. ϕ is the camera width angle of the view in degrees. Λ is the camera height angle of the view in degrees. M_r is the motor step size in degrees. The PTC must have an acceptable

tolerance that accepts a few error pixels to avoid the oscillation and not push the motor beyond its capability.

3. Experimental Results

This section describes the ACTS implementation, pros and cons of the system, and the new improvements compared to the previous systems.

To implement our ACTS, we used OpenCV [18] and DLIB [19] image processing and pattern recognition libraries. We used two servo motors to build the PTB. Each servo motor contains a potentiometer that can provide feedback on the motor's current location. The camera provides the digital frames, and the computer processes all the acquired data to track the object's location. The computer sends the required speed commands and the directions to rotate the camera.

In our ACTS, the object of interest is initialized automatically using motion detection. The ROII detects the object motion and creates a template image for the moving object. However, in some cases, the motion detection may not detect the correct size of the object or may detect a background noise instead. If the motion is not detected correctly, it will produce an incorrect template. Examples of incorrect object detection are shown in figure 3. The green bounding box represents the detected moving object. The second row images in the figure represent the detected motion contour. To decrease the effect by the background noise and avoid detecting the wrong size of the object, the dimensions of the motion contour can be defined with a limitation. For example, if the contour width is < 40 pixels, and the height is < 40 pixels, then neglect the detected motion. By using such criteria, the small moving or swinging objects in the scene can be neglected, and the object's real motion size can be approximated. If there are multiple moving objects at the same time, the ROII considers the largest motion contour as the object of interest. The ROII motion detection is shown in Figure 4.

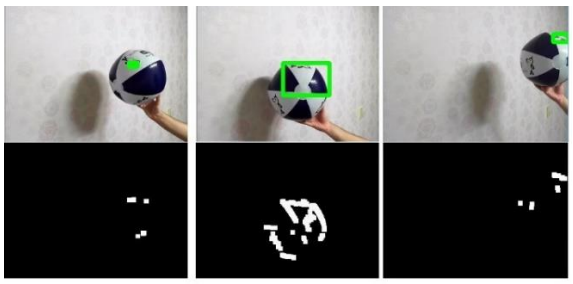


Figure 3. Sample Images of Incorrect Object Size Detection

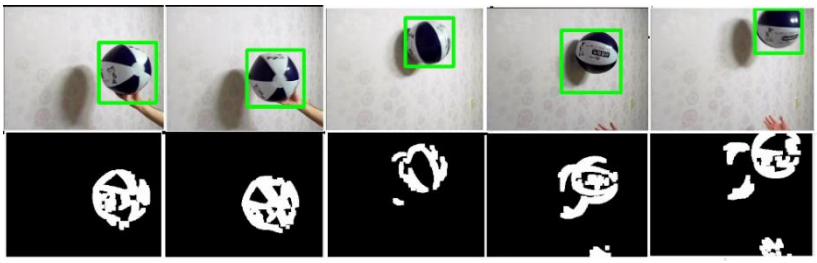


Figure 4. ROII Motion Detection

The VTT of our system can track a fast object due to the high FPS processing speed. The VTT is so reliable that it can keep tracking the object with minimal drifting for long periods. Figure 5 shows sample frames with the input to the correlation tracker, the

correlation output, and the calculated correlation peak to side lobe ration (PSR) value. At the first two frames, the object is detected, and the correlation output shows a strong white spot at the center of the tracked object. A high PSR value indicates that the object is successfully detected.

In the third frame, the object has collided with another object that blocks the vision. At the collision instant, the correlation output image is not a strong white spot at the center of the tracked object, and the PSR drops below 7, indicating that the object is not within the view. In such a case, the ROII needs to work again to initialize the tracked object. However, if an object blocked the vision suddenly, the PSR will drop below 7 very fast. A critical case can occur if an object blocks the vision of the tracked object slowly. The PSR might not drop below 7 because the tracker may adapt part of the collided object as the new object, causing the PSR value not to drop significantly. Eventually, the system will track the collided object instead of the object of interest. Although such a problem is out of the scope of this paper, it might require more discussion in the future.

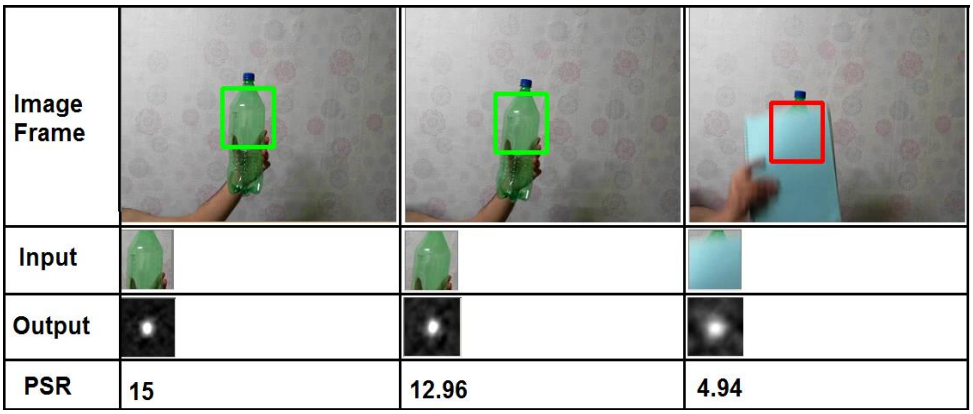


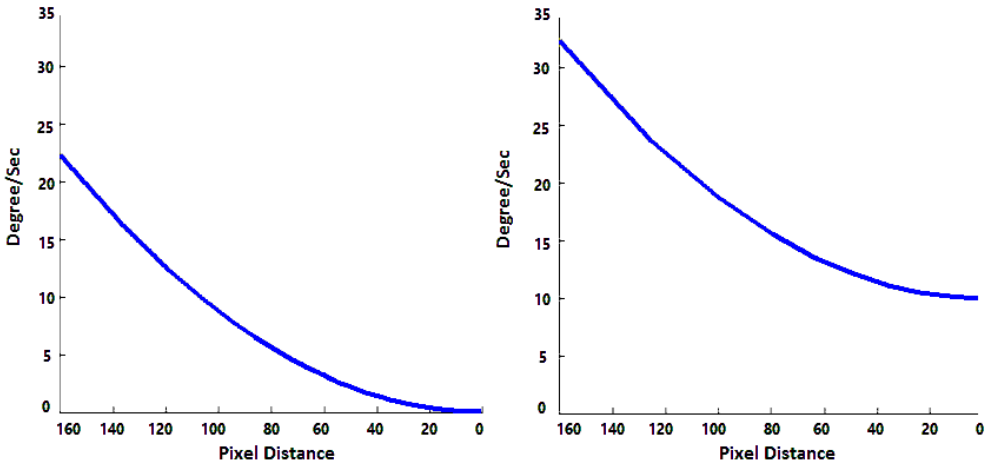
Figure 5. Object Tracking and Collision Detection

The object tracking and scale detection are shown in Figure 6. To verify the accuracy of the implemented scale estimation algorithm, a test scenario of a tracked object that is located at different distances from the camera is performed. As the object moves closer and farther from the camera, its scale changes. The VTT detects the current object scale and surrounds it with a green rectangle. The system shows high accuracy results for the algorithm and low processing requirements.

Figure 6. Scale Changes Estimation

The PTC maintains the tracked object within the center of the camera. To test the speed changes using the proposed closed loop controller, we simulated equations (4) and (5) and also recorded the results. The speed generated by the pan for a fixed and a moving

object is shown in Figure 7. We placed a static object to the end left of the camera and recorded the speed generated by the PTC as shown in (A) of Figure 7. We repeated the test for a moving object at 10 degrees/second and recorded the generated speed as shown in (B) of Figure 7. The parabolic curve shape is due to the squared value of the object distance from the camera center. The squared value provides a fast increase of the generated speed by the PTC as the object distance from the camera center increases. The curves show that the speed varies according to the object speed, pan-tilt speed, and the distance from the camera's center.



(A) **(B)**

Figure 7. The PTC Speed Generation

Practically, the motor might not be able to stop at the center pixel of the tracked object due to its step size. However, equations (4) and (5) will try to rotate the motor to the tracked object's exact center, but the motor might not able to stop at the destination point and might pass over it. Thereafter, it will generate an opposite speed to direct the motor again backwards and keep oscillating around the center. The oscillation effect for the three different frame resolutions is shown in Figure 8. In Figure 8 (A), the resolution was 320 × 240; in (B), the resolution was 640 × 340; and in (C), the resolution was 1280 × 720. As the image resolution increases or as the zoom level increases, the pixel errors also increase. Equations (8) and (9) describe the maximum error and its relation to the image resolution. The red bounding lines in the figure represent the maximum possible error by the PTC. However, two extra pixel errors might result from the VTT, therefore, the error exceeded the bounding lines by two pixels. In Figure 8, only the pan axis error is shown because the tilt axis has the same effect.

To avoid such oscillation, the PTC needs to know the motor's step size limitation to provide the system with an acceptable error tolerance. In our system, the maximum error of the pan direction was 4.4 pixels for an image width of 320 pixel and 1 degree stepper motor resolution while the camera width angle of the view was 72 degree. We also supply the PTC with a criterion that if the error is less than 4.4 pixels, do not rotate the PTB. The oscillation around the object center no longer existed.

Finally in Figure 9, sample images of the system tracking are shown. After initializing the target object, the VTT sends the tracked object coordinates to the PTC. The PTC then sends the destination and speed commands to the PTB. In the figure, the tracked object is always around the center of the camera.

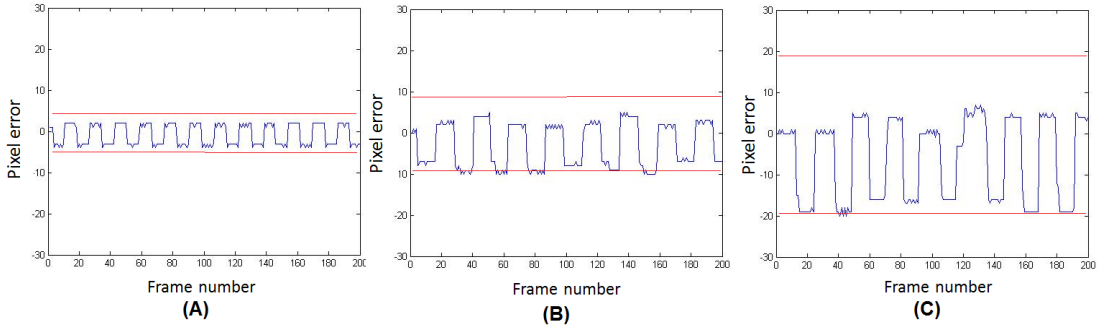


Figure 8. Oscillation around the Tracked Object Due to the Motor Limitation. The Red Line shows the Maximum Error Boundaries

Figure 9. The ACTS Tracks the Ball and Maintains it within the Camera Center while Moving in the View

4. Conclusion

In this paper, we presented an ACTS that is suitable for indoor monitoring and tracking such as pets or other moving objects of interest. The proposed ACTS is comprised of three blocks: ROII, VTT, and PTC. The ROII is responsible for the tracked object initialization; the VTT is responsible for the visual tracking of the object, and the PTC rotates the camera smoothly and accurately according to the acquired object's location. In this paper, we implemented the recent image processing technology for tracking an object, and then we implemented an accurate scale estimation technique to detect the tracked object's size. The system can track an object with minimum drifting, and it is robust enough to overcome intensity changes, object deformation, scale changes, object fade, and collision. The presented closed loop PTC provides a smooth and accurate rotation toward the tracked object. We tested each block and presented the results. According to the obtained results, we can say that the system is very suitable for indoor monitoring. The system can process 140 FPS for a 320 × 240 frame resolution. The high FPS enables the system to track fast objects. The low processing complexity for the VTT makes it possible to rebuild the ACTS using inexpensive hardware and processors. However, in this work, we assumed that the background is static or contains minor motion during the template initialization. As a future work, the ROII can be improved by enabling it to initialize a template in the presence of a noisy background. By improving the ROII, the system will be able to run in noisy and complex environments.

Acknowledgments

This work was supported by the 2016 Advanced Research Center Fund (MPEES-ARC) of Myongji University.

References

[1] Murray, Don, and Anup Basu., "Motion tracking with an active camera", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 16, no. 3, **(1994)**, pp. 449-459.
[2] Montera, Dennis A., Steven K. Rogers, Dennis W. Ruck, and Mark E. Oxley, "Object tracking through adaptive correlation", Optical engineering, vol. 33, no. 2, **(1994)**, pp. 294-302.
[3] Lewis, J. P., "Fast normalized cross-correlation", Vision interface. vol. 10, no. 1, **(1995), pp. 120-123**.
[4] Ahmed, Javed, Ahmad Ali, and Asifullah Khan, "Stabilized active camera tracking system", Journal of Real-Time Image Processing, vol. 11, no. 2, **(2016)**, pp. 315-334.
[5] Singla, Nishu, "Motion Detection Based on Frame Difference Method", International Journal of Information & Computation Technology, vol 4, no. 15, **(2014)**, pp. 1559-1565.
[6] Bolme, David S., J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui., "Visual object tracking using adaptive correlation filters", Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE, **(2010)**, pp. 2544-2550.
[7] Danelljan, Martin, Gustav Häger, Fahad Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking", British Machine Vision Conference, Nottingham, BMVA Press, **(2014)** September 1-5.
[8] Ahmed, Javed, and M. Noman Jafri, "Best-match rectangle adjustment algorithm for persistent and precise correlation tracking", Machine Vision, ICMV 2007, International Conference on. IEEE, **(2007)**, pp 91-96.
[9] Massimo Piccardi, "Background subtraction techniques: a review", IEEE International Conference on Systems, vol. 4, **(2004)**, pp. 3099 – 3104.
[10] Yu-Fei Ma, Hong-Jiang Zhang, "Detecting motion object by Spatio-Temporal Entropy," IEEE International Conference on Multimedia and Expo ISBN, **(2001)** August 22.
[11] Wei Shuigen, Chen Zhen, Dong Hua, "Motion detection based on temporal Difference method and optical flow field", Second International Symposium on Electronic Commerce and Security, vol. 2, no. , **(2009)**, pp. 85 – 88.
[12] Hamilton, Eric, "JPEG file interchange format," C-Cube Microsystems, **(1992)** September.
[13] Ross, David A., Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," International Journal of Computer Vision, vol. 77, no. 1, **(2008)**, pp. 125-141.
[14] Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie, "Visual tracking with online multiple instance learning", Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on. IEEE, **(2009),** pp. 983-990.
[15] Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer, "Kernel-based object tracking", IEEE Transactions on pattern analysis and machine intelligence, vol. 25, no. 5, **(2003)**, pp. 564-577.
[16] Yahya Imad Mohammed, and Jong Myung Rhee, "A New Visual Pet Activities Monitoring System Design," Advances in Computer Science and Ubiquitous Computing. Springer Singapore, **(2015)**, pp. 445-450.
[17] Mir-Nasiri, Nazim, "Camera-based 3D Tracking System," TENCON 2005-2005 IEEE Region 10 Conference. IEEE, **(2005)** November 21, pp. 1-6.
[18] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. "O'Reilly Media, Inc.", **(2008)** September 24.
[19] King, Davis E., "Dlib-ml: A machine learning toolkit," The Journal of Machine Learning Research 10, **(2009)** 10 July, pp. 1755-1758.

Authors

Yahya Imad Mohammed, received his B.Sc. degree in Computer Engineering from University of Baghdad, Iraq, in 2013. After graduation, he joined a private company as an information technology (IT) specialist to manage environment monitoring stations installation and networking in cooperation with the Iraqi ministry of environment for one year. Later, he worked as an IT specialist and a

team leader for Global Cleaners private company. Currently he is a master degree student at Myungji University in South Korea.

Jong Myung Rhee, received his PhD from North Carolina State University, USA, in 1987. After 20 years at the Agency for Defense Development in Korea, where he made noteworthy contributions to C4I and military satellite communications, he joined DACOM and Hanaro Telecom in 1997 and 1999, respectively. At Hanaro Telecom, which was the second largest local carrier in Korea, he served as Chief Technology Officer (CTO) with a senior executive vice-president position. His main duty at Hanaro Telecom was a combination of management and new technology development for high-speed Internet, VoIP, and IPTV. In 2006, he joined Myongji University, and currently, he is the Vice President of Myongji University as well as a full professor in the Information and Communications Engineering Department. His recent research interests are centered on military communications and smart grids, including ad-hoc and fault-tolerant networks. He is a member of IEEE and IEICE.