

Web-Based Object Tracking Using Collaborated Camera Network

Abul K. M. Azad^{1*}, Mohammed Misbahuddin²

¹Department of Technology, Northern Illinois University, DeKalb, IL, USA

²Department of Electrical Engineering (Graduate Student), Northern Illinois University, DeKalb, IL, USA

Email: *aazad@niu.edu, misbah4064@gmail.com

How to cite this paper: Azad, A.K.M. and Misbahuddin, M. (2018) Web-Based Object Tracking Using Collaborated Camera Network. *Advances in Internet of Things*, 8, 13-25.

<https://doi.org/10.4236/ait.2018.82002>

Received: December 30, 2017

Accepted: April 24, 2018

Published: April 27, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The paper presents a web based vision system using a networked IP camera for tracking objects of interest. Three critical issues are addressed in this paper. First is the detection of moving objects in the foreground; second is the control of pan-tilt-zoom (PTZ) IP cameras based on object location; and third is the collaboration of multiple cameras over the network to track objects of interests independently. The developed system utilized a network of PTZ cameras along with a number of software tools for this implementation. The system was able to track a single and multiple objects successfully. The difficulties in the detection of moving objects are also analyzed while multiple cameras are collaborating over a network utilizing PTZ cameras.

Keywords

Internet Protocol Camera, Networked Camera, Vision Tracking, Tracking Handover, Image Processing, Collaborated Tracking

1. Introduction

An Internet protocol (IP) network camera allows remote viewing and monitoring from anywhere and anytime over the Internet. This is a very promising technology in the field of video surveillance. With the pan-tilt-zoom (PTZ) feature, an IP camera allows surveillance of large areas thereby reducing the number of fixed cameras required to cover a given area. An IP camera responds to commands through its integrated web server [1]. In most of the cases, the PTZ-IP cameras are either manually operated or programmed under timed operation, even when multiple cameras are deployed within an area. This makes it difficult to utilize them to their highest potential.

The issues for using PTZ-IP cameras for tracking are well summarized in ref-

erence [1]. The author highlighted the issue of improper response time within the camera's web server, very low frame rates, and difficulty in image processing due to constantly changing backgrounds. A number of studies have been conducted in the area of tracking multiple objects using networked cameras. A system was designed that is able to discover spatial relationships among the camera's fields of view and use this information to correspond to different perspective views of the same object [2]. In another effort, researchers tracked multiple moving objects using a single camera [3]. They used image segmentation and then identified humans in the image by computing the size of the blobs obtained. This is efficient only if the moving object is within the field of vision of the camera. However, using a PTZ camera for tracking multiple objects creates the challenge of locking on a single object of interest. Horesh and his co-authors reported a system where they use trajectories of objects moving around to predict and then detect the object's next location [4]. They divided the ground surface into a large grid and used network flow to track the trajectories of multiple moving objects for differentiation. In one effort, researchers showed how they are tracking people using skin color segmentation [5]. They locked onto their targets using PTZ cameras and tracked their motion. However, as the system depended on skin color detection, their system was viable for tracking only two individuals. Chong and his co-workers tracked objects utilizing multiple PTZ cameras in collaboration [6]. In this work, each camera module had its own detection and tracking mechanism; however, the processing was done on a central network. This allowed the cameras to communicate among each other and allowed tracking of multiple objects. They use Kalman Consensus filters to address the issue of multiple object tracking. But it could not prioritize among the multiple targets. One effort reports a technique in which they were able to track objects across multiple cameras without overlapping views [7]. They utilized appearance model and space-time cues like velocity, location, and time to identify the moving target across multiple cameras. However, their system did not address the issue of using a PTZ camera and presence of multiple objects in the frame.

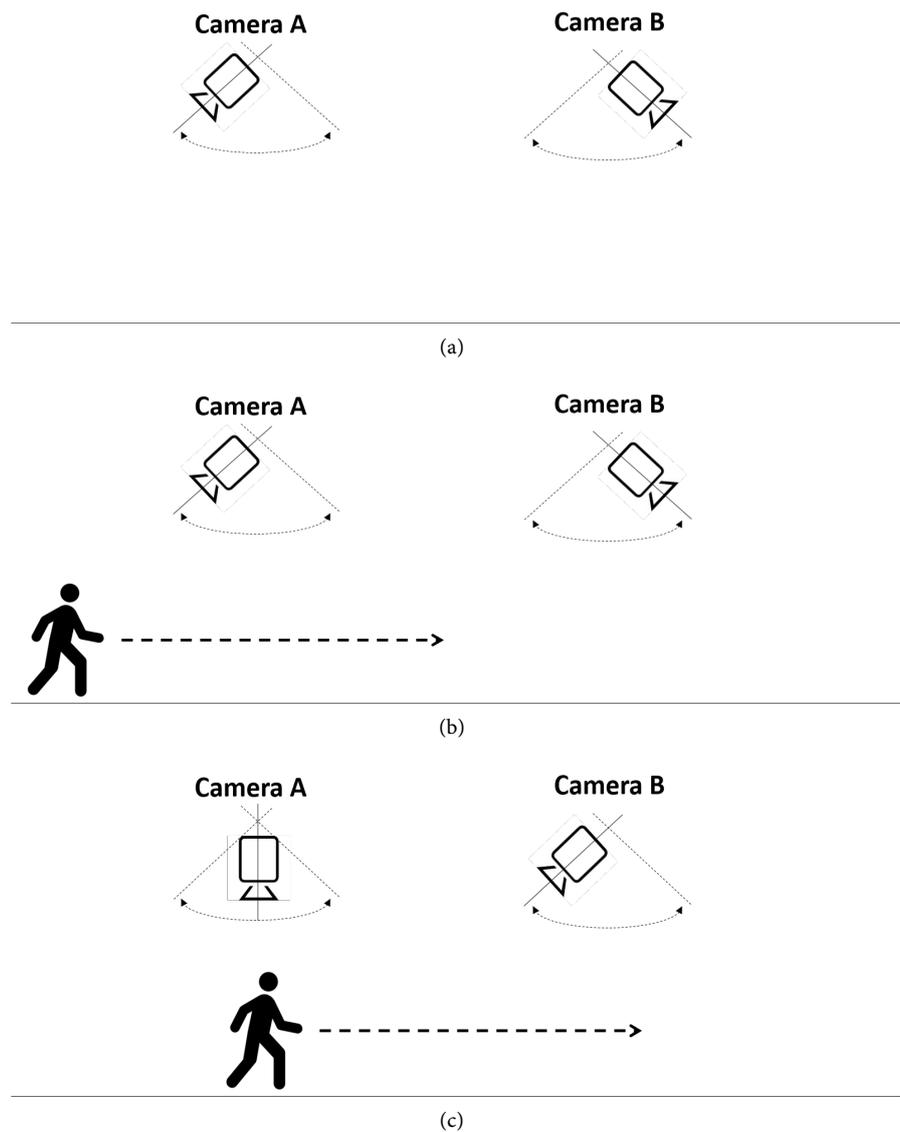
This paper describes a PTZ-IP surveillance system that overcomes some of these challenges. With the proposed technique, PTZ-IP cameras were programmed to detect and track moving objects and at the same time collaborate with other PTZ-IP cameras in the network so that the tracking can be done effectively and efficiently. In addition, the technique also addresses the challenge of identifying objects of interest that need to be tracked. The feature identifies the target of interest among a group of moving objects. Priority can be set based on identity, direction or motion, and time of a day. A centralized approach is proposed in which multiple PTZ-IP cameras are tracked an object of interest across the network. This paves a way for an advanced video surveillance in which an object of interest can be detected and tracked over a network of multiple PTZ-IP cameras. A feature described as "handing-off" tracking to the next PTZ-IP camera has also been introduced in which once a moving object is de-

tected in the field of next camera in the network, the current camera hands off the tracking of that object.

2. System Design Strategy

Within the designed system, once the object is identified, the camera control algorithm changes the pan and tilt to track and follow the movement of the object. This process is running as a feedback loop that controls the speed of motion of the camera based on the speed of the moving object. When an object is detected within the Field of View (FOV) of Camera A, then Camera B pans itself towards the FOV of Camera A. The intention is to follow the moving object when it leaves the FOV of Camera A and enters the FOV of Camera B. In this way, cameras within the network try to track the object as long as the object is within the networked area. The tracking process is illustrated in **Figure 1**.

The system implementation involves a number of activities that includes control



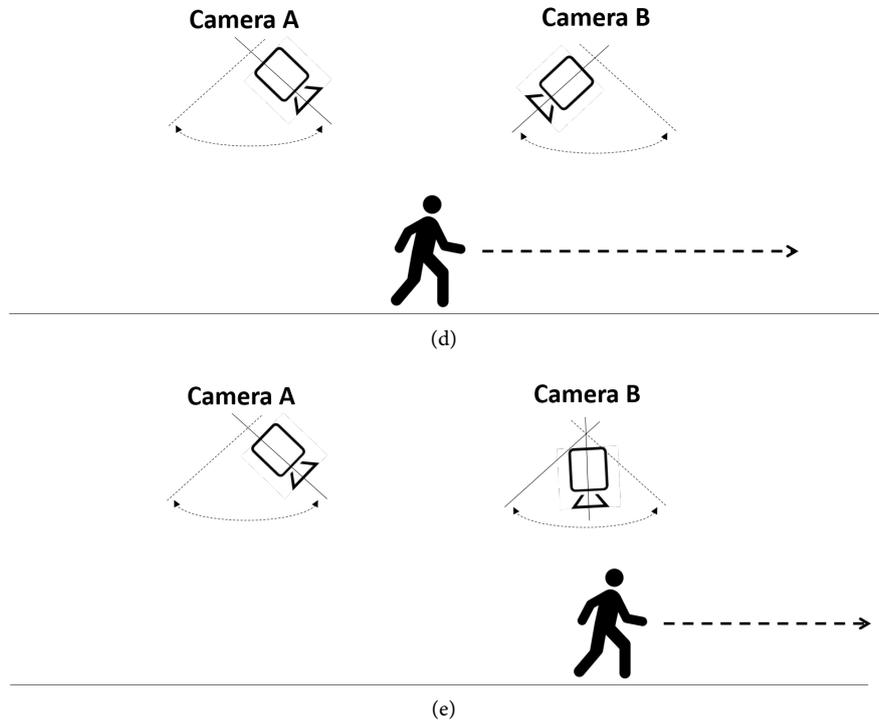


Figure 1. Collaborated detection and tracking between network cameras A and B. (a) Camera A and B in idle mode searching for moving objects; (b) Camera A detects a moving object; (c) Camera B moves towards camera A's FOV and detects the moving object; (d) Hand-off of tracking from camera A to camera B. (e) Camera B starts tracking a moving object and Camera A focuses onto the FOV of camera B.

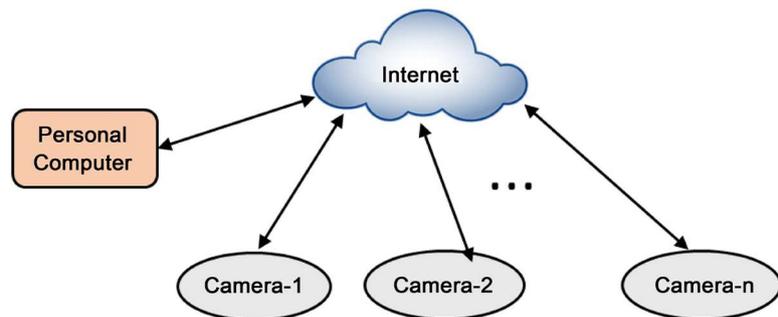


Figure 2. System level block diagram.

of PTZ-IP cameras, moving object detection and tracking, tracking management, collaborated tracking, and prioritizing multiple object tracking. A system block diagram is shown in **Figure 2**. The identification of moving objects involves motion detection, background subtraction, region sampling, morphological filtering and blob detection. Two PTZ-IP Foscam cameras were used for the developed system. The same algorithm can be adopted and integrated for any number of PTZ-IP cameras.

3. Implementation

The implementation of the designed system involves a series of activities, such as

image gathering, moving object detection, tracking, and handing over the object tracking to the next camera within the network. **Figure 3** shows a flowchart of the tracking activities for a given camera. The section describes the details of each activity.

3.1. Control of PTZ-IP Cameras

Information about the installation and use of a Foscam PTZ-IP camera can be found from the manufacturer [8]. It automatically assigns itself a local IP that can be accessed using a network browser. Logging into the camera system using the provided IP allows access to the manufacturer-developed graphical user interface (GUI) through its web server. Controlling the PTZ-IP camera through the GUI is very simple and user-friendly; however, the challenge is to automate the camera control through the newly developed image processing system.

The pan and tilt of the camera were controlled by an image processing unit developed using C++ (on Visual Studio). The program ensured that the tracking object was at the center of the FOV of the camera. **Figure 4** shows how the pan and tilt of the camera were adjusted to make the center of the FOV coincide with the object's position. **Figure 4(a)** shows the initial position of the object and the camera's FOV where the object is at one corner of the camera's FOV. **Figure 4(b)** shows how the camera has automatically panned toward its left to make the object's location coincide with the center of the camera's FOV. Finally, in **Figure 4(c)** the camera has successfully matched its center with the object's location.



Figure 3. Object tracking flowchart.

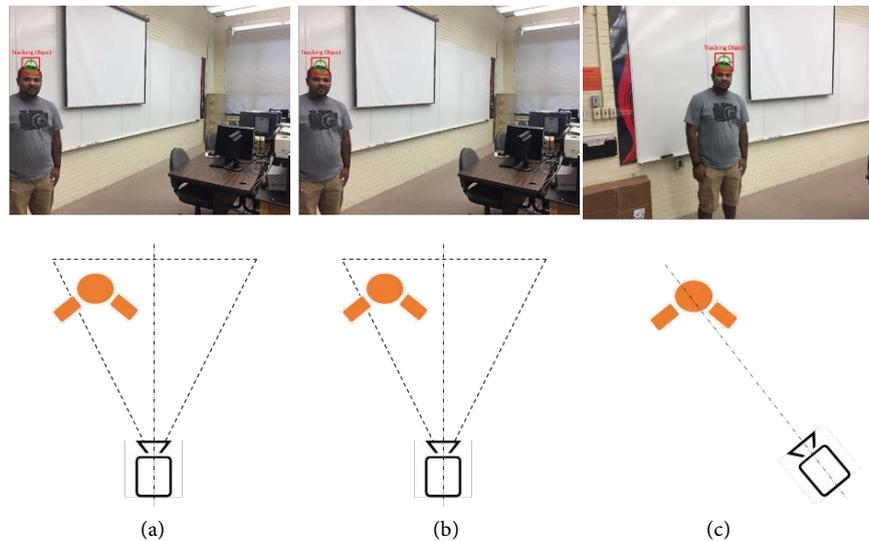


Figure 4. Auto-adjustment of pan and tilt. (a) Object is away from the center of the camera's FOV; (b) Camera turned left in an attempt to bring object to its FOV's center; (c) Camera successfully aligns its FOV's center with object's position.

3.2. Moving Object Detection

Background modeling is utilized for detection and tracking of a moving object. Research has presented different approaches for background modeling by background detection, subtraction, and motion cues [9] [10] [11]. For the proposed system we subtracted consecutive frames from the camera to identify moving objects (Figure 5). Subtraction of two images is performed straightforward in a single pass. The difference of image $P_1(i, j)$ and image $P_2(i, j)$ is given by Equation (1) where $Q(i, j)$ is the output image [12].

$$Q(i, j) = P_1(i, j) - P_2(i, j) \quad (1)$$

The subtracted image f was sent through a morphological filter to remove noise and unwanted motion cues. Morphological filtering involved creating a threshold output (Figure 6) followed by erosion and dilation process with an 8×8 filter (Figure 7) [13].

The erosion of an image $f(x)$ by a structural window s is denoted by $f \ominus s$,

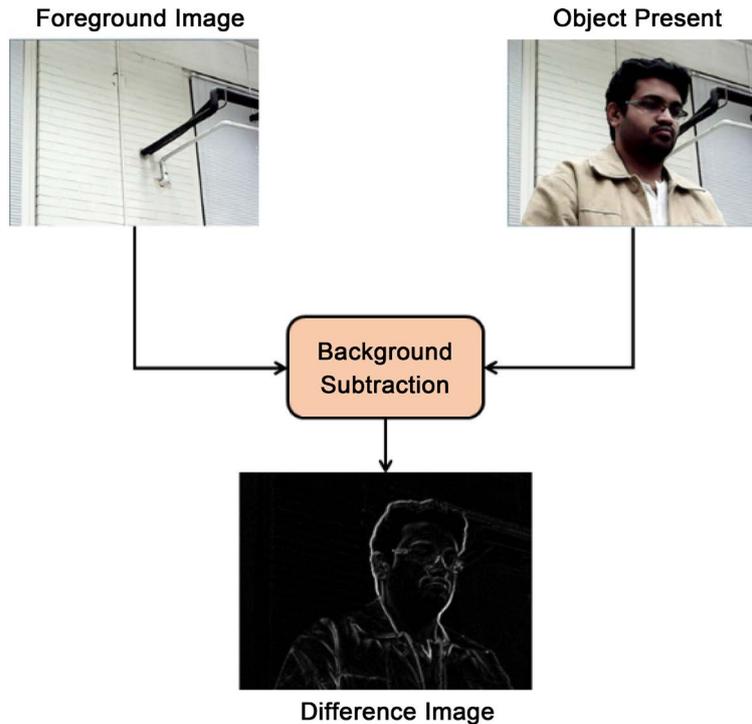


Figure 5. Background subtraction.



Figure 6. Threshold output.



Figure 7. Filtered output (after Erosion and Dilation).



Figure 8. Moving object detected through contour detection.

where θ denotes the erosion between f and s . Resultant image $g(x)$ can be presented by Equation (2) [14].

$$g = f \theta s \quad (2)$$

The dilation operation is performed to add pixels to the boundaries of the objects present in the image. The number of pixels added to the objects depends on the size and shape of the structuring element defined to process the image. The dilation of the image $g(x)$ by a structural element s results in the image $h(x)$. The relationship is provided in Equation (3) [14].

$$h = g \oplus s \quad (3)$$

The Eroded and Dilated images are sent for contour detection. The largest contour detected was given the identity of the moving object. **Figure 8** shows the moving object detected as a result of the contour detection process. In order to get more understanding on contour detection, the reader can read papers by Pablo *et al.* [15] and Richard S. [16].

3.3. Tracking Management

Once the moving object is detected, the PTZ-IP camera has to align its center FOV to match the object's location. This is an autonomous process where the camera has to continuously track the moving object in the scene. The process was implemented through passing commands to the decoder unit of the camera web server.

Figure 9 shows the steps involved in the pan and tilt control of the cameras. Depending on the initial location of the detected object, the camera has to either

pan left or right and also tilt up or down. The requirement commands are sent to the decoder unit to initiate the needed pan or tilt motion. Once initiated, a delay unit introduces an appropriate amount of delay depending on the speed of the moving object. After the required movement of the camera (in terms of pan and tilt) a stop command is sent to the decoder to stop the motion.

3.4. Collaborated Tracking

This section refers a system in which the autonomous moving object tracking by an IP-PTZ camera is extended over a network, in which the cameras communicate with each other and provide effective tracking of the moving object in a wide area. **Figure 1** in section 2 illustrates how the cameras follow a moving object from one point to another. For example, Camera A detects the moving object and starts tracking, in the meantime Camera B moves its FOV to align with the FOV of Camera A in such a way that the moving object is also detected by Camera B. Once the object moves away from the FOV of Camera A, Camera B jumps in and takes control of tracking the object. **Figure 10** shows how the two cameras coordinate in detecting the object of interest.

3.5. Multiple Object Tracking

One of the challenges faced in this technique is Multiple Objects Tracking (MOT). Various studies done on MOT show different approaches for attempting static camera images [17] [18] [19]. However, the challenges in MOT are more toward a PTZ control of the camera. To analyze and detect the high priority targets, judgments can be made based on several parameters: direction of

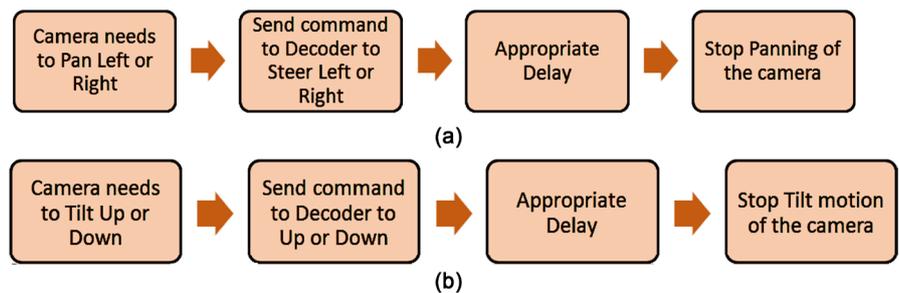


Figure 9. Instructions flow chart of camera control. (a) Steps for pan control; (b) Steps for tilt control.



Figure 10. Camera 1 and Camera 2 focusing on the same object.

motion of the object, date and time, identity of the object under surveillance, and skeptical objects (Figure 11).

Direction of motion of the object provides an indication of whether the object is moving into the building or outside the building. Based on the priority, either objects entering the premises or leaving the premises can be tracked and selected for the PTZ camera tracking system. If the need to track objects entering the premises is more important, then PTZ camera tracking can be triggered for those objects that are moving towards the premises. This is done by determining optical flow, in which the relative position of the object is tracked over a period of time and the direction of motion is computed. Berthold and Brian provide a detailed methodology of how optical flow can be administered [16] [20], where they demonstrated that a moving object in a video can be detected by tracking the brightness of the object over the two-dimensional plane of the image. They are based on the fact that when an object moves, its brightness remains constant irrespective of its position in the image frame. By tracking the brightness over the plane of the image can give the direction and position of the object. By denoting image brightness at a point (x, y) in an image at time t , by $E(x, y, t)$. When the object moves we can derive the following equation, considering that the brightness of a particular point in the pattern is constant:

$$Exu + Eyu + Et = 0 \quad (4)$$

where Ex , Ey , Et is the change in brightness along x-plane, y-plane, and time t . This can be further computed to obtain the direction of the movement.

Date and time of the process also mark an important event. If it is a weekend or a holiday, then the PTZ cameras might be more useful in tracking objects by a conventional way of regular pan and tilt triggered uniformly. That way any kind of multiple object movement on a holiday or weekend is considered a high surveillance zone. Accordingly, depending on the time of day, the PTZ camera control can be revised based on the requirements. In other words, the PTZ camera surveillance can be activated only on weekends, holidays, and other special occasions.

Another approach to sort objects is to use identity tracking, which is similar to face recognition. As an example, a group of people regularly pass through a camera system and their faces are recorded within the system. If an unknown

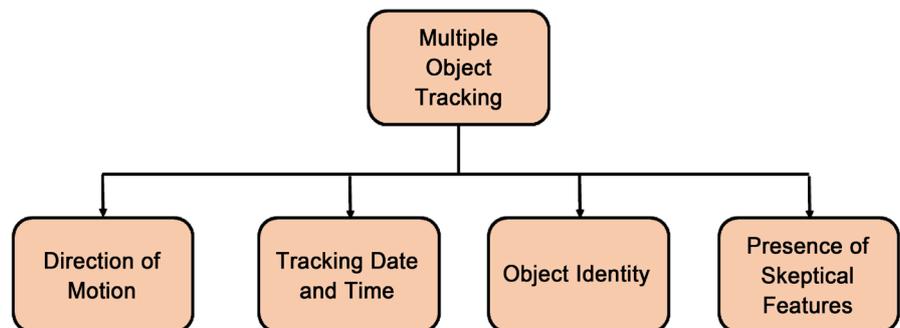


Figure 11. Multiple object tracking protocol.

person passes the area, then the tracking system will be activated and start tracking the face. In this way camera will only track an unknown person and will not act on other passersby. This can be implemented by creating Haar Classifiers to train the system to recognize concerned objects. Szeliski explains the various object detection systems in detail [16]. Haar-like feature based detection is a machine learning approach where a cascade function is trained from a lot of positive and negative images. This cascade function is then used to detect objects in an image.

The system can be designed to trigger the tracking system with the presence of a skeptical feature—consider a criminal retention facility, where the inmates wear orange color jump suits. Presence of an individual wearing an orange colored jumpsuit anywhere near the exit door can be considered a skeptical feature and will activate the tracking system. Similarly, other features of interest can be defined and coded into the tracking system. By converting a RGB image into a HSV image and by using H (Hue), S (Saturation), and V (Variance) dimension to easily detect relevant colors in an image. Hue denotes the color in the image, Saturation denotes the dominance of that color and Variance denotes the brightness of the color. This way a particular color can be isolated easily by using the relevant Hue values. Once the color has been isolated in an image, morphological operations like erosion and dilation can be performed on the image which cleans out any noise (explained in section 3-2). This is then passed onto a blob detection process which helps in determining the location of the color in the image.

4. Software Integration

To implement the developed system a number of software were utilized, including Matlab, Visual Studio, OpenCV, and a web browser. **Figure 12** shows a flowchart demonstrating the interaction among the software packages. MATLAB is used to capture IP camera video stream directly and convert it into images using FFmpeg, where FFmpeg is a freeware package used to access multimedia digital files (like videos). The images extracted by MATLAB are then used by Visual Studio (with OpenCV installed on the system) to perform image

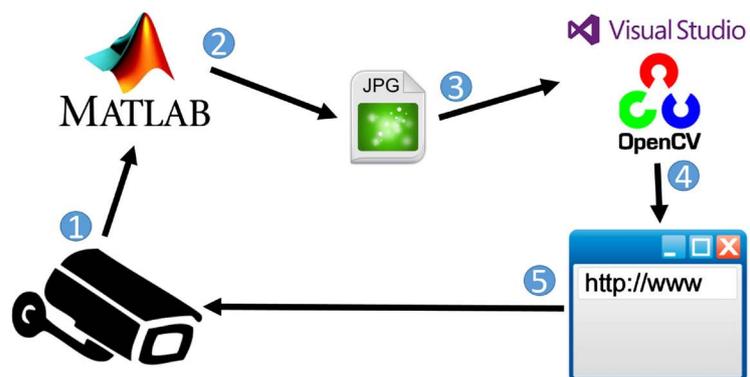


Figure 12. Software interaction for object tracking.

manipulation exercises to obtain the tracking direction.

The steps involved in this process start with Step 1 in which the IP camera is connected to the network streams live video data on its secured password controlled web server. This account information is supplied to the MATLAB via program codes, which allows MATLAB to access the camera feed directly.

```
imshow(I) >> I=getimage(strmatch(fname,['web
http://www.IPCameraWebServer/username/password.jpg']))
```

Once the image files are created, MATLAB saves them in a common folder that is also shared with Visual Studio. These image files are then pulled by Visual Studio and image processing is performed using OpenCV functions. The direction of the motion of the target is accessed using OpenCV, and accordingly, the pan and tilt motion of the camera is carried out. As an example, if the target moves toward the right, the camera needs to follow accordingly and pan toward the right. This information is relayed to the PTZ IP camera using a web browser. The web browser is used, as it directly allows access to the IP camera's server.

```
system("open
http://www.IPCamera/username/password/PTZCodeForMovement");
```

The code was initially run on a webcam to test the functionality of the image processing system. Once detection ran successfully, the IP camera feed was given to the image processing code. Based on the direction of movement, corresponding decoder control code was executed on the IP camera to adjust the pan and tilt motion so that it could track the moving object. The same camera control code was also run on another IP camera through a bridge control between the two IP cameras, which allowed communication between the two networked cameras, provided both are run on the same network. Once the object went out of the field of vision of one camera, then the other camera's pan and tilt were changed to align with the field of vision of the previous camera. This allowed collaborative tracking between networked IP cameras.

5. Conclusion

The article describes the development of a surveillance system utilizing multiple PTZ-IP cameras. This addresses some of the shortcomings reported by other researchers. A feature described as "handing-off" tracking to the next PTZ-IP camera has been introduced in which once a moving object is detected in the field of the next camera in the network, the current camera hands off the tracking of that object. The system can be implemented with any number of networked cameras allowed by software and hardware capacity while they collaborate to track an object within the area of coverage with those cameras. The system has the capacity to identify a specific object of interest among a group of objects and track this accordingly. The criterion for object of interest is pro-

grammable and can be tied to an alarm to alert authorities. A number of software tools were utilized to implement the system.

As FOSCAM server side is a closed application and access to the camera's image feed is not open. The system relied on retrieving the feed from the snapshots on a web-browser. This involved using MATLAB to retrieve the image and save it locally, which can then be picked up by OpenCV and processed. However, this process involved redundant actions, excessive use of software platforms resulting in time delays. This can be improved by using open cameras allowing the system to retrieve images directly. The future development may involve training the system to detect special features and track them accordingly. The system can be designed to track any anomalous behavior in the video feed and flag it for further investigation.

Acknowledgements

The authors would like to thank the NSF for its support for the reported work (NSF TUES project, award number DUE-1140502).

References

- [1] Parisa, D.Z.V. and Bilodeau, G.-A. (2011) People Tracking Using a Network-Based PTZ Camera. *Machine Vision Applications*, **22**, 671-690. <https://doi.org/10.1007/s00138-010-0300-1>
- [2] Sohaib, K., Omar, J., Zeeshan, R. and Mubarak, S. (2001) Human Tracking in Multiple Cameras. *IEEE International Conference on Computer Vision*, 7-14 July 2001, Vancouver, 331-336.
- [3] Cai, Q. and Aggarval, K. (1996) Tracking Human Motion Using Multiple Cameras. *Proceeding of the 13th International Conference on Pattern Recognition*, Vienna, 25-29 August 1996, 68-72.
- [4] Shitrit, H.B., Berclaz, J., Fleuret, F. and Fua, P. (2013) Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**, 1614-1627. <https://doi.org/10.1109/TPAMI.2013.210>
- [5] Stillman, S., Tanawongsuwan, R. and Essa, I. (1998) A System for Tracking and Recognizing Multiple People with Multiple Cameras, Georgia Institute of Technology. *Graphics, Visualization and Usability Center*, Technical Report # GIT-GVU-98-25, August.
- [6] Ding, C., Song, B., Morye, A., Farrell, J.A. and Roy-Chowdhury, A.K. (2012) Collaborative Sensing in a Distributed PTZ Camera Network. *IEEE Transactions on Image Processing*, **21**, 3282-3295. <https://doi.org/10.1109/TIP.2012.2188806>
- [7] Javed, O., Rasheed, Z., Shafique, K. and Shah, M. (2003) Tracking across Multiple Cameras with Disjoint Views. *Proceedings of the 9th IEEE International Conference on Computer Vision*, 13-16 October 2003, Nice, 950-957. <https://doi.org/10.1109/ICCV.2003.1238451>
- [8] Foscam, IP Camera CGI V1.27 (2017) http://www.foscam.es/descarga/ipcam_cgi_sdk.pdf
- [9] Lalondea, M., Fouchera, S., Gagnon, L., Pronovost, E., Derenne, M. and Janelle, A. (2007) A System to Automatically Track Humans and Vehicles with a PTZ Camera. *Proc. SPIE 6575, Visual Information Processing XVI*, 30 April 2007, Orlando, FL.

- <https://doi.org/10.1117/12.718446>
- [10] Kang, S., Paik, J., Koschan, A., Abidi, B. and Abidi, M.A. (2003) Real-Time Video Tracking Using PTZ Cameras. *Proceedings of SPIE 6th International Conference on Quality Control by Artificial Vision*, **5132**, 103-111.
<https://doi.org/10.1117/12.514945>
- [11] Kaur, N. (2012) Real Time Automatic Object Tracking by Pan-Tilt-Zoom Cameras in an IP-Surveillance System. *International Journal of Computational Engineering Research*, **2**, 63-69.
- [12] Pixel Subtraction (2017) <https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm>
- [13] Efford, N. (2000) Digital Image Processing—A Practical Introduction Using Java. Pearson Education Limited, Pearson.
- [14] OpenCV (2017) Eroding and Dilating.
https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html
- [15] Arbelaez, P., Maire, M., Fowlkes, C. and Malik, J. (2011) Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 898-916.
- [16] Szeliski, R. (2010) Computer Vision: Algorithms and Applications. Springer-Verlag, London.
- [17] Xiang, Y., Alahi, A. and Savarese, S. (2015) Learning to Track: Online Multi-Object Tracking by Decision Making. *International Conference on Computer Vision*, Chile, 7-13 December 2015, 1-9. <https://doi.org/10.1109/ICCV.2015.534>
- [18] Possegger, H., Mauthner, T., Roth, O.M. and Bischof, H. (2014) Occlusion Geodesics for Online Multi-Object Tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 1306-1313.
- [19] Chari, V., Lacoste-Julien, S., Laptev, I. and Sivic, J. (2015) On Pairwise Costs for Network Flow Multi-Object Tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 5537-5545.
- [20] Horn, B.K.P. and Schunck, B.G. (1981) Determining Optical Flow. *Artificial Intelligence*, **17**, 185-203.