Real-time Smart Situation Awareness based on Camera Array and Machine Learning

Fenghui Yao^{a,*}, Guifeng Shao^a

^aTennessee State University affiliation, 3500 John A. Merritt Blvd, Nashville, TN 37215, USA

*Corresponding Author: fyao@tnstate.edu

Abstract

This paper proposes a real-time situation awareness system based on a surveillance camera array and machine learning techniques. The surveillance camera array consists of multiple PTZ cameras which are divided into two classes: large area cameras and local area cameras. Large area cameras are used to detect whole moving targets (e.g. persons, cars) and local area cameras are used to detect a specific area of the moving targets (e.g. person faces, car license plates). The behaviors of the detected moving targets are recognized to be normal or abnormal by combining short-term and long-term video sequence analyses. Furthermore, the detected faces are fed to a face recognition subsystem to determine whether the person is known or not. The experimental results show the proposed situation awareness system is effective and useful.

Keywords: situation awareness, behavior recognition, face recognition, machine learning

1. Introduction

Situational awareness or situation awareness (SA) is the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their future status. Situation awareness has been recognized as a critical, yet often elusive, foundation for successful decision-making across a broad range of situations, many of which involve the protection of human life and property, including aviation, air traffic control, ship navigation, health care, and emergency response. To surveil a wide area, such as tracking a vehicle traveling through a network of roads in a city or analyzing the overall activities happening in an airport, multiple sensors such as electro-optical (EO) or infrared (IR) cameras are used because the view of a single camera is finite and

limited by scene structures. Many situation awareness systems based on multiple EO/IR sensors have been developed⁽¹⁻⁶⁾. Situation awareness systems are based on a multidisciplinary field related to computer vision, pattern recognition, signal processing, communication, embedded computing, and image processing. The key technologies relevant to situation awareness systems from the perspective of computer vision are: (i) multi-camera calibration, (ii) topology of a camera network, (iii) object re-identification, (iv) multi-camera tracking, and (v) multi-camera activity analysis.

Fig. 1. Situation system awareness architecture.

Camera calibration is a fundamental problem in computer vision and is indispensable in many video surveillance applications. There is a vast amount of literature on calibrating camera views with respect to a 3D world coordinate system⁽⁷⁻⁹⁾. The topology of a camera network identifies whether camera views are overlapped or spatially adjacent and describes the transition time of objects between camera views. The knowledge of the topology is important in tracking objects across camera views for object re-identification⁽¹⁰⁻¹¹⁾. This is the process of matching two image regions observed in different camera views and recognizing whether they belong to the same object or not,

purely based on appearance information without spatiotemporal reasoning⁽¹²⁻¹³⁾. Multi-camera tracking is used to track the behavior of objects across camera views⁽¹⁴⁻¹⁵⁾. Multi-camera activity analysis can automatically recognize different types of activities and detect abnormal activities in a large area by fusing information from multiple camera views⁽¹⁶⁻¹⁷⁾. Activity analysis and object recognition are key components in any situation awareness system. This paper proposes a real-time situation awareness system based on a camera array and machine learning techniques that performs both behavior analysis and target recognition. The remainder of this paper is organized as follows. Sections 2 introduces the architecture of the proposed system. Section 3 discusses the behavior recognition and object recognition capabilities. Section 4 describes the experiment result, and lastly, Section 5 presents the conclusion and future works.

Fig. 2. Process flow for behavior recognition.

2. System Architecture

The overall architecture of the system is shown in Fig. 1. The system employs a camera array with multiple PTZ (pan-tilt-zoom) cameras which are divided into two classes: large area cameras (zoomed out) and local area cameras (zoomed in). The videos from the camera array are fed to the change detection sub-system, which detects the targets (e.g., persons, cars). If the detected target is a person, the local area camera is controlled to zoom in and detect the person's face. The image of the person's face is then sent to the face recognition sub-system. If the detected face is unknown, it will be added to the face database. Simultaneously, the moving target is tracked by using template matching, and its trajectory is extracted as motion information. If the motion pattern is not recognized in behavior database, it will be recorded as a new behavior. The situation semantics

generation sub-system generates the scene semantics based on the object the system recognized.

3. Implementation

The implementation of the proposed system contains (i) behavior recognition, (ii) face recognition, and (iii) situation semantics generation.

3.1 Motion Behavior Recognition

The behavior recognition process flow is shown in Fig. 2 and is summarized in the following.

- (1) The target detection is based on Yolo⁽¹⁸⁾. Yolo is a pre-trained CNN object detector which uses the COCO⁽¹⁹⁾ database. COCO is a large-scale object detection, segmentation, and captioning dataset, containing 88 objects. Fig. 3 shows some examples of detection results at an airport. The detected objects are marked by the rectangles.

Fig. 3 Object detection results by Yolo

Fig. 4. (a) Objects detected at i -th frame. (b) Objects detected at $(i+1)$ -th frame.

- (2) The object tracking is based on template matching. Let O_{im} and O_{jn} denote the objects detected at two consecutive image frames, i and j , where $m = 0, 1, \dots, M$, and $n = 0, 1, \dots, N$, and M and N are the total number of objects detected at frames i and j , respectively. The template matching procedure is as follows:
 - (i) For object O_{im} and O_{jn} , if their labels are different, set the matching measure to 0, and there is no need to do the matching.
 - (ii) If their labels are the same, O_{jn} is resized to the size of O_{im} , denoted as $\hat{O}_{jn}$. The correlation coefficient between O_{im} and $\hat{O}_{jn}$ is calculated. The maximal

correlation coefficient is considered as the match for O_{im} and $\hat{O}_{jn}$, where $m = 0, 1, \dots, M$, and $n = 0, 1, \dots, N$.

- (3) The matched object trajectory that contains the center (x_c, y_c) and width and height size (w, h) of the matched object are recorded to the motion pattern.
- (4) The behavior recognition is based on the motion pattern analysis. It consists of short-term motion recognition and long-term motion recognition. The short-term motion recognition is conducted at 4-second interval based on the motion analysis. The long-term motion recognition uses the motion pattern for a period between 10 to 20 seconds.
- (5) Dilation and erosion operations are applied to the motion pattern generated in step (3). The detected motion pattern is registered to motion database for CNN training.
- (6) The trained CNN is used for motion recognition.

3.2 Face Recognition

The face recognition sub-system contains (i) face detection, (ii) face feature detection, and (iii) face recognition.

Once a human target is detected, the system will use the local area camera to detect the face. The face detection is based on MTCNN⁽²⁰⁾ which is a deep cascaded multi-task framework that exploits the inherent correlation between detection and alignment to boost up their performance.

The face feature detection employs VGGFace2 to generate face embeddings. The VGGFace2, which has been trained by using 3.31 million images of 9131 subjects, accepts 224x224 input images and outputs 2048 face embeddings⁽²¹⁾.

The embeddings are then normalized. The normalized face embeddings for the same person are averaged to make a standard face embedding for that person. The face recognition calculates the cosine distance between the input face embeddings and the standard face embeddings. If the matching measure is less than the threshold, it is considered as a matched face.

3.3 Situation Semantics Generation

Situation semantics generation attempts to provide a solid theoretical basis of reasoning about common-sense and real-world situations. This project generates the semantics of a video stream from the camera array. For example, the video stream could show that a person is breaking a car window. The system in this project will analyze the video and generate

the situation semantics: “A person is breaking a car window”. Situation semantics generation employs SimpleNLG⁽²²⁾. It is the most widely used open-source natural language generation (NLG) library. SimpleNLG is used to generate grammatically correct English sentences. For example, if the system detects a person from the input video, it generates a word “person”, and if the system further detects the change of the person’s location via motion pattern analysis, it generates another word “walk”. When SimpleNLG receives the words “person” and “walk”, it can generate a sentence like “a person is walking”. Furthermore, based on the face recognition result, the system can identify who is breaking the car window.

Fig. 5. Camera array that contains 2 cameras.

4. Experiment Results

This project’s system is built on a Windows 10 platform using the Python programming language. Fig. 5 shows the camera array that contains two Wanscam metal PTZ IP Dome outdoor cameras. These wireless PTZ cameras have full HD 1080P (1920x1080) resolution and clear night vision (80 meters).

Fig. 6. Motion patterns.

Behavior recognition experiment:

To develop the behavior recognition algorithm, 12 scenarios were developed related to a person and a car in a

parking lot. They are summarized in Table 1 and illustrated in Fig. 6. Each scenario was recorded 10 times.

Table 1. Scenarios for behavior recognition.

- (1) A person walks to a car and gets inside the car.
- (2) A person gets off from a car and walk away.
- (3) A person gets off from a car and walk to another direction.
- (4) A person passes by a car.
- (5) A person passes by a car in an opposite direction.
- (6) A person looks around the car.
- (7) A person looks around the car in an opposite direction.
- (8) A person robs a car.
- (9) A person drops an item on the ground.
- (10) A person picks up an item from the ground.
- (11) A person walks to a car and drives it away.
- (12) A car comes and a person gets off from a car.

Fig. 7 shows the tracking results for one recording of scenario 1 in Table 1. (a) shows the object detection result at frame 40. (b) shows the object detection result at frame 120. (c) is the recognition result from frame 0 to 40. (d) is the recognition result from frame 81 to 120. (e) is the generated motion pattern, the white line is the trajectory of a person, red line is the trajectory of a car, and blue line is the trajectory of a backpack.

Fig. 7. (a) Object detection result of for one of scenario 1 at frame 40. (b) Object detection result at frame 120. (c) Recognition result from frame 0 to 40. (d) Recognition result from frame 81 to 120. (e) Generated motion pattern, red - car, white - person; blue - backpack.

Fig. 8. shows a part of tracking result. The first column is the frame number. The second, third, and fourth columns display

information of the detected objects in the frames, Each column is a tuple of that object's label, center, width and height.

23	['car', 438.5, 204.5, 351, 123]				
24	['car', 439.0, 204.5, 352, 123]				
25	['car', 439.0, 204.5, 352, 123]				
26	['car', 438.5, 204.5, 351, 123]	['person', 717.0, 310.0, 94, 234]			
27	['car', 439.0, 204.5, 352, 123]	['person', 716.0, 322.0, 94, 232]			
28	['car', 439.0, 205.0, 352, 124]	['person', 700.0, 318.5, 108, 247]			
29	['car', 438.5, 205.0, 351, 124]	['person', 697.0, 312.0, 100, 238]			
30	['car', 438.5, 205.0, 351, 124]	['person', 690.5, 308.5, 87, 239]			
31	['car', 439.0, 205.0, 352, 124]	['person', 686.5, 304.5, 83, 253]			
32	['car', 438.0, 205.0, 352, 124]	['person', 665.0, 301.5, 96, 273]			
33	['car', 438.0, 205.0, 352, 124]	['person', 656.5, 295.5, 97, 259]	['handbag', 629.5, 351.0, 49, 74]		
34	['car', 438.0, 204.5, 348, 125]	['person', 655.5, 304.0, 97, 250]	['handbag', 617.5, 346.0, 47, 76]		
35	['car', 436.5, 204.5, 349, 125]	['person', 644.5, 302.0, 105, 248]	['handbag', 606.0, 344.0, 50, 76]		
36	['car', 437.0, 204.5, 348, 125]	['person', 639.0, 295.0, 98, 250]	['handbag', 598.0, 337.0, 50, 76]		
37	['car', 434.5, 204.5, 345, 125]	['person', 629.5, 287.0, 95, 236]	['handbag', 587.5, 327.0, 47, 72]		
38	['car', 433.5, 204.5, 341, 125]	['person', 627.0, 276.5, 92, 233]	['handbag', 582.5, 312.0, 49, 76]		
39	['car', 432.0, 205.0, 338, 124]	['person', 612.5, 272.0, 87, 240]	['handbag', 576.5, 304.5, 45, 71]		

Fig. 8. Part of tracking result. The first column is the frame number; the remaining columns are tuples of the detected objects' labels, centers, widths and heights.

Face detection and semantics generation experiment:

Fig.9 (a) – (c) shows the face image database for Person 1, 2, and unknown. Fig. 9 (d) shows the face recognition rate for the test image set 1, 2 and 3. Test image set 1 contains 10 images of Person 1, test image set 2 contains 10 images of Person 2, and test image set 3 contains 10 images of unknown persons.

Fig. 10 shows the experiment that uses a camera array to detect the target. Fig. 10. (a) shows that a person and three cars are detected by the large area camera of the camera array. (b) shows a person's face is detected by the local area camera of the camera array. (c) shows the semantics generated by the system based on the detection result.

	Person 1	Person 2	Unknown
Image set 1	90%	0%	20%
Image set 2	0%	100%	0%
Image set 3	0%	0%	30%

(d)

Fig. 9. (a) –(c) Face image database. (b) Face recognition result.

5. Conclusions

This paper describes a real-time situation awareness system based on a camera array and deep learning techniques. The average processing time for target detection and recognition as well as face detection is 833 milliseconds per

frame. The scene understanding result, i.e., the generation of situation semantics: “The system detected a person and three cars”, takes 306 milliseconds on average. In total, the system takes 1139 milliseconds to process and understand one frame. It can potentially be improved by using a computer cluster and GPUs. Moreover, the system currently employs a small face database and a small behavior database, both of which can be expanded upon in future works.

(a)

(b)

(c)

Fig.10. Experiment result by using a prototype camera array. (a) A person and 3 cars were detected by the large-area camera; (b) The person’s face is detected by the local-area camera; (c) The semantics generation result.

6. Copyright

This paper, submitted to ICIAE, is the original and unpublished research. It is not submitted to other conferences or journals.

References

- (1) R. T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade: “Algorithms for cooperative multisensor surveillance,” Proc. of the IEEE, Vol. 89, pp. 1456–1477, 2001.
- (2) R. S. Feris, B. Siddiquie, J. Petterson, Y. Zhai, A. Datta, L. M. Brown, and S. Pankanti: “Large-Scale Vehicle Detection, Indexing, and Search in Urban Surveillance Videos,” IEEE Trans. on Multimedia, Vol. 14, No. 1, pp. 28–42, February 2012.
- (3) S. Y. Jin, Y. S. Jeong, C. Park, K. J. Oh, and H. J. Choi: “An Intelligent Multi-Sensor Surveillance System for Elderly Care,” Smart Computing Review, Vol. 2, No. 4, pp.296-307, August 2012.
- (4) S. Saxena and R. K. Singh: “A Survey of Recent and Classical Image Registration Methods,” International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.7, No.4 (2014), pp.167-176.
- (5) M. Valera, and S. A. Velastin: “Intelligent distributed surveillance systems: A review,” IEE Proceedings. No. 20041147, pp.193–204, 2004.
- (6) X. Wang: “Intelligent multi-camera video surveillance: A review,” Pattern Recognition Letters 34 (2013) 3–19.
- (7) G. A. Jones, and J. R. Renno, P. Remagnino: “Auto-calibration in multiple-camera surveillance environments,” Proc. of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2002.
- (8) R. Pflugfelder, and H. Bischof: “Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views,” IEEE Trans. PAMI 32, 709–721. 2010.
- (9) B. Triggs: “Camera pose and calibration from 4 or 5 known 3d points,” Proc. of IEEE International Conference on Computer Vision, Vol. 1, pp. 278 - 284, Kerkyra, September 20-27, 1999.
- (10) V. Kettner, and R. Zabih: “Bayesian multi-camera surveillance,” Proc. of CVPR’99, Vol. 2, June 23-25, 1999. Fort Collins, CO.
- (11) X. Wang, K.T. Ma, G. Ng, and E. Grimson: “Trajectory analysis and semantic region modeling using a nonparametric bayesian model,” Proc. of CVPR’08, 23-28 June 2008, Anchorage, AK.
- (12) E. D. Cheng, and M. Piccardi: “Matching of objects moving across disjoint cameras,” Proc. IEEE Int. Conf. on Image Processing, 2006. October 8-11. 2006, pp. 1769 – 1772, Atlanta, GA, USA.
- (13) P. E. Forssen: “Maximally stable colour regions for recognition and matching,” Proc. of CVPR’07, 17-22 June 2007, Minneapolis, MN.
- (14) R. Eshel, and Y. Moses: “Homography based multiple camera detection and tracking of people in a dense crowd,” Proc. of CVPR’08, June 23-28, 2008, Anchorage, AK.
- (15) C. Kuo, C. Huang, and R. Nevatia: “Inter-camera association of multi-target tracks by on-line learned

- appearance affinity models,” Proc. of 11th European Conference on Computer Vision, 2010, Sept. 5-11, pp. 383-396, Crete, Greece.
- (16) P. Smith, N. Lobo, and M. Shah: “Temporalboost for event recognition.” Proc. of IEEE Int’l Conf. on Computer Vision, 2005, Oct. 17-21, Vol. 1, pp. 733 – 740, Beijing, China.
- (17) X. Wang, X. Ma, and W. E. L. Grimson: “Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, pp. 539-555, 2009.
- (18) J. Redmon, A. Farhadi, “YOLOv3: An Incremental Improvement,” arXiv:1506.02640v5, May, 2016.
- (19) T. Lin, M. Maire, S. Belongie, et al: “Microsoft COCO: Common Objects in Context,” arXiv:1405.0312v3, Feb., 2015.
- (20) K., Zhang, Z. Zhang, Z. Li and Y. Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10):1499–1503, 2016.
- (21) Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman: “VGGFace2: A dataset for recognising faces across pose and age,” arXiv:1710.08092v2, May, 2018.
- (22) Simplenlg project: <https://pypi.org/project/simplenlg/>