Understanding Images of Graphical User Interfaces: A new approach to activity recognition for visual surveillance

Li Yu
Lehigh University
Bethlehem, PA 18015
Tel: 1-610-758-4081
E-mail: liy3@cse.lehigh.edu

Terrance E. Boult
University of Colorado at Colorado Springs
Colorado Springs, CO 80933-7150
Tel: 1-719-262-3510
E-mail: tboult@cs.uccs.edu

ABSTRACT
A fundamental problem in surveillance systems is the specification of "activities of interest". While various activity recognition systems have been developed, they have used complex hand-coded representations. What is of interest to a particular surveillance system user can vary greatly, and the security forces using the system are not, in general, advanced computer users.

This paper presents a novel paradigm for specifying and recognizing activities in a surveillance system by visualizing tracking results, and using computer vision techniques to interpret the images presented by their Graphical User Interfaces. Representation of activities of interest can be easily drawn by users. Not only is the drawing-based specification of activities of interest easier than previous approaches, but when a "rule" fires, it is easy to explain "why", by showing the operator the associated drawings. This approach also permits a new type of system integration, where we integrate the "display" of sensors rather than trying to develop a complex communication protocol.

KEYWORDS: User interfaces, visualization, activity specification and recognition

1. INTRODUCTION

The current state of the art in visual surveillance depends on using Graphical User Interfaces (GUIs) to communicate between the sensors and the human operators so the human operator can infer the activities. This goal is achieved by exploiting the human's visual ability to take in vast amount of graphical information, and analyze it into reasonable semantics and infer from that display. Well-designed GUIs may contain massive, accurate, and easy-to-understand information. "A picture is worth a thousand words."

Recognizing activities is a fundamental goal of visual surveillance systems. Currently systems use humans to interpret their system output and recognize activities of interest. There has been considerable work on automating this process [13, 15, 6]. These approaches are powerful for recognizing temporally and spatially complex activities, but the specification of activity of interest requires special knowledge, and is not easy for users, such as security guards, to understand and use those mathematical models to specify and recognize human activities.

In past work we have developed techniques for detecting and locating targets in the GPS coordinate system [3], and this paper builds on the output of such a system. In particular we presume an underlying tracking system detects, classifies, tracks and locates the targets. The resulting target data is sent to a GUI that draws the objects' trajectories on a 2D overhead image or map. If the targets were displayed such that past-locations were displayed as smaller (and shrinking) icons, then a single glance at the GUI would convey direction and crude speed of motion allowing the user to recognize the above without watching the video itself. By observing the map-oriented GUI, a human user can trivially recognize a wide range of that activities such as Entry in a restricted area, Running, PickUp, DropOff, Loiter, Crowding, etc. A drop-off event is displayed on the GUI in Figure 1.

While totally integrated systems are often ideal, the integration of multiple different sensors subsystems can be costly. Thus it is relatively common to exploit the computer's ability to display multiple windows. For example, a radar system can run together with a visual tracking system. Ideally they two would be "integrated" and the targets superimposed on the same map. However the effort needed to integrate is often considerable and may require information the sensor manufactures are not willing

to provide. Thus they may be run independently but displayed simultaneously, e.g. the video surveillance window adjacent to the raw radar "output" window. The two GUIs can be observed together allowing the human to recognize activity patterns of interest that cannot be found by just watching one of the GUIs.

A human operator is quickly fatigued, and not as efficient as machines when performing voluminous routine tasks. Thus, we seek to use of computer vision techniques to recognize these activities. While previous work has sought representations that analyze the "internal state" produced by the vision sub-systems, our goal is to understand the state of running systems by analyzing the semantics contained in their GUIs, with what we call a UI-GUI-based system.

In this paper, we present an approach to recognizing activities by understanding the images of GUIs using appearance-based recognition algorithm. A visual representation of activities is designed and used in both visualization of the tracking results and user specification of activities of interest. Recognition system is trained from user-specified representations. Training on real scene is not necessary, but because it just takes GUI images, training on real data from the running systems' GUIs can be used as training sets as well.

An appearance-based recognition algorithm [12] is used to match images from the running GUI with user-specified images. As an example, we interpret a tracking system using this approach. The tracker detects multiple moving targets in the scene, and generates events of the moving objects. An event includes the object's identifier, its type, its current position, and the time of occurrence of the event. On its GUI, trajectories of moving objects are displayed on a top-view map of the region according to the events.

There has been research on specifying activities with graphical representation for video retrieval system [1, 4, 16]. "Motion-based sketch-drawing" is used to formulate queries for motions in [1, 4], with which users can specify a moving path together with temporal features such as arrival time and duration. QBE (Query By Example) is another technique used for specifying motion-based queries [16].

Programming by Demonstration (PBD) is related to our research in that it explores interaction techniques for non-programmer users to create generalized programs. Tinker [10] is designed for beginners in Lisp language. Chimera [9] enables users to build graphical editing macros from the graphical command history. Eager [5] provides an application-independent approach to handle repetitive tasks. Anticipation highlight is used as the interaction technique that minimally intrudes users normal operations. In [2, 14], pixel data from graphical user interfaces (GUIs) are explored to describe users' intentions. This approach facilitates a PBD system to be integrated into existing environments and can obtain some information that is otherwise unavailable.

Design of an effective GUI is crucial to our work. A poorly designed GUI can make it difficult for our vision system to recognize activities and would likely make it difficult for a human user to infer activities of interests. Because our approach is based on visual appearance of the GUI images, we expect the relationship to be synergistic. GUI changes that make it easier for UIGUI to distinguish behaviors should make them easier for a human user too. There is a large body of work on effective GUI design, [7] gives a review on different information visualization applications, categorized with data types. The Information Mural [8] displays large information spaces by using a 2D, reduced representation, which fits an entire space into a screen. In [11], a map display is developed for information retrieval, which reveals semantic relationship of documents, and facilitates navigating, and perceptual inference on retrieval interfaces.

While the UI-GUI approach has wide application potential, this paper focuses on specifying and recognizing a collection of spatio-temporal events as depicted in a surveillance system's GUI, and on the integration of surveillance system data.

2. VISUAL REPRESENTATION OF ACTIVITIES

The visual representation is used in both visualizing tracking results, and specifying activities of interest. A well-designed visual representation should be not only easy recognizable but also easy to specify. Generally speaking, we represent activities as a sequence of images of the state of the GUI. This allows users to specify the patterns in a graphical way. They can get the GUI into a desired state either by running it on real data, or using a "drawing" package (or extension of that GUI) to place Object/Icons from the GUI at the desired location on the screen. Because the representation is inherently consistent with the GUI, it is ideal for feedback and adaptation by the user. If a specified rule does not match a user's expectations, the incorrect model of an activity can be readily adjusted by observing the actual activity displayed on the GUI and updating the drawings.

2.1 Trajectories

The basic building block in our graphical representation is the movement of a single object over a certain period of time, which is represented as a trajectory. A trajectory is an object's moving history over a time frame. Rather than just showing current location, a short history is graphically presented. It is composed of a sequence of icons, which incorporates spatio-temporal features within the period of time. The location of the icon is the location of the object at a specific time, and the size of an icon represents time since the target was at that location. The larger the icon size, the more recent the target was at that position. The visual attributes of an icon, such as color and shape denote the type of the moving object. Using a particular shape of icons for objects may improve the recognition rate, but our major concern is the spatial arrangement of trajectories, so we in our initial tests used simple dots and different colors for different types of objects. More recent work used icons that may improve discrimination between similar trajectories involving different types. Implicit in the spatio-temporal nature of the trajectory, more subtle semantics can be inferred. Decreasing distance between the icons indicates a slowdown trend. And object's direction is from its trajectory's "little" end to "big" end. A "trajectory" can contain no tail, which means that the object is static. Figure 1 shows a drop-off event which contains a human and car trajectory.

2.2 Concurrent Trajectories

A scenario may consist of concurrent trajectories happening within a period of time. There are often objects that are interacting with one another. A drop-off event contains a car and human trajectory (Figure 2). The user defines the spatial scope of interest for the group of trajectories. It is the active area that is supposed to receive events from the running tracker. It also handles the problem of translation-independent. By the size of the scope, users can specify that an activity is of interest if it happens anywhere in a large area or can restrict it to a very small area. This enhances our representation in two aspects. First, it makes the spatial definition of activities fuzzy, which means it can tolerate users' inaccurate input of activities' locations. Second, it helps to define a pattern of activity that could happen in an uncertain region.

3. ACTIVITY RECOGNITION

Activity recognition is performed by comparing an online test image with stored training images. Both test and training images contain trajectories of an activity, and the trajectories are aligned with the center of images to handle translation independence.

3.1 Image Set Generation and Training

Training images will be generated for combination of different variations of trajectory parameter. Since the parametric appearance algorithm [12] used can recover an object by coarsely sampling the space of the object's appearances, it is not necessary to generate every possible pattern of activities. Training data are directly generated from users' drawings by perturbing icon locations (e.g. by Gaussian distribution). Rotation is applied to a trajectory if it is orientation independent. However, by perturbing a trajectory, some unintended "activity" images might be included. The resulting expanded training set can be viewed and edited by the user.

3.2 Real-Time Recognition

In real-time, the tracker keeps track of multiple moving objects. An event happens if the tracked object moves beyond a predefined distance from its previous position. An event contains the moving object's type, position, time, etc. The position is translated to the local coordinates of the GUI, and an icon is displayed on the GUI for each event. Icons represent an object's moving history as described in section 2.1. The display of the GUI changes, as old icons fade out and new icons are displayed. A test image is constructed by collecting icons in a region of interest. An activity is recognized if the test image is close to one of the training images representing the activity in the eigenspace [12].

4. PRELIMINARY EXPERIMENTS

The experiments here are meant to demonstrate some of the concepts and highlights of the issues. The task in these demonstrations is to detect three activities of interest (CAR-WEST, DROP-OFF, and PICK-UP) in a test set with distracters, and find out how well the system can discriminate similar patterns of activities.

The visual representation of an object in the GUI is an important element for our activity classification. We can use different visual representations as long as they discriminate different objects and the trail of a moving object well. To demonstrate it, we carry out this experiment twice with the same geo-spatial test data, but with two different GUI representations of objects. In the first experiment, all objects are represented by dots. Different types of objects are denoted using different colors. A green dot represents a car, and a red dot a person. In the second experiment, icons with the same color are used to represent different types of objects. (This leaves color to be used for other things, e.g. threat level). The experiment shows similar results with dot and icon representation, which means that distinctive colors and shapes have similar visual effect for discrimination. Thus we just analyze the experiment with icon representation.

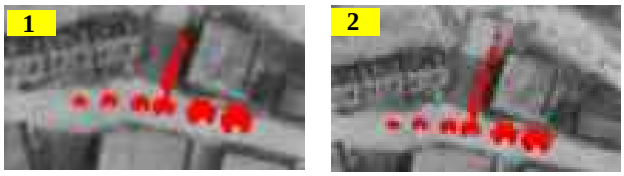

Figure 2. Two distracters in test data – 1. ambiguous precise pickup, 2. distinctive precise-pickup.

As an example of the idea of synergistic feedback, we have a "precise pickup" as a distracter in the test data, where the car and person were moving to arrive at the pickup point at the same time (instead of one waiting for the other), as shown in Figure 2. Note that this PICK-UP is hard to differentiate from DROP-OFF. To illustrate how different design of UI can influence the detection rate, we used another precise PICK-UP with a longer human trajectory. Shown as distinctive precise-pickup in Figure 2, it looks more different from DROP-OFF in that walking directions are more distinctive with longer trajectories. We used these two patterns of precise-pickup alternatively in the test data, and experiment results shows that the false alarm rate of distinctive precise PICK-UP is much lower than that of ambiguous precise PICK-UP.

5. DISCUSSION

This paper introduced a new paradigm for activity modeling -- Understanding Images of Graphical User Interfaces (UI-GUI). We interpret activities as a collection of spatio-temporal events as depicted in the images of the systems' GUIs. Easy-to-use graphical representations of activities are provided, which can be used to train the recognition engine. Activities of interest can be recognized in real time using an appearance-based recognition algorithm. An example of surveillance task was discussed in detail.

Activity recognition using new paradigm is not dependent upon the GUI of running applications, but does depend on it providing a visual distinction of the activities of interest. A poorly designed GUI can influence the interpretation process. If the activities of interest are only weakly distinguishable the UI-GUI performance degrades gracefully. An example compared icon-based GUIs and "colored-dots".

Currently, we do not deal with temporal segmentation of long-term activities, for example a human activity that takes several phases to finish. One approach currently being explored is to adopt a state-based approach, with each state representing one phase of a complex activity. An activity then becomes a sequence of recognitions, and when "partial" of the activity detects its event within its spatial scope it adds an "icon" representing that activity on the GUI. In this way the human user could also see the partial activity state, and the GUI based approach can continue to find the longer-term activity.

Our preliminary work shows the promise of UI-GUI in activity recognition. However the UI-GUI approach is not limited to activities in surveillance systems - it is expected to have a broad impact.

6. REFERENCES

1. Ahanger, G. Benson, D. and Little, T. D. C. Video Query Formulation. *In Proceedings of Storage and Retrieval for Images and Video Databases II, IS&T/SPIE Symposium on Electronic Imaging Science and Technology*, pages 280---291, San Jose, CA, February 1995.
2. Amant, R. S., Lieberman, H., Potter, R., and Zettlemoyer, L. Visual Generalization in Programming by Example. *Communications of the ACM* 43(3):107--114. http://citeseer.nj.nec.com/amant00visual.html
3. Boult T. E., Micheals R. J., Gao X. and Eckmann M., Into the woods: visual surveillance of non-cooperative and camouflaged targets in complex outdoor settings, *the Proceedings of the IEEE*, Oct 2001.
4. Chang, S.-F. Chen, W. Meng, H. Sundaram, H. and Zhong, D. VideoQ: an automated content-based video search system using visual cues. *In Proceedings of ACM Multimedia* 1997, Seattle, November 1997.
5. Cypher, A. Eager: Programming Repetitive Tasks by Example. *In Proceeding of ACM CHI'91 Human Factors in Computing Systems* (1991), pp. 33-39.
6. Davis, J. and Bobick, A. The representation and recognition of action using temporal templates. *Proceedings Computer Vision and Pattern Recognition* (CVPR'97). pp.928-934. 1997.
7. Geisler, G. Making Information More Accessible: A Survey of Information Visualization Applications and Techniques. Technical report, Information and Library Science at University of North Carolina, January 1998.
8. Jerding, D. F., Stasko, J. T. The Information Mural (Draft), Graphics, Visualization, and Usability Center, College of Computing, Georgia Institute of Technology Technical Report GIT-GVU-96-xx, March 1996.
9. Kurlander, D., Feiner, S. A History-based Macro by Example System. *In Proceedings of the ACM Symposium on User Interface Software and Technology* (1992), pp 99-106.
10. Lieberman H. Tinker: A Programming by Demonstration System for Beginning Programmers. *Watch What I Do: Programming by Demonstration*. MIT Press, Cambridge, MA. 1993.
11. Lin, X. Map Displays for Information Retrieval. *Journal of the American Society for Information Science*, 48(1), 40-54, 1997.

12. Nayar, S. K. Murase, H. and Nene, S. A. Parametric appearance representation. *In Early Visual Learning.* Oxford University Press, February 1996.
13. Oliver, N., Rosario, B., Pentland, A. A Bayesian Computer Vision System for Modeling Human Interactions. *IEEE Transactions On Pattern Analysis And Machine Intelligence.* Vol. 22, No. 8, August 2000.
14. Potter, R., Shneiderman, B., Bederson, B. Pixel Data Access for End-User Programming and Graphical Macros. CS-TR-4019, UMIACS-TR-99-27.
15. Yuri A. Ivanov, Aaron F. Bobick. Recognition of Visual Activities and Interactions by Stochastic Parsing. *IEEE Transactions On Pattern Analysis And Machine Intelligence.* Vol. 22, No. 8, August 2000.
16. Zhang, H. J. Low, C. Y. Smoliar, S. W. Wu, J. H. Video parsing, retrieval and browsing: an integrated and content-based solution, *Proceedings of the third ACM international conference on Multimedia*. January 1995.