UC Berkeley
Faculty Research

Title
Contextual Activity Visualization from Long-Term Video Observations

Permalink
https://escholarship.org/uc/item/4q09629b

Authors
Morris, Brendan
Trivedi, Mohan Manubhai

Publication Date
2010-08-01

Contextual Activity Visualization from Long-Term Video Observations

Brendan Tran Morris and Mohan Manubhai Trivedi
University of California, San Diego
August 2010

Contextual Activity Visualization from Long-Term Video Observations

Brendan Tran Morris and Mohan Manubhai Trivedi, *University of California, San Diego*

The Contextual Activity Notification Visualization Analysis System (Canvas) provides a user interaction interface for instantaneous feedback of contextual processing units that enable high-level semantic extraction and understanding.

I ntelligent monitoring of environments has progressed rapidly in the past 10 years.[1] Major technological advancements have pushed the field toward ever-more complex environments. The decreased price of video cameras, a primary sensor for surveillance applications, along with improved quality has enabled the use of multiple cameras in more varied spaces. Vast amounts of data can be transmitted efficiently because of high-quality video-compression techniques and improved wireless communication, which facilitates flexible setup and configuration. Most importantly, the research community has made great strides in providing intelligence to these spaces. Low-level problems such as object detection and tracking are possible in real time, making common surveillance tasks straightforward, such as monitoring a sensitive area for unauthorized entry. Intelligent monitoring now seeks to provide situational awareness for a semantically meaningful understanding of environment activity.

The key to accurately understanding an environment is incorporating the needs of the monitoring system's user. A human must be included in the analysis loop for critical decisions because these decisions must be based on a deep understanding of the environment and monitoring situation.

Unfortunately, due to vast amounts of streaming information, limited attention, and distributed awareness, human operators cannot accurately and effectively monitor large areas and networks. Automated computational techniques are vital for the monitoring process to help highlight and guide user attention to relevant areas, thus relieving tedious concentration on noncritical information. The challenge is to distill the volumes of monitoring information into a manageable quantity and present it to users so they can make appropriate decisions in a sufficient amount of time.

In this article, we present the Contextual Activity-Notification Visualization Analysis System (Canvas), which is used to develop advanced monitoring techniques, integrate cameras installed around the University of California, San Diego (UCSD) campus and centralize information.[2] Our work focuses on building an upgradeable framework for simple user interaction through an accessible visualization. Rather than present a

user with raw sensor data from the physical world, we introduce visualization layers to abstract the internals of monitoring algorithms and provide a clean consumable computational output. CANVAS provides a flexible backbone that lets us improve vision algorithms while providing a seamless visualization interface. This ultimately improves the effectiveness of the monitoring by focusing attention and presenting only the most relevant information. The visualization is built on Web technology to make the information available anywhere, anytime.

System Description, Framework, and Functionalities

The block diagram in Figure 1 depicts CANVAS's major components. The system's central goal is to provide users with ubiquitous access. This is reflected by the archival block located in the center of the diagram. A database collects and stores data that is accessible through a standard Internet connection for quick retrieval. Most of the database storage is devoted to data collection from the connected sensors. Any number of sensors can be hooked into the database. Typical sensors are video and audio devices that each have specialized data-extraction techniques, such as position estimates via tracking or object descriptors.

The archive data is used to train computational modules in the learning block. Example modules can distinguish different types of objects (such as pedestrians from vehicles), model highway traffic flow, and compactly represent activity through trajectory learning. The models are archived and used for live analysis, where current sensor readings are used in conjunction with the trained models to describe the scene's
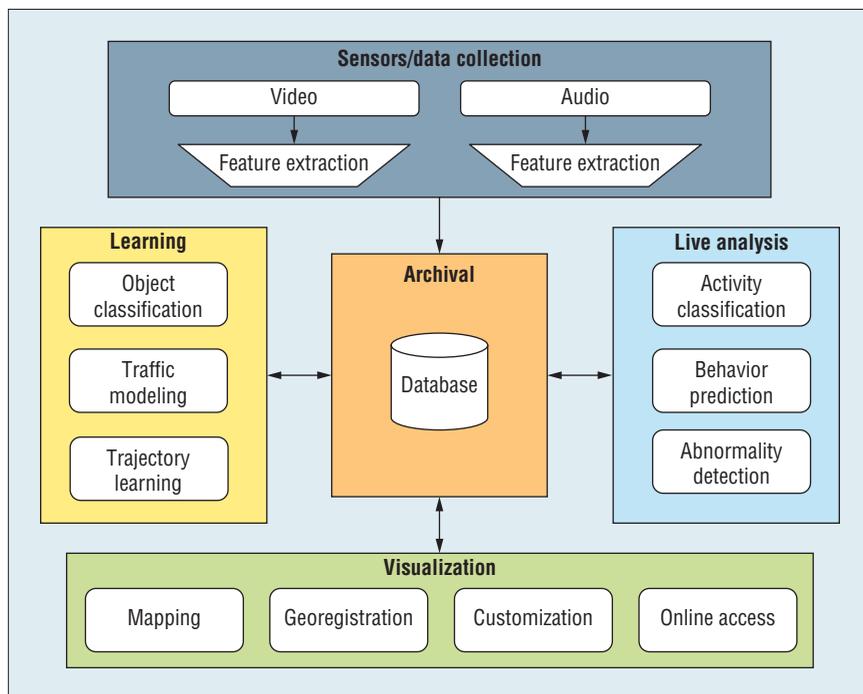

Figure 1. CANVAS monitoring diagram. The monitoring framework relies on a layer of physical infrastructure that includes cameras and other sensors. The raw sensory data is archived in a database for retrieval. The notification layer relies on the visualization module to provide necessary data in real time. Users can customize data and modify results. A hidden layer, connecting the physical world to visualization, incorporates the analysis modules and associated learning devices necessary to provide users with contextual information.

current state. The results of live analysis can be wired back into the database as added supplementary features—for example, a trajectory has position and velocity as well as an object description. Finally, the database contents are made available to users through the visualization module. Users can query the database to retrieve relevant information and have the display updated in real time.

All the modules except data collection pass information in both directions from the archival block, which lets the system dynamically change over time. Models developed through learning techniques can query more recent data to update and refine results, which in turn modifies the live-analysis output. Modification is even possible through the visualization module. In this case, users can customize results to present the most relevant information for their task.

These end user goals can help dictate which types of analysis are necessary.

Information Archive

The heart of CANVAS is the database archival system. We implemented a MySQL relational database system to provide access to organized information tables to multiple users. The widespread use of MySQL has led to the development of many libraries to connect with the database from different programming languages and operating systems. This operational flexibility allows virtually any machine with a network connection to communicate with the database and access its data.

The archival block's main goal is to timestamp and store sensor data that provides measurements about the state of the monitored world. As the database is updated, a historical context emerges that is necessary

Figure 2. University of California, San Diego video network. A network of video cameras around campus provides coverage of different environments. Both rectilinear pan-tilt-zoom (PTZ) and omnidirectional cameras monitor highway vehicle traffic and the close interactions of people and vehicles on campus.

for accurate scene understanding. The centralized database allows for a fluid design because it can grow and adapt to new information types and requests as necessary. Training databases, used for learning, can be separated and maintained as subsets of the full database. New information and measurement types can be included with the addition of new sensors or computational modules. This adaptation is necessary for long-term use because monitoring needs can change over time.

We split the database into three main partitions: sensors, models, and data. The first partition holds information about all the connected sensors. Each camera sensor is denoted by its type (such as pan-tilt-zoom [PTZ] or omnidirectional), location (latitude and longitude), and information for mapping (PTZ setting and conversion from image to world coordinates). We can quickly integrate new cameras into the CANVAS system by including this sensor information. The model partition maintains the learning results used during

live analysis. This partition denotes the model functionality and the parameters necessary for analysis. The last database partition deals with the raw sensor data.

A set of secondary databases are populated by video processing for use by the learning modules. The measurement database holds information describing the appearance of each detected object for type classification. Tracking information, including location, speed, and acceleration, is stored in the tracks database for trajectory learning. The traffic-modeling module relies on information stored in the highway statistics database, which includes vehicle flow, density, and speed logged every 30 seconds. Finally, the live database is automatically updated using current data to provide information for visualization.

Data Collection and Sensors

The data collection front end provides CANVAS with meaningful and useful signals. All the low-level data generation and extraction happens within this block. We designed

sensor-specific filters to extract measurements or features from raw sensors. Some filters are simple and merely pass the raw measurement onto the database (such as inductive loop sensors), while more complicated filters require processing (such as tracking for motion description and measurements of object size and shape).

Video cameras are our primary sensors. Figure 2 shows a map of UCSD along with a few of the many camera nodes situated around campus. A variety of environments, both indoor and outdoor, as well as different coverage and different objects of interest are present. Using the principle of distributed interactive video arrays (DIVAs),[2] we monitor highway traffic along Interstate 5, human-vehicle interactions on campus roads, and people indoors using both PTZ controllable and wide-area-covering omnidirectional cameras. The networked cameras stream video for remote processing, while nonstreaming cameras require a local machine to capture and send analysis data along a network link.

Learning and Analysis

Although the learning module usually operates as an offline process and analysis is needed in real time, the two modules are closely linked. Live analysis relies on the learned models to make sense of sensor data and understand the monitoring scene. This section describes a number of learning techniques and the questions that can be answered during live analysis using the model database.

For each learning module, we created a training database by extracting the needed information from the archival database. A training database is accumulated by collecting the appropriate signals over a sufficient time period. Analysis models can be created by applying learning algorithms to the compiled data. Database maintenance updates training data for adaptive models, which more accurately represent the monitoring scene's current configuration.

Analysis modules are essential for effective monitoring because they ease the cognitive load on human observers. Multiple analysis tasks can be run in parallel on multiple video feeds, which is difficult for humans.

Object Classification

Classification identifies the type of detected object based on its visual signature. Using the 2001 US Department of Transportation National Household Travel Survey for guidance, we identified the seven most often occurring vehicle types in highway streams: sedan, pickup, SUV, van, semi, truck, and bike. This detailed real-time fleet composition is a missing management component essential for estimating emissions or infrastructure load assessment.[3] On campus, detected objects are marked as either a car, pedestrian, biker,

skateboarder, or a group of people. This classification helps identify critical situations when vehicles and people interact in close proximity.

An object's similarity to examples in the training database determines its unique type identification.[4] Each object has measurements m_i taken and transformed into comparable features using linear discriminant analysis (LDA): $o_i = \mathbf{W}m_i$.

An object's appearance o_i is summarized after applying the feature transformation matrix $\mathbf{W}$, learned during the training phase, to the raw measurement m_i. The class similarity

> Analysis modules are essential for effective monitoring because they ease the cognitive load on human observers.

w_c, computed using a weighted K-nearest neighbor (wKNN) technique, assesses how similar object i is to class type c. The similarity metric,

$$w_c = \sum_{\substack{d=1 \\ o_d \in D_c}}^{K} \frac{1}{\|o_i - o_c\|},$$

compares the object example to its K closest training examples in the training set D_c for object type c. The object type is labeled as the class c that has the highest weight over the extent of tracking for T frames:

$$L_T = \arg\max_c \sum_{t=1}^{T} ln \frac{w_c^t}{\sum_c w_c^t}.$$

Traffic Modeling

Highway management relies on knowledge of the location and number of vehicles in the transportation network. The essential measures of flow (number of vehicles/time), density (number of vehicles/distance), and speed (miles per hour) estimates are generated in 30-second intervals by visual vehicle tracking. Traffic models emerge through aggregation over time and location that describe varying highway conditions. Unlike loop detectors, we can also compile the statistics based on vehicle type because of object classification, as Figure 3a shows. This allows for fine-grained analysis of the effects of commercial and private vehicles on highway control, environmental emissions impact studies, and road wear and tear.

We can track daily speed variations using historical measurements. Figure 3b shows the speed fluctuations over the course of a week. Notice the significant slowdown during the Friday evening commute not seen on other days. These daily speed profiles are used to indicate the motion state of vehicles during online tracking by the bounding-box color: {speeding, normal, slow, stopped} = {blue, green, yellow, red} (see Figure 3c). Rather than relying on posted speed limits, speeding vehicles are recognized based on the historical driving conditions. For example, during congestion situations, dangerous speeds are significantly lower than the posted limit.

Trajectory Learning

Recently, one of the most popular techniques for automated surveillance and monitoring is trajectory learning (see the "Trajectory Learning for Intelligent Monitoring" sidebar on page 59). This technique makes it easier to monitor larger video networks because activity models are learned
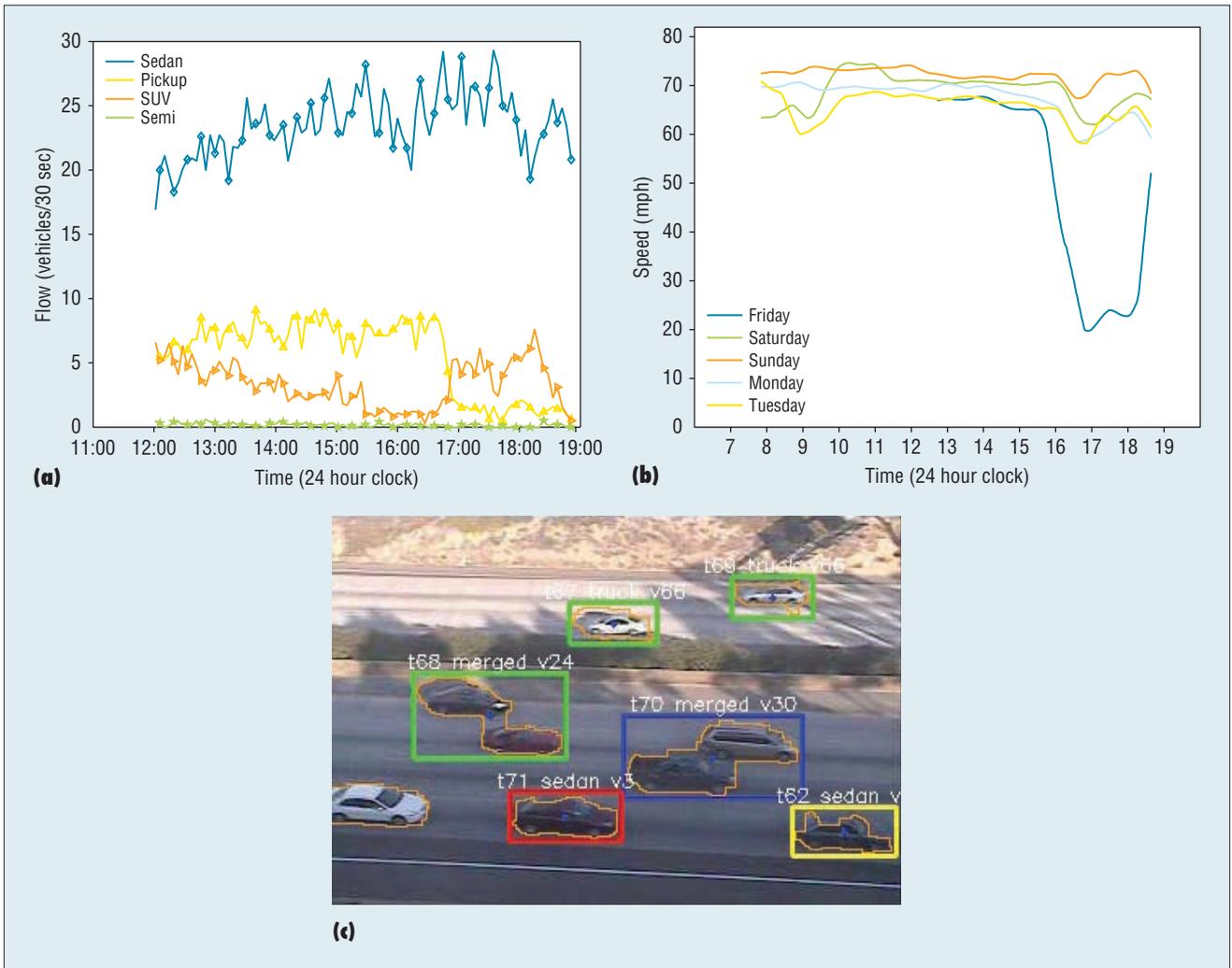
Figure 3. Traffic modeling. (a) Northbound highway flow for specific vehicle types. (b) Southbound speed characteristics for different days of the week. (c) Speed profiling based on daily models: {speeding, normal, slow, stopped} = {blue, green, yellow, red}. Commuter congestion causes differing characteristics in either highway direction. At this hour, the normal southbound speed is significantly slower than northbound.

automatically without the need for manual specification.

Trajectory learning begins by collecting a training database of object trajectories. The training database is clustered by comparing the similarity between tracks[5] to represent the typical scene behaviors. An activity's spatiotemporal properties are compactly and probabilistically represented with a hidden Markov model (HMM), where each HMM state indicates position and dynamic information (velocity). The activity HMMs are inserted into the model database and

used for live monitoring in order to classify current activity, predict future behavior, and detect abnormal events. (The "Theory of Trajectory Learning" sidebar on page 60 highlights the theoretical steps of trajectory learning and analysis.)

Activity Classification. The activity models learned from trajectory analysis help describe and indicate which activity most likely generated the track. During live analysis, the model with the highest likelihood explains what an object was doing while

under observation. Transmitting just the model label can help compress a trajectory into a single, low bandwidth description.

Behavior Prediction. Using the trajectory models, we can also predict future behavior. Instead of evaluating an entire track, the likelihood of a partial trajectory is evaluated to generate a prediction label that is updated and refined with each new video frame. This activity-level prediction extends further in time than a standard one-step prediction (Kalman prediction)
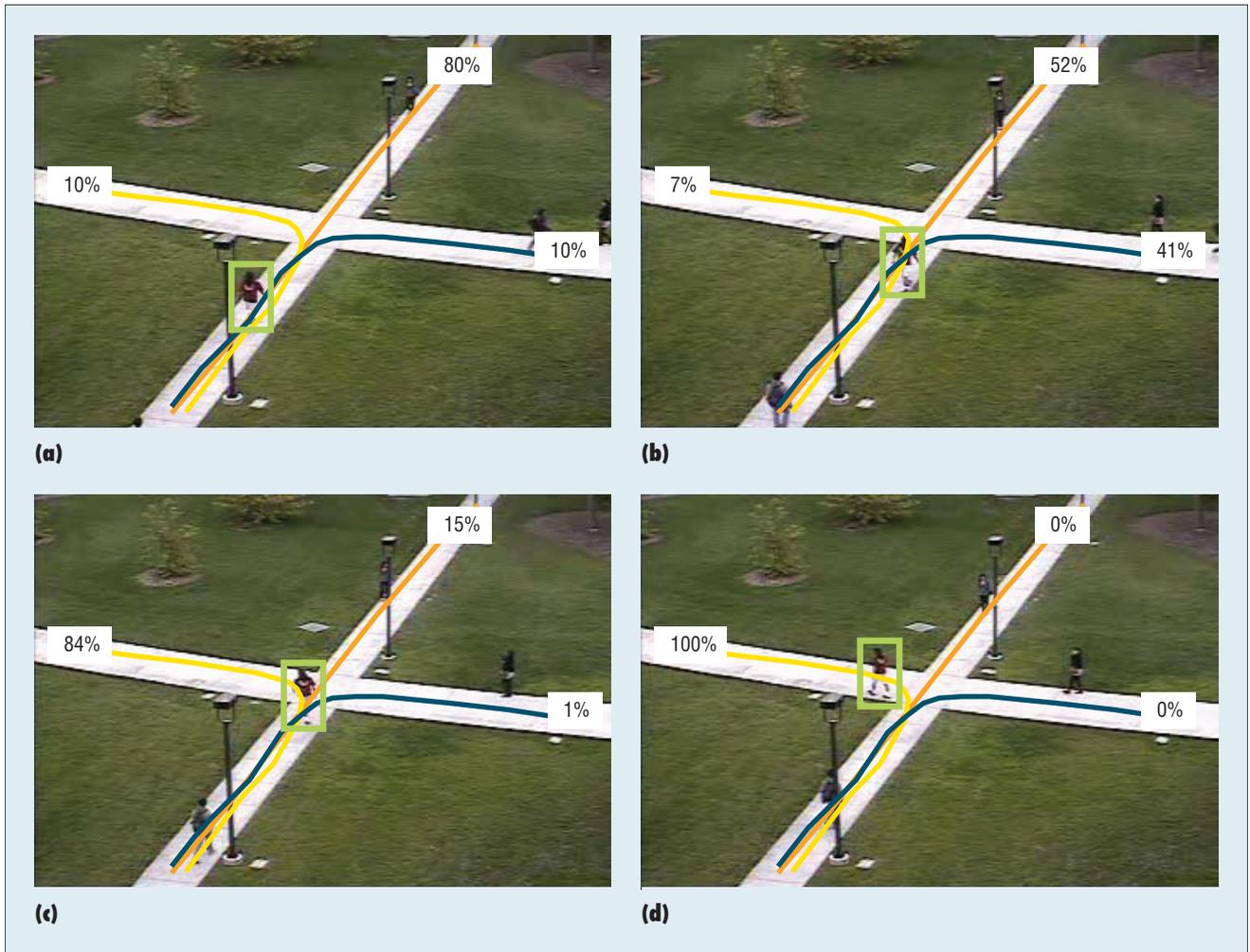
Figure 4. Historically relevant long-term activity prediction. The best three predicted paths are displayed with an associated confidence. (a) Initial estimates of future action. (b, c) Path probabilities change based on local measurements and as more tracking data is accumulated. (d) The final path indicates movement seconds into the future, which could not be accurately estimated using just local motion models.

because it leverages acceptable behaviors rather than relying on a simple motion model, which can drift.

Figure 4 displays a predication example, showing the three most-likely activities with their associated confidence. As more data is collected, a better picture of future behavior is formed, as denoted by the adjusted path confidences.

Abnormality Detection. In many monitoring situations, the most interesting events are unexpected. These atypical occurrences indicate unexplained activities that require further

examination. A trajectory is deemed to be anomalous if it does not fit any of the activity models well (low likelihood).

During live tracking, an incomplete trajectory can be evaluated for its current abnormality state in order to promptly detect when an unusual deviation occurs. The image sequence in Figure 5 shows the set of activities at an intersection and marks a typical behavior with a green box. When the person cuts across the lawn, the anomaly is noted immediately as a red bounding box. Figure 6 illustrates the typical vehicle routes

and examples of anomalous trajectories extracted from automatic postprocessing.

Visualization

The visualization block's main goal is to provide a common environment for displaying the live analysis modules. The visualization environment presents an immersive and interactive display that preserves the context of the information sources. Simultaneous access to different data sources lets the user control the area, scale, and information of interest without changing the surrounding
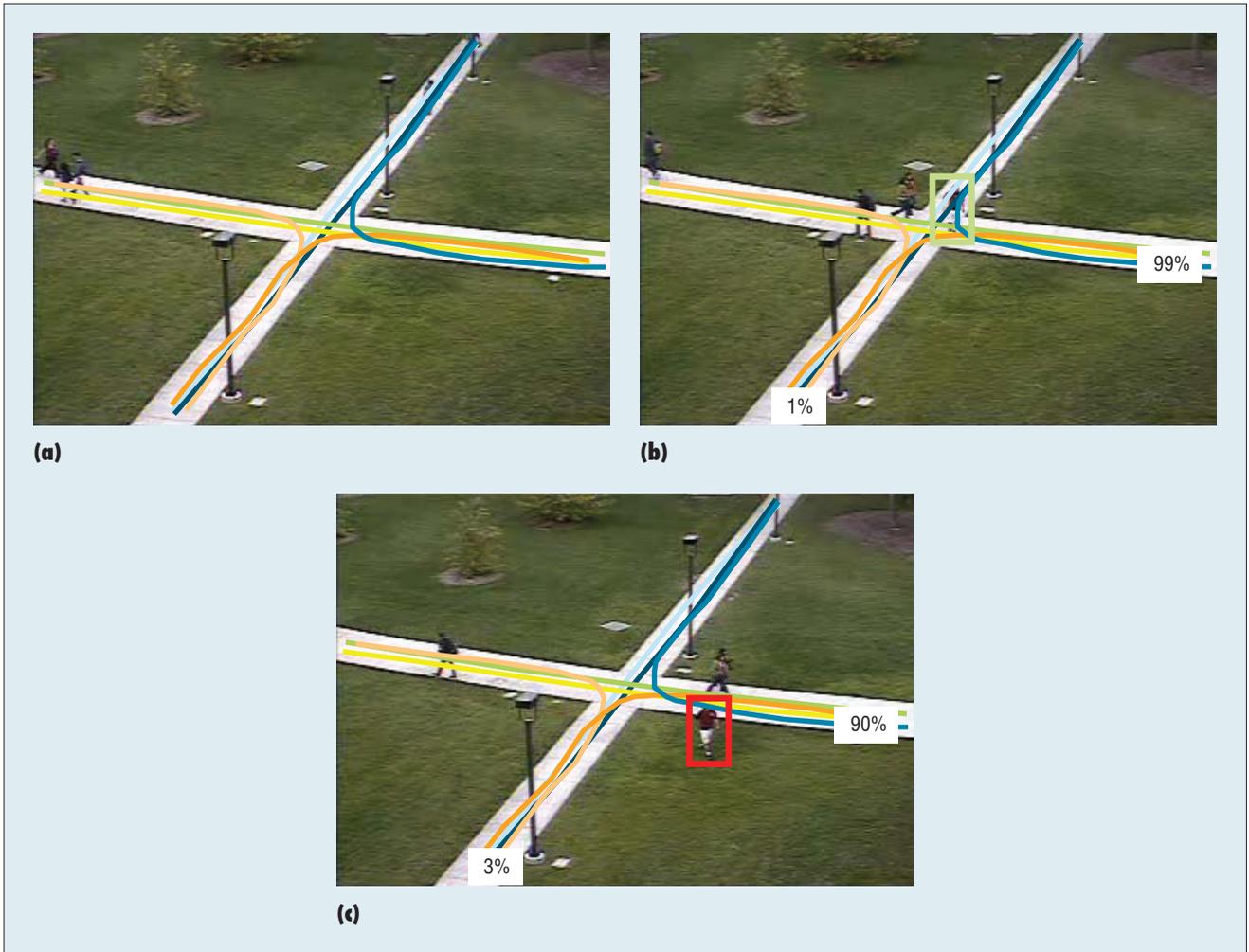
Figure 5. Real-time detection of abnormal activities. (a) The set of all typical motion routes for a crosswalk. (b) Detected man in green starting a turn at the intersection. (c) When the man leaves the path, the anomaly is marked by the red bounding box.

environmental context, enabling a cohesive picture that provides the user with complete situational awareness.[6] Awareness is realized through functional display layers built for each analysis module, where each additional visualization layer provides a more detailed picture of the monitoring state.

While providing expansive environmental context, we take care to avoid distractions that can detract from the principle monitoring task.[7] Instead of overloading the display with many annotations, we distill information and visualize it using icons and avatars (see Figure 7a). This filtered view uses automatic highlighting to limit the cognitive load on users and help focus their attention on the locations most likely to be interesting.[8]

The visualization block indicates the location of sensors with respect to one another, gives access to raw video feeds, presents pertinent analysis results, and provides a user interface to navigate, query, and customize the display.

Mapping

Although the real world is 3D, we do not contextualize information in a 3D environment because this would limit usage to locations with complete 3D graphic models.[9] Instead, we use a 2D map representation of the environment. A map provides surrounding environmental context, which helps users comprehend spatial relationships between objects, increasing situational awareness.[10] We built the user display using the Google Maps API because it is a familiar interface (often used for directions) and its wide coverage makes it applicable to most outdoor locations. The environmental context is available through different modalities, such as aerial imagery or through geographical information system (GIS) type layers depicting structures and areas of interest. The API also supports user interaction with the use of draggable markers and other line-drawing tools.
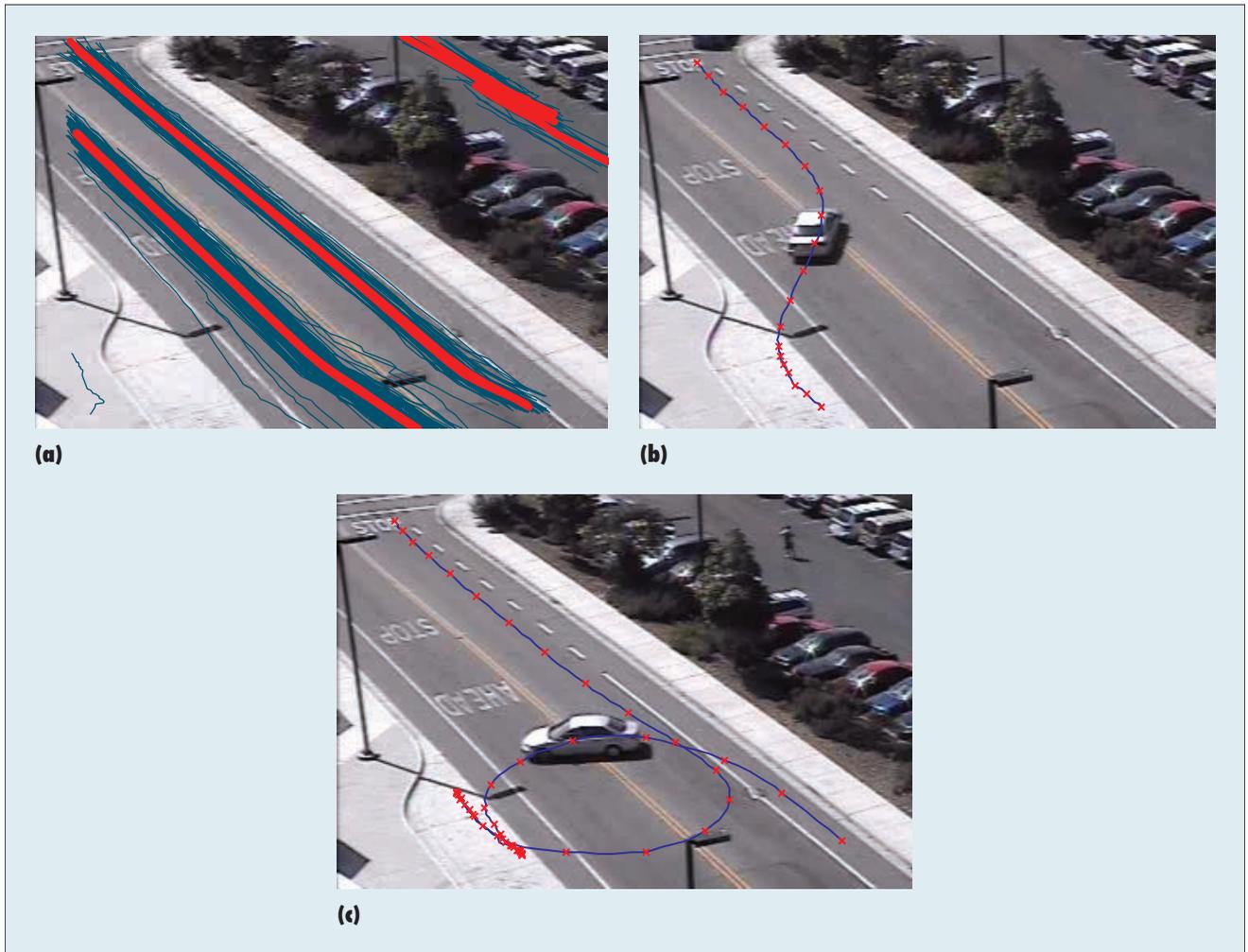
Figure 6. Example showing typical behaviors learned from trajectories and abnormal activities that were automatically detected. (a) Training trajectories are in blue and the learned activities are overlayed in red. (b) A vehicle stops in the bus turnout. (c) A loop is performed through the bus turnout. The red hashed blue lines show example abnormalities.

Georegistration

To properly visualize analysis results on the map, the outputs must be properly aligned to the map coordinates. Therefore, we transform sensor coordinates into GPS latitude and longitude coordinates using a georegistration process, which requires calibration between the sensor and map spaces. Image-based calibration is learned through a homography transformation H, mapping the image-pixel locations on the ground plane (such as the road) x_{im} to its corresponding latitude and longitude coordinates on the map X_{GPS}: $X_{GPS} = H x_{im}$.

A camera's homography can be found by using a GPS receiver to collect the latitude and longitude coordinates of specific image points. H can be estimated using the corresponding coordinates with the four-point algorithm for a planar scene.[11]

Customization

Another design principle of the visualization module is to present information to a user only when needed. Complex environments are filled with activities and events that might be irrelevant for most users. A successful service will provide user-specific information to answer the most-relevant questions. An example of this design paradigm is personalized traffic reports that generate travel estimates given a user-specific commuter route.[12]

This design paradigm called for a simple interface that would abstract the database connection and communication from a user. The CANVAS display customization is available through buttons that overlay results onto the map. In this way, the appropriate SQL commands are generated by the Web page rather than by the operator, removing the need for training.

The user interface presents clickable controls to select camera feeds, change environmental context (see map layer in Figure 7a and aerial imagery in Figure 7b), and display analysis results.

Figure 7. Canvas visualization page. (a) A campus street is monitored using two slightly overlapping cameras. The output of object classification and tracking is marked on the map. Icons indicate the object type and are placed on the map based on camera georegistration information that converts image coordinates to GPS latitude and longitude. (b) Environmental context is presented using an aerial highway image. The detected vehicles are marked with car icons, which appear in the different lanes.

Trajectory Learning for Intelligent Monitoring

Technological advances in hardware, compression, and wireless transmission coupled with greater societal acceptance has led to widespread deployment of video cameras. These cameras stream vast amounts of information that need to be analyzed continually. Without computer-assistive technologies, such data would be impossible for human operators to process without errors or omissions of critical events due to inattention, fatigue, or boredom.

Trajectory learning is one of the key techniques for automatic activity analysis in surveillance systems. Trajectory descriptors have been used successfully for video indexing and retrieval and are used increasingly along with data-mining and machine-learning techniques to understand activity.[1–3] Rather than define activities of interest, models are built in an unsupervised fashion based on observed data. Through careful observation of motion, typical actions reveal an underlying scene structure, which can be extracted in three basic steps.[4] First, objects are tracked and trajectories collected. The trajectories are compared and then grouped by clustering. Finally, each cluster of trajectories is compactly summarized by a modeling technique and stored for future comparison.

The advantages of automatic surveillance based on trajectory learning are as follows:

- *Increased flexibility.* Motion is a low-level feature that can be extracted in a variety of indoor or outdoor surveillance environments.
- *Reduced reliance on expert operators.* Activities are not defined by hand but by data.
- *Principled methods for determining atypical activities.* Anomalies are statistically determined and data-driven.
- *Real-time implementation.* Activity models are typically simple for fast comparison and can be evaluated as data arrives.

Using the trajectory models, it is possible to classify observed activities, detect abnormal activities, and make better long-term predictions on future activities by leveraging historical data. In addition, all this analysis can be performed in real time, which ensures a timely detection response.

References

1. N. Anjum and A. Cavallaro, "Multifeature Object Trajectory Clustering for Video Analysis," *IEEE Trans. Circuits Systems Video Technologies*, vol. 18, no. 11, 2008, pp. 1555–1564.
2. L. Patino et al., "Extraction of Activity Patterns on Large Video Recordings," *Computer Vision*, vol. 2, no. 2, 2008, pp. 108–128.
3. C. Piciarelli, C. Micheloni, and G. Luca Foresti, "Trajectory-Based Anomalous Event Detection," *IEEE Trans. Circuits Systems Video Technology*, vol. 18, no. 11, 2008, pp. 1544–1554.
4. B.T. Morris and M.M. Trivedi, "A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance," *IEEE Trans. Circuits Systems Video Technology*, vol. 18, no. 8, 2008, pp. 1114–1127.

The two major user customization/selection modalities are *video-feed selection* and *map layers*. Video-feed selection is used to initialize raw video streams from up to two live feeds. The map layers provide the common map-based visualization of results. Map scale and navigation is controlled through the Google Map API, and computational layers are created for the analysis modules (traffic flow, classification results, and trajectory analysis). We created a layer for each analysis type and camera pair. Figure 7 shows two different classification layers. Figure 7a shows a classification layer denoting humans and vehicles on campus, while Figure 7b shows vehicle tracking.

Further customization is possible with advanced users who design specialized computational layers. Similar to GIS software, a user would define the queries necessary to extract pertinent information as well as define any visualization layers. An example is a zone alert to monitor a sensitive region. The advanced user would specify a polygon in the image and search the tracks database for objects within this region.

Online, Mobile Access

The visualization block's final goal is to provide access to information wherever it is needed through remote access. This allows more convenient monitoring because it does not have to occur on site. The Canvas visualization was built on Web technology to be platform independent and portable, relieving the need to design or compile different versions of the code for specific platforms. Besides remote availability, design in Web-based technologies makes it possible to realize mobile, portable access and help fulfill the promise of a ubiquitous age where the rapid developments in mobile handset and network technologies can bring customized management services to all people.[13] The increasing popularity of mobile applications on cellular phones indicates the desire for instant connectivity and functionality.

Evaluation

Over a single day the total accuracy for classification of eight different vehicle types was 78 percent for the Interstate 5 scene (see Figure 3c). Table 1 presents the accuracy for each hour of the day with sufficient lighting for vehicle detection. The performance degraded due to shadows cast during the mid-morning hours, but this could be ameliorated using shadow-suppression techniques.

The highway traffic statistics module, named Vector,[4] performed quite well. Comparison of the Vector

Theory of Trajectory Learning

Trajectory dynamics analysis provides low-level situational awareness to a range of surveillance applications. Typical motion is repetitive, which allows event analysis in the context of historically meaningful motions.[1,2]

Learning

Learning activities include the following:

- *Tracking.* Objects are tracked and trajectories, F_i, are collected into a training database. $F = \{f_1, \ldots, f_t\}$, where $f_t = [x, y, u, v]$ (note that xy is the position and the associated velocities are uv).
- *Clustering.* Trajectories are clustered into similar groups, where each grouping is indicative of a typical activity. Similarity is measured using trajectory-specific distance measures $D(F_i, F_j)$ that are designed to handle the varying lengths of trajectories.[3]
- *Cluster validation.* The number of activities in a scene is unknown a priori and must be estimated based on the similarity of clusters.[1]
- *Modeling.* Each activity cluster is probabilistically modeled for inference. A trajectory's spatiotemporal properties are encoded in a hidden Markov model (HMM) $\lambda = (A, B, \pi)$. The likelihood $P(F|\lambda_i)$ of a trajectory being a realization of activity λ_i can be computed using the forward-backward algorithm.[4]

Analysis

Using the automatically learned models, we can describe the current activity in a scene in real time.

- *Activity classification.* A trajectory is classified based on the most likely model to generate it:

$$\lambda^* = \underset{k}{\arg\max}\ p(\lambda_k | F)$$
$$= \underset{k}{\arg\max}\ p(F|\lambda_k)\, p(\lambda_k)$$

- *Path prediction.* Long-term predication is made based on expected activities. The predicted activity changes based on the amount of data available at the current time:

$$\hat{\lambda} = \underset{j}{\arg\max}\ p(\lambda_j | w_t \hat{F}_{t+k})$$

where w_t is a windowing function and $\hat{F}_{t+k}$ is the trajectory up to the current time t as well as k predicted future tracking states.

- *Abnormality detection.* An atypical trajectory is identified because it does not fit any learned model well. The detection sensitivity is controlled by an adjustable threshold L_{λ^*} that can be learned during training:

$$p(\lambda^* | \hat{F}) < L_{\lambda^*}$$

References

1. W. Hu et al., "A System for Learning Statistical Motion Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, 2006, pp. 1450–1464.
2. B.T. Morris and M.M. Trivedi, "Learning, Modeling, and Classification of Vehicle Track Patterns from Live Video," *IEEE Trans. Intelligent Transportation Systems*, vol. 9, no. 3, 2008, pp. 425–437.
3. B. Morris and M.M. Trivedi, "Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE CS Press, 2009, pp. 312–319.
4. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, 1989, pp. 257–286.

statistics module with hand-counted flow shows an error count of less than two vehicles over a 30-minute period (see Figure 8a). Longer-term comparison with Berkeley's Performance Measurement System (PeMS) shows strong correlation with loop detectors, the standard traffic management sensor (see Figure 8b). Notice the large flow disturbance at 18:00 that is closely tracked.

The HMM-based trajectory modeling procedure was able to accurately classify vehicles into the correct lane on either side of the Interstate 5 scene.

Table 2 shows the performance in each lane in either direction. The northbound direction has slightly lower performance because the lanes appear closer in the image due to projective distortion.

Table 3 displays further results for prediction and abnormality detection. The Cross experiment considered a traffic intersection similar to the scene shown in Figure 5 but viewing cars. The accuracy is lower in this situation because of more complex behaviors and also a larger number of activities, but it still achieves a high level of performance.

Future work will provide customized feedback to users. Feedback can be provided to users through infrastructure communication. The proliferation of GPS-enabled devices provides a new way of detecting people and vehicles on much larger scale. We can improve tracking by fusing GPS and visual tracks and using them for more advanced situational assessments. Connecting Canvas to other devices could allow, for example, warnings to be issued to a pedestrian's phone or notifications to drivers in potentially dangerous situations.

Table 1. Percentage accuracy for hourly test clips.

| Time	Sedan	Pickup	SUV	Van	Semi	Truck	Bike	Other	Total	No. of trajectories evaluated
06:21	94.9	59.5	81.9	31.6	50.0	33.3	0	97.5	81.5	405
07:19	96.2	32.5	83.2	06.7	66.7	25.0	100	98.0	84.7	497
08:17	61.2	33.3	91.9	14.3	50.0	50.0	100	96.7	97.6	530
09:15	53.2	38.7	82.3	53.9	37.5	23.1	100	96.8	63.7	444
10:13	36.8	26.7	77.3	26.7	71.4	40.0	0	93.9	51.0	357
11:11	63.4	47.2	90.4	28.0	66.7	33.3	–	89.6	68.6	417
12:09	86.0	71.7	82.6	48.0	50.0	37.5	100	96.9	80.0	432
13:08	95.6	76.3	83.5	39.1	100	50.0	–	97.9	87.0	393
14:06	96.9	77.8	84.2	18.2	–	66.7	100	94.9	86.2	449
15:04	96.0	76.4	81.9	23.1	100	09.1	100	100	85.4	492
16:02	97.1	66.2	76.0	24.0	100	55.6	100	100	85.7	553
17:00	99.1	65.5	62.0	03.6	–	0	100	94.5	83.0	630
17:45	89.0	75.9	52.8	10.0	–	100	67.0	97.7	76.0	297
18:45	96.0	57.9	73.9	10.5	–	100	50.0	97.9	84.6	382
19:43	95.0	77.8	78.5	0	100	100	–	100	86.5	222
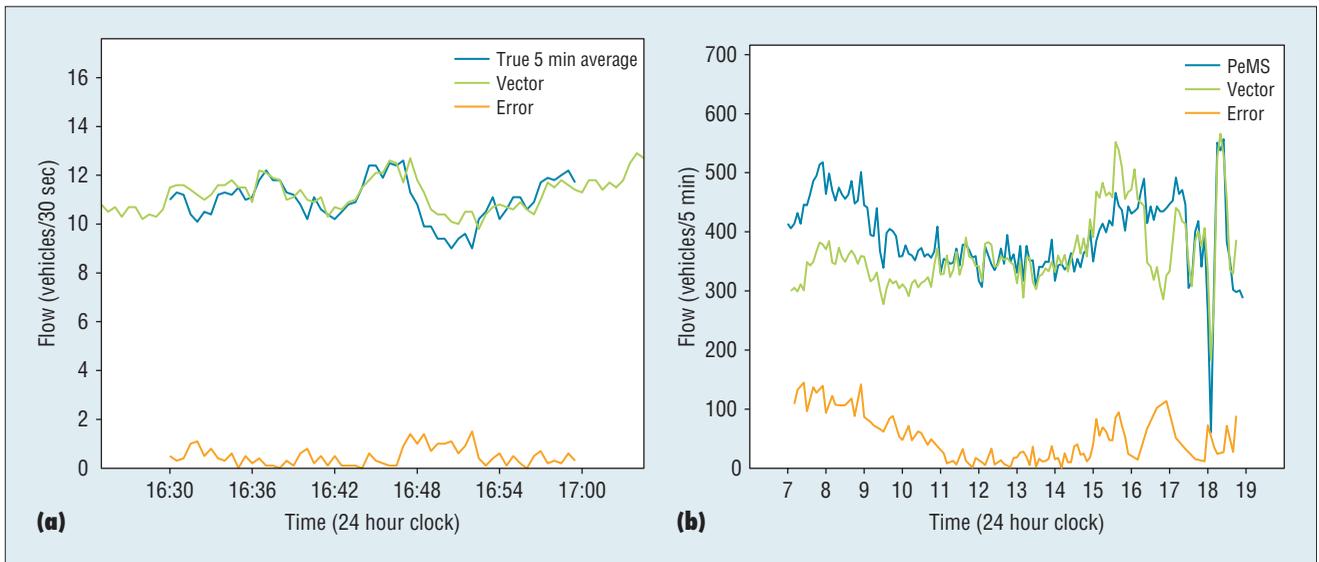
Figure 8. Evaluation results. (a) Comparison with true lane flow over 30 minutes. (b) Flow comparison with PeMS loop detector data.

Table 2. Interstate 5 lane classification performance.

Interstate direction	Lane 1 (%)	Lane 2 (%)	Lane 3 (%)	Lane 4 (%)	Total (%)
South	98.7	100	96.2	97.6	98.0
North	100	91.7	84.4	94.6	93.0

Table 3. Trajectory learning experimental results.

Experiment	N_{lanes}	Lane assignment		Abnormality detection		Lane assignment		Live prediction		Unusual event detection	
		R/N*	Accuracy (%)	R/N*	Accuracy (%)	R/N*	Accuracy (%)	R/N*	Accuracy (%)	R/N*	Accuracy (%)
Cross	19	9,191/ 9,500	96.7	168/ 200	84	35,197/ 41,871	84.1	35,077/ 41,871	83.8	830/ 999	83.1
Interstate 5	8	879/ 923	95.0	–	–	14,045/ 14,876	94.4	13,859/ 14,876	93.2	–	–

*R is the correctly numbered examples, and N is the total number of test examples.

THE AUTHORS

Brendan Tran Morris is a postdoctoral researcher with the Computer Vision and Robotics Research Laboratory at the University of California, San Diego. His research interests include intelligent surveillance systems, recognizing and understanding activities in video through machine learning, and in-vehicle behavior prediction for driver assistance and safety. Morris has a PhD in electrical and computer engineering from UCSD. Contact him at b1morris@ucsd.edu.

Mohan Manubhai Trivedi is a professor of electrical and computer engineering and the founding director of the Computer Vision and Robotics Research Laboratory at the University of California, San Diego. His research interests include machine and human perception, distributed video systems, multimodal affect and gesture analysis, human-centered interfaces, intelligent driver assistance, and transportation systems. He is a fellow of IEEE and the Society of Photo-Optical Instrumentation Engineers (SPIE). Contact him at mtrivedi@ucsd.edu.

References

1. H.M. Dee and S.A. Velastin, "How Close Are We to Solving the Problem of Automated Visual Surveillance?" *Machine Vision and Applications*, vol. 19, no. 5, 2008, pp. 329–343.
2. M.M. Trivedi, T.L. Gandhi, and K.S. Huang, "Distributed Interactive Video Arrays for Event Capture and Enhanced Situational Awareness," *IEEE Intelligent Systems*, vol. 20, no. 5, 2005, pp. 58–66.
3. "Traffic Monitoring Guide," Office of Highway Policy Information, US Dept. of Transportation, 2008, www.fhwa. dot.gov/ohim/tmguide/.
4. B.T. Morris and M.M. Trivedi, "Learning, Modeling, and Classification of Vehicle Track Patterns from Live Video," *IEEE Trans. Intelligent Transportation Systems*, vol. 9, no. 3, 2008, pp. 425–437.
5. B. Morris and M.M. Trivedi, "Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE CS Press, 2009, pp. 312–319.
6. T.B. Hall and M.M. Trivedi, "A Novel Interactivity Environment for Integrated Intelligent Transportation and Telematic Systems," *Proc. IEEE Conf. Intelligent Transportation Systems*, IEEE Intelligent Transportation Soc., 2002, pp. 396–401.
7. S.J. Landry, T.B. Sheridan, and Y.M. Yufik, "A Methodology for Studying Cognitive Groupings in a Target-Tracking Task," *IEEE Trans. Intelligent Transportation Systems*, vol. 2, no. 2, 2001, pp. 92–100.
8. M.A. Goodrich et al., "Supporting Wilderness Search and Rescue using a Camera-Equipped Mini UAV," *J. Field Robotics*, vol. 25, nos. 1–2, 2008, pp. 89–110.
9. A. Calbi, C.S. Regazzoni, and L. Marcenaro, "Dynamic Scene Reconstruction for Efficient Remote Surveillance," *Proc. IEEE Int'l Conf. Advanced Video and Signal-Based Surveillance*, IEEE CS Press, 2006, pp. 99–104.
10. J.L. Drury et al., *Comparing Situation Awareness for Two Unmanned Aerial Vehicle Human Interface Approaches*, tech. report 06-0692, Mitre, 2006.
11. Y. Ma et al., *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer, 2005.
12. G. Chockalingam, "California Wireless Traffic Report," 2009, http://traffic. calit2.net/.
13. E.-S. Ryu and C. Yoo, "Towards Building Large Scale Live Media Streaming Framework for a U-City," *Multimedia Tools and Applications*, vol. 37, no. 3, 2008, pp. 319–338.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*