Personal Driving Diary: Constructing a Video Archive of Everyday Driving Events

M. S. Ryoo, Jae-Yeong Lee, Ji Hoon Joung, Sunglok Choi, and Wonpil Yu
Electronics and Telecommunications Research Institute, Daejeon, Korea
{mryoo,jylee,jihoonj,sunglok,ywp}@etri.re.kr

Abstract

In this paper, we introduce the concept of personal driving diary. *A personal driving diary is a multimedia archive of a person's daily driving experience, describing important driving events of the user with annotated videos. This paper presents an automated system that constructs such multimedia diary by analyzing videos obtained from a vehicle-mounted camera. The proposed system recognizes important interactions between the driving vehicle and the others from videos (e.g. accident, overtaking, ...), and labels them together with its contextual knowledge on the vehicle (e.g. its physical location on the map) to construct an event log. A novel decision tree based activity recognizer that incrementally learns driving events from first-person view videos is designed. The constructed diary enables efficient searching and event-based browsing of video clips, which helps the user to retrieve videos of dangerous situations and analyze his/her driving habits statistically. Our experiment confirms that the proposed system reliably generates driving diaries by annotating learned vehicle events.*

1. Introduction

A *personal driving diary* is a multimedia archive of a person's daily driving experience. It illustrates important driving events of the user, providing recorded videos of the events and describing when and where the events have occurred. Figure 1 shows an example driving diary. The driving diary will not only enable interactive search of video segments with important vehicle events such as accidents, but also help the user to analyze his/her driving habits and patterns (e.g. dangerous overtaking and sudden stops) statistically for safer driving. The user will be able to retrieve and examine an event log (i.e. a diary) with videos taken from his/her vehicle, and use it for various purposes.

This paper presents an automated system that generates such multimedia diary by analyzing videos obtained from a vehicle-mounted camera. The objective is to construct a system that automatically annotates and summarizes obtained first-person viewpoint videos, enabling fast, efficient, and used-oriented browsing (and analysis) of driving events. The trend of mounting video cameras on vehicles is growing rapidly (e.g. 'black box cameras' for accident recording [13]), and most of vehicles will equip cameras observing the front in the near future corresponding to the societal interests. Our motivation is to provide a personal summary of vehicle events by utilizing such cameras, and develop an efficient way of searching important video segments. In this paper, we design and implement a novel system integrating various components including visual odometry, pedestrian detection, vehicle detection, tracking, and activity-level event recognition/logging. Several existing computer vision methodologies are combined with our newly designed activity recognition component, reliably generating video diaries for drivers.

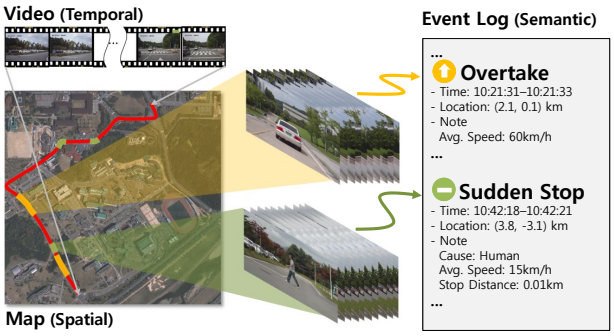


Figure 1. An example personal driving diary.

Notably, we designed our personal driving diary system to have an interactive learning property. Instead of limiting the system to only analyze predefined events, the proposed methodology enables interactive additions of user-specific events based on his/her necessity. That is, a user may add new events to be annotated in the future interactively without trying to retrain the entire system. Our system detects and labels interactively learned events, constructing a driving diary tailored for the user.

The contribution of this paper is the introduction of the concept of personal driving diaries. We present a novel paradigm that everyday driving experience of drivers can be annotated and archived, and discuss methodologies for the generation of event-based personal driving diaries from first-person view videos. The personal driving diary constructed by our system will enable efficient searching (and retrieval) of vehicle events. Even though there has been previous attempts to apply computer vision algorithms for vehicle-mounted cameras (e.g. [6]), a system to analyze vehicle activities (i.e. events) from them has not been studied in depth previously. Furthermore, we extend our previous event recognition methodology for incremental learning of novel events. Our event recognition methodology which enables capturing of personal statistics will benefit other types of life-logging systems as well.

2. Related works

Life-logging. Life-logging systems using wearable cameras have been developed to record a person's everyday experiences [8, 7, 4]. Hori and Aizawa [8] utilized multiple sensors (e.g. cameras, GPS, brain-wave analyzer, ...), automatically logging videos based on various keys from systems components such as a face detection and a GPS localization. Doherty et al. [4] also used a wearable camera. They have classified each image scene (i.e. frame) into a number of simple event categories using image features (e.g. SIFT), showing a potential that videos can be annotated based on user events.

However, most of previous life-logging systems focused on the elementary recording of entire video data [12], instead of constructing an interactive diary composed of videos of specific events. Previous systems attempted to construct general purpose achieves by relying on the index created by extracting simple image-based features, rather than performing a video-based analysis to interpret activity-level (i.e. complex) events. Furthermore, an ability to interactively add new event categories and videos has been very limited in previous life-logging systems.

Human activity recognition. Human activity recognition is a computer vision methodology essential for analyzing videos. Particularly, activity recognition methodologies utilizing spatio-temporal features from videos have obtained a large amount of interests [11, 5, 10]. However, even though previous systems successfully recognized events from videos with various settings (e.g. backgrounds), little attempts have been made to analyze activity videos from moving first-person view cameras. Furthermore, previous systems were designed to learn activities using off-line training, preventing interactive learning of complex events.

Vehicle cameras. As described in the introduction, increasing number of vehicles are equipping cameras for safety and accident recording purposes these days [13]. Various pedestrian detection algorithms have been developed and adopted for vehicle-mounted cameras [6], in order to support safer driving of drivers. However, most of the previous works limited themselves to accident prevention using simple per-frame detection, and did not attempt to analyze events from the videos.

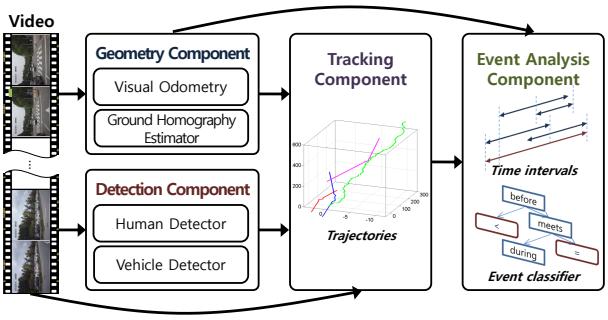


Figure 2. An overall architecture of our driving diary system.

3. Framework

In this section, we present an overall framework for our personal driving diary system. The idea is to provide a complete system architecture, so that an implemented system is installed on a mobile camera system (e.g. a black box camera or a smart phone) to annotate videos taken from a driving vehicle. Various computer vision techniques are designed and adopted to extract semantic information from first-person view videos containing vehicle events.

Our driving diary system is composed of four components: geometry component, detection component, tracking component, and event analysis component. These components obtain visual inputs (i.e. videos) from the camera and interact each other to analyze events involving the driving vehicle itself, other vehicles, and pedestrians. Figure 2 illustrates the overall architecture.

The geometry component uses a visual odometry algorithm to measure the self-motion of the camera. That is, the trajectory of the driving vehicle is obtained with respect to its initial global position, enabling our diary to record the vehicle's location on the map and provide an appropriate browsing interface. The detection component detects pedestrians and vehicles at every image frame of the input video. In addition, based on the geometrical structure of the scene analyzed by the geometry component, it estimates locations (i.e. bounding boxes) of the detected objects in global world coordinates. The tracking component applies object tracking algorithms to obtain trajectories of detected pedestrians and vehicles.

Finally, our event analysis component annotates all ongoing events from continuous streams of videos using the vehicle's self-trajectory from the geometry component and

the other trajectories from the tracking component. High-level events such as 'overtaking' and 'sudden stopping caused by pedestrians' are recognized hierarchically using trajectory-based features. Our event detection component allows interactive additions of new driving events. Events are annotated together with the driving vehicle's location and other contextual information.

As a result, our system converts an input driving video into a diary of semantically meaningful events. A user interface has been designed so that the user retrieves videos of interesting events from the diary. As discussed above, an interface to add new event to be annotated in the future is supported by our system as well. In the following section, we discuss each of the components in detail.

4. System

4.1. Geometry component

The geometry component localizes the driving vehicle and estimates planar homography of the ground. Visual odometry calculates relative pose between two adjacent images, which is accumulated for global localization [9] (Figure 4). Locally invariant features are extracted each frame, whose matching is performed using KLT optical flows. In addition, the geometric relation (i.e. an essential matrix) is estimated using a five-point algorithm with an adaptive RANSAC [2]. Estimating a ground plane using regular patterns on the ground (e.g. lane and crosswalk) enables global localization of other objects on it. Our geometry component computes a mapping from image coordinates to metric coordinates for objects.

4.2. Detection component

The detection component detects pedestrians and vehicles, and estimates their locations at every image frame (Figure 3). The estimated locations of the objects in image coordinates are transformed into global coordinates based on the information from the geometry component. We adopt histogram of gradients (HOG) features [3] and apply a sliding windows method to detect pedestrians. Furthermore, we implemented the sliding windows to be more efficient by filtering out windows with little vertical edges. We focused on the fact that a pedestrian is an upright person who is walking, and he/she produces a fair number of vertical edges. For the vehicle detection, we apply the Viola and Jones' method [15] to detect rear-view of appearing vehicles. Three types of vehicles (sedans, SUVs, and buses) are detected as a result.

4.3. Tracking component

Our tracking component maintains a single hypothesis for each object, and relies on color appearance model of the objects for the tracking. Results from the detection component are matched with the maintained object hypotheses using a greedy algorithm described in [16]. Similarity between each detection result and each object hypothesis is computed using its position, size, and color histogram. Next, these similarities are sorted to check for the valid match. For each of unmatched detections, a new hypothesis is created and a color template is built from the corresponding image region (i.e. bounding box) with an elliptic mask. Whenever the match fails, a template tracker is applied to update the unmatched object hypotheses. The color template is updated only when the hypothesis succeeds to match with a detection result. The actual trajectories are generated by applying extended Kalman filters (EKFs) with a constant velocity model in global world coordinates (e.g. Figure 4).

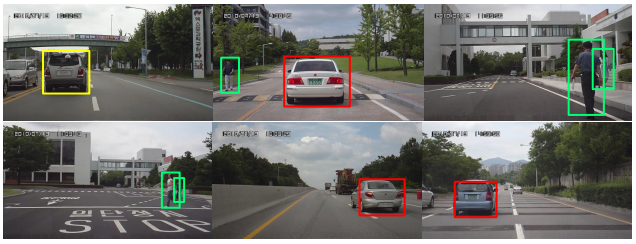


Figure 3. Example pedestrian/vehicle detection results obtained from our detection component.

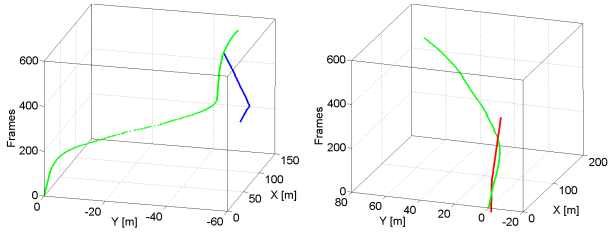


Figure 4. Example trajectories. Green trajectories show the driving vehicle's tracks estimated using visual odometry, blue is for a pedestrian, and red is for a vehicle. The left trajectories are from a 'sudden stop', and the right ones are from an 'overtaking'.

4.4. Event analysis component

The role of the event analysis component is to label all ongoing events of the vehicle given a continuous video sequence. In contrast to previous logging systems, we designed our event analysis component to recognize complex events learned interactively: We extended the previous approach of spatio-temporal relationship match (STR-match) [10] that obtained successful results on human activity recognition, so that events are learned and recognized in an additive fashion. Learned driving events are represented in terms of simpler sub-events, and they are recognized by hierarchically analyzing the relationships among the detected sub-events.

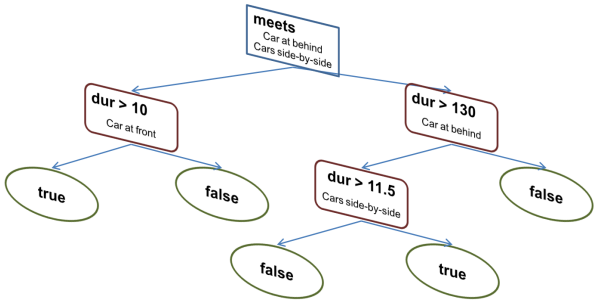


Figure 5. Example spatio-temporal relationship (STR) decision tree of a driving event 'overtake'. The left child of a node is activated when the relation corresponding to the node is true, and the right child is activated other wise.

First, nine types of elementary sub-events that consist complex vehicle events are recognized, which serve as building blocks of our hierarchical event detection process: 'car passing another', 'car passed by another', 'car is at front of another', 'car at behind of another', 'cars side-by-side', 'accelerating', 'decelerating', 'vehicle stopped', and 'pedestrian in front'. These sub-events are used as the system's vocabulary to describe complex driving events. The system recognizes the sub-events using four types of features extracted from local 3-D XYT trajectories of the driving vehicle and the other objects (i.e. pedestrians and vehicles): 'orientation', 'velocity', 'acceleration', and 'relative XY coordinate of the interacting vehicle'. Time intervals (i.e. pairs of starting time and ending time) of all occurring sub-events are recognized, and are provided to the system for the further analysis.

We implement a decision-tree version of the STR-match to recognize vehicle activities from detected sub-events, while making it possess an interactive learning ability. Our event analysis component learns decision-tree classifiers from training examples, automatically mining important spatio-temporal patterns (i.e. relationships) among sub-events. That is, we statistically train an event detector per activity, which will make the videos containing the corresponding event to reach a leaf node with the 'true' label when tested with the decision tree.

Our STR decision tree is a binary decision tree where each node of it corresponds to a predicate describing a condition of a particular sub-event (e.g. its duration *greater* than a certain threshold) or a relationship between two sub-events (e.g. a time interval of one sub-event must occur *during* the other's). Allen's temporal predicates [1] (*equals*, *before*, *meets*, *overlaps*, *during*, *starts*, and *finishes*) and their inverse predicates are adopted to describe relations between two sub-events. These predicates not only describe that certain sub-events must occur in order for the activity to occur, but also describe necessary temporal relations among the sub-events' time intervals.

The recognition is performed by traversing the tree from the root to one of its leaves, sequentially testings whether its sub-event detection results satisfy the predicate of each node. If it does, the recognition system traverses to the left child of the node. Otherwise, it must go to the right child. Figure 5 shows an example STR decision tree learned from training video sequences. The decision tree illustrates that in order for a driving event of 'overtaking' to occur, its sub-events 'car at behind of another car', 'cars side-by-side', and 'car at front of another car' must occur while satisfying a particular structure.

The decision trees are learned by iteratively searching for the predicate which maximizes the *gain* given the sub-event detection results of training sequences. The new node (i.e. the predicate) providing the maximum information gain is added to the tree one by one based on training examples. The gain of the decision tree caused by adding a new predicate to one of its leaves is defined as follows:

$$Gain(S, N) = Entropy(S) - \sum_{v} \frac{|S_v|}{|S|} Entropy(S_v) \tag{1}$$

where v is a binary variable, S is a set of training examples, and S_v is the subset of S having value v for node N. Here, the entropy is defined as:

$$Entropy(S) = -p_0 log_2(p_0) - p_1 log_2(p_1) \tag{2}$$

where p_0 is the fraction of negative examples in S, and p_1 is the fraction of positive examples in S. If S is divided into two sets of an identical size, the entropy is 1 and we have the gain of 0.

Essentially, our learning algorithm is searching for the predicate that divides the training examples into two sets whose size difference is the greatest (i.e. most unbalanced). Each of the left child and the right child of the added node either becomes a leaf node that decides that the driving event has occurred, or becomes an intermediate node waiting for another predicate to be added. A greedy search strategy is applied to find the STR decision tree that provides maximum gain given training videos.

In order to make our STR decision tree learning incremental (i.e. in order to enable interactive addition of user-specific events), we take advantage of the incremental tree induction (ITI) method [14]. The ITI method is incorporated into our STR tree learning process, which recursively updates the trees after each addition of a new video example to ensure the optimum gain. That is, our trees allow a user to feed videos of the new event to be annotated.

As a result, our system recognizes complex vehicle events (e.g. overtaking) incrementally learned from training videos. The personal driving diary is constructed by concatenating annotated driving events while describing other context including locations of the vehicle, vehicle tracking histories, and/or pedestrian tracking histories.

5. Experiments

In this section, we evaluate the accuracy of the personal driving diary generated by our system. Our driving diary is an event-based log of the user's driving history, implying that the correctness of the diary must statistically be evaluated by measuring the event annotation performance. For our experiment, we constructed a new dataset with driving video scenes taken from a first-person view camera attached to a vehicle (Subsect. 5.1). Using this dataset involving various types of driving events, we tested our system's ability to annotate time intervals of ongoing events (Subsect. 5.2).

5.1. Dataset

Our dataset focuses on six types of common driving events which are semantically important: long stopping, overtake, overtaken, sudden acceleration, sudden stop - pedestrian, and sudden stop - vehicle. A 'long stopping' describes the situation which the driving vehicle was staying stationary for more than 15 seconds. A 'sudden stop - pedestrian' indicates that the car was suddenly stopped because of the pedestrian ahead, and a 'sudden stop - vehicle' corresponds to an event of the car being stopped by another car in its front.

We have collected more than 100 minutes of driving videos from a vehicle-mounted camera. The camera was mounted under the rear-view mirror, observing the front. The dataset is segmented into 52 scenes, where each of them contains 0 to 3 events. As a result, a total of 60 event occurrences (i.e. 10 per event) has been captured by our dataset, and their time interval ground truths are provided.

5.2. Evaluation

We measured the event annotation accuracies of our system using a leave-one-out cross validation setting, similar to [5]: Among 60 event occurrences in our dataset, we select one event occurrence as test data and use the other 59 event occurrences as positive/negative training examples. This testing process is repeated for the 60 rounds, and the system performances have been averaged for these 60 rounds to provide the overall event annotation accuracy. In addition, a separate set of labeled pedestrian images and vehicle images were used for training the detection component.

For each round, the event analysis component takes advantage of the given training examples to learn the spatio-temporal decision tree classifiers. In order to test the incremental property of our learning, the training videos have been provided to the system sequentially. We measured whether the annotation was correct for the testing event occurrence, while counting the number of false positive annotations. In our experiment, an event annotation is said to be correct if and only if the overlap between the detected time interval and the ground truth interval overlaps more than 50%. Otherwise, it is treated as a false positive.

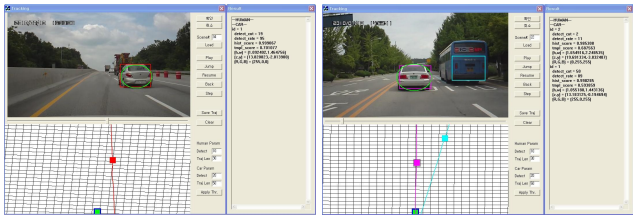


Figure 6. Video retrieval interface of our diary.

Table 1. Event annotation accuracies of the system.

Events	Accuracy	False positives
Long stopping	1.0	0.00
Overtake	0.9	0.00
Overtaken	0.9	0.00
Sudden acceleration	0.9	0.01
Sudden stop - pedestrian	0.8	0.02
Sudden stop - vehicle	1.0	0.01
Total	0.917	0.04

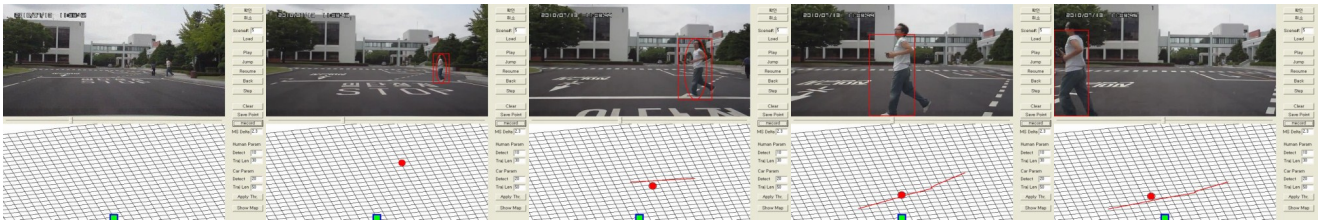
Table 1 shows the event detection accuracies of our system. 'Accuracy' describes the ratio of correctly annotated driving events among the testing events. 'False positives' shows the average number of false annotations generated per minute. We are able to observe that our system successfully annotates ongoing events in continuous video streams, reliably constructing appropriate personal driving diaries. In Figure 6, our system interface describing retrieved videos, locations of the vehicle on the map, and pedestrian/vehicle trajectories are illustrated. In addition, example videos of important driving events annotated using our system is shown in Figure 7.

6. Conclusion

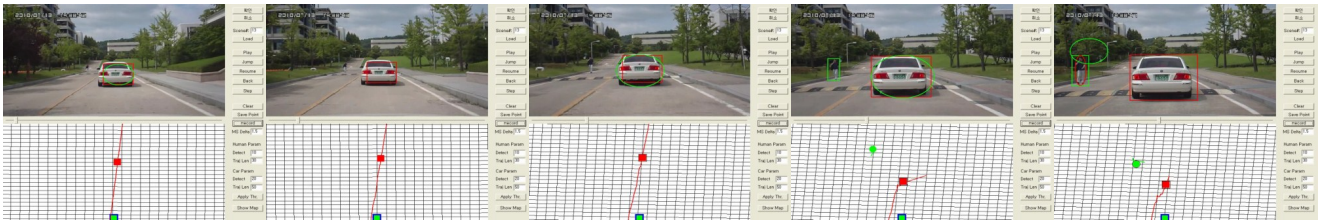
We introduced the concept of *personal driving diary*. We proposed a system that automatically constructs event-based annotations of driving videos, enabling efficient browsing and retrieval of users' driving experiences. The experimental results confirmed that our system reliably generates a multimedia achieve of driving events. Our driving diary enabled statistical analysis of users' driving habits based on vehicle events, and provided videos of important driving events, global locations of the vehicle, and trajectory histories of interacting pedestrians/vehicles.

Acknowledgments

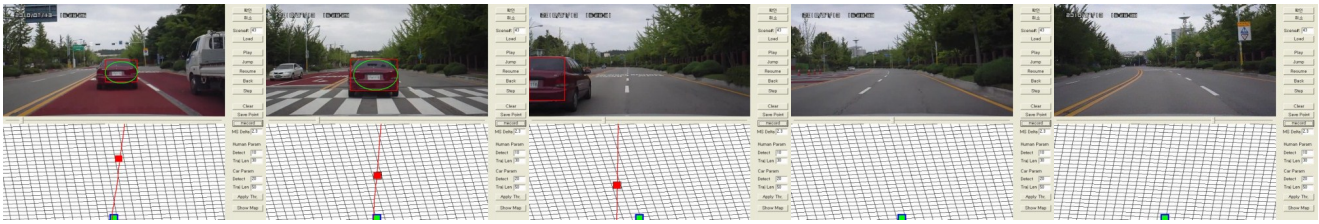
This work was supported partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT). [The Development of Low-cost Autonomous Navigation Systems for a Robot Vehicle in Urban Environment, 10035354]



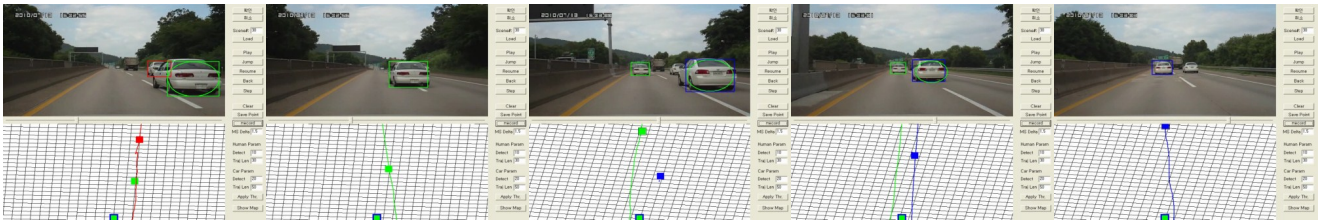
(a) A vehicle sudden stop event caused by a pedestrian



(b) A vehicle sudden stop event caused by another vehicle in the front



(c) A sequence of two other vehicles overtaking the driving vehicle



(d) A sequence of two vehicle overtaken event - two other vehicles are overtaking the driving vehicle

Figure 7. Example video sequences of annotated driving events. First-person view videos of various driving events are shown.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] S. Choi and W. Yu. Robust video stabilization to outlier motion using adaptive RANSAC. In *IROS*, 2009.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] A. R. Doherty, C. O. Conaire, M. Blighe, A. F. Smeaton, and N. E. O'Connor. Combining image descriptors to effectively retrieve events from visual lifelogs. In *ACM MIR*, 2008.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE Workshop on VS-PETS*, 2005.

[6] T. Gandhi and M. M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE T ITS*, Sept 2007.

[7] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell. Passive capture and ensuing issues for a personal lifetime store. In *ACM CARPE, in conjunction with ACM MM*, 2004.

[8] T. Hori and K. Aizawa. Context-based video retrieval system for the life-log applications. In *ACM MIR*, 2003.

[9] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR*, 2004.

[10] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[11] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, 2004.

[12] A. J. Sellen and S. Whittaker. Beyond total capture: A constructive critique of lifelogging. *Communications of the ACM*, 53(5):70–77, May 2010.

[13] US Patent 20040201697A1. "Black-box" video or still recorder for commercial and consumer vehicles, 2004.

[14] N. C. Utgoff, P. E. abd Berkman and J. A. Clouse. Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29:5–44, 1997.

[15] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[16] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, 2006.