Video Scene Analysis of Interactions between Humans and Vehicles Using Event Context

M. S. Ryoo[*]
Robot Research Department
ETRI
Daejeon, Korea
mryoo@etri.re.kr

Jong Taek Lee[*]
CVRC / Department of ECE
University of Texas at Austin
Austin, TX, U.S.A.
jongtaeklee@mail.utexas.edu

J. K. Aggarwal
CVRC / Department of ECE
University of Texas at Austin
Austin, TX, U.S.A.
aggarwaljk@mail.utexas.edu

ABSTRACT

We present a methodology to estimate a detailed state of a video scene involving multiple humans and vehicles. In order to automatically annotate and retrieve videos containing activities of humans and vehicles, the system must correctly identify their trajectories and relationships even in a complex dynamic environment. Our methodology constructs various joint 3-D models describing possible configurations of humans and vehicles in each image frame and performs maximum-a-posteriori tracking to obtain a sequence of scene states that matches the video. Reliable and view-independent scene state analysis is performed by taking advantage of *event context*. We focus on the fact that events occurring in a video must contextually coincide with scene states of humans and vehicles. Our experimental results verify that our system using event context is able to analyze and track 3-D scene states of complex human-vehicle interactions more reliably and accurately than previous systems.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*

General Terms

Algorithms, Experimentation, Performance

Keywords

Dynamic scene analysis, scene state tracking, event context

1. INTRODUCTION

Automated and continuous estimation and analysis of humans, objects, and their relative states has long been a goal of artificial intelligence, robotics, and computer vision. Particularly, in computer vision, detection and tracking of humans from closed-circuit television (CCTV) videos taken in various environments have widely been studied in the last several decades, proposing numerous promising approaches. Recent works [13, 8] have shown successful results on human detection and tracking in crowded environments.

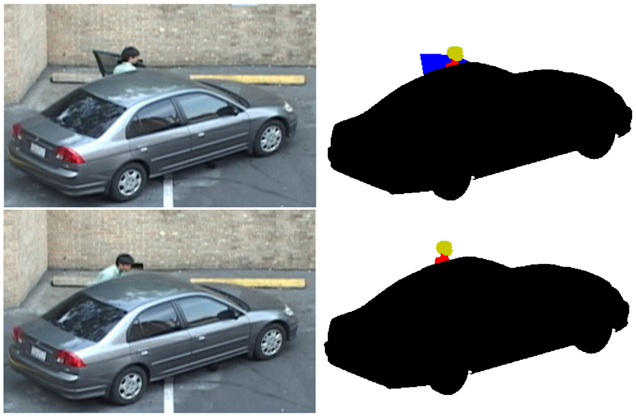


Figure 1: Example 3-D scene models of two different image frames. Detailed analysis of configurations of humans and vehicles using event context must be performed, in order to correctly distinguish states of two images with similar appearances.

However, tracking of humans itself is insufficient to analyze complex interactions between humans and other objects, vehicles for example. In order to annotate and retrieve videos containing activities of humans and vehicles, complex movements of humans/vehicles and their relationships in a dynamic environment (e.g. a crowded parking lot) must be analyzed. The system must be able to identify detailed 3-D states of all objects appearing in each frame. Such an analysis is particularly essential for many important applications including surveillance and military systems.

Our goal is to develop a methodology which is able to estimate the detailed state of a dynamic scene involving multiple humans and vehicles. The system must be able to track their configuration even when they are performing complex interactions with severe occlusion such as when four persons are exiting a car together. The motivation is to identify 3-D states of vehicles (e.g. status of doors) and their relations with persons, which is necessary to analyze complex human-vehicle interactions (e.g. breaking or stealing a vehicle). In addition, our methodology aims to identify the seat of each person entering the vehicle (e.g. driver's or passenger's), estimating his/her role and position even when

[*]These two authors contributed equally to the paper.

CIVR '10, July 5-7, Xi'an, China
Copyright ©2010 ACM 978-1-4503-0117-6/10/07 ...$10.00.

he/she is not visible. The challenges are derived from significant human-human occlusion and human-vehicle occlusion, which previous human tracking systems had difficulties handling. Figure 1 illustrates such difficulties.

Even though there have been previous attempts to process videos of humans and vehicles, they have focused on recognition of simple human-vehicle interactions. Instead of performing detailed scene analysis in a complex environment, they either assumed that the interactions are performed in simple environments which have no (or little) occlusion [4, 9], or assumed that human-manual corrections of tracking objects [5] are provided. The system proposed by Lee *et al.* [6] was able to perform view-independent recognition of a single person getting out of (or into) a vehicle, but it was limited in processing crowded human-vehicle interactions with two or more people. This is due to their inability to analyze states of scenes composed of multiple objects, failing to process complex events composed of several fundamental human-vehicle movements (e.g. 'door open,' 'person get in,' or 'person get out').

In order to achieve our goal, we design a probabilistic algorithm to track humans and analyze their dynamic relationships with vehicles using *event context*. We focus on the fact that many simple events can be detected based on low-level analysis, and these detected events must contextually coincide with human/vehicle state tracking results. That is, simple events (e.g. a person approaching a vehicle) detected during interactions can be used as key *features* (e.g. it may be a sign of the person opening the door next) for more robust tracking. Our approach updates the probability of a person (or a vehicle) having a particular state based on these basic events observed. The probabilistic inference is made for the tracking process to match event-based evidence. The event influences an interval of states, making a certain set of states more probabilistically favorable then the others for each time frame. For example, tracking a person occluded by a door is difficult without any contextual knowledge, but the detection of the event 'a person opening the door and going into the car' may help the system analyze his/her movements in these frames.

Our tracking problem is formulated as a Bayesian inference of finding the sequence of scene states with the maximum posterior probability. The scene state includes individual object states (humans and vehicles), object-object occlusions, and specific parameters of objects, e.g. door position and status. Our system estimates and tracks scene states frame-by-frame using Markov Chain Monte Carlo (MCMC), measuring the appearance similarity between hypothetical 3-D scene models and the observed image. The appearance of the scene state is described in terms of joint 3-D models and its projection is compared with the real image. In addition, as mentioned above, our probabilistic framework uses event-based cues to update the prior probability of object states, tracking highly occluded human-vehicle interactions (e.g. a person opening a door) reliably. In order to handle an event which is only detected 'after' its occurrence, we propose an algorithm to correct past frames by traversing past time frames.

The rest of the paper is organized as follows: Section 2 discusses previous works. Section 3 presents a detailed description of scene states. In Section 4, we formulate the tracking problem as a dynamic Bayesian inference. Section 5 describes our 3-D human/vehicle joint models to evaluate scene states. The concept of event context, which is required for a reliable scene analysis, is introduced here as well. We present how we use a MCMC algorithm to solve the formulated tracking problem in Section 6. Section 7 shows our experimental results. Section 8 concludes the paper.

2. PREVIOUS WORKS

Tracking. In previous tracking solutions following a Bayesian framework, trajectories of objects are modeled as a sequence of scene states describing the location of objects [11, 1, 3, 12, 13]. Zhao *et al.* [13] presented a model-based approach for segmentation and tracking of humans in crowded situations following a Bayesian framework. They computed prior probabilities and joint likelihoods using 3-D human models and calculated the posterior probability. Because of the enormous complexity of solution space, they used a data-driven MCMC for efficient sampling of the posterior probability to search for the optimal solution.

Ryoo and Aggarwal [8] presented the observe-and-explain paradigm for optimal tracking under severe occlusion. The limitation of most of the previous human tracking systems following the hypothesis-and-test paradigm [1, 13] is that they are required to maintain an exponentially growing number of hypotheses over frames if they do not apply pruning. Under severe occlusion, pruning can result in significant tracking performance reduction, and the system was able to overcome such limitations. However, the system only tracks humans without considering any other objects or their relations. The system was unable to analyze interactions between humans and vehicles.

Human-Vehicle Interaction. In addition, there have been several attempts to analyze interactions of humans and vehicles. Ivanov and Bobick [4] used stochastic context-free grammars to recognize human activities involving vehicles. Joo and Chellappa [5] recognized activities in a parking lot such as picking up and dropping off. Similarly, Tran and Davis [10] proposed an approach using Markov logic networks to recognize vehicle-related events for surveillance. However, the above-mentioned works have focused on the 'detection' of simple events rather than analyzing complex scenes with severe occlusion. Park and Trivedi [7] presented an approach to analyze moving-object interactions between humans and vehicles, but scenes they analyzed were limited to simple interactions with little occlusion as well.

Lee *et al.* [6] proposed a system to recognize human-vehicle interactions such as exiting and entering. A shape-based matching with a 3-D vehicle model is performed to detect a vehicle, and regions-of-interest(ROIs) are extracted from four door regions of the detected vehicle next. Under the assumption that the interactions occur in the ROIs, their system extracts motion and shape features in ROIs and analyzes them to classify interactions. However, since they did not consider spatial organizations (e.g. occlusion) between door ROIs, their system was unable to process interactions such as 'two persons coming out of the car from doors on the same side'. Furthermore, similar to [4, 5, 10], they did not attempt to analyze detailed scene configurations of objects. They did not take advantage of event context and were unable to analyze human-vehicle interactions in complex environments.

The main contribution of this paper is that our methodology makes a detailed 3-D scene analysis for videos of dynamic interactions between humans and vehicles. We propose a new view-independent scene state tracking methodology using joint 3-D models of humans and vehicles, designed for complex scenes with severe human-vehicle occlusion. We also believe that our paper is the first paper to take advantage of event context for analyzing and tracking such scenes.

3. DEFINITION OF SCENE STATES

In this section, we define the 'state', S, of each scene. A state is a complete description of objects' locations, their internal parameters, and relationships among them in each scene image. The level of detail in the scene state definition directly influences the system's level of understanding image frames, and is important for constructing a scene analysis system. Throughout this paper, our system interprets a video as an observation generated by a particular sequence of scene states (Figure 2), and searches for the sequence that best describes the dynamics of objects and their relationships in the video.

In many of the previous tracking paradigms (e.g. [11, 1]), each state is modeled as a set of independent objects (with particular parameters) present at each frame. Recently, the tracking paradigm has been extended to explicitly consider occlusion among humans [13, 8]. However, these previous systems only consider relative depth-ordering among humans, limiting themselves on analyzing detailed states of human-vehicle interactions such as "one car is parked in a parking lot, its front left door is fully opened, and a person is in the middle of getting out of the car through the door."

In our approach, we extend the definition of the scene state so that it can describe scene conditions more specifically. A scene state, S, is composed of the term C describing individual object states and the term R specifying object-object spatial relationships: $S = (C, R)$. The object state C is a set of c_ks, each describing the object class, tracking ID., and the class-specific parameters of the kth object: $c_k = (cls_k, id_k, param_k)$ where cls_k is the class of the object and id_k is its ID. Because there are two classes of objects (i.e. a human and a vehicle) and they have different object properties, the parameters for two classes ($param_k$) are defined differently. R is defined as a spatial relationship of all objects in C. R is composed of multiple rs, each describing the spatial relationship between two different objects (i.e. whether they are occluded, they are close to each other, or they have any spatial relationships): $R = \cup_{c_i \neq c_j} r_{(i,j)} = \cup_{c_i \neq c_j}(type, c_i, c_j)$. For example, $r_{(1,2)} = (occ, c_1, c_2)$ illustrates that the object c_2 is occluded by c_1. As a result, our scene state not only describes the locations of individual objects but also specifies their relative dynamics.

4. BAYESIAN FORMULATION

We formulate the tracking process of human-vehicle interactions as a Bayesian inference of computing the posterior probabilities of scene states:

$$S^{max}_{(1,2,...)} = argmax_{S_{(1,...,n)}} P(S_{(1,...,n)}|O_{(1,...,n)}),$$

where S_i is a scene state at frame i, O_i is an observation at frame i, and n is the number of frames observed. That is, we want to compute the optimum sequence of scene states that matches with the observations best. $P(S_{(1,...,n)}|O_{(1,...,n)})$ can further be enumerated as the multiplication of prior probability and image likelihood:

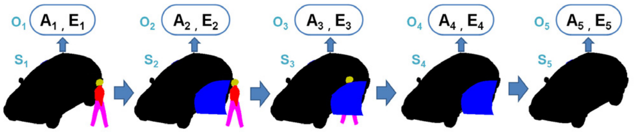


Figure 2: Example scene state transitions of 'a person entering a car'. Each S_i is a scene state, and (A_i, E_i) corresponds to an observed image frame. The goal of our system is to identify a sequence of states correctly describing the video.

$$P(S_{(1,...,n)}|O_{(1,...,n)}) = P(O_{(1,...,n)}|S_{(1,...,n)}) \cdot P(S_{(1,...,n)}).$$

For an efficient searching of the maximum-a-posteriori (MAP) of a scene state in all frames, $S^{max}_{(1,...,n)}$, we make a Markov assumption:

$$\begin{aligned} &P(O_{(1,...,n)}|S_{(1,...,n)}) \cdot P(S_{(1,...,n)}) \\ &\quad = P(O_1, ..., O_n|S_1, ..., S_n) \cdot P(S_1, ...S_n) \\ &\quad = P(O_n|S_n) \cdot P(O_1, ..., O_{n-1}|S_1, ..., S_{n-1}) \\ &\qquad \cdot P(S_n|S_{n-1}) \cdot P(S_1, ...S_{n-1}) \\ &\quad = P(O_n|S_n) \cdot P(S_n|S_{n-1}) \\ &\qquad \cdot P(O_{(1,...,n-1)}|S_{(1,...,n-1)}) \cdot P(S_{(1,...,n-1)}) \end{aligned} \quad (1)$$

Therefore,

$$\begin{aligned} &argmax_{S_{(1,...,n)}} P(S_{(1,...,n)}|O_{(1,...,n)}) \\ &\quad = \{argmax_{S_n} P(O_n|S_n) \cdot P(S_n|S_{n-1}), \\ &\qquad argmax_{S_{(1,...,n-1)}} P(S_{(1,...,n-1)}|O_{(1,...,n-1)}) \end{aligned} \quad (2)$$

From Equation (2), $P(O_n|S_n) \cdot P(S_n|S_{n-1})$ needs to be calculated to search MAP scene state in frame n. Intuitively, $P(O_n|S_n)$ is the likelihood between the observed image and the scene state at frame n, and $P(S_n|S_{n-1})$ describes the transition probability.

We further extend our Bayesian formulation to take advantage of 'event context' for reliable and detailed tracking of scene states. As mentioned in the previous sections, event detection results can be treated as an important feature that benefits the tracking process greatly. The state tracking problem must be formulated so that it takes into account the fact that occurrences of events must meet with the states of the scenes during the event. For example, if the event of the person getting out of the car is clearly occurring, then there is little possibility that the person was out of the scene during this event.

While an observation O corresponds only to an image appearance A in most of the previous systems, we extend the Bayesian tracking formula so that certain events between a vehicle and a human change the prior probabilities of objects. Therefore, observation O is defined to include both appearance A and event E.

$$\begin{aligned} P(O_n|S_n) \cdot P(S_n|S_{n-1}) &= P(A_n, E_n|S_n) \cdot P(S_n|S_{n-1}) \\ &= P(A_n|S_n) \cdot P(E_n|S_n) \cdot P(S_n|S_{n-1}) \\ &\propto P(A_n|S_n) \cdot P(S_n|E_n) \cdot P(S_n|S_{n-1}) \quad (3) \end{aligned}$$

That is, we assume $P(E_n)$ is uniformly distributed. In Equation (3), $P(A_n|S_n)$ represents the similarity between an in-

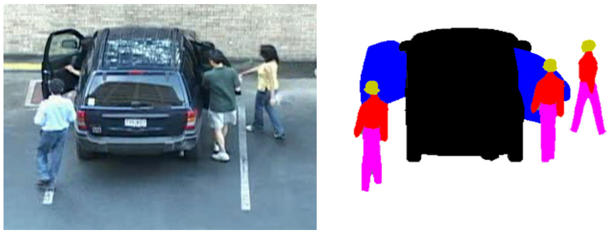


Figure 3: An example appearance of a projected 3-D scene state (right image) corresponding to an input image (left one). The 3-D scene model is constructed based on S_n, and is used for the appearance likelihood computation.

put image and an object model. $P(S_n|E_n)$ represents the prior probability of an image frame n in a particular state S_n, given an occurrence of an event E. If object states are assumed to be independent on events as in previous systems, $P(S_n|E_n)$ is the same as $P(S_n)$. $P(S_n|S_{n-1})$ shows the conditional probability of scene states in continuous two frames.

By solving the formulated Bayesian inference problem, we are able to estimate the most probable sequence of scene states. Each of the probability terms described in this section is modeled more explicitly in the following section.

5. PROBABILISTIC MODELING

In this section, we present the method to compute Bayesian probability 'given' each scene state and image frame (e.g. Figure 3). We present a 3-D scene (human and vehicle) model which is used for calculating appearance likelihood, and introduce our 'event context' that influences states' prior probabilities for contextual inference. The methodology to search for the optimal scene state based on these models will be discussed in Section 6.

5.1 Appearance Likelihood, $P(A_n|S_n)$

Our comprehensive definition of a scene state enables the system to construct a virtual appearance of the scene given its state. We use a 3-D model of a human (or a vehicle) to represent an appearance of each individual object c_k. The motivation is to estimate the optimal appearance of an individual object c_k in the scene as a 2-D projection of its 3-D model, so that it can be compared with the real image to measure the appearance likelihood. Furthermore, the appearance of multiple overlapped objects are modeled by considering the spatial relationship of the objects R. Figure 3 shows an example 2-D projection of a 3-D scene model consists of several 3-D human and vehicle models. We take advantage of such appearance model to compare it with a real image to measure the state likelihood. The camera parameters for the projection are assumed to be known.

5.1.1 3-D Vehicle Model

Our system assigns a 3-D model for each vehicle appearing in the scene. Based on the parameters of the vehicle state, a snapshot of the 3-D vehicle model is computed at each frame to obtain its virtual appearance. A vehicle is described with the following parameters: (x, y, *size*, *orient*, *tilt*, *type*, *door*). x and y are the center xy-coordinates of the vehicle, *size* is the resize factor of an 3-D template image, *orient* is the orientation of the vehicle, *tilt* is the tilt angle of the vehicle, *type* is the type of the vehicle (e.g. sedan and sport utility vehicle), and *door* is the parameters of all doors to describe how far the doors are open (closed, partially opened, and fully opened). The orientation and tilt angle of a vehicle are quantized and sampled for 5 degrees. Sample 2-D projection images of a 3-D vehicle model with an opened door are shown Figure 4(a).

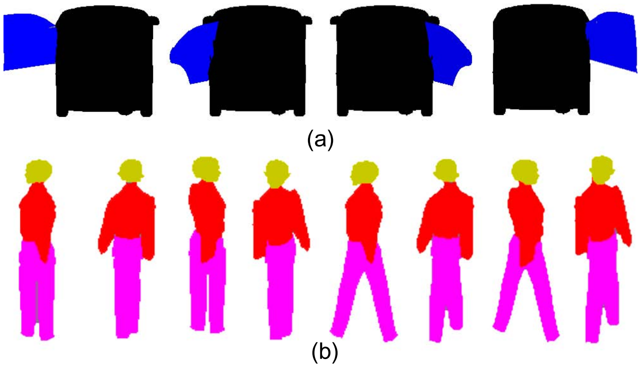


Figure 4: (a) 2-D projections of 3-D vehicle models representing door opening states. (b) 2-D projections of 3-D human models. The left four images are from a standing model and the right four images are from a walking model.

5.1.2 3-D Human Model

Similar to our 3-D vehicle model, a 3-D model is assigned per person in the scene. A human is described with the following parameters: (x, y, *size*, *orient*, *tilt*, *type*, *color_histogram*, *velocity*). x, y, *size*, *orient*, and *tilt* of a human are defined similar to those of a vehicle. Two *types* of human models are used: walking and standing. In addition to the 3-D human shape model, a color histogram is used to detect and distinguish human objects [13] in order to handle non-rigid human appearances. For human objects, we calculate *color_histogram* on three regions of humans such as a head, an upper body, and a lower body. The *velocity* is also calculated for tracked human objects to be applied in Kalman filtering. The orientation and tilt angle of a human are digitized and sampled for 90 degrees and 5 degrees, respectively. Each 3-D human model at a frame is generated based on these parameters. Sample 3-D human models of two types are presented in Figure 4(b).

5.1.3 Human-Vehicle Joint Model

A human-vehicle joint model is constructed per scene by considering the spatial relationship (e.g. occlusion) R of humans and vehicles. We construct a complete 3-D scene model composed of multiple 3-D object models, so that its 2-D projection may be compared with the real image. A 3-D scene model essentially is a set of 3-D human and vehicle models whose relative spatial relationships are described with R.

The process to obtain a projection of a joint scene model (given a particular scene state) is as follows: 1) Build a blank canvas whose size is the same as the real image for representing a scene model. 2) Choose object c_k which does not occlude any non-chosen object, based on R. 3) Draw

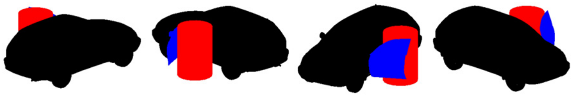


Figure 5: Example occlusion types generated based on the simulation. Representative occlusion types describing relationships among human, door, and vehicle body are presented.

the 2-D projection of the object c_k. 4) Repeat 2) and 3) until all objects are drawn. That is, we are essentially drawing all objects into a blank image in a particular order so that an occluded object is drawn before the object occluding it. Drawing each object can be done using the 3-D human/vehicle individual models. Note that spatial relationship R specifies which object is occluded by which, enabling the overall joint model projection process.

To construct a complete projection of a 3-D scene model, object-object spatial relationship (R) should be calculated. The spatial relationship between humans or between vehicles can be obtained based on xy-coordinates of the objects. Based on the following criteria, we build R for each S_n using its C value. The two criteria for deciding relations of human-human occlusion and vehicle-vehicle occlusion are: 1) If the feet of person c^p_{k1} are located under the feet of person c^p_{k2} and two people are overlapped in an image, person c^p_{k1} occludes person c^p_{k2}. 2) If the center of vehicle c^v_{k1} is under the center of vehicle c^v_{k2}, vehicle c^v_{k1} occludes vehicle c^v_{k2}. Human-vehicle occlusion is more complex to process compared to the other two types of occlusion, due to the existence of doors. A relation between an overlapped human and vehicle (i.e. which is occluding which) is estimated by comparing C with several simulated occlusion types. As shown in Figure 5, we construct several representative occlusion types with a rough simulation, and compare which occlusion type matches the given C of the scene S_n best. The depth order of the best matching occlusion type is chosen to be the relation between the human and the vehicle.

5.1.4 Joint Image Likelihood

Here, we present how we actually compute the appearance likelihood based on the projection of the joint model described above. We compare the expected appearance (i.e. 2-D projection) generated from the 3-D scene model with a real image. We measure the distance between the image and the model for each object c_k, and sum them to compute the state-image distance. That is, assuming conditional independence among appearances of non-occluded object regions given the 3-D scene model, we can calculate $P(A_n|S_n)$ as $\prod_{c_k} P(A_n|M(c_k))$, where $M(c_k)$ is a non-occluded region of object c_k obtained in Section 5.1.3. $P(A_n|M(c_k))$ can be measured by calculating the ratio of the number of foreground pixels of $M(c_k)$ to the number of foreground pixels on the region $(P(FL_k\ |M(c_k))$ and pixel-wise color distances $(P(CL_k\ |M(c_k)))$. Thus, $P(A_n|S_n)$ can be calculated as shown in Equation (4).

$$\begin{aligned} P(A_n|S_n) &= \prod_{c_k} P(A_n|(c_k,R)) = \prod_{c_k} P(A_n|M(c_k)) \\ &= \prod_{c_k} \{P(FL_k\ |M(c_k)) \cdot P(CL_k\ |M(c_k))\} \end{aligned} \quad (4)$$

5.2 Dynamics Likelihoods, $P(S_n|E_n) \cdot P(S_n|S_{n-1})$

In this subsection, we model two probability terms that influence the posterior probability, $P(S_n|E_n)$ and $P(S_n|S_{n-1})$. Intuitively, the former corresponds to the probability of the 'event context' supporting the states, and the latter specifies the influence of the previous frame state to the current state. We discuss how we model each of these terms describing scene dynamics.

5.2.1 Event Context, $P(S_n|E_n)$

As we have formulated in Section 4, the probability of the scene in a particular state S_n is highly dependent on its event context. The occurrence of an event at a particular time interval (i.e. a pair of a starting time and an ending time) suggests that the states within the interval must follow a particular distribution; the state sequence must contextually agree with the event. Here, we model such probabilistic distribution of the interval's states for each event class (i.e. type). The goal is to assign scene states that match event detection results with higher probability values.

Let a pair (t_s, t_e) be a time interval of an event e. Then, we model the distribution $P(S_n|E_n = e)$ for all states of $t_s < n < t_e$ to have a distribution learned from training examples of the event e. Similar to the case of appearance likelihood computation, we assume conditional independence among objects in the scene as follows:

$$P(S_n|E_n = e) = \prod_{c_i} P(c_i|E_n = e) \cdot \prod_{c_j} P(c_j|E_n = null) \quad (5)$$

where c_i are the objects involved in the event e, and c_j are the other objects. We assume that the event time intervals do not overlap, meaning that there's only one (or no) event going on at a particular time frame.

We model each $P(c_i|E_n = e)$ based on training data. We assume that all states within the event's interval show an identical probability distribution, ignoring their temporal order. Given a set of example state sequences corresponding to the event intervals, $P(c_i|E_n = e)$ is learned by considering all observed ground truth states to be sampled from the same distribution. More specifically, we model $P(c_i|E_n = e)$ to have a 3-dimensional distribution where the first dimension specifies whether the object c_i is in the scene and the other two dimensions specify the relative XY-coordinates of the object. As a result, the system makes certain spatial locations more probabilistically preferable than others for the object during the event interval. Our event context has an effect of narrowing down the state search space, making the scene state tracking process more efficient and reliable.

In principle, our proposed methodology is able to cope with any number of events as long as their state distributions can be learned. However, in this paper, we have chosen the three events which most effectively benefits the scene tracking process for computational efficiency. The defined events are 1) a person gets out of a vehicle, 2) a person approaches and opens a door of a vehicle, and 3) a person is sitting inside a car. For example, in the case of the third event, the distribution of the locations of the person c_k during the event's time interval will be modeled to be centered at the seat. Thus, our event context consideration process will update all $P(S_n|E_n)$ within the interval so that it penalizes the states representing the location of the c_k to be somewhere else. All of this is done by learning the distributions based on training examples.

We discuss more about how we actually detect events' time intervals and take advantage of them in Section 6.

5.2.2 *Previous State,* $P(S_n|S_{n-1})$

The term $P(S_n|S_{n-1})$ describes the probability of the objects (i.e. humans and vehicles) in a certain scene state S_n, given their state at the previous frame $n-1$. Our system's consideration on the previous state is done in a straight forward fashion. Similar to previous tracking algorithms [13, 8], our system assumes linear movements of objects. Based on the XY velocity of the object, the distribution of $P(S_n|S_{n-1})$ is modeled to have a Gaussian distribution centered at the expected location using the previous state.

6. MAP SEARCHING BY MCMC

In this section, we present an algorithm to search the scene state S_n^{max} providing the highest posterior probability at time frame n. What we presented in Section 5 is a method to compute the posterior probability of each scene state S_n, and we now search for the optimum state among them. A trivial approach is to perform brute force searching. However, the high dimensionality of our solution space requires a fast maximum-a-posteriori (MAP) searching algorithm. Markov Chain Monte Carlo (MCMC) has been widely used in complex tracking systems for efficient MAP searching. We apply the following three procedures to search MAP.

6.1 Markov Chain Monte Carlo Dynamics

Our MCMC algorithm searches for the best scene state at each frame. It randomly applies one of the predefined moves to S_n, iteratively updating the S_n for hundreds of rounds while searching for the one with the highest probability. We have adopted a Metropolis-Hastings algorithm with reversible jumps [2]. At each iteration, our Metropolis-Hastings algorithm applies a randomly selected move to an individual object state c_k of S_n to obtain S', which will either be discarded or accepted as the new S_n. The initial value for S_n is set to be S_{n-1}, and is iteratively updated. The prior probability of selecting a human as c_k to update is 0.9 and that of selecting a vehicle is 0.1. The list of MCMC sampling moves are as follows:

1. **Object addition hypothesis.** Randomly select a vehicle or person to be added in the scene. All parameters of an object are randomly chosen from prior object parameter distribution, except for the position (x, y). The center position of an object will be randomly located on the foreground pixels.

2. **Object update hypothesis.** Change parameters of objects based on their prior probability distributions. For human objects, the values of x, y, *size*, *type*, and *orient* are updated. The other parameters are automatically calculated using the knowledge of the camera model and the ground plane. For vehicle objects, the values of x, y, *size*, *orient*, *type*, and *door* are updated as well.

3. **Object removal hypothesis.** Randomly select a vehicle or a person to be removed from the scene.

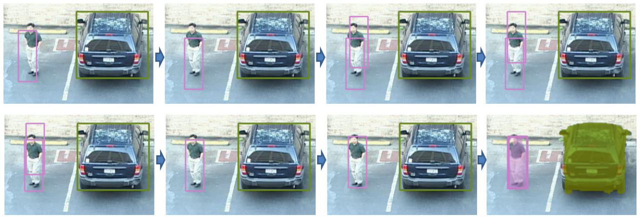


Figure 6: Example candidate scene states, S', obtained during our MCMC iteration. Various MCMC sampling moves have been sequentially applied to search for an optimal scene state, S_n^{max}.

At every iteration of the dynamics, the system updates object-object spatial relationship (R) from the updated individual object states (C). Therefore, the system can obtain a new scene state (S') and calculate $P(O_n|S') \cdot P(S'|S_{n-1})$. We accept the scene state S' for S_n if the $P(O_n|S') \cdot P(S'|S_{n-1})$ is larger than $P(O_n|S_n) \cdot P(S_n|S_{n-1})$. The experimental results are obtained after 200 iterations. Figure 6 shows an example iteration of our MCMC process.

6.2 Event Detection

In order to search for the scene state providing the maximum posterior probability, events occurring during human-vehicle interactions must be detected. The detected events will enable the computation of the dynamics likelihood probability using event context (i.e. Subsection 5.2), making our system able to track detailed scene states. In principle, any of the existing activity recognition methodologies can be adopted for the detection of events. In our implementation, events are recognized using a rule-based elementary detector with a simple criterion; our elementary detector is activated (i.e. it detects an event) by checking whether the previous state S_{n-1} satisfies the encoded rules of the event. That is, we say that the event is occurring if the rules are satisfied and use this information as an event context to compute the state probabilities.

Note that the detector is activated at a particular time point, instead of fully providing events' intervals. In general, the detector is activated either at a starting time or an ending time of the event depending on its characteristics. No exact time interval is provided, and most events are detected 'after' the event has occurred. This implies that the probability computation using the event context presented in Subsection 5.2 is difficult in a standard forward inference process. It is not capable of recalculating the past states even if the system later finds that an event has occurred in the past frames. This situation occurs commonly for the detectors which are difficult to compute exact occurring time intervals (e.g. traditional hidden Markov models), and hence we present a forward/backward probability updating process in the following subsection. The motivation is to dynamically update future (or past) frames that are expected to be within the time interval until the event conditions are violated.

The detailed detection criteria of our three events, "a person getting out of a car," "a person approaching and opening a door of a vehicle," and "a person sitting inside a car" are as follows:

1. **A person getting out of a car.** The event of "a

person getting out of a car" is detected at time t_e, which is the ending frame of the event's time interval. The detection rules are 1) a new person appears near a door d and 2) the door d is open. That is, we assume that the new person came out from the door.

2. **A person approaching and opening a door of a vehicle.** The event of "a person approaching and opening a door of a vehicle" is detected at time t_s, which is the starting time frame of the event's time interval. The detection rules are 1) a person from outside the scene boundary approaches a door d (i.e. their distance becomes small) and 2) the door d was closed at t_s. The event continues until the person disappears or the distance between the person and the vehicle becomes larger than a threshold.

3. **A person sitting inside a car.** The event of "a person sitting inside a car" is detected at frame t_s (i.e. starting time), when the following conditions are satisfied: 1) a person c_k disappears near a door d at frame t_s and 2) the door d was opened at frame t_s. The event continues until the person reappears from the door.

6.3 Updates with Backward Tracking

As mentioned in the previous subsection, many events tend to be detected 'afterwards', making the MCMC-based MAP state computation with event context difficult. What we present in this subsection is a methodology to support our event context-based scene state tracking by compensating for such late detections using a backward re-tracking process.

We say that an event has a forward characteristic if it is detected at its starting time, and has a backward characteristic if it is detected at its ending time. Basically, unless an event having a backward characteristic occurs, our system progresses the computation of MAP states in a forward direction using the MCMC-based algorithm presented in Subsection 6.1. This process is similar to hidden Markov models or other sequential state models. The system assumes that no event is going on, if no forward event has been detected (it may later correct it if an event with a backward character is detected afterwards). If a forward event e is detected at frame t_s, the system records that the event is starting to occur from the frame t_s and considers the event context for each frame n such that $t_s < n$. This event context consideration (i.e. $E_n = e$) is applied for future frames, as long as the conditions of the event are satisfied, influencing $P(S_n|E_n)$.

The backward probability update process is described as follows. Once a backward event is detected, our system initiates the tracking process in the backward direction, starting from the frame t_e where the event is detected. That is, we update (or re-estimate) the scene states of frame n such that $n < t_e$. Leaving non-related objects c_js unchanged, the system recalculates $P(c_i|E_n = e)$ for event related objects c_is at frame n and recomputes $P(S_n|O_n)$ to search for the MAP state. This backward traversal process is continued until the event conditions are violated. For example, in the case of the event 'a person getting out of a vehicle', the backward traversal is continued until the person disappears in the backward tracking process (i.e. until the system reaches the frame where he/she comes out of the vehicle for the first time). Figure 7 shows an example backward tracking process. For computational efficiency, we concatenate the backward process for a certain amount of frames (i.e. delays the initiation of the backward tracking mode), so that the backward updates can be done at once without having a duplicate update process.

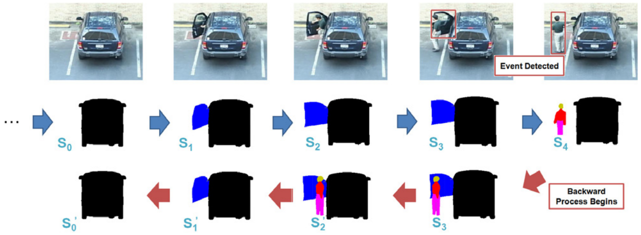


Figure 7: An example backward tracking process initiated by the event 'a person exiting a car'. The event triggers the backward tracking, successfully correcting previous scene states to contextually agree with the event.

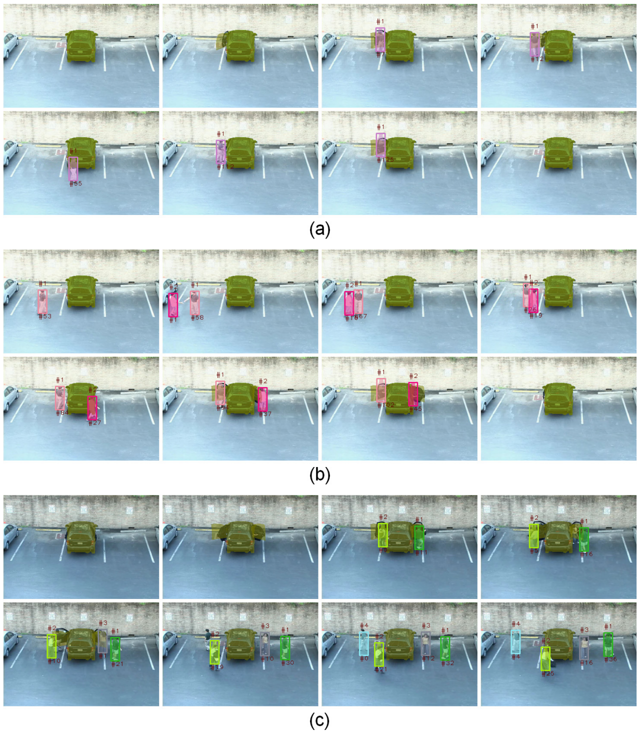
7. EXPERIMENTAL RESULTS

We tested the system's ability to track scene states from videos of humans interacting with a vehicle. We generated a dataset of 20 video sequences for our experiments. Each video sequence includes one vehicle and one to four interacting persons. Each person either enters into or gets out of the car (or both) in a video at least once. The videos were filmed at 12.5 frames per second with the resolution of 360 by 240 pixels. Five different actors participated in the experiment, and a total of 2535 frames have been collected.

In each sequence, an actor interacts with a vehicle at least once and at most twice. In the first 12 sequences, each actor appearing in the scene (note that there can be 1 to 4 actors) performs both 'entering' and 'exiting' interactions. In the other 8 sequences, only one interaction is performed per actor. Among 20 sequences, 6 videos were taken with a single actor, another 6 videos contain two actors, and the other 8 videos were taken with four actors. As a result, a total of 36 entering and 35 exiting interactions are performed. The videos with four actors are particularly challenging, since multiple persons participate in the interaction with the vehicle body and doors, occluding each other as we can observe from Figure 8(c).

We have measured the tracking accuracies of all persons appearing in the videos. For each person, the system estimates his/her trajectory using our approach and compares it with its ground truth trajectory. The tracking process at each time frame is said to be correct if the tracked bounding box of the person overlaps more than 75% of the ground truth bounding box. For each estimated trajectory, we find the longest interval in which the object is correctly tracked. We define the tracking accuracy as the length of this longest interval divided by the length of the entire ground truth trajectory. The tracking accuracies of persons are averaged to measure the mean accuracy of our system.

We have compared our system with a baseline system similar to [13], which considers occlusion among persons and uses MCMC to solve the tracking problem. This system does not take advantage of a 3-D vehicle model or event context, and tracks objects purely in terms of human models. The objective of this implementation is to compare our system with others to confirm the advantage of our new sys-


(a)

(b)

(c)

Figure 8: An example of tracking results on humans interacting with a vehicle in various environments: (a) one person exits and enters a car, (b) two people enter a car, and (c) four people exit a car.

Table 1: Composite interaction recognition results

Scene condition	Avg. Tracking Accuracy		Number of Frames
	Previous system	Our system	
1 person	85.4 %	92.0 %	852
2 persons	85.2 %	93.3 %	788
4 persons	67.5 %	81.5 %	895
Total	**79.1 %**	**88.7%**	2535

tem using event context.

Table 1 shows the overall tracking accuracies of the two systems. Our approach clearly outperforms the baseline. The previous method performed particularly worse for videos with four persons. This is due to its inability to analyze detailed scene states with severe human-vehicle occlusion. We are able to observe that the use of 3-D scene state models and event context benefits the system greatly. The tracking accuracy of one-person scenes and that of two-person scenes are observed to be similar. This result is because of the fact that the occlusions in two-person scenes are not severe: each of them usually gets in or out of the car from a different direction. Therefore, the difficulty of tracking humans in one-person scenes was similar to the one in two-person scenes.

Figure 8 shows example tracking results of human-vehicle interaction videos. Actors appearing in the videos are tracked very accurately by our improved tracking system. Tracking of one person in Figure 8(c) failed at the beginning of his appearance, but the system was able to recover quickly.

8. CONCLUSIONS

We have presented a methodology for analyzing a sequence of scene states from videos of human-vehicle interactions. We developed a probabilistic framework for scene state tracking using 3-D scene models, identifying detailed configurations of humans and vehicles appearing in videos. Furthermore, we introduced the concept of event context which benefits the scene state analysis process greatly. Interplays between event detection and state tracking are explored probabilistically, providing better results in the experiments.

Ack.: This work was supported partly by the Texas Higher Education Coordinating Board under award no. 003658-0140-2007, and partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT) [10035354 The Development of Autonomous Navigation Systems for a Robot Vehicle in Urban Environment].

9. REFERENCES

[1] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE T PAMI*, 18(2):138–150, 1996.

[2] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE T PAMI*, 22(8):809–830, 2000.

[4] Y. Ivanov, C. Stauffer, A. Bobick, and W. Grimson. Video surveillance of interactions. *IEEE Workshop on Visual Surveillance*, 0:82, 1999.

[5] S. W. Joo and R. Chellappa. Attribute grammar-based event recognition and anomaly detection. In *CVPRW*, 2006.

[6] J. T. Lee, M. S. Ryoo, and J. K. Aggarwal. View independent recognition of human-vehicle interactions using 3-d models. In *IEEE Workshop on Motion and Video Computing (WMVC)*, 2000.

[7] S. Park and M. M. Trivedi. Analysis and query of person-vehicle interactions in homography domain. In *ACM International Workshop on Video Surveillance and Sensor Networks (VSSN)*, pages 101–110, New York, NY, USA, 2006. ACM.

[8] M. S. Ryoo and J. K. Aggarwal. Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. In *CVPR*, June 2008.

[9] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 2:2246, 1999.

[10] S. D. Tran and L. S. Davis. Event modeling and recognition using markov logic networks. In *ECCV*, pages 610–623, 2008.

[11] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

[12] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE T PAMI*, 26(9):1208–1221, 2004.

[13] T. Zhao, R. Nevatia, and B. Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE T PAMI*, 30(7):1198–1211, 2008.