Why Did the Person Cross the Road (There)? Scene Understanding Using Probabilistic Logic Models and Common Sense Reasoning

Aniruddha Kembhavi, Tom Yeh, and Larry S. Davis

University of Maryland, College Park
anikem@umd.edu, tomyeh@umiacs.umd.edu, lsd@cs.umd.edu

Abstract. We develop a video understanding system for scene elements, such as bus stops, crosswalks, and intersections, that are characterized more by qualitative activities and geometry than by intrinsic appearance. The domain models for scene elements are not learned from a corpus of video, but instead, naturally elicited by humans, and represented as probabilistic logic rules within a Markov Logic Network framework. Human elicited models, however, represent object interactions as they occur in the 3D world rather than describing their appearance projection in some specific 2D image plane. We bridge this gap by recovering qualitative scene geometry to analyze object interactions in the 3D world and then reasoning about scene geometry, occlusions and common sense domain knowledge using a set of meta-rules. The effectiveness of this approach is demonstrated on a set of videos of public spaces.

Keywords: Scene Understanding, Markov Logic Networks.

1 Introduction

We build on recent research in appearance-based object recognition and tracking [1,2,3,4], recovery of qualitative scene geometry from images and video [5,6,7], and probabilistic relational models for integrating common sense domain models with uncertain image analysis [8], to develop a video understanding system that can identify scene elements (cross walks, bus stops, traffic intersections), characterized more by qualitative geometry and activity than by intrinsic appearance. The domain models we use are naturally specified by humans, and characterize scene elements in terms of geometric relationships (sidewalks are found along roads and are parallel to roads) and activity relationships (people walk on sidewalks, wait and possibly queue for a bus).

These domain models are related to image analysis (appearance, tracking, motion) by representing them as probabilistic logical models (Markov Logic Networks). These logical models, however, describe *what typically happens* in the scene and not *what is visible* in some video of that scene. We bridge this gap using two methods. First, we recover qualitative scene geometry to analyze object interactions in the 3D world rather than the 2D image plane. Second,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 693–706, 2010.

we utilize a set of meta-rules that capture general rules about scene geometry and occlusion reasoning and fuse them with common sense domain knowledge to detect these scene elements in videos taken from arbitrary viewpoints. This involves reasoning about unobserved events and inferring their occurrence based on other observations. Figure 1 provides an overview of our system.

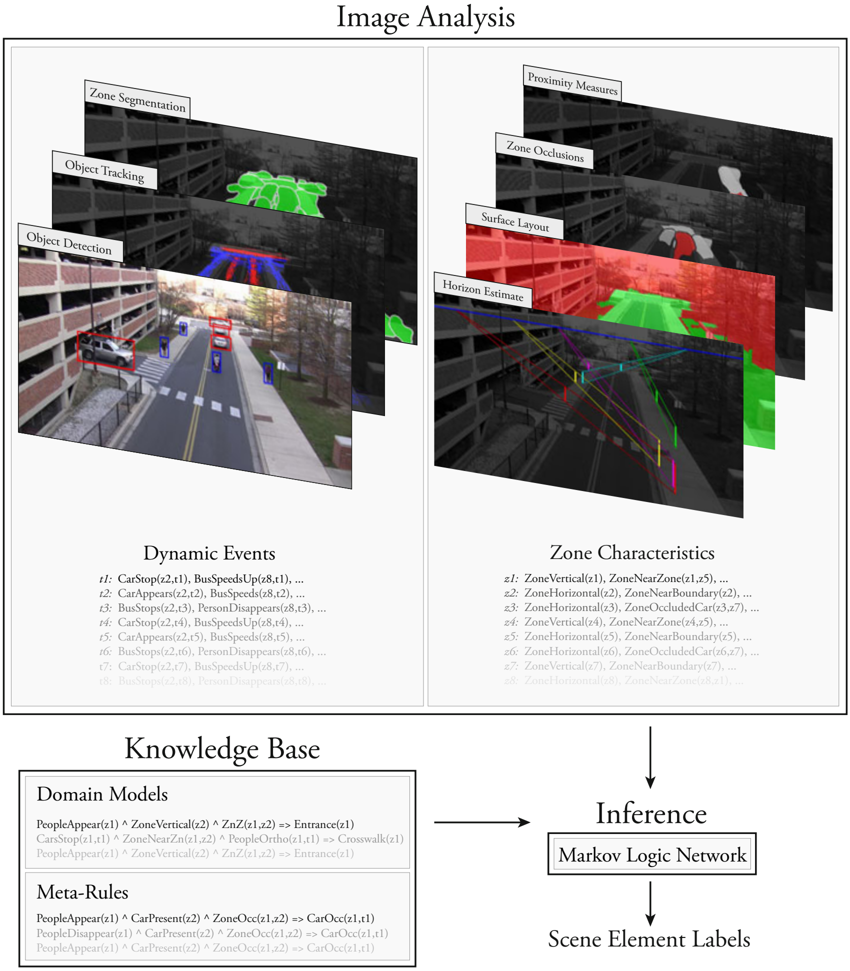


Image Analysis
Zone Segmentation
Object Tracking
Object Detection
Proximity Measures
Zone Occlusions
Surface Layout
Horizon Estimate
Dynamic Events
t1: CarStop(z2,t1), BusSpeedsUp(z8,t1), ...
t2: CarAppears(z2,t2), BusSpeeds(z8,t2), ...
t3: BusStops(z2,t3), PersonDisappears(z8,t3), ...
t4: CarStop(z2,t4), BusSpeedsUp(z8,t4), ...
t5: CarAppears(z2,t5), BusSpeeds(z8,t5), ...
t6: BusStops(z2,t6), PersonDisappears(z8,t6), ...
t7: CarStop(z2,t7), BusSpeedsUp(z8,t7), ...
t8: BusStops(z2,t8), PersonDisappears(z8,t8), ...
Zone Characteristics
z1: ZoneVertical(z1), ZoneNearZone(z1,z5), ...
z2: ZoneHorizontal(z2), ZoneNearBoundary(z2), ...
z3: ZoneHorizontal(z3), ZoneOccludedCar(z3,z7), ...
z4: ZoneVertical(z4), ZoneNearZone(z4,z5), ...
z5: ZoneHorizontal(z5), ZoneNearBoundary(z5), ...
z6: ZoneHorizontal(z6), ZoneOccludedCar(z6,z7), ...
z7: ZoneVertical(z7), ZoneNearBoundary(z7), ...
z8: ZoneHorizontal(z8), ZoneNearZone(z8,z1), ...
Knowledge Base
Domain Models
PeopleAppear(z1) ^ ZoneVertical(z2) ^ ZnZ(z1,z2) => Entrance(z1)
CarsStop(z1,t1) ^ ZoneNearZn(z1,z2) ^ PeopleOrtho(z1,t1) => Crosswalk(z1)
PeopleAppear(z1) ^ ZoneVertical(z2) ^ ZnZ(z1,z2) => Entrance(z1)
Meta-Rules
PeopleAppear(z1) ^ CarPresent(z2) ^ ZoneOcc(z1,z2) => CarOcc(z1,t1)
PeopleDisappear(z1) ^ CarPresent(z2) ^ ZoneOcc(z1,z2) => CarOcc(z1,t1)
PeopleAppear(z1) ^ CarPresent(z2) ^ ZoneOcc(z1,z2) => CarOcc(z1,t1)
Inference
Markov Logic Network
Scene Element Labels

Fig. 1. System overview. Our scene understanding system consists of an image analysis module (Section 3) that takes an input video and outputs a set of events and zone characteristics as observational evidence, a knowledge base (Section 4) that stores human elicited domain models and general rules about scene geometry and occlusion as a set of first-order logic rules, and an inference engine (Section 5) based on Markov Logic Networks that uses the logic rules and observational evidence to infer the labels of visible scene elements.

As an example, consider a model for a bus-stop. This model might indicate that people wait and queue at a bus stop, a bus stops at the bus stop, the doors to the bus open, people leave the bus through the doors, then the people waiting enter the bus through the doors, the doors close, and finally the bus leaves. From the viewpoint in Scenario 1 (refer to Figure 2), all of the activities associated with this bus stop model are observable. Scenario 2 shows a bus stop seen from another viewpoint, in which the bus occludes the people waiting to board, and the bus doors are not visible. In this case, our system reasons about this occlusion, and determines that what we expect to observe are that the people waiting for the bus will be gone when the bus leaves, and that new people will be seen after the bus leaves.

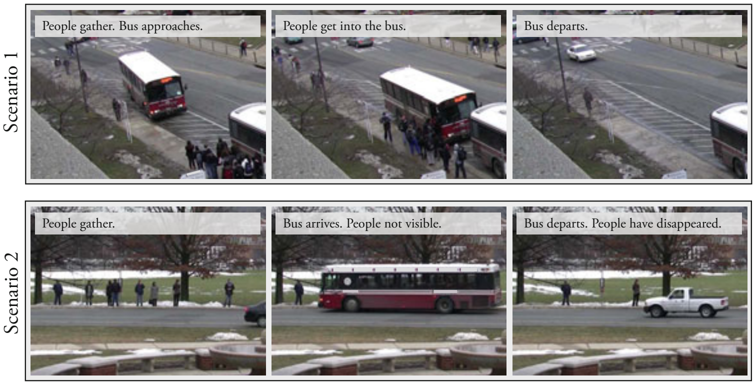

Scenario 1
People gather. Bus approaches.
People get into the bus.
Bus departs.
Scenario 2
People gather.
Bus arrives. People not visible.
Bus departs. People have disappeared.

Fig. 2. Two bus stops observed from different viewpoints. In Scenario 1, all activities associated with a typical bus stop model are observable. In Scenario 2, the bus occludes people departing and entering the bus.

We demonstrate our video understanding framework on a dataset of videos of public spaces. These video sequences were collected using cameras overlooking scenes from varying viewpoints. Each contains multiple scene elements of interest, such as bus stops, traffic intersections, stop signs, crosswalks, garage entrances, etc. Our system is able to correctly identify a large number of these scene elements described by the human elicited domain models.

2 Related Work

Methods to categorize scenes from single images by completely bypassing the tasks of image segmentation and object detection are described in [9,10,11]. Oliva et al. [9] represented holistic image structure using low level features that captured the degree of naturalness, openness, ruggedness, etc. whereas Fei-Fei et al. [10] represented scenes as bags of codewords of texture measures. More recently, there have been attempts to jointly solve the tasks of object recognition and scene classification [12,13,14,15]. Bosch et al. [13] detected objects and then used the object distribution for scene classification. Murphy et al. [14] combined the holistic image representation of [9] with local object detectors using a tree-structured graphical model. Li et al. [15] proposed a framework to deal with three problems simultaneously: object detection, segmentation and scene categorization.

There has also been progress in recovering surface orientations [5,7] and occlusion boundaries [16], given just a single image. Recently, Hoiem et al. [17] proposed a framework in which estimates of surface orientations, occlusion boundaries, objects, camera viewpoint and relative depth are combined, enabling automatically reconstructed 3D models.

Research in the domain of scene understanding from videos has mostly focused on building models of motion patterns of objects and using these to detect anomalous behaviors [18,19,20,21]. While Hu et al. [20] propose a parametric approach to model typical scene behaviors, Saleemi et al. use non-parametric density functions. Building such typical behavior models can help to improve foreground detection, detect areas of occlusion and identify anomalous motion patterns. There have also been attempts to learn activity based semantic region models for locations such as roads, paths, and entry/exits, most notably by Makris et al. [19] and Swears et al. [22]. Both these approaches involved designing a detector for every scene element.

Research in object category recognition has typically focused on building visual classifiers trained on annotated datasets. Recently however, there has been a growing interest in building object category models directly from human elicited descriptions [23,24,25]. Such approaches have the potential to learn unseen object categories based on their descriptions in terms of known visual attributes.

3 Image Analysis

Our scene understanding framework has three components: an image analysis module, a knowledge base and an inference module (refer to Figure 1 for a system overview). The image analysis module first segments the scene into a set of neighborhoods called *zones*. It then analyzes appearance characteristics of each zone as well as motion properties of objects passing through them, to generate a set of zone attributes that characterize local scene geometry and capture occlusion relationships between zones. A set of dynamic events is then generated for every zone, at every time instant, to describe the behavior of objects in the scene. The knowledge base consists of domain models describing the scene elements of interest, as well as a set of meta-rules that capture general knowledge about scene geometry and occlusion. The inference module, based on Markov Logic Networks (MLN), integrates events generated by the image analysis component with the rules in the knowledge base to label scene elements. The knowledge base and inference module are described in Sections 4 and 5 respectively. The components of the image analysis module are described below.

3.1 Detection and Tracking

We detect and track three classes of objects: humans, cars and buses. Detection is carried out using the object detection method proposed in [2][1]. For the purposes of human detection, we directly used a trained model provided along with the code, which was trained on the INRIA pedestrian dataset [1]. The car detector is trained using the Caltech Car Rear Training Set and the ETHZ Car Side Training Set [26]. The bus detector is trained using images from Bing Image Search. A two level association based tracking method is used to link object detections into tracks. At the low level, detections are linked to form tracklets using appearance and proximity features. At the second level, these tracklets are associated into longer tracks using appearance and motion features. Figure 3b shows car and human tracks obtained for one of the videos in our dataset.

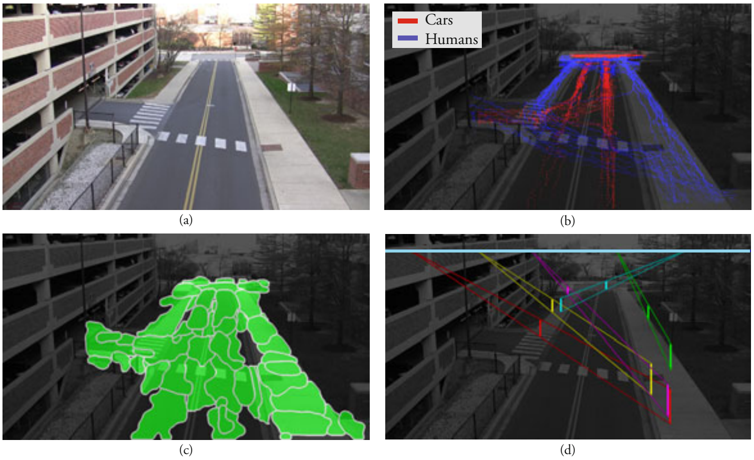


Fig. 3. Components of the image analysis module. (a) Background image for Scene I. (b) Trajectories (Sec 3.1). (c) Zones (Sec 3.2). (d) Horizon line estimate (Sec 3.3).

3.2 Zone Segmentation

The MLN based reasoning module utilizes events generated by the image analysis framework to assign labels to each part of the scene. To avoid performing inference at the pixel level, we segment the scene spatially into a set of zones, and perform inference on each zone. Zone segmentation groups pixels based on their appearance, location and the motion characteristics of objects passing through them. This results in a set of zones in which objects display distinct behaviors. Examples include locations where people gather and stand still for a long time (at bus stops), locations where vehicles drive in specific directions (along drive lanes), locations where cars and people cross each other (at cross walks), etc.

[1] Code obtained: `http://www.umiacs.umd.edu/~schwartz/softwares.html`

We begin by obtaining a background image by simply constructing an image for which a pixel $p(i,j)$ is the median of all pixels in the video at that location. This image is oversegmented by an image segmentation algorithm [27] to create a set of superpixels[2]. A set of features are computed for each superpixel, including: (1) Appearance - 3 histograms (one each for R,G,B) (2) Motion - Velocity magnitude histogram and velocity orientation histograms (weighted by magnitude) for each class of passing objects. An affinity matrix that includes the similarity between all pairs of superpixels is created for each feature. The distance metric used for all histograms is the Earth Mover's Distance (EMD). In addition, a location based affinity matrix is also created. This captures the minimum Euclidean distance between all pairs of superpixels and is calculated efficiently using the distance transform. Spectral clustering is then used to group superpixels into zones. We used the self-tuning method proposed by Zelnik-Manor et al. [28][3], since it automatically selects the scale of analysis as well as the number of clusters. Figure 3c shows zones obtained for one of the scenes in our dataset.

3.3 Scene Geometry Analysis

Surface Layout. An estimate of the scene surface layout supports reasoning about the location of many scene elements. For example, entrance and exit zones (such as doors into buildings) are typically located where horizontal and vertical surfaces meet. We obtain a rough surface layout using the method of [5][4] which classifies pixels into three primary classes: *horizontal*, *vertical* and *sky*. This estimate uses information extracted from individual images. However, we also have the additional knowledge of object trajectories that can help us obtain better surface estimates. Our meta-rules (discussed in Section 4) encode common sense knowledge about surfaces such as: *Objects are supported by a horizontal surface. Objects might appear out of and disappear into vertical surfaces.* Such rules allow us to correct some of the erroneous surface estimates provided by [5]. Figure 4 shows a surface layout before and after inference by our system.

Proximity Measures. Models of scene elements typically contain predicates corresponding to notions of proximity in the world, such as *nearby*, *far away*, *next to*, etc. Distances measured directly in the image plane, however, do not maintain these scene proximity relationships. Under a unit aspect ratio perspective camera model, we show how to compare segment lengths measured at different parts of the image based on their *true lengths* in the 3D world. We break the problem down into two components: segments parallel to the camera axis (lengths along a column of pixels) and segments parallel to the camera image plane (lengths along a row of pixels), shown in Figure 5.

[2] Code obtained: `http://www.wisdom.weizmann.ac.il/~ronen/index_files/segmentation.html`

[3] Code obtained: `http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html`

[4] Code obtained: `http://www.cs.uiuc.edu/homes/dhoiem/`

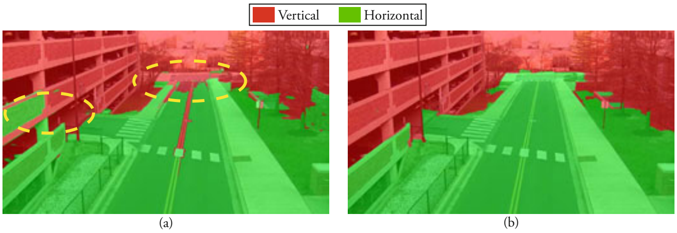


Fig. 4. Surface layout estimates before and after inference by our system. The road visible in the far distance is erroneously labeled as a vertical surface (in (a)), but corrected after inference (in (b)), due to the presence of objects passing over it.

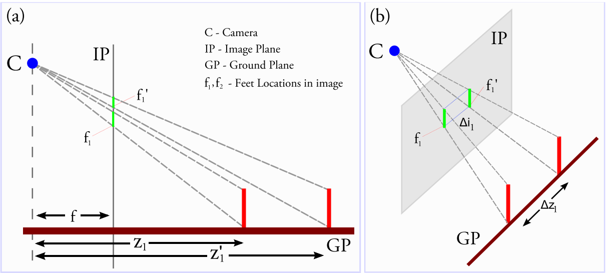


Fig. 5. Schematic relating image plane distances to ground plane distances

Consider Figure 5a. As in [6], we translate our image co-ordinates (u, v) to $(\hat{u}, \hat{v})$ so that $\hat{v} = 0$ for every point on the horizon line and $\hat{v} > 0$ below the horizon line. In this new co-ordinate system f_1 represents the foot location in the image of a person at a distance z_1 from the camera and f_1' is the foot location when the person takes a step $\varDelta z_1$ parallel to the camera axis to be located at a distance z_1' from the camera. Now, $f_1 z_1 = f_1' z_1' = f y_c$. Consider a person at a second location in the scene taking a step $\varDelta z_2$. This gives us: $f_2 z_2 = f_2' z_2' = f y_c$. A little algebra yields $\frac{(f_1'-f_1)f_2 f_2'}{(f_2'-f_2)f_1 f_1'} = \frac{\varDelta z_1}{\varDelta z_2}$. Now consider Figure 5b. Here the person moves from foot location f_1 to a new location f_1' parallel to the camera image plane. One can obtain: $\varDelta i_1 y_c = \varDelta z_1 f_1$, where $\varDelta i_1$ represents the image plane distance between the two feet locations. For a second person at a new location, we obtain: $\varDelta i_2 y_c = \varDelta z_2 f_2$. This yields $\frac{\varDelta i_1}{\varDelta i_2}\frac{f_2}{f_1} = \frac{\varDelta z_1}{\varDelta z_2}$. Given the horizon line, the above equations relate distances (segment lengths) measured at different locations in the image plane, based on the true 3D measurements. Measures such as *nearby*, *far away*, etc., when defined at one location in the image, can be thus transformed to equivalent measures at other locations.

The horizon line is estimated using the method of Lv et al. [29]. Consider two vertical poles of the same height in the scene. The two lines joining their foot locations and head locations, respectively, intersect at a point on the horizon line. Thus, three non-coplanar poles of the same height uniquely determine the horizon line. In practice, we have a large number of people walking through each scene. Each pair of detections (from the same human track) provides us with an estimate of a point lying on the horizon line. A least squares estimate of many such detection pairs yields a good horizon line estimate (shown in Figure 3d).

Zone Transitions. While the distance measures described above help define notions of proximity in the scene, they do not capture the restrictions imposed on object trajectories due to the scene layout. For example, a sidewalk is located adjacent to a road, yet vehicles typically do not traverse between roads and sidewalks. We characterize typical traffic patterns in the scene in terms of the average transition times of objects between one zone and another. These patterns are represented as transition matrices, one for each object class. Zone pairs that do not have any traffic flowing between them, are assigned a large transition time by default. Figure 6 shows examples of proximal zones. Note that cars typically conform to fixed directions, where as people walk along paths in both directions.

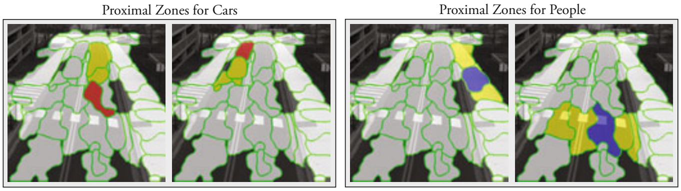


Fig. 6. Examples of proximal zones based on zone transition matrices. (a) Vehicles travel from red zones onto yellow zones within a short time span. (b) People walk from blue zones onto yellow within a short time span.

Directionality. User descriptions of scene elements often involve spatial prepositions which provide a notion of directionality, such as *in front of*, *behind*, *to the left of*, etc. Under the assumption that objects move in the direction in which they are facing, we define four directions with respect to the motion of the object: left, right, front and behind. Furthermore, some zones in the scene exhibit a single dominant direction of motion (based on the objects that pass through them). This is especially true of zones located on the road, on which vehicles strictly follow a single direction of motion. The four directions defined above are also noted for such zones, with respect to the centroid of the given zone.

3.4 Zone Occlusion Relationships

As objects move through the scene, they occlude different areas of the scene as well as objects present at those locations. This is a common source of errors in a typical computer vision system. Knowledge about typical occlusion areas can provide valuable information to the scene understanding framework. For example, people trajectories ending at a location suggest the presence of a doorway to a building at that location. However, the observation of a vehicle parked nearby, with the knowledge that it may causes occlusions at the former location, can prevent such an inference error. We represent occlusion relationships between zones using a binary matrix OC (one for each object class). For every object that passes through a zone z_i, we determine zones in the scene that intersect the object bounding box in the image plane (indicating potential occlusions), while the object was within z_i. If a zone z_j consistently undergoes occlusion by objects in z_i, the indicator variable $OC(z_i, z_j)$ is set to 1.

3.5 Event Generation

Short time spans of 20 frames are grouped together to form a temporal window. A set of dynamic events is generated at every zone within each temporal window. These events characterize the location, motion and trajectory of objects in a given zone during the given window. This results in a large set of evidence ground atoms passed to the inference module throughout the duration of the video sequence. In addition, the image analysis module also generates a set of zone characteristics and inter zone relationships, as described above. These are also represented as evidence atoms and passed on to the inference module.

4 Knowledge Base

The knowledge base consists of two components: a set of scene element models and a set of meta-rules that capture information about scene geometry, occlusion reasoning as well as common sense knowledge that applies to many domains. We begin with a description of our approach to represent uncertain knowledge, and then proceed with outlining the two components of our knowledge base.

4.1 Knowledge Representation

Knowledge is represented as first order production rules. The rules are represented in clausal form, whereby each rule is a conjunction of clauses and each clause is a disjunction of literals. Rules are constructed using variables such as *zone*, *time*, etc. Some variables are typed. Such variables have mutually exclusive and exhaustive values. For example, the typed variable *appearPersonReason* signifies an explanation for the birth of a person track and must take one of the following values: {*TrackingFailure, OcclusionByCar,...*}.

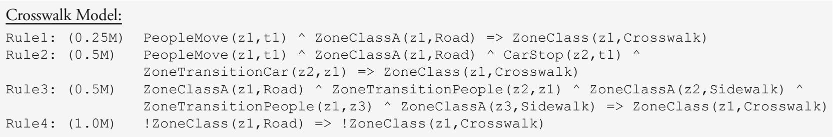
We use two types of predicates. The first represents events in the video and are associated with a particular zone and time instant (*PersonAppear(zone,time)*). The second represents properties of individual zones (*ZoneIsVertical(zone)*), relationships between zones (*ZoneNearZone(zone, zone)*) and relationships between time instants (*ShortlyAfter(time, time)*). These predicates need only be calculated once for the entire video sequence.

Each rule in our knowledge base is associated with a weight that indicates its confidence. We use three degree of confidence for rules of absolute certainty ($weight = M$), for ones with lesser certainty ($0.5M$) and for rules that may be true a very small fraction of times ($0.25M$). One may infer the certainty of a human elicited rule by frequency adverbs such as always, never, rarely, etc.

Some of the predicates generated by the image analysis module, such as *ZoneIsVertical(zone)*, have a confidence value associated with them. Such uncertain predicates are integrated into the first order rules using the method employed in [8]. Consider a predicate P with a weight w. We introduce a dummy observation predicate O_P along with a rule $O_P \rightarrow P$ and associate the weight w with this rule. The predicate O_P does not have any weight associated with it.

4.2 Scene Element Models

Each scene element is described by a logical model comprising a set of first order rules. These logical models describe a scene element on the basis of *what typically happens* in a scene at that element. For example, the logical model for a crosswalk consisting of logic rules with confidence measures is given in Figure 7[5]. The numbers in parentheses represent the weight assigned to each rule (recall that M represents the highest weight assigned in the knowledge base). The presence of people walking on the road indicates that they might be passing over a crosswalk (Rule 1). However, pedestrians often disobey laws and cross the road at other locations. The presence of a car waiting for people to cross the road is a stronger indication of a crosswalk and is thus assigned a higher weight (Rule 2).



Crosswalk Model:

```
Rule1: (0.25M)  PeopleMove(z1,t1) ^ ZoneClassA(z1,Road) => ZoneClass(z1,Crosswalk)
Rule2: (0.5M)   PeopleMove(z1,t1) ^ ZoneClassA(z1,Road) ^ CarStop(z2,t1) ^
                ZoneTransitionCar(z2,z1) => ZoneClass(z1,Crosswalk)
Rule3: (0.5M)   ZoneClassA(z1,Road) ^ ZoneTransitionPeople(z2,z1) ^ ZoneClassA(z2,Sidewalk) ^
                ZoneTransitionPeople(z1,z3) ^ ZoneClassA(z3,Sidewalk) => ZoneClass(z1,Crosswalk)
Rule4: (1.0M)   !ZoneClass(z1,Road) => !ZoneClass(z1,Crosswalk)
```

Fig. 7. First order logic rules representing a crosswalk model

4.3 Meta-Rules

In addition to the scene element models, the knowledge base also consists of a set of meta-rules, which encode information relating to scene geometry, occlusion handling, common failures of low level computer vision modules as well as common sense knowledge about the world. They only need to be written once, but are then widely applicable over a large number of domains. For instance, consider the scene element *Building Entrance/Exit.* Entrances and exits are typically characterized as sources and sinks of person tracks. There are however, a variety of situations that may lead to an initiation of a person track such as: exiting a vehicle, tracker identity switching, occlusion within a group of people, etc. Our meta rules encode such possibilities. This enables the inference module to reason about plausible explanations when it encounters a new person track. This reduces the number of false locations that might be labeled as an entrance-exit.

[5] Other models provided at: `http://www.umiacs.umd.edu/~ani`

5 Inference Using Markov Logic Networks

There has been a growing interest in problems related to knowledge representation and learning in domains that are rich in relational as well as probabilistic structure. Markov Logic Networks (MLN) are one such representation that combine first order logic with probability theory in finite domains [30]. They support the specification of statistical models using intuitive and understandable first order rules. A first order knowledge base, by itself, is often impractical to use for real world problems. Each rule in such a knowledge base is a hard constraint. A world that does not satisfy a single formula gets assigned a zero probability. MLNs attempt to relax these hard constraints using weights for each formula. The probability of a world is dependent upon the number of formulae that the world satisfies and the weights assigned to those formulae. MLNs can also be viewed as a template for constructing ordinary Markov networks. Given a set of formulae and constants, a MLN produces a Markov network. Based on the constructed network, marginal distributions of events given the observations can be computed using probabilistic inference. We use the Alchemy system [31] to represent our rules and perform inference on the resulting MLN[6].

5.1 Local Inference Procedures

The image analysis module generates a large number of evidence ground atoms within every temporal window, for every zone in the scene. Over the entire video, the number of ground atoms gets prohibitively large, rendering inference intractable. However, the spatio temporal interactions between objects, that characterize the scene elements of interest are sufficiently local in nature, both spatially and temporally. For instance, consider the crosswalk model in Figure 7 described by the interaction between people walking on the crosswalk and vehicles waiting on the road adjacent to it. Interactions between objects at locations far away from the crosswalk do not affect inference about the given zone. Likewise, interactions between people and vehicles at the crosswalk, at other times in the video, are largely independent of the current interaction.

We break down the large inference problem into smaller ones, carried out in every zone and at regularly spaced time instants. For every such spatio temporal location, the inference procedure takes into consideration events generated at a set of neighboring zones and time instants. For each zone, votes for each label, which are generated over the duration of the video, are aggregated to determine the final scene element label associated with that zone.

6 Experiments

We demonstrate our scene understanding framework on a dataset of 5 videos of public spaces, totaling over 100,000 frames (about 58 minutes). The video data

[6] Code available: `http://alchemy.cs.washington.edu/`

has been collected using cameras overlooking scenes from varying viewpoints. Each scene contains a large amount of pedestrian, car and bus traffic passing through it. Over the entire dataset, the number of pedestrians, cars and buses is approximately 700, 500 and 25 respectively. The data has been collected in high definition mode (1920x1080 pixels). Figure 8 shows some representative frames.

The scene elements that we seek to identify are: Road, Sidewalk, Other Path (other paths taken by people, which are not sidewalks), Bus-stops, Stop-sign Zones, Crosswalks, Entrances-Exits for People (typically buildings) and Entrances-Exits for Vehicles (typically garages). Figure 8 shows the labels assigned to different regions of the scenes. The system is able to correctly identify a large number of the scene elements using the human elicited domain models.

Our scene understanding framework is effectively able to reason about the scene geometry and occlusions to identify scene elements from widely varying viewpoints. Recall the example of a bus-stop observed from two viewpoints (Figure 2). Scene

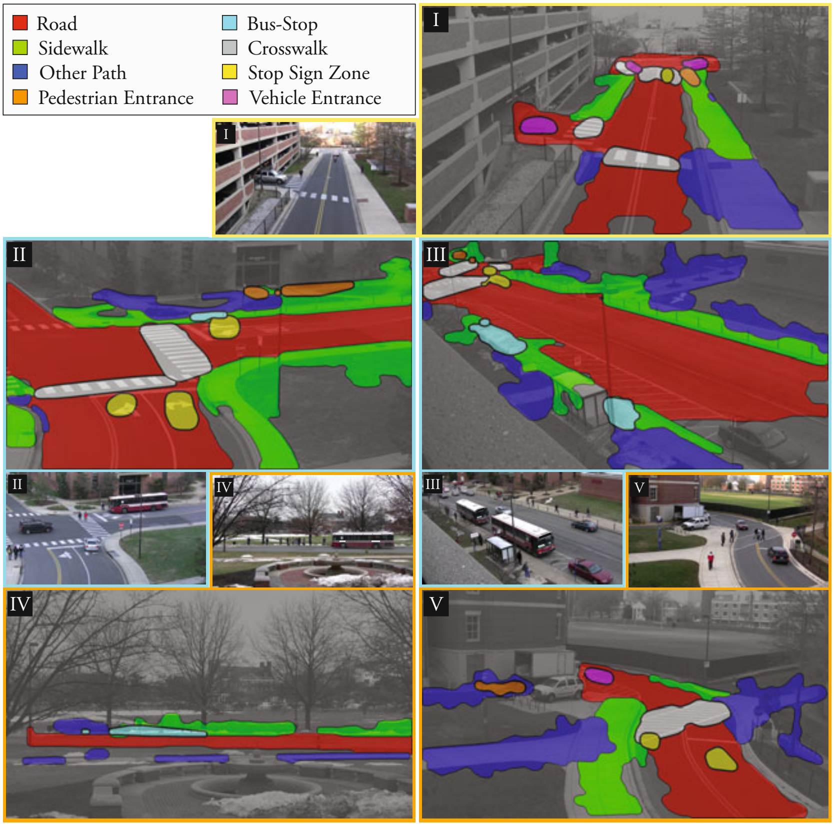


Fig. 8. Scene element labels determined by our system for Scenes I-V along with a representative image from each scene

III contains a view of a bus-stop in which we are able to observe people entering and exiting the bus. Scene II and IV, on the other hand, contain views of bus-stops in which the doors of the bus are not visible. The system reasons about people that might have entered and exited the buses that stopped at the location and correctly identifies all bus stops. Two locations are marked as bus-stops in Scene III, since buses stop one behind the other in this scene.

Pedestrian crosswalks are also correctly identified in all scenes, with the exception of a partially visible crosswalk in Scene II. These include the three crosswalks visible in the far distance in Scene III. A fair number of people tend to cross roads at locations other than crosswalks. However, cars do not always stop for such jaywalking violations. The system correctly identifies crosswalk locations using this additional information and suppresses the false alarms. Vehicle and pedestrian entrances are identified on the basis of track appearances and disappearances into vertical surfaces. Scene I shows a correctly identified garage entrance. The other detections in Scene I are not garage entrances, but they correspond to locations in the scene (away from the image boundary and close to vertical surfaces) where cars enter and exit the camera frame. Scene V shows a loading dock correctly marked as an entrance/exit for people. We fail to detect one of the doorways in Scene III (primarily due to a leafless, yet occluding tree), but another entrance in the distance away is correctly determined.

Roads, Sidewalks and Other Paths are also identified in each scene. Sidewalks are defined to be paths adjacent to roads and parallel to them on which people walk. Zones are considered parallel to one another if the orientations of objects passing through them are similar. Stop-sign zones are also detected in the scenes. The system does not merely depend on locations where cars stop-and-go, but also uses information such as *Stop zones are located adjacent to cross-walks and at intersections.* Scene V shows a false alarm caused by cars frequently stopping at a busy crosswalk. Such false alarms can be reduced by analyzing a larger amount of data, spanning different times of the day.

Acknowledgements. This research was partially supported by ONR grant N00014-09-10044.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
2. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. In: ICCV (2009)
3. Lampert, C., Blaschko, M., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE PAMI (2009)
4. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
5. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV (2007)
6. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: CVPR (2006)

7. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions (2009)
8. Tran, S.D., Davis, L.S.: Event Modeling and Recognition Using MLNs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 610–623. Springer, Heidelberg (2008)
9. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001)
10. Fei-fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
11. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. IEEE PAMI (2008)
12. Gupta, A., Kembhavi, A., Davis, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. IEEE PAMI (2009)
13. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via plsa. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
14. Murphy, K., Torralba, A., Freeman, W.T.: Using the forest to see the trees: A graphical model relating features, objects, and scenes. In: NIPS (2003)
15. Li, L.-J., Socher, R., Fei-Fei, L.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: CVPR (2009)
16. Charless, X.R., Ren, X., Fowlkes, C.C., Malik, J.: Figure/ground assignment in natural images. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 614–627. Springer, Heidelberg (2006)
17. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop on scene interpretation. In: CVPR (2008)
18. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE PAMI (2000)
19. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Trans. Systems, Man, and Cybernetics (2005)
20. Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: A system for learning statistical motion patterns. IEEE PAMI (2006)
21. Saleemi, I., Shafique, K., Shah, M.: Probabilistic modeling of scene dynamics for applications in visual surveillance. IEEE PAMI (2009)
22. Swears, E., Hoogs, A.: Functional scene element recognition for video scene analysis. In: Workshop on Motion and Video Computing (2009)
23. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer(2009)
24. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
25. Wang, J., Markert, K., Everingham, M.: Learning models for object recognition from natural language descriptions (2009)
26. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV (2008)
27. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR (2007)
28. Zelnik-manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS (2004)
29. Lv, F., Zhao, T., Nevatia, R.: Camera calibration from video of a walking human. IEEE PAMI (2006)
30. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning (2006)
31. Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Domingos, P.: The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA (2007)