Pattern Recognition Letters 32 (2011) 878–889

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Augmenting video surveillance footage with virtual agents for incremental event evaluation

C. Fernández [*], P. Baiget, F.X. Roca, J. Gonzàlez

Computer Vision Centre, Edifici O. Campus UAB, 08193 Bellaterra, Spain

ARTICLE INFO

Article history:
Available online 14 October 2010
Communicated by R. Davies

Keywords:
Human behavior analysis
Smart video surveillance
Benchmarking
Ontologies

ABSTRACT

The fields of segmentation, tracking and behavior analysis demand for challenging video resources to test, in a scalable manner, complex scenarios like crowded environments or scenes with high semantics. Nevertheless, existing public databases cannot scale the presence of appearing agents, which would be useful to study long-term occlusions and crowds. Moreover, creating these resources is expensive and often too particularized to specific needs. We propose an augmented reality framework to increase the complexity of image sequences in terms of occlusions and crowds, in a scalable and controllable manner. Existing datasets can be increased with augmented sequences containing virtual agents. Such sequences are automatically annotated, thus facilitating evaluation in terms of segmentation, tracking, and behavior recognition. In order to easily specify the desired contents, we propose a natural language interface to convert input sentences into virtual agent behaviors. Experimental tests and validation in indoor, street, and soccer environments are provided to show the feasibility of the proposed approach in terms of robustness, scalability, and semantics.

1. Introduction

Evaluating human activities in image sequences is commonly required by applications seeking to recognize and understand video events, such as surveillance, automatic retrieval of video content, and advanced human–computer interfaces. Specifically, it is desirable to account for occurrences observed in localized areas of interest, and to a feasible extent, identify arbitrarily complex behaviors. Such a high-level evaluation requires prior steps of segmentation and tracking, which have been intensively researched during the last years. Given the great number of available alternatives, there exists an increasing need to compare and evaluate performance on such systems.

The field of tracking evaluation assesses the capability of trackers to estimate the location of moving agents over time in image sequences, under different environmental conditions. A considerably large number of datasets have been published to provide researchers with standardized sequences, in order to evaluate and compare tracking approaches. Some datasets for the field of event/activity recognition include the CLEAR dataset (Stiefelhagen et al., 2006), the ViSOR project (Vezzani and Cucchiara, 2008), the BEHAVE interactions test case scenarios,[1] the CAVIAR test case scenarios,[2] the HumanEva dataset,[3] or the VS-PETS benchmark data. [4]

Nevertheless, since the construction of datasets is invariably sequence-oriented, these repositories often aim to solve specific difficulties in fixed contexts, sometimes resulting on an overadaptation of trackers to the scenes. Thus, it becomes difficult to compare two different image sequences in terms of tracking complexity. Moreover, new recordings, even from the same scenario, are exposed to different conditions due to changes of illumination, weather, or configuration of the scenario. To avoid the effort-consuming and not fully controllable task of acquiring new sequences for tracking evaluation, we propose instead to incorporate virtual agents and objects to already recorded scenes, which allow us to scale at will the complexity of a scene – e.g., occlusions, crowds, splitting/merging –.

This paper contributes with a tool that increases the complexity of image sequences in terms of occlusions and crowds, in an scalable and controllable manner. To avoid having to deal with computer graphics techniques, a natural language interface allows testers to easily incorporate virtual agents to the recorded scenes and control their developments from high-level. This will require (i) having the occlusions automatically handled by a scene composition task; (ii) having virtual agents develop their instructed activ-

* Corresponding author. Tel.: +34 935811828; fax: +34 935811670.
E-mail addresses: perno@cvc.uab.es (C. Fernández), pbaiget@cvc.uab.es (P. Baiget), xavir@cvc.uab.es (F.X. Roca), poal@cvc.uab.es (J. Gonzàlez).

[1] http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/

[2] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[3] http://vision.cs.brown.edu/humaneva/index.html

[4] http://www.cvg.rdg.ac.uk/PETS2006/data.html

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.09.027

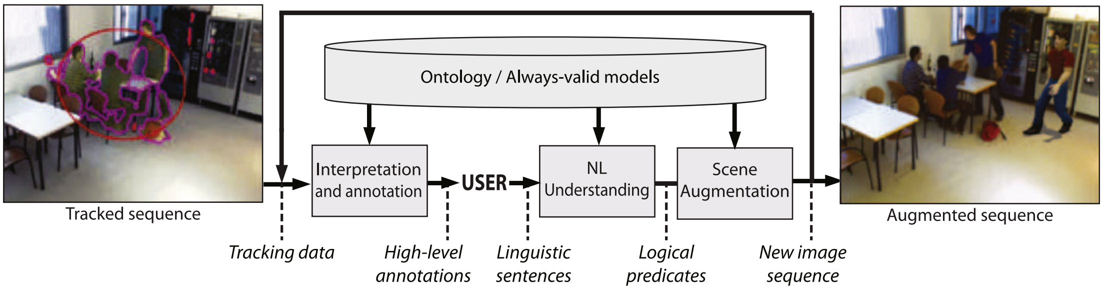


Fig. 1. Modular diagram of the components involved in the presented framework.

ities while reacting to events in the original recording; and (iii) having precise ground truth available to evaluate segmentation and tracking processes, i.e., the silhouettes, trajectories, and high-level behaviors of the synthetic agents.

Considering the features listed above, the proposed system extends and enhances the architecture described in Fernández et al. (2008) to facilitate the test and refinement of modules devoted to activity analysis from image sequences. Research in this area can benefit from a unified methodology to test or compare the performance and range of the algorithms in a controllable manner.

The presented system follows the architecture shown in Fig. 1, which includes three main modules and a series of a priori models. This framework builds upon the fields of computer vision, knowledge representation, computational linguistics, and computer graphics; similar inter-field collaborations are reviewed in Section 2. Section 3 discusses the representation of spatiotemporal knowledge undertaken by the system by means of an ontology. After that, Sections 4 and 5 present the tasks for interpretation and annotation of behavior and virtual agent modeling for scene augmentation, respectively. Section 6 shows experimental results and evaluation with augmented scenes in indoor, street, and sports environments, validated in terms of segmentation, tracking, and event recognition. Finally, Section 7 draws the concluding remarks. Since the natural language understanding task is not directly linked to the aim of the paper, a brief description of this module is included as an appendix at the end of the document.

2. Related work

Our framework aims at augmenting sequences with synthetic data, by accomplishing several tasks: implementation of ontological schemes, recognition of human activity and interactions in image sequences, interpretation of whole scene situations, and generation of linguistic descriptions. Regarding the main goal, similar techniques use synthetic data towards scene augmentation (Black et al., 2003; Qureshi and Terzopoulos, 2005; Taylor et al., 2007). Nevertheless, although these approaches permit users to scale the scene in terms of number of simultaneous tracks, and evaluate tracking and fixation capabilities, they do not consider adding complex behaviors for the virtual agents, and do not evaluate the performance at a scene interpretation level. We especially aim at this level of evaluation.

Ontology schemes have also been used in the literature for the representation of video events, for example (Francois et al., 2005; Ma and Mc Kevitt, 2004). The ontology proposed in this paper presents novelty in the distribution of video events, ranging from metric-temporal events – i.e., basic events with no complexity, and simply linked to movement or pose recognition – to events involving multiple agents/objects and requiring complex interpretations. As shown, this structure distributes adequately the knowledge processing to different modules and makes the annotation task easier.

Identifying human activities in image sequences requires to build proper behavior models that can be easily associated to observations obtained from tracking systems. A large variety of such approaches exist in the literature. On the one hand, several approaches use probabilistic models to generate behavior patterns. Hidden Markov Models (Brand and Kettnaker, 2000) and several variants (Galata et al., 2001; Oliver et al., 2000) have been studied in the last years, showing reasonable performance in selected environments. In (Buxton, 2003), Buxton reviews progress in generative models for advanced Cognitive Vision Systems (CVS) to explain activities in dynamic scenes, observing applications such as education, smart rooms, and also surveillance systems. Kojima et al. report some approaches based on concept hierarchies of actions to recognize interesting elements and developments in a scene, particularly people and object interactions (Kojima et al., 2002).

The study of interactions among moving objects is faced using statistical approaches for high-level attention and control. Most approaches do not emphasize the contextual properties of analyzed behaviors; instead, we define an independent stage to analyze the evolution of situations and their contextualization. Another modeling paradigm tries to automatically learn behavior models based on properties of specific regions of the scenario (Piciarelli and Foresti, 2006; Mokhber et al., 2007; Qian et al., in press). Nevertheless, learning methods do not address the conceptual ambiguity between the image sequence and its possible interpretations.

Scene interpretation is traditionally achieved by top-down methods, which make use of prior semantic knowledge to construct the behavior models. However, such methods usually depend on the scenario and on the expertise of the human designer. A behavior recognition framework is proposed by Brémond et al. (2006): for each tracked actor, the behavior recognition module performs three levels of reasoning, viz states, events, and scenarios. An early conception of artificial CVS was introduced by Nagel (1988), who has actively investigated for decades the field of CVS and Image Sequence Evaluation applied to vehicular traffic surveillance (Nagel, 2004). He tackles the high-level analysis of visual occurrences using fuzzy logic inference engines, and derives the results to the generation of NL textual descriptions. Recently, Gonzàlez applied this architecture to enlarge the domain of a CVS towards the analysis of general human behaviors in image sequences, in what has been called Human Sequence Evaluation (HSE) (Gonzàlez et al., 2009). Our proposed system builds upon the HSE scheme, where information flows between the lowest levels, image acquisition and segmentation, and the highest ones, NL interactions with end-users.

Several contributions also propose NL interfaces to affect the behavior of virtual agents, to let humans interact with smart environments, or to create augmented reality scenes. For example, (Bindiganavale et al., 2000) introduces an architecture to allow external users to input immediate or persistent instructions using natural language, and see the agents' resulting behavioral changes in the graphical output of the simulation. (Nijholt et al., 2009) discusses the modeling and simulation of interacting participants in virtual meeting rooms and smart home environments, using multi-modal capturing techniques that include verbal instructions. (Irawati et al., 2006) describes an Augmented Reality (AR) multi-modal interface that allows users to arrange virtual furniture in a virtual room, using a combination of speech and gestures from a real paddle.

Douze and Charvillat (2006) have recently joined computer vision techniques with AR, where moving targets are tracked from image sequences and merged into other real or virtual environments. Nevertheless, the method does not consider the animation of behavioral virtual agents in the resulting sequence. Klein and Murray (2007) adapt SLAM algorithms developed for robotic exploration into AR: using images from calibrated hand-held cameras, they collect thousands of feature points used to estimate a dominant ground-plane. This information permits them to add behavioral virtual objects over the ground-plane. The method keeps a correct estimation of the ground-plane as the camera moves, thereby maintaining a consistent existence of virtual objects in the image sequence.

Next section starts the description of our framework, by detailing the organization of the semantic knowledge involved.

3. Spatiotemporal and semantic knowledge

When structuring the semantic concepts required for the interpretation of occurrences, we use *event* concepts as central elements from which to build the rest of the knowledge resources. These events are organized linearly, ranging from basic actions identified by vision processes – e.g., an agent appears, moves fast, sits down –, to uncertain, intentional knowledge based on high-level behaviors – e.g., a group of people talks friendly; a soccer player scores after receiving the ball –. The set of events is organized as the central taxonomy of the ontology in our system; a significative set of these concepts can be found in Table 1.

Other concepts related to the events, such as the possible types of agents or objects participating in the occurrences, or the locations where these are developed, are included into the ontology in additional taxonomies. We link situations to the rest of the concepts by means of ontological *constraints* that restrict the validity of the situations to the specific domain. For instance, the situation *use_vending_machine* requires a person to be at a particular location (*vending_machine*), so it is modeled with two constraints: *is_agent (pedestrian)* and *is_at_location (pedestrian, vending_machine)*. In the case of a *chase*, we need at least an agent and a patient: *is_agent (pedestrian)*, *is_patient (pedestrian)*. These hierarchies of concepts and their constraints conform the terminological part of the ontology.

Apart from the concepts, the ontology also stores the instances of concepts that have been detected in the scene, i.e., information regarding the world state and the individuals existing on it. Once the abstract events, constraints, and entities are satisfied for a certain world state, these concepts are instantiated as *event instances*, *constraint instances*, and *entity instances*, respectively. For example, for the *theft* event instance in Table 2, the constraint instances required by the ontology are a thief, *is_agent (ped2)*; a victim, *is_patient (ped1)*; and a stolen item, *has_object (obj1)*. Keeping track of the instances is mandatory for the NL understanding task, to identify references to the agents involved in the scene.

4. Interpretation and annotation of behaviors

The intelligent management of situations builds upon an complete and detailed knowledge of particular scenarios, in order to facilitate complex semantic explanations that are valid in concrete domains. Inspecting a variety of different discourse domains we observe a series of characteristics:

1. *Temporal discontinuity*: The spatiotemporal data observed in time-variant scenes is valid only during limited intervals of time.
2. *Sensory gap*: Estimating quantitative values from observed image sequences involves uncertainty.
3. *Semantic gap*: More uncertainty is included from associating conceptual attributes to geometric quantities, for example *abnormal behavior* speed, due to the inherent vagueness of many terms.

The vision algorithms applied continuously over the recordings produce an extensive amount of geometric data. A process of abstraction is performed in order to extract and manage the relevant knowledge derived from the tracking processes, see (Baiget et al., 2009) for additional information. This knowledge is provided in form of spatiotemporal predicates expressing uniquely basic

Table 1
Sample concepts from the *situation*/*event* taxonomy of the ontology.

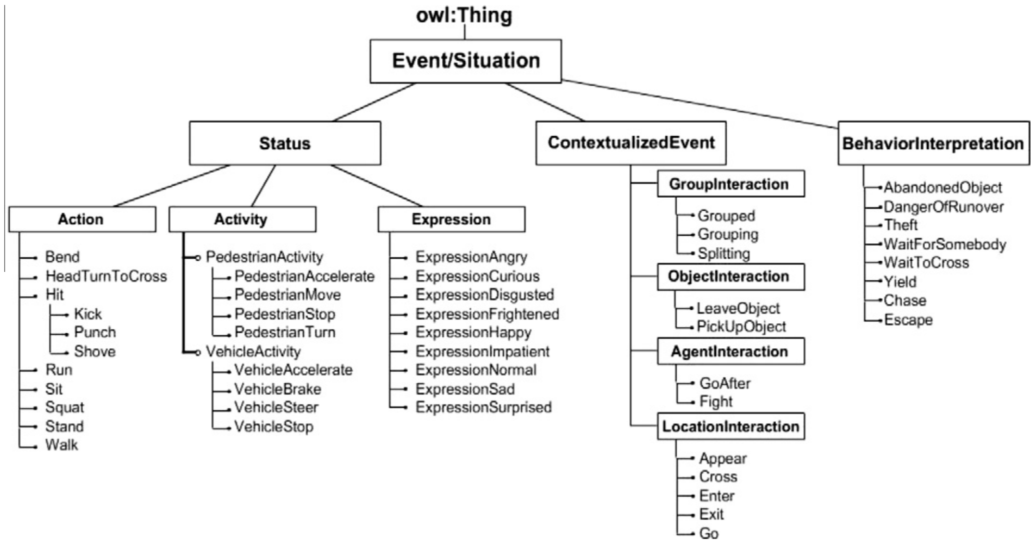


Table 2
Possible instances of entities (left) used in event indexes (right). For a *theft* to be indexed, *ped2*, *ped1*, and *obj1* must accomplish a certain semantic context.

Entity	Entity instance		Event type	Event instance
Pedestrian	*ped2*		Spatiotemporal	*walk (ped2, fast)*
Vehicle	*veh1*	⇒	Interaction	*appear (ped2, sidewalk)*
Location	*sidewalk*		Interaction	*pick_up (ped2, obj1)*
Object	*obj1*		Interpretation	*theft (ped2, ped1, obj1)*
Descriptor	*fast*		Interpretation	*danger_of_runover (veh1, ped2)*

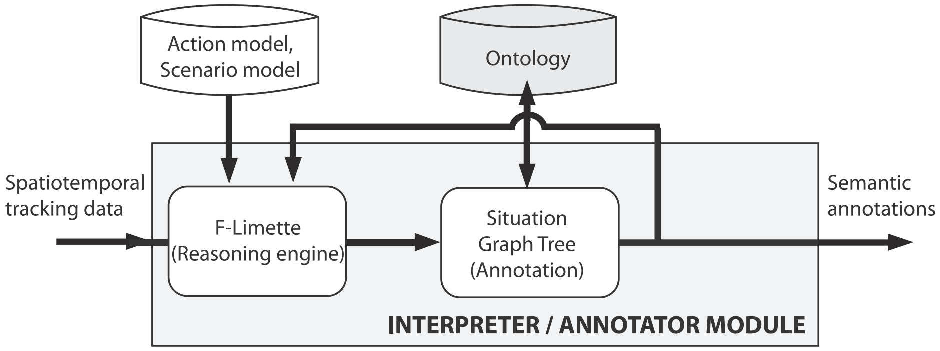


Fig. 2. Scheme of the interpreter module. This module conceptualizes new motion data, identifies events using a priori models, and carries out a situational analysis.

spatiotemporal developments. They facilitate a schematic conceptual representation of knowledge which is time-indexed and incorporates uncertainty.

Hence, in addition to the prior knowledge of the locations, it is desirable to find a conceptual framework that exploits these additional particularities of the data. A convenient solution to address these issues is to represent quantitative knowledge by means of fuzzy logic predicates. To this end, we use the Fuzzy Metric-Temporal Logic (FMTL) formalism (Schäfer and Brzoska, 1996), which consists of a rule-based inference engine in which conventional logic formalisms are extended by a fuzzy and a temporal components. In terms of notation, FMTL is similar to the well-known reasoning engine PROLOG (Colmerauer, 1990). However, the temporal and spatial components of FMTL make it a suitable tool to represent observed events in image sequences.

Nevertheless, some guidelines are needed to establish more complex relations of cause, effect, precedence, grouping, interaction, and in general any reasoning performed with time-constrained information at multiple levels of analysis. We use the high-level conceptual predicates defined in the ontology to express semantic relations among entities, at a higher level than metric-temporal relations. The tool which has been chosen to enable

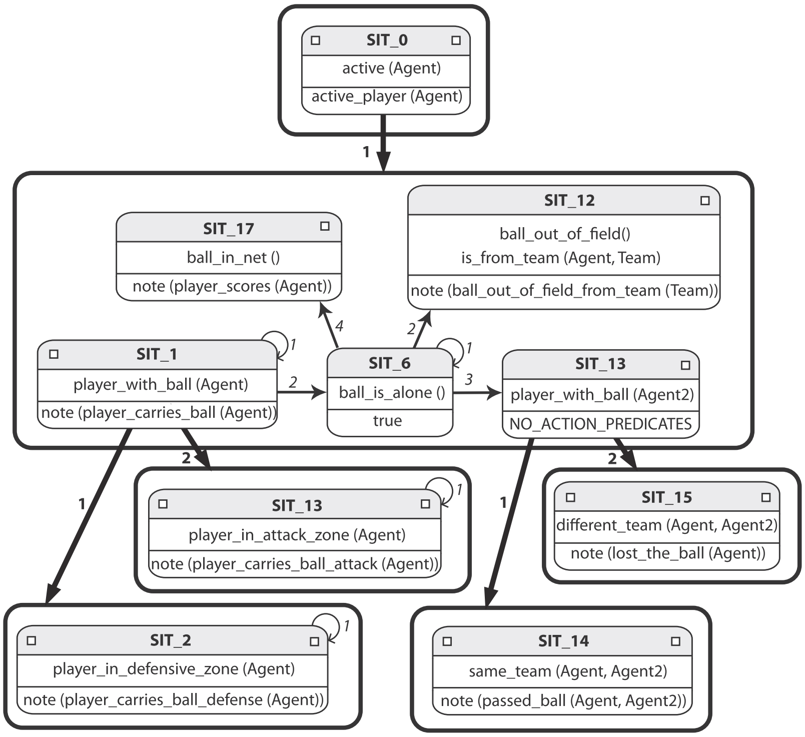

SIT_0
active (Agent)
active_player (Agent)
1
SIT_17
ball_in_net ()
note (player_scores (Agent))
SIT_12
ball_out_of_field()
is_from_team (Agent, Team)
note (ball_out_of_field_from_team (Team))
4
2
1
1
SIT_1
player_with_ball (Agent)
note (player_carries_ball (Agent))
2
SIT_6
ball_is_alone ()
true
3
SIT_13
player_with_ball (Agent2)
NO_ACTION_PREDICATES
2
1
SIT_13
player_in_attack_zone (Agent)
note (player_carries_ball_attack (Agent))
1
1
2
SIT_15
different_team (Agent, Agent2)
note (lost_the_ball (Agent))
SIT_2
player_in_defensive_zone (Agent)
note (player_carries_ball_defense (Agent))
1
SIT_14
same_team (Agent, Agent2)
note (passed_ball (Agent, Agent2))

Fig. 3. This SGT is evaluated to perform as a virtual commentator for soccer. When a set of conditions applies, its `note` reaction predicate generates a semantic annotation.

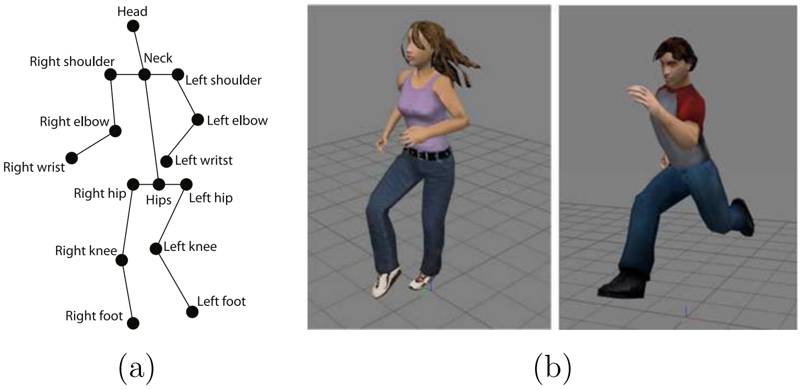


Fig. 4. (a) Generic human body model represented using a stick figure similar to Cheng and Moura (1999), here composed of twelve limbs and fifteen joints and (b) Different human models performing *dancing* and *running* actions.

behavior modeling, recognition, and synthesis of such predicates is the Situation Graph Tree (SGT), see (Nagel, 2004; Gonzàlez et al., 2009). The SGT is a hierarchical classification tool used to describe behavioral activity of agents in terms of situations they can be in. These trees contain a priori knowledge about the admissible sequences of occurrences in a defined domain.

The semantic knowledge related to an agent at a given point of time is contained in a series of *situations* (Nagel, 1988), the nodes of the hierarchical graph, see Fig. 3. Each situation evaluates a set of conditions in form of FMTL predicates, and reacts generating a new predicate once all the conditions are asserted. This new predicate varies according to the application: for event recognition, it is a high-level interpretation of the asserted situation, e.g., *an agent crosses the street* or *steals an object to another agent*; for virtual agent generation, it is a response action, e.g., *stop if a car is crossing*.

Each modeled situation is distributed along the tree-like structure of an SGT by means of the *particularization*, *prediction*, and *self-prediction edges*. Particularization edges instantiate more specific situations when certain conditions are accomplished. Prediction edges inform about the following admissible states within a situation graph from a given state, including the maintenance of the current state by means of self-prediction edges. The conjunction of these edges allows experts to define a map of admissible paths through the set of accepted situations. An example of SGT for basic commentation of soccer matches is shown in Fig. 3.

SGTs recognize the instantiated situations of an observed agent by applying a *graph traversal*. The goal of the traversal is to determine the most particular situation that can be instantiated by considering the collection of asserted FMTL predicates at each time step. These predicates are generated as a fuzzy discretization of the spatiotemporal data acquired by the tracking systems. The traversal of the SGT is applied by considering the knowledge encoded in the form of prediction and particularization edges. Fig. 2 depicts the interaction between the SGT and the fuzzy reasoner; a deeper explanation is detailed in Arens and Nagel (2003).

The reaction predicates are notes describing the content of the situations, one per time-step, as a result of the continuous evaluation of the SGT. Persistent notes are finally grouped along the temporal interval in which they have been a constant output. As a result, the whole sequence is split in cohesive time-intervals defined by the start of each semantic tag. Thus, we obtain sequences of interpretations (event recognition) and virtual agent reactive behaviors (scene augmentation) from tracked data.

5. Virtual agent modeling

Predicates are obtained from NLU as explained in Section A. In essence, they represent goals to be reached by some virtual entity in the scenario. To accomplish this task, we adapt the FMTL+SGT framework presented in Section 4 towards the creation of synthetic instances of agent trajectories. As explained before, the SGT is evaluated given the quantitative information obtained from tracking at each frame step, which instantiates situations and raises reactions, like the annotations of observed behavior already described. Here, on the other hand, these reactions are used to produce synthetic behaviors. Given an initial configuration of a virtual agent, the system recursively generates the activities for the agent within its context.

Generating synthetic trajectories requires adapting prior knowledge from Table 1, initially designed for behavior recognition. Thus, each virtual agent behavior is modeled using three different generation processes, according to the level of abstraction: action (*walk, bend*), contextualized events (*going to vending machine, accelerating*), and behavior (*entering a crosswalk*).

5.1. Virtual action

A human action is defined as a discrete sequence of movements of body parts. In this work we use a human model based on the stick figure, see Fig. 4. The learned sequence of movements for a particular action is called the prototypical action or *p-action*, defined as a cubic spline in a PCA space, where each point $p \in [0,1]$ corresponds to the mean postures of several performances, see Fig. 5. Using *p*-actions, we can model both cyclic actions, e.g. *walk* or *run*, and non-cyclic ones, e.g., *wave* or *bend* (Gonzàlez et al., 2009).

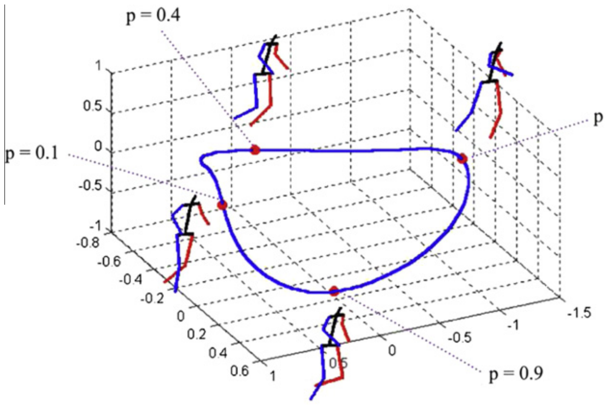


Fig. 5. *p-actions* computed in the *aRun aSpace* (Gonzàlez et al., 2009). By varying the parameter pose *p* we move along the manifold, temporally evolving the human body posture along the prototypical performance of a learnt action.

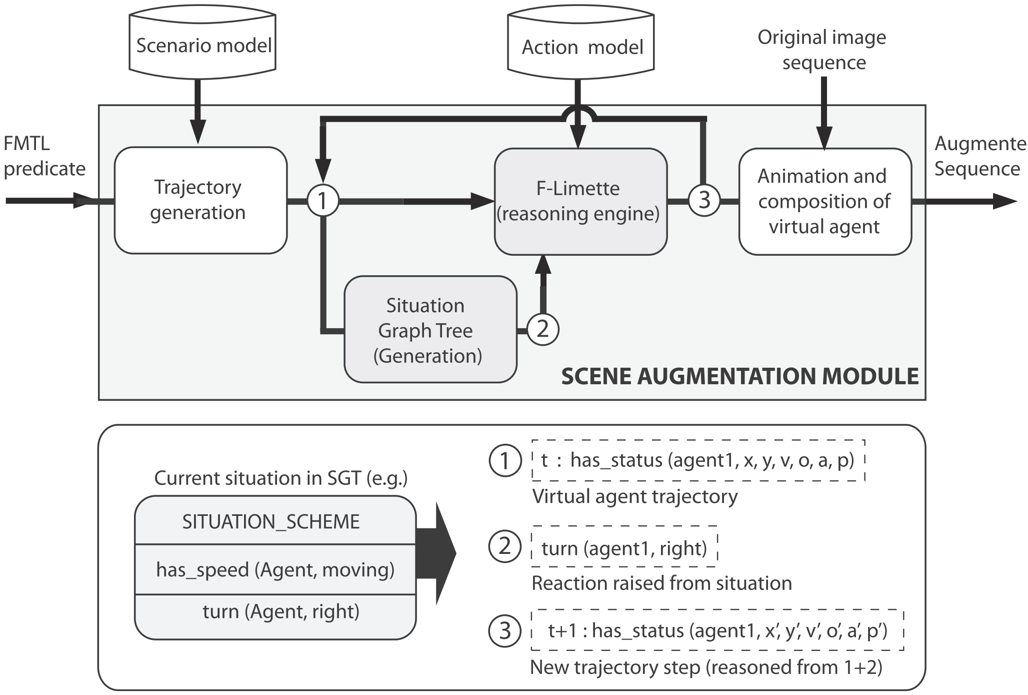

Scenario model
Action model
Original image sequence
FMTL predicate
Trajectory generation
1
F-Limette (reasoning engine)
3
Animation and composition of virtual agent
Augmented Sequence
Situation Graph Tree (Generation)
2
SCENE AUGMENTATION MODULE
Current situation in SGT (e.g.)
SITUATION_SCHEME
has_speed (Agent, moving)
turn (Agent, right)
1
t : has_status (agent1, x, y, v, o, a, p)
Virtual agent trajectory
2
turn (agent1, right)
Reaction raised from situation
3
t+1 : has_status (agent1, x', y', v', o', a', p')
New trajectory step (reasoned from 1+2)

Fig. 6. Scheme of the AR process. When considering the pictured situation scheme, the numbered predicates can be found in the corresponding states of the information flow.

5.2. Generation of contextual events

Next, in order to adapt its motion online, accomplishable goals must be specified for an agent in the scenario. Such objectives require predicates to adapt the agent trajectory, so it is valid in future time steps. For instance, given an agent with state vector[5] $s_t^{Ag} = (x_t, y_t, v_t, o_t, a_t, p_t)$, the predicate *go_to_location (Ag, Location)* computes the shortest trajectory $\left\{s_{t+1}^{Ag}, \ldots, s_n^{Ag}\right\}$ to arrive to *Location* and infers its next position (x_{t+1}, y_{t+1}) according to the current speed value v_t.

5.3. Behavior generation

Virtual agent behaviors need to be defined considering the previously mentioned interactions. Reactions to the situations, used to annotate observed events, are also employed to modify the agent status for future frame steps. These generated status predicates are returned to the SGT as a feedback, and the reasoner considers them in following evaluations of the SGT.

This recursive procedure is depicted in Fig. 6. The traversal starts with an initial status of a virtual agent, containing its position, orientation, speed, and action at the very first time. Then, for each time step t, the traversal uses the current agent status s_t^{Ag} to generate the next one s_{t+1}^{Ag}. In the example in Fig. 6, the situation instantiated at time t generates the action predicate *turn (agent1, right)*. This predicate modifies the agent status so that the agent will be turning to the right in the following time steps. The computation of s_{t+1}^{Ag} is based on s_t^{Ag} and the modeled *p*-action. The semantic concept *right* is converted into a numerical value by combining the current orientation o and speed v, and is used to generate the new position (x', y'), speed v', and orientation o' for the next time step. The obtained values are used to construct the agent status for the time step $t + 1$ and will be used as input for time step $t + 2$ in a subsequent traversal loop. More accurate physical rules can be elaborated by simply defining them as FMTL predicates.

Action predicates like *turn* determine particular movements and actions for a virtual agent. This is achieved by modifying its position, velocity, orientation, and action. For example, `accelerate (Agent, Value)` modifies the velocity of the agent for the next time step. The fuzzy concept *Value* describes the discrete value of speed that the agent will take in a future time step. Collisions can be avoided by evaluating the distance between the virtual agent and the rest of the object via `has_distance` predicates, computed over the estimated positions. Agent interactions are tackled in the same fashion, given that each agent behavior is provided by a dedicated SGT traversal. Obstacles have to be defined in the conceptual models provided a priori.

6. Experimental results

The described system has been tested on three different scenarios from indoor, street, and sports scenes, in which original image sequences have been incremented with virtual agents. We have validated the approach regarding segmentation, tracking, and event recognition. In the first case, we compare the performance of 3 state-of-the-art background substraction techniques when segmenting original and augmented sequences. We also evaluate how different tracking algorithms perform at location entrances, occlusions, and location exits. Finally, we present lists of semantic annotations for the original and augmented sequences of each scenario, to demonstrate that the creation of virtual agents modifies in a controlled manner the initial performances of the tracking algorithms.

Surveillance scenarios represent either open or closed environments, each type entailing different events of interest: in street and soccer scenarios, cameras cover wide regions and agents occupy a narrow part of the image, the analysis thus being focused on their silhouettes and trajectories. On the other hand, indoor scenarios typically deal with small environments that contain a richer set of objects to interact with (chairs, tables, vending machines). Regarding open scenarios, urban and sport environments also differ: soccer scenes are more constrained in terms of number of agents and expected behaviors, which are based on well-known

[5] The state vector incorporates values for ground-plane position, velocity, orientation, type of action, and percentage in the sequence of the action.

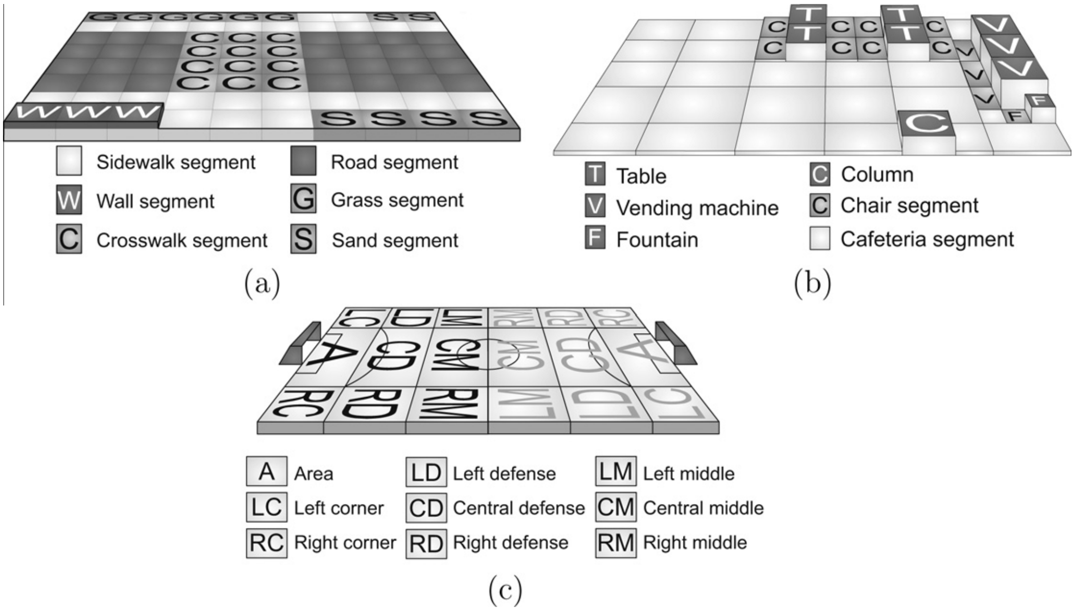


Fig. 7. Conceptual models for the scenarios used to test behavior interpretation: (a) street, (b) indoor, and (c) soccer.

rules; urban street scenes contain less prior information, and thus require a detailed semantic description of the scenario elements to improve interpretations.

In order to study pedestrian behaviors and vehicle–pedestrian interactions, we use the *HERMES outdoor* sequence.[6] Secondly, we focus on the analysis of sport video sequences to obtain descriptions of matches; to this end, a soccer sequence from the VS–PETS 2003 database [7] has been used to evaluate the behavior of soccer players. The third sequence depicts an indoor cafeteria scenario, in which we analyze a vandalic indoor behavior. A conceptual model of the environment has been designed for each application domain, see Fig. 7. The scenario is represented in ground-plane coordinates, which allows the reasoning system to work with accurate 3D information extracted from calibrated cameras.

6.1. Evaluation of segmentation

Given that the mask of a virtual agent presents a synthetic – thus perfect – chroma, it requires to undergo a process of noise generation to present the characteristics of the camera output image. Several works have studied the estimation and generation of camera noise; here, we follow the noise model of a CCD camera as described in (Liu et al., 2006):

$$I = f(L + n_s + n_c) + n_q, \tag{1}$$

where f is the camera response function, L the irradiance, n_s the irradiance-dependent (photon) noise, n_c the independent noise before gamma correction, and n_q the quantization/amplification noise, which is usually ignored. Noises have zero mean, $L\sigma_s^2$ variance for n_s, and σ_c^2 variance for n_c.

We evaluate segmentation comparing three techniques on background substraction for shadow detection, Ariel Amato and Mozerov (2008), Seo et al. (1997), Stauffer and Grimson (2000), with and without the incremental presence of virtual agents. The original soccer sequence has been augmented twice, by adding five virtual agents each time. Estimated average values of $L\sigma_s^2$ and σ_c^2 for the original frames have been 0.036 and 0.050, and synthetic noise has been generated for each pixel in the masks. Fig. 8 shows silhouette segmentation errors at pixel-level evaluated in terms of percentage of false positives (FP) and false negatives (FN), over a random selection of 20 frames taken from the sequence. A new ground truth has been computed by joining the existing manual annotations with the ones automatically generated for virtual agents.

As it can be seen, the addition of virtual agents generally reduce the FP rate. This happens because the lack of chromatic problems in the silhouettes of virtual agents tends to ease the identification of foreground. The first technique does not reduce errors as much as the other two when scaling the sequence. On the other hand, we also observe an increment of the FN rate, given the lack of precision of the algorithms to accurately segment the new silhouettes, caused mainly by occlusions, camouflage, and clutter.

6.2. Evaluation of tracking

The evaluation of tracking algorithms has been tested incrementally, too. The original *HERMES outdoor* sequence has been augmented with 30, 60, and 120 virtual agents appearing and disappearing over the whole timeline. Two different trackers have been tested: a blob tracker from the OpenCV library,[8] which is well-known to the computer vision community, and a real-time tracker based on segmentation on a static background (Roth et al., 2009). The ground truth labeling has been obtained manually using a touch screen.

Virtual agents were instructed to randomly follow behaviors like walking by the sidewalk, waiting for someone, or crossing the street. Fig. 9 compares several frames showing the results of the two trackers from both the original and the incrementally augmented sequences. We can see that the performance drops as the crowd increments.

In order to validate the detection of simple tracking events, we account the detection of simple tracking events compared to manual annotation. The results are depicted in Table 3.

[6] The indoor and street sequences presented here are part of the dataset recorded for the HERMES Project (IST 027110, http://www.hermes-project.eu), which are publicly available at http://iselab.cvc.uab.es/tools-and-resources.

[7] http://www.cvg.rdg.ac.uk/PETS2006/data.html

[8] http://opencv.willowgarage.com/

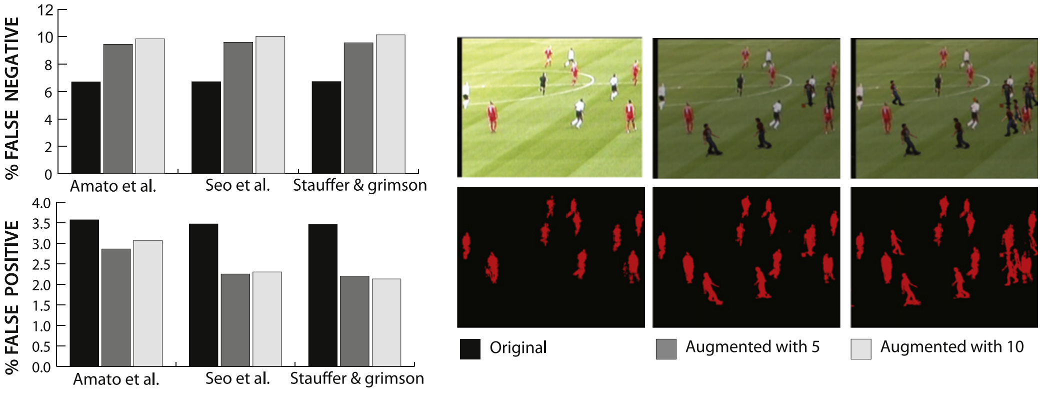


Fig. 8. Silhouette segmentation errors for the original and augmented sequences, in terms of percentage of false positive (FP) and false negative (FN) pixels.

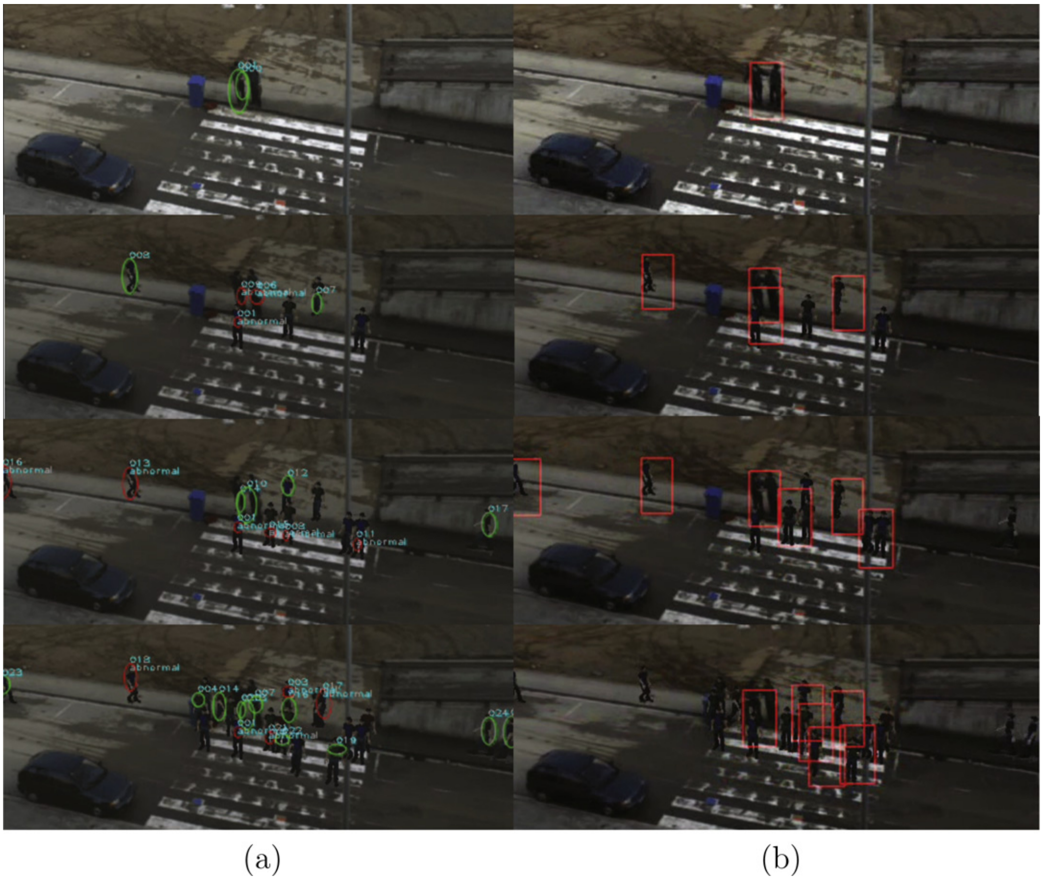


Fig. 9. Top–down: Tracking results of the original sequence, sequence augmented with 30, with 60, and with 120 virtual agents, for (a) the OpenCV blob tracker and (b) the real-time tracker (Roth et al., 2009).

Table 3
Detection of simple events in the first 17 s of the street sequence, for original (*Or.*) and augmented versions with 30, 60, and 120 virtual agents.

Events	Ground truth				Tracker (Roth et al., 2009)				Blob tracker			
	Or.	30	60	120	Or.	30	60	120	Or.	30	60	120
Enter scene	2	10	19	33	2	9	13	15	2	12	15	28
Exit scene	0	0	0	1	1	3	8	10	0	8	8	9
Start occlusion	1	4	10	29	2	6	8	16	1	4	10	27
End occlusion	1	4	9	13	2	5	7	8	1	3	9	23
Enter crosswalk	0	5	10	18	0	5	11	22	0	5	5	10
Exit crosswalk	0	0	0	1	0	1	4	4	0	1	3	3

We observe that tracking performance decreases as the crowd of virtual agents is formed, especially for the second tracker, which is optimized for real-time performance rather than multiple target tracking. For this tracker, crowded context tends to have bounding boxes slide away until lost. The blob tracker performs better on acquiring and holding on targets,

The *entering/exiting scene* events were successfully recognized. However, due to camouflage, the number of occlusions detected is higher than the annotated ground truth. Finally, most of the events *entering/exiting crosswalk* have been detected.

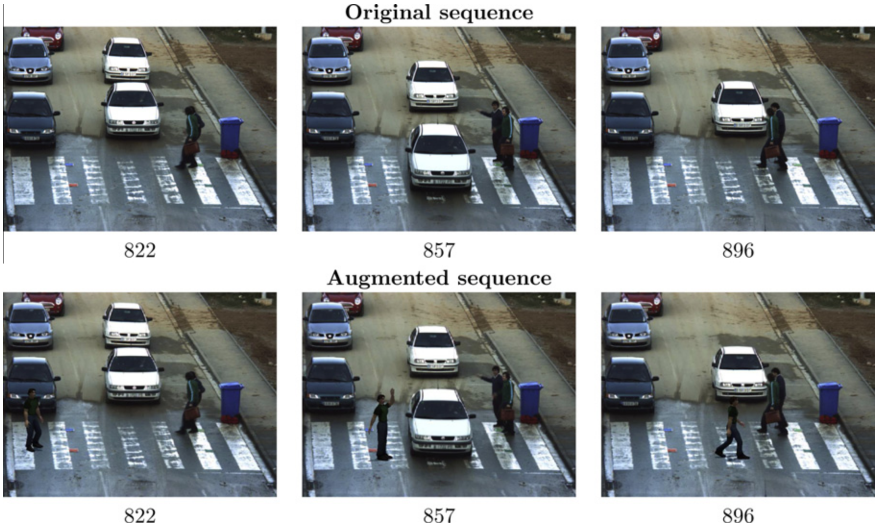
6.3. Evaluation of event recognition and annotation

Table 4 show sample frames and annotations from the original and augmented versions of the *HERMES outdoor* sequence. In the original sequence, pedestrians *Ag1* and *Ag2* stop before entering the crosswalk and an approaching vehicle (*Ag3*) does not give way, thus generating the event *danger_of_runover*. The textual description provided by the user has been: *"A new pedestrian appears by the lower left side at frame 800. The pedestrian enters the crosswalk. He leaves by the lower right side*". The purpose of this experiment is to study the capability of a behavior analysis system to recognize target situations such as runovers. In addition, since virtual agents are aware of what is happening in the original sequence, they can react to previously recognized events. As it can be observed in the second annotation table, the new virtual agent (*Ag4*) is correctly tracked, and a new predicate *danger_of_runover* is generated at frame 855. After having reacted to the environment, this agent retakes the demanded behavior.

The second experiment is shown in Table 5. Its purpose is to demonstrate that an addition of virtual agents may affect the behavior analysis annotations. Events in the original sequence describe the normal development of a match: player *B_4* passes the ball during frames 195–284, this is intercepted by *A_2* at frame 295. In the augmented sequence, virtual agents have been added as components of *team B* to interfere in the activity analysis. Each virtual agent has been given two instructions: *"A new player appears in the* $\langle zone \rangle$ *at frame 180*", with $\langle zone \rangle$ being one of $\{CD, CM, LD, \ldots\}$ as described in Fig. 7(c), and *"He chases after the ball*". In the new sequence, virtual agent *B_7* runs towards *B_4* at frame 195. At frame 284, *B_7* seems to have captured the ball, so the system interprets a correct pass between team B members: *passed_the_ball (B_4, B_7)*. However, the virtual agent does not own the ball – the trajectory of an original element of the sequence cannot be affected –. *A_2* finally takes the ball, and the system interprets that the virtual agent lost it, asserting *lost_the_ball (B_7)*.

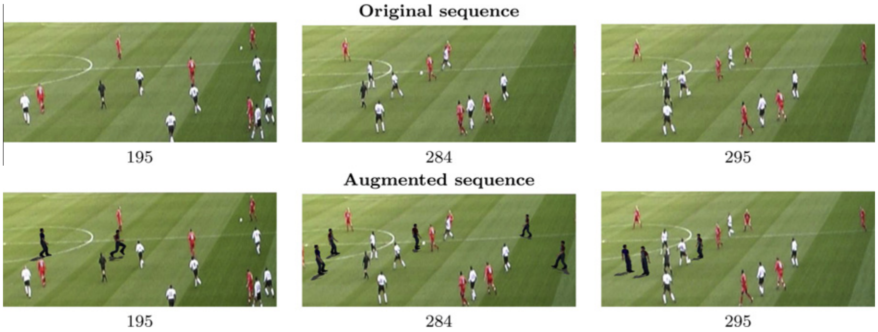
The list of events in the *HERMES indoor* scenario describe interactions among agents, objects, and locations, and also interpretations of complex behaviors and occurrences. In the original scene, three persons and two objects are shown interrelating among them and with elements of a cafeteria such as a vending machine, chairs, and tables. The instantiated events include agents

Table 4
Semantic annotations obtained for the frame interval [805, 875] of the street HERMES sequence, for the original and augmented sequences.



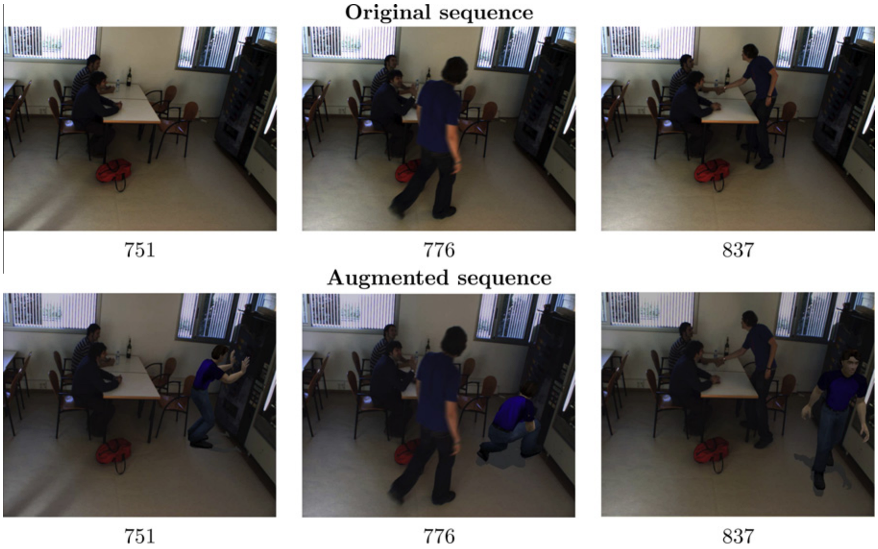
Start	Event (original sequence)	#	Event (augmented sequence)	#
806	Appear (Ag3, left)	1	Appear (Ag3, left)	1
810	Enter (Ag1, crosswalk)	2	Enter (Ag1, crosswalk)	2
824	Danger_of_runover (Ag3, Ag1)	3	Danger_of_runover (Ag3, Ag1)	3
825	Stop (Ag1)	4	Stop (Ag1)	4
825	–		**Enter (Ag3, crosswalk)**	5
828	Brake_up (Ag3)	5	Brake_up (Ag3)	6
828	Danger_of_runover (Ag3, Ag2)	6	Danger_of_runover (Ag3, Ag2)	7
838	Back_up (Ag2)	7	Back_up (Ag2)	8
838	–		**Stop (Ag3)**	9
842	Stop (Ag2)	8	Stop (Ag2)	10
852	Accelerate (Ag3)	9	Accelerate (Ag3)	11
855	–		**Danger_of_runover (Ag4, Ag3)**	12
871	Appear (Ag4, left)	10	Appear (Ag4, left)	13
872	Exit (Ag3, right)	11	Exit (Ag3, right)	14
891	Give_way (Ag4, crosswalk)	12	Give_way (Ag4, crosswalk)	15
895	–		**Walk (Ag3, crosswalk)**	16
898	Walk (Ag2, crosswalk)	13	Walk (Ag2, crosswalk)	17

Table 5
Semantic annotations obtained for the frame interval [187,465] of the VS–PETS image sequence and its augmented scene.



Start	Event (original sequence)	#	Event (augmented sequence)	#
187	Player_carries_ball_defense (B_4)	1	Player_carries_ball_defense (B_4)	1
279	Threw_the_ball (B_4)	2	Threw_the_ball (B_4)	2
284	–		**Passed_ball (B_4, B_7)**	3
296	Lost_the_ball (B_4)	3	**Lost_the_ball (B_7)**	4
296	Player_carries_ball_defense (A_2)	4	Player_carries_ball_defense (A_2)	5

Table 6
Sequence of semantic annotations obtained for the frame interval [700,900] of the indoor HERMES sequence.



Start	Event (original sequence)	#	Event (augmented sequence)	#
702	–		**On_location (Ag3, vending_machine)**	1
714	–		**Vandalize_vending_machine (Ag3)**	2
716	Appear (Ag3, entrance1)	1	Appear (Ag4, entrance1)	3
742	–		**Use_vending_machine (Ag3)**	4
755	On_location (Ag3, cafeteria)	2	On_location (Ag4, cafeteria)	5
782	On_location (Ag3, table2)	3	On_location (Ag4, table2)	6
882	Meet (Ag3, Ag1)	4	Meet (Ag4, Ag1)	8
890	–		**Exit (Ag3, entrance2)**	9

appearing and leaving, displacements among the different scenario regions, sit down and stand up actions, normal interaction with a vending machine, and violent behaviors such as kicking or punching the vending machine.

In this third and last experiment, shown in Table 6, a virtual agent interacts with the elements of the scenario while partially occluding the real agents in the scene. This example takes advantage of a closer position of the camera to provide more detailed sequences of actions, so that specific combinations of gestures can be detected by activity analysis systems due to the increased resolution of the agents. The input text in this case has been *"A new person enters by the first entrance at frame 650. He pushes the vending machine. He takes a drink from it. He leaves by the second entrance*". In the augmented sequence, particular behaviors like *vandalize_vending_machine (Ag3)* at frame 714 or *use_vending_machine (Ag3)* at frame 742 can be correctly detected by the system by means of pose estimation algorithms.

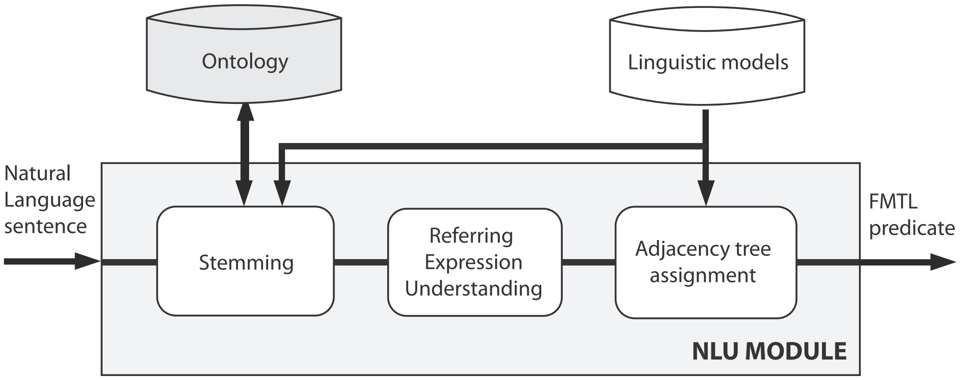

Ontology
Linguistic models
Natural Language sentence
Stemming
Referring Expression Understanding
Adjacency tree assignment
FMTL predicate
NLU MODULE

Fig. A.10. Scheme of the NLU module. Sentences written by the user are individually converted into conceptual predicates that will generate the augmented sequences.

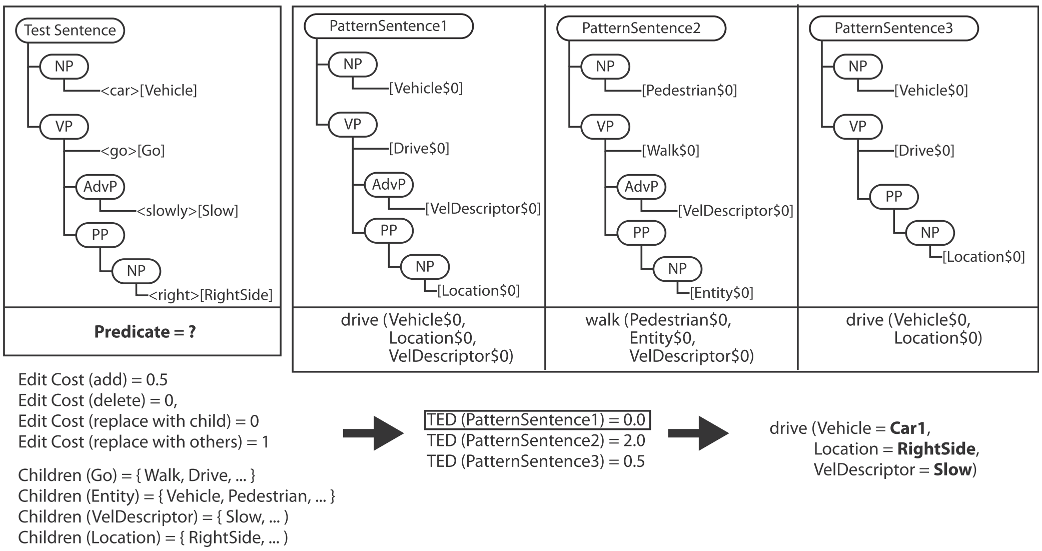

Test Sentence
NP
<car>[Vehicle]
VP
<go>[Go]
AdvP
<slowly>[Slow]
PP
NP
<right>[RightSide]
Predicate = ?
PatternSentence1
NP
[Vehicle$0]
VP
[Drive$0]
AdvP
[VelDescriptor$0]
PP
NP
[Location$0]
drive (Vehicle$0, Location$0, VelDescriptor$0)
PatternSentence2
NP
[Pedestrian$0]
VP
[Walk$0]
AdvP
[VelDescriptor$0]
PP
NP
[Entity$0]
walk (Pedestrian$0, Entity$0, VelDescriptor$0)
PatternSentence3
NP
[Vehicle$0]
VP
[Drive$0]
PP
NP
[Location$0]
drive (Vehicle$0, Location$0)
Edit Cost (add) = 0.5
Edit Cost (delete) = 0,
Edit Cost (replace with child) = 0
Edit Cost (replace with others) = 1
Children (Go) = { Walk, Drive, ... }
Children (Entity) = { Vehicle, Pedestrian, ... }
Children (VelDescriptor) = { Slow, ... }
Children (Location) = { RightSide, ... }
TED (PatternSentence1) = 0.0
TED (PatternSentence2) = 2.0
TED (PatternSentence3) = 0.5
drive (Vehicle = Car1, Location = RightSide, VelDescriptor = Slow)

Fig. A.11. A test sentence is compared to a collection of pattern trees, each one associated to a generic predicate. The predicate of that pattern with a lowest TED specializes its predicate with information from the sentence.

7. Conclusions

We have presented a framework that may benefit research in the fields of segmentation, tracking, and behavior analysis. The system presented here can add virtual agents to available recordings using the presented framework, in order to evaluate the limitations of segmentation, tracking, and behavior understanding processes in terms of agent scalability, occlusion handling, and agent interaction. Users do not require expertise in computer graphics, given that the behavior of the virtual agents is controlled by NL sentences. Experimental tests and validation in indoor, street, and sports environments have showed the feasibility of the proposed approach.

The system semantically indexes the observed events. The taxonomy of events provides the space and validity of possible annotations for video sequences of a domain. The SGT acts as a content classifier, which semantically characterizes the temporal structure of video sequences. Thus, the resulting predicates can be identified as high-level semantic indexes, which facilitate further applications such as search engines and query-based retrieval of content. This scheme has been applied to video-surveillance. Future work will be devoted to reduce the amount of prior knowledge that needs to be specified to a given scenario. The automatic extraction of conceptual knowledge related to the scenario constitutes an interesting line of research nowadays, and would relax the requirements to apply the proposed system into new scenarios.

Acknowledgments

This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDI-Video project, and by the Spanish MEC under projects CICYT ERINYES TIN2009-14501-C02 and CONSOLIDER-INGENIO 2010 MIPRCV CSD2007-00018. We especially want to thank Prof. H.-H. Nagel for his creative discussions, guidance, and support.

Appendix A. Linguistic user interaction

This contribution incorporates a Natural Language Understanding (NLU) module that enables end-users to augment video sequences with virtual actors, in order to obtain complex augmented scenes.

NLU is typically considered a process of hypothesis management, in which given a textual NL input, the most appropriate interpretation out of a set of possibilities has to be chosen. In our case, the ontology of Table 1 specifies the domain of validity undertaken by the universe of possible user queries, and reduces them to a handleable space of situations.

The general operations conducted by the NLU module are shown in Fig. A.10. First, each sentence of the user is processed by a stemming algorithm based on parsing rules, and its contents are linked to concepts from the global ontology.[9] After that, the specific context of the sentence is found by relating the required referring expressions to entity instances, e.g., "*this agent*", "*the second person*", and "*last pedestrian*" are expressions that refer to specific agents. Lastly, the interpreted sentence is analyzed at a syntactic/semantic level, and its contents are assigned to the most suitable action predicate in order to generate virtual agents in the scene. Further information about these processes is detailed in Fernández (2010).

By linking each lemma to an ontological concept, we reduce the amount of interpretations of an input sentence to those admissible. However, it is possible that more than one word is directed to the same concept, e.g., pedestrian/person/walker→Pedestrian. In order to enhance the recognition of words, and to avoid extra scaling the coverage of the linguistic models, further lexical disambiguation is accomplished relying on the WordNet lexical database (Fellbaum, 1998). Lists of closely related words are retrieved using semantic metrics based on relationships such as synonymy and hypernymy. New candidates are evaluated to determine the ontological nature of an unknown word; as a result, the word is linked to a number of domain concepts that can explain it.

On the other hand, the assignment of linguistic content to an action predicate is achieved by (i) parsing the sentence into a dependency tree, and (ii) measuring its distance to a series of pattern trees, each one associated to a predicate from the ontology. This is done by a Tree Edit Distance (TED) algorithm (Bille, 2005) constrained by the ontology: the concepts at the leaves of the trees are aligned to each other and compared, and their disagreement is penalized. Penalties are high for absences, null for particularizations, and for generalizations they depend on the levels of difference in the hierarchy of concepts. Fig. A.11 depicts an example in which the concept *Car* augments the distance with pattern tree 2 having *Pedestrian* at the corresponding leaf, but specializes the general concept *Vehicle* at the same position in patterns 1 and 3 with distance zero. The pattern tree with lowest distance to the test tree is considered the best interpretation, and the fields of its associated predicate are particularized with specific information from the sentence.

References

Amato, A., Mozerov, M., 2008. Background subtraction technique based on chromaticity and intensity patterns. In: 19th International Conference on Pattern Recognition (ICPR'2008), Tampa, Florida, USA.

Arens, M., Nagel, H.-H. 2003. Behavioral knowledge representation for the understanding and creation of video sequences. In: Proceedings of the 26th German Conference on Artificial Intelligence, pp. 149–163.

Baiget, P., Fernández, C., Roca, X., Gonzàlez, J., 2009. Generation of augmented video sequences combining behavioral animation and multi-object tracking. Comput. Animation Virtual Worlds 20 (4), 473–489.

Bille, P., 2005. A survey on tree edit distance and related problems. Theor. Comput. Sci. 337 (1-3), 217–239.

Bindiganavale, R., Schuler, W., Allbeck, J.M., Badler, N.I., Joshi, A.K., Palmer, M., 2000. Dynamically altering agent behaviors using natural language instructions. In: AGENTS '00: Proceedings of the fourth International Conference on autonomous agents. ACM, New York, NY, USA, pp. 293–300.

Black, J., Ellis, T., Rosin, P. 2003. A novel method for video tracking performance evaluation. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), Citeseer, pp. 125–132.

Brand, M., Kettnaker, V., 2000. Discovery and segmentation of activities in video. IEEE Trans. PAMI 22 (8), 844–851.

Brémond, F., Thonnat, M., Zuniga, M., 2006. Video understanding framework for automatic behavior recognition. Behav. Res. Meth. 3 (38), 416–426.

Buxton, H., 2003. Learning and understanding dynamic scene activity: A review. Image Vis. Comput. 21 (1), 125–136.

Cheng, J., Moura, M., 1999. Capture and represention of human walking in live video sequences. IEEE Trans. Multimedia 1 (2), 144–156.

Colmerauer, A., 1990. Introduction to Prolog III. Commun. ACM 33 (7), 68–90.

Douze, M., Charvillat, V., 2006. Real-time generation of augmented video sequences by background tracking. Comput. Animation Virtual Worlds 17 (5), 537–550.

Fellbaum, C., 1998. WordNet: An electronic lexical database. MIT press Cambridge, MA, Cambridge, Massachusetts.

Fernández, C. 2010. Understanding image sequences: the role of ontologies in cognitive vision, Ph.D. thesis, Universitat Autònoma de Barcelona, Barcelona, Spain.

Fernández, C., Baiget, P., Gonzàlez, J. 2008. Cognitive-guided semantic exploitation in video surveillance interfaces. In: First International workshop on tracking humans for the evaluation of their motion in image sequences (THEMIS'2008), in conjunction with BMCV, vol. 1, Leeds, UK, pp. 53–60.

Francois, A., Nevatia, R., Hobbs, J., Bolles, R., Smith, J., 2005. VERL: An ontology framework for representing and annotating video events. IEEE Multimedia 12 (4), 76–86.

Galata, A., Johnson, N., Hogg, D., 2001. Learning variable-length markov models of behavior. Comput. Vis. Image Understanding 81 (3), 398–413.

Gonzàlez, J., Rowe, D., Varona, J., Roca, X., 2009. Understanding dynamic scenes based on human sequence evaluation. Image Vis. Comput. 27 (10), 1433–1444.

Irawati, S., Green, S., Billinghurst, M., Duenser, A., Ko, H. 2006. Move the couch where?: developing an augmented reality multimodal interface. In: Proceedings of ISMAR, vol. 6, Citeseer, pp. 183–186.

Klein, G., Murray, D. 2007. Parallel tracking and mapping for small AR workspaces. In: Proceedings of Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan.

Kojima, A., Tamura, T., Fukunaga, K., 2002. Natural language description of human activities from video images based on concept hierarchy of actions. Int. J. Comput. Vis. 50 (2), 171–184.

Liu, C., Freeman, W., Szeliski, R., Kang, S. 2006. Noise estimation from a single image. In: IEEE CVPR, pp. 901–908.

Ma, M., Mc Kevitt, P. 2004. Visual semantics and ontology of eventive verbs. In: Proceedings of the First International Joint Conference on Natural Language Processing, pp. 278–285.

Mokhber, A., Achard, C., Milgram, M., 2007. Recognition of human behavior by space-time silhouette characterization. Patt. Recognit Lett. 29 (1), 81–89.

Nagel, H.-H., 1988. From image sequences towards conceptual descriptions. Image Vis. Comput. 6 (2), 59–74.

Nagel, H.-H., 2004. Steps toward a cognitive vision system. AI-Magazine 25 (2), 31–50.

Nijholt, A., Zwiers, J., Peciva, J., 2009. Mixed reality participants in smart meeting rooms and smart home environments. Personal Ubiquitous Comput. 13 (1), 85–94.

Oliver, N., Rosario, B., Pentland, A., 2000. A Bayesian computer vision system for modeling human interactions. IEEE Trans. PAMI 22 (8), 831–843.

Piciarelli, C., Foresti, G.L., 2006. On-line trajectory clustering for anomalous events detection. Patt. Recognit Lett. 27 (15), 1835–1842.

Qian, H., Mao, Y., Xiang, W., Wang, Z., 2010. Recognition of human activities using SVM multi-class classifier. Pattern Recogn. Lett. 31 (2), 100–111.

Qureshi, F., Terzopoulos, D. 2005. Towards intelligent camera networks: A virtual vision approach. In: Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 177–184.

Roth, D., Koller-Meier, E., Van Gool, L., 2009. Multi-object tracking evaluated on sparse events. Multimedia Tools Appl., 1–19.

Schäfer, K., Brzoska, C., 1996. *F*-Limette: Fuzzy logic programming integrating metric temporal extensions. J. Symbolic Comput. 22 (5-6), 725–727.

Seo, Y., Choi, S., Kim, H., Hong, K.-S. 1997. Where are the ball and players? soccer game analysis with color-based tracking and image mosaic. In: Proceedings of ICIAP'97, vol. II, pp. 196–203.

Stauffer, C., Grimson, W., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Patt. Anal. Mach. Intell. 22 (8), 747–757.

Stiefelhagen, R. et al., 2006. The CLEAR 2006 Evaluation. In: Stiefelhagen, R., Garofolo, J.S. (Eds.), Multimodal technologies for perception of humans, 1st Int. evaluation workshop on CLassification of Events, Activities and Relationships (CLEAR), vol. 4122. LNCS, Springer, pp. 1–44.

Taylor, G., Chosak, A., Brewer, P. 2007. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In: IEEE CVPR, pp. 1–8.

Vezzani, R., Cucchiara, R., 2008. ViSOR: VIdeo Surveillance On-line Repository for annotation retrieval. In: IEEE International Conference on Multimedia and Expo. Hannover, Germany, pp. 1281–1284.

[9] To avoid excessive ambiguity when resolving the meaning of the inputs, this module accepts uniquely single (not compound) sentences from the end-users.