Video-Based Multi-Agent Traffic Surveillance System

Bruno Abreu, Luis Botelho, Andrea Cavallaro, Damien Douxchamps, Touradj Ebrahimi, Pedro Figueiredo, Benoit Macq, Benoit Mory, Luis Nunes, Javier Orri, Maria José Trigueiros, Ana Violante

MODEST Consortium
ADETTI (Lisbon, Portugal); EPFL (Lausanne, Switzerland); LEP (Paris, France); UCL (Louvain, Belgium)
http://www.tele.ucl.ac.be/MODEST/
Tel +351 217903906 – Fax +351 217903099

Luis.Botelho@iscte.pt

Abstract

This paper describes Monitorix, a video-based traffic surveillance multi-agent system. Monitorix agents are grouped in four tiers, according to the kind of information processing they perform: the sensors and effectors tier, the objective description tier, the application assistant tier, and the user assistant tier. The video analysis algorithms use an adaptive, data-driven, application independent approach to extract features from the video raw data. In spite of the diversity of agent tasks, adaptive learning algorithms are used in most cases. The integration of video analysis algorithms and agent technology is made via a special middle agent called Proxy. Monitorix is a fully decentralised multi-agent system living in a FIPA Platform and using FIPA Agent Communication Language. The Tracking of vehicles across non-overlapping cameras is performed by the Tracker agent, using a traffic model and learning algorithms that tune the model parameters.

Keywords Traffic Monitoring and Control, System Architectures, Imaging and Vision Enhancement, Surveillance, Multi-Agent System

1 Introduction

Monitorix is a video-based traffic surveillance multi-agent system that integrates video analysis and agent technology.

The Monitorix system has several cameras placed along a highway. The visual scopes of the cameras don't overlap. Each camera is attached to a processing unit (using a video capture card) that hosts a set of image algorithms (VOK, Video Object Kernel) that segment the scene and generate a sequence of descriptions. The descriptions of each camera are fed into a collection of agents running in a different computer. These agents perform higher level analyses of the descriptions and generate alarms signalling the occurrence of abnormal or otherwise relevant situations.

During the development of Monitorix we faced the following questions. What kind of video processing approach should be used? How to integrate video algorithms and agent technology? What is the most suitable agent architecture? How should agents communicate? How to track vehicles across several cameras whose scopes don't overlap?

We have decided to use data-driven, application independent video algorithms for feature extraction.

The VOK and the agents are integrated through a middle agent called Proxy. VOK runs as a server in the processing unit of the camera; Proxy runs as a client in a different computer.

Monitorix is a fully decentralised multi-agent system with respect to control and communication. Agents that need information request it from agents that provide it. The interaction of an agent with the others is controlled by a BDI[1]-like architecture [5] implemented by a production system [1].

Agents communicate using FIPA ACL[2] messages with SL[3] contents [4][2]. ACL is a communication language based on the speech-act theory [10]. SL is a content language based on a modal logic with belief, goal, intention, uncertainty and action operators [3]. Each agent defines a set of predicates, functions and actions that may be referred in the messages that it receives.

Vehicles are tracked across cameras by the Tracker agent, using an improvement of a traffic model presented in [7], whose parameters are continuously updated by learning algorithms.

At the task level, although the exact approach varies with the specific task, adaptive learning techniques are a common denominator in all levels of the system. Background extraction is adaptive. Statistical change detection is adaptive. The classification of mobile objects uses competitive learning algorithms. The computation of typical trajectories uses statistical adaptation. The tracking of mobile objects from one camera into the next updates the parameters of its prediction model, using a

Proceedings of the 2000 Intelligent Vehicles Conference, the Ritz-Carlton Hotel, Dearborn, MI, USA, October 4-5 2000

[1] BDI – Beliefs, Desires and Intentions
[2] ACL – Agent Communication Language
[3] SL – Semantic Language

combination of symbolic learning and genetic algorithms.

Section 2 describes the general architecture of the multi-agent system. Section 3 describes each agent of the system. Section 4 describes the video algorithms. Section 5 presents results. Finally, section 6 presents conclusions and discusses future research topics.

2 System Architecture

We have designed an open and generic multi-tier architecture for Surveillance Systems. Monitorix is a special case of this architecture.

The first tier of the generic architecture, **the Sensors and Effectors Tier**, consists of representatives (i.e., proxies) of the sensors and effectors, which deliver sensor output and receive requests for changes in the sensor parameters and/or actions to be performed in the environment by the effectors. The Monitorix system has only one agent of this tier, in each camera: the camera Proxy.

The task of the second tier, **the Objective Description Tier**, is the construction of an objective and possibly distributed representation of the observed physical environment. This tier refines the sensor information producing higher-level semantic descriptions independent of a particular application. This might help the sensors by providing feedback based on the conclusions regarding the coherence of the sensor's outputs. The majority of the Monitorix agents belong to this tier: LocalSite, Classifier, Behaviour and Tracker agents (see Figure 1 and Section 3 for details).

The third tier, **Application Assistant Tier**, analyses the information of the previous tiers in order to produce actions and decisions defined by the specific application. Typically, this will result in inputs to effectors. In the Monitorix system this could result in the generation of alarms when a car is driving dangerously. This tier should be capable of determining that a particular observed zigzag was not an instance of dangerous behaviour, but was due to a momentary obstruction of one lane.

In the fourth tier, **User Assistant Tier**, the user establishes application profiles that condition the interactions in lower tiers. There may be several types of user functionalities. An example from Monitorix is the generation of global traffic reports.

This architecture is an intelligent multi-agent system, where each of the functionalities is performed by one or more agents. The global task is realised by the co-operation among the agents, which is achieved by communication. The whole system relies on an agent **platform** that performs generic tasks on behalf of the agents, providing the means for decentralised co-operation mechanisms. We have chosen a **FIPA platform** because it is both a well-specified and an open platform.

In Monitorix, we demonstrate the effectiveness of this decentralised information analysis architecture by building at least one agent for each tier, as we can see in Figure 1.

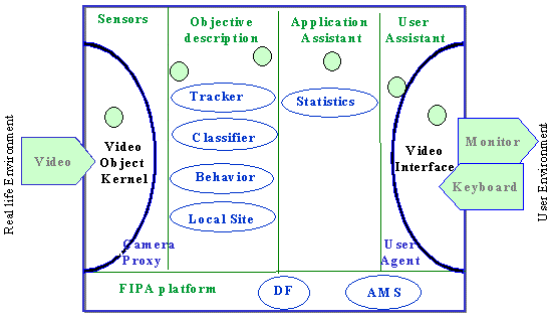

Figure 1 - Monitorix Architecture

A specific agent interaction protocol was adopted in the Monitorix architecture. According to this protocol, whenever possible, agents send persistent requests to providers, instead of having to repeat the same requests over and over. This is called the information-subscription protocol [2].

This architecture is **open** (agents communicate in a standard language; any agent can be plugged into the system and exchange information with other agents, with the help of ontology agents if needed); **flexible** (at any moment, we can add new functionalities or even new tiers, without changing previous work); and **scalable** (in general, the complexity of this system has a linear growth with more inputs, i.e. adding more sensors or agents to the system implies only a linear increase in communication since an agent doesn't usually communicate with all other agents).

3 Agents Description

In this section, we describe the agents of all tiers of the Monitorix system.

3.1 FIPA Platform Agents

A FIPA platform is a set of services provided by platform agents and by the platform itself. These services ease the development and the operation of multi-agent systems. The platform has two mandatory agents: the AMS (Agent Management System), and the DF (Directory Facilitator). The AMS provides platform management related services such as the registration of agents in the platform and a white pages service. A FIPA platform has exactly one AMS. The DF provides a yellow pages service. A platform must have at least one DF. All agents in the platform should register their services with the DF. Whenever an agent wants to know which agent provides a given service, it asks the DF. Besides, the AMS and the DF, the platform provides a message transport service. The agents in the platform just have to call an API that allows them to send messages to other agents in the same or in other platforms. In FIPA compliant platforms, agents communicate in FIPA ACL.

3.2 Objective Description Tier

This tier of Monitorix comprises the following agents: LocalSite, Classifier, Behaviour and Tracker.

LocalSite agent

The main goal of this agent is to serve as the repository of information regarding a site partially monitored by a camera. The information is maintained and made available to other agents in the form of instances of MPEG-7 like descriptors and description schemes, which describe the site. The type of information maintained regards the topology of the road system under surveillance, the main aspects of the road geometry, information about the division of the roads into lanes and their types (allowed traffic directions, whether it is legal to change to another lane, etc.), as well as information about the legal speeds allowed and the actual speeds practiced, and the typical trajectories of each class of vehicle. This agent also maintains information about which road segments are visible through the camera.

All information maintained by the LocalSite agent is available for all the other agents. Most notably the Behaviour and the Tracker agents resort to road topology and geometry in order to perform their tasks.

The LocalSite agent also provides the Proxy agent with the information necessary for it to label object descriptions with the lane in which they are located whenever they are observed.

The Behavior agent occasionally informs the LocalSite of updated typical trajectories and actual speeds of on sight lane segments.

Classifier agent

The Classifier agent sends classifications of mobile objects to any agent that requests that information. The classification may have one of the following values: CAR, TRUCK, MOTORBIKE, PERSON, TOOLARGE (possible under-segmentation error) and MOBILE_OBJECT (object not classified).

Each classification has a confidence factor in the interval [0..1], in which 0 means "no confidence", and 1 stands for "maximum confidence".

Each Classifier receives information regarding the size of the bounding box that involves each of the mobile objects present in a snapshot. This information is sent by the Proxy agent.

The Classifier uses tracking information generated by the Tracker agent and by other Classifiers about previous classifications of each mobile object (if they exist), which can improve the results of the classification procedure.

The classification methods are based on the matching of the observed size of a mobile object with previously gathered information of typical sizes for objects of all classes.

The class templates are updated using an unsupervised learning procedure, which initialises units with the typical measures foreseen for each class, and uses the object descriptions with higher confidence as inputs. This procedure may prove useful during the camera calibration stage and to provide better clustering of the object descriptions.

The adaptation of the class templates is based on well-known algorithms for competitive learning [6].

Behaviour agent

The main goal of the Behaviour agent is to build a description of the behaviour of mobile objects and forward it to any agent that requests it.

The Behaviour agent determines the typical trajectories of each class of mobile objects, describes the motion of individual mobile objects in terms of lane changes, speed and position, and detects specific events like zigzag driving, wrong lane driving and stopped vehicles.

To determine typical trajectories, the Behaviour uses position information sent by Proxy. The algorithm adjusts the trajectory whenever new position information is received.

The Behaviour also requests the Proxy to send it the speed of each mobile object. This allows determining the speed changes of each mobile object.

The Behaviour internal data structures organise mobile objects according to the classification received from Classifier.

Periodically, the Behaviour updates the minimal and maximum speeds for each lane and mobile object class. It also processes the number of lane changes. This is also done when mobile objects leave the scope of the camera.

Tracker agent

The tracker Agent is a multi-process agent, with the purpose of tracking the vehicles trajectory and speed, in order to predict their position, in the moment they will be observed by the next camera. The purpose of this process is to provide predictions to the Tracker agent in the next camera so that it can more easily identify the vehicles observed there.

The prediction process is composed by two algorithms: the prediction algorithm and the learning algorithm. The prediction algorithm is based on a traffic model with the following structure: $V_{t+1}=V_t+d\times(U(D)-V_t)$. V_n is the speed in the n^{th} instant of time, d is the braking or accelerating factor. This factor is dependant on U(D), the security speed which is dependent on the distance to the next vehicle. If the current speed is higher than U(D), then d is negative (the vehicle has to brake). Otherwise, d is positive and the vehicle will speed-up. Other factors were introduced: Lane Factor and Maximum Speed Factor.

The Lane Factor reflects the lane preferences of the vehicle. The Behaviour agent sends the lane occupation rate of each vehicle to the Tracker agent. The Tracker uses this information to predict if the vehicle will change lane or not. The prediction depends not only on the lane preferences of the vehicle but also on possible speed gains it may have if it changes lane and on a security factor.

The Maximum Speed Factor is just a limit for the speed of that vehicle.

The Learning Algorithm is a hybrid of a genetic algorithm and a symbolic algorithm (Candidate Elimination). The purpose of this algorithm is to readjust the factors of the model in order to improve the predictions. The Candidate Elimination Algorithm learns a set of rules that can be used to determine the qualitative update of each of the model parameters (i.e., higher, same, lower). This algorithm learns through examples tagged as negative or positive according to whether or not they meet given criteria. These examples are a set of attributes with their values. The algorithm creates two hypothesis spaces: the General and the Specific. Then it changes these spaces in order to generalise or constrain the values of the attributes according to new observations. The output is the intersection between the two spaces (the General with the Specific), which gives a set of attribute-value pairs reflecting the tendencies of the adjustments to be made for each parameter (e.g. ((break, higher), (acceleration, lower), (security-speed, same))). These tendencies are used to constrain the search space of the genetic algorithm, which through recombination and mutation computes the new values of the model parameters.

3.3 Application Assistant Tier: Statistics agent

The Statistics agent computes the frequencies of each class of vehicle and also the frequencies of each kind of behaviour. In the near future, this agent will also compute a pollution index based on the class and speed of observed vehicles. For the moment this is the only application agent.

3.4 User Assistant Tier: UserAgent

The UserAgent is the agent that represents the user within Monitorix. It takes the information from the other agents, processes and delivers it upon user requests.

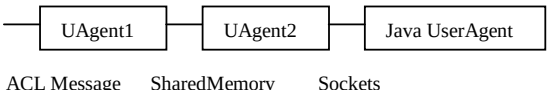

Figure 2 - UserAgent Architecture

The UserAgent is made up of three independent components (Figure 2). UserAgent1 interacts with the other agents via ACL. It carries out all information requests from the user side to the other agents in the system. The received information is stored in a shared memory until the UserAgent2 retrieves it. Thus, UserAgent1 works as an information repository. The use of shared memory between UserAgent1 and UserAgent2 provides a fast communication channel.

UserAgent2 takes information from UserAgent1 and prepares it to be sent to the Java UserAgent, which implements the graphical browser. The UserAgent2 acts as a wrapper of the Java UserAgent. Doing so, if one wants to change the browser, a simple modification in UserAgent2 would be required but UserAgent1 will remain unchanged. The communication with the browser is made via TCP/IP sockets, which allows having the browser on any computer connected to the network.

3.5 Sensors and Effectors Tier: Proxy

The Proxy agent implements the interface between the agents and the VOK algorithms, which may be seen as the sensors of the system. From the point of view of the agents, the VOK does not exist, all they see is the Proxy.

Conceptually, the interface service provided by the Proxy consists of two tasks: the delivery of image descriptions to the agents that subscribe them, and the delivery of feedback information provided by the agents to the VOK algorithms. Currently, the Proxy provides only the description delivery service.

The image is segmented in Snapshots, which contain a set of mobile objects. A mobile object is described by a set of attributes, namely *object-id*, *position*, *orientation*, *speed*, *colour* and *size*. Each of these attributes is itself described by a compound data structure. For instance, the position of an object contains its co-ordinates (x, y, z) and the road lane. All the physical attributes and sub-attributes of mobile objects are uncertain attributes (due to inaccuracies of the input information).

Agents that want to receive information from the VOK, ask the Proxy to define a filter that takes a Proxy object description as input and returns a requester object description as output. After the filter has been successfully defined, the requester agent asks the Proxy to apply the filter to the mobile object descriptions provided by the VOK and send it the results through information messages.

4 Video Object Kernel

Monitorix agents are fed with object descriptions shaped by the description scheme proposed in [9] for the representation of mobile objects. This description scheme is built from descriptors such as size, motion and position. Instances of these descriptors are extracted from 2D-object using three-dimensional reconstruction of objects.

In order to extract the moving objects from the input video, a segmentation procedure is required. A segmentation process leads to a partition of an image or a video sequence into regions according to a given criterion. Monitorix aims at locating mobile objects in the observed scene, thus a change detector can naturally drive the segmentation. Change detection analysis provides a classification of the pixels in the video sequence into one of two classes: moving objects (foreground) and background. Moving objects generate changes in the image intensity, and therefore motion detection is related to temporal change

detection. However, the relationship between motion and temporal changes is not unique. Temporal changes in two successive images occur not only in correspondence to moving objects, but also in two areas referred to as uncovered background and overlap of the same object. The uncovered background does not belong to a moving object, but it is detected as temporally changed. The overlap of two successive object images is hard to be detected as change, when the object is not sufficiently textured. These problems are less critical when the temporal changes are computed between the current image of the sequence and a reference frame that represents only the scene background. For this reason, Monitorix uses an adaptive background reference frame.

The adaptive computation of the background frame is an iterative process that refreshes, at an instant *n+1*, the background obtained from *n* previous frames of the sequence with the incoming *(n+1)* frame. The background accretion method uses a *blending formula* that weights pixels of the incoming frame, according to their chances to belong (or not) to the background (i.e. inlier/outlier discrimination). This is done by computing an error map that takes into account both changes with respect to the previously computed background and with respect to the previous frame, as shown in Figure 3.

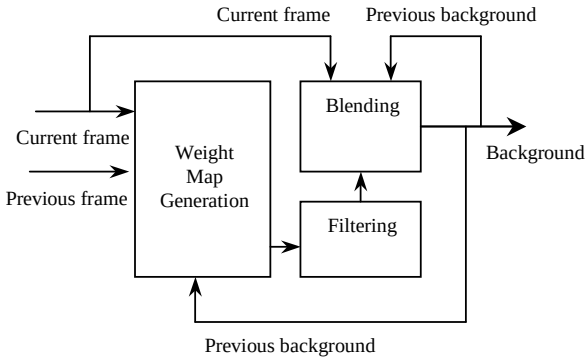

Figure 3 – Background adaptation

This adaptive refreshment of the background has two main advantages. It allows the change detection algorithm to rely on an effective reference frame and increases the robustness to slow changes in the illumination conditions (e.g. sunsets). Moreover, all the frames of the video sequence do not need to be used in this process: in our case, we refresh the background every two frames (the refreshment rate is thus half the video frame rate).

Besides the perturbation in the temporal changes introduced by a moving object, camera noise also heavily influences the results of the segmentation. In fact, due to the noise introduced by the acquisition process, a large number of pixels that do not correspond to a change in the real world appear as changed in the sequence. An *ad hoc* thresholding strategy is required to discriminate between pixels changed due to noise or to a real change. Thresholds have to be tuned manually according to the sequence characteristics and often need an update along the sequence itself. This main drawback limits this approach for an automatic application such as advanced video based surveillance. Moreover if the contrast of moving objects is not sufficiently high compared to the camera noise, there might not exist a unique threshold able to get rid of noise and preserve, at the same time, the motion information. To overcome these problems, an approach based on a statistical decision rule [11] is adopted to identify the moving objects. This compares through a significance test the statistical behaviour of a small neighbour (chosen as a square window) of each pixel position in the difference image with a model of the noise that could affect the difference image. This method does not require hand tuning of any threshold and does not increase severely the computational load when compared to simple threshold techniques.

Figure 4 - Descriptors visualisation

With the hypothesis of plane motion and rigid objects, one can infer from the 2D extracted objects their three-dimensional shape using a structure-from-motion scheme. We start with the segmented objects as input and estimate a dense sub-pixel motion field within the masks. The process used here is a growing-region Block Matching Algorithm (BMA) where the seeds are chosen from an interest map, computed with the Moravec operator [8].

From this motion information, the structure of the object can be determined without scale. The corresponding shape is then matched with the 2D mask to yield a scaled information.

From the resulting cloud of 3D points, we extract the 3D size (a 'bounding box'), position and speed. Colour is also extracted using specific techniques. Typical results are shown in Figure 4. From this motion field, the 3D structure of the rigid moving

object can be computed and given as input to the agents of the system.

5 Experiments and Results

Video sequences have been recorded in order to test the system. A static camera overlooking the highway was calibrated using on-scene reference points. Those consisted in calibration poles placed at known locations. Although the raw resolution of the camera was 704x576, it was reduced to the 352x288 CIF format to match the hardware currently available on most highways.

The development of Monitorix is not yet complete, although everything reported in this paper has already been implemented. Nevertheless, we have some partial results obtained by evaluating the Classifier agent. The preliminary results observed so far issued a 10% error in the classification process with a small set of very noisy data.

6 Conclusions

We have presented a road monitoring system developed on the basis of generic tools defined by MPEG-7 and FIPA standardisation bodies. MPG7-like descriptions and FIPA ACL messages with extended SL contents though not specifically developed for surveillance proved very useful for this application.

The main advantage of using agents on top of the video algorithms is twofold: the tasks performed by each agent can be incrementally sophisticated without that affecting the rest of the system; autonomous programs can be independently developed and tested.

Current efforts are being done towards a real-time implementation of the system. One approach to tackle this issue is to improve the performance of the system by implementing feedback loops between the agents and the video algorithms for the extraction of descriptions. Using feedback information and also model-based reasoning will enable the video algorithms to more rapidly focus their attention in more relevant objects.

The implementation of feedback loops between agents and video algorithms will be our next step. This will clearly reveal new advantages of integrating agent technology with video analysis. The main achievement will be the possibility of using low-level, application independent algorithms guided by high level model-based knowledge.

Acknowledgements

This work is partially funded by the ACTS 304 Project and by UNIDE/ISCTE. The authors would like to thank all the present and past members of the Modest project, which have directly or indirectly contributed to this work.

References

[1] Botelho, L.M. "A Control Structure for Agent Interaction". Proceeding of the Intelligent Vehicles Conference. Submitted. 2000.

[2] Botelho, L.M.; Lopes, R.; Sequeira, M.M.; Almeida, P.; and Martins, S. “Inter-agent communication in a FIPA compliant intelligent distributed dynamic-information system”. *Proc. 5th International Conference on Information Systems Analysis and Synthesis (ISAS99).* 1999.

[3] Bretier, P.; and Sadek, D. "A rational agent as the kernel of a cooperative spoken dialogue system: implementing a logical theory of interaction". *Proc.3th ATAL Workshop*. 1996

[4] Foundation for Intelligent Physical Agents. FIPA 97 Specification, Version 2.0 Part 2 "Agent Communication Language".1998

[5] Georgeff, M.P.; and Rao, A.S. "The semantics of intention maintenance for rational agents". *IJCAI'95,* p704-710. 1995

[6] Haykin S.,"Neural Networks: a comprehensive foundation". Prentice Hall. 1999

[7] Helbing, D.; and Huberman, B. A. "Coherent moving states in highway traffic". *Nature* **396**, 738-740. 1998

[8] Moravec H.P., “Towards automatic visual obstacle avoidance”. *IJCAI'77, pp. 584.* 1977.

[9] Piscaglia, P.; and Mory, B. ACTS AC304 project MODEST. “Xeno-Object Description Scheme”. P134, MPEG-7 Evaluation Ad Hoc meeting.1999

[10] Searle, J.R. 1969. Speech Acts. Cambridge University Press, 1969

[11] Ziliani, F.; and Cavallaro, A. "Image Analysis for Video Surveillance based on Spatial Regularization of a Statistical Change Detection". *Proc. 10th Int. Conf. on Image Analysis and Processing, Venice, pp. 1108-1111*.1999