A Machine-Vision System to Detect Unusual Activities Online at Vehicular Intersections

Un Sistema de Visión por Computadora para Detectar en Línea Actividades Inusuales en Intersecciones Vehiculares

Sandra Luz Canchola Magdaleno, Joaquín Salas Rodríguez, Hugo Jiménez Hernández, José Joel González Barbosa and Juan B. Hurtado Ramos
CICATA Qro. IPN
scanchola@ipn.mx, salas@ieee.org, hugojh@gmail.com, jgonzalezba@ipn.mx, jbautistah@ipn.mx

Article received on January 04, 2008; accepted on August 28, 2008

Abstract
In this article, we present a real-time machine-vision system to detect vehicles running on red light or performing forbidden turns at crossroads. The system operates during daytime by receiving video streams from two different sources. One of them is a camera viewing the crossroads to detect unusual activity, while a second camera watches the semaphore to keep synchrony with the traffic controller. The system performance and reliability have been tested on a real vehicular intersection during extended periods of time.
Keywords: Machine vision, real-time systems, unusual activity detection, automatic surveillance.

Resumen
En este artículo, presentamos un sistema de visión por computadora para detectar, en tiempo real, vehículos pasándose el alto o realizando vueltas prohibidas en cruceros viales. El sistema opera durante el día recibiendo secuencias de video de dos diferentes fuentes. Una de ellas observa el crucero para detectar actividad inusual, mientras que la segunda monitorea el semáforo para mantener la sincronía con el controlar de tráfico. El desempeño y confiabilidad del sistema han sido probados en una intersección vehicular real durante períodos de tiempo que abarcan días.
Palabras clave: Visión por computadora, sistemas en tiempo real, detección de actividad inusual, vigilancia automática.

1 Introduction

In this paper, we present a real-time machine-vision system to detect unusual activity at a semaphore-controlled vehicular intersection during daytime. The system includes two cameras. While one of the cameras oversees the crossroads to detect unusual activity, the other one watches a nearby semaphore to keep the first camera running in synchrony with the traffic lights. This kind of arrangement is especially suitable when interacting with the controller electronics is troublesome or expensive. Crossroads are an important part of the modern transportation infrastructure because (Fuerstenberg and Roessler 2006): a) they redirect the vehicular flow, b) people spend a lot of time at intersections, and c) they are the place where many accidents occur. We are interested in improving crossroad operation, in particular, through the detection of abnormal activities. We are mainly interested in activities such as vehicles making forbidden turns or running on a red light. The machine-vision system described is an extension of a previously presented model by Salas *et al.* (2007). In this paper, we describe the operation of the synchronization subsystem and its subsequent implementation in real time.

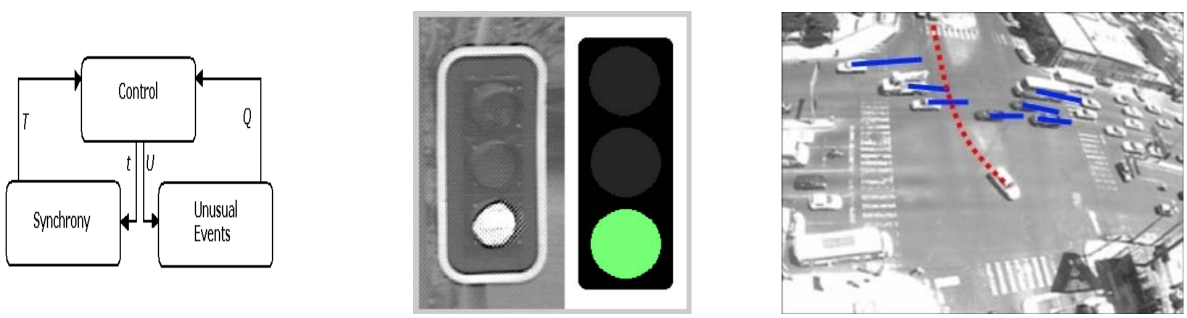
Detecting and tracking vehicles is a challenging problem due to a variety of factors that include the presence of shadows and the partial occlusion of moving objects. Kato *et al* (2004, 2002) and Kamijo *et al.* (2000) used Hidden Markov Models (HMM) and Markov random fields (MRF) to classify pixels into shadows, foreground, and background, even in the presence of occlusion. This low-level approach can be combined with high-level reasoning, as Cucchiara *et al.* (2000) have reported. During daytime, these authors use spatio-temporal analysis. At nighttime, they detect cars by the analysis of their headlights. This multilevel tracking model has also been explored by

Veeraraghavan *et al.* (2003). At low-level, they track blobs using heuristics to compensate for blob splitting and merging. At high level, they use Kalman filtering to track blobs.

There are many important applications that can result from monitoring vehicular traffic. Dailey *et al.* (2000) have estimated traffic speed using a sequence of images taken from an uncalibrated camera. They rely on a number of geometric, physical, and even legal constraints to simplify the problem. Such a technique could simplify the deployment of applications. Nonetheless, under some circumstances, the calibration can be achieved naturally using the scene intrinsic properties. For instance, Schoepflin and Dailey (2003) detect the speed of vehicles in a rectilinear roadway. Based on vehicle trajectories, they detect scene vanishing points to calibrate camera parameters. Also, Angel *et al.* (2003) use images collected from a helicopter to estimate speeds, travel times, densities, and queuing delays. Under some circumstances specialized hardware could be employed to solve specific tasks. For instance, Yamada and Soga (2003) developed an integrated single-chip visual sensor which detects direction and velocity of motion on a focal plane.

Different strategies have been utilized to detect unusual activity in video. For instance, while Jung *et al.* (2001) represent trajectories with polynomial models, Brand *et al.* (1997) discover patterns of activity by inferring the internal structure of an HMM. Also, Bioman and Irani (2005) interpret abnormal activity detection as the problem of building up the current image from pieces of data of previously extracted examples. Irregularities are detected when not enough evidence is found to compose the current observation. Similarly, Xiang and Gong (2005) base their approach on both, the discovery of the natural grouping of activity patterns and the introduction of an accumulating expression, where the relative importance of the visual information is taken into account for the detection of abnormalities. Instead of analyzing the resulting behavior, Dee and Hogg (2004) use a model of psychological function, where they attempt to deduce the cause of the behavior by comparing their observations with a model of those goals and obstacles known to be on the scene. On the other hand, Stauffer and Grimson (1999) learn patterns of activity by constructing a hierarchical classifier representing individual trajectories. Furthermore, Zhong *et al.* (2004) extended Stauffer and Grimson's method by considering spatio-temporal templates and their description in a color histogram as prototypes for a co-occurrence matrix.

In this paper, we describe a machine-vision system to detect, in real-time, unusual activities at vehicular intersections. The system illustrated in Fig. 1 is described in the rest of the document. Firstly, in §II, we review some of the techniques used. Then, in §III, we describe the module that deducts the traffic light status, useful for system synchronization. Next, in §IV, we explore the module that detects unusual events. In §V, we show experimental evidence concerning the performance of the system. Finally, we summarize our development, discuss the results and point out future lines of study.



(a) System Flow (b) Synchronization (c) Unusual Events

Fig. 1. The control unit iterates between two modules: synchronization and unusual events. Based on the elapsed time t, the change of the green light from off to on marks the beginning of the traffic controller light cycle. Once the time cycle T is estimated, the appropriate usual event space descriptor U is sent to the Unusual Events module which returns a description of the crossroads state Q

2 Preliminaries

Central in our method is the use of a mixture of Gaussians (MOG) to model both the intensity and the direction of motion at each particular pixel. Given a set of n observations, $\gamma_1,...,\gamma_n$, and a family $\Im$ of probability density functions on $\Re$, the problem is to find the probability density $f(\gamma)\in\Im$ that is most likely to have generated the given observations. In this scheme, each member of the family $\Im$ has the same general Gaussian form. Each member is distinguished by different values of a set of parameters Γ, Duda *et al.* (2001). That is,

$$f(\gamma;\Gamma)=\sum_{k=1}^{K} p_k\, g(\gamma;\mu_k,\sigma_k), \tag{1}$$

where $g(\gamma;\mu_k,\sigma_k)$ is a *one*-dimensional Gaussian function and $\Psi=(\psi_1,...,\psi_a)=((p_1,\mu_1,\sigma_1),...,(p_a,\mu_a,\sigma_a))$ is a $3a$ -dimensional vector containing the mixing probabilities p_a as well as the mean μ_a and standard deviation σ_a of the Gaussian function a in the mixture. When a new observation γ_i is available, it is compared against the parameters of the Gaussian models. After a considerable number of frames have been processed the MOG consists of a set of Gaussians along with their respective weight. The MOG is then pruned to eliminate Gaussians that have little support.

Another problem we face in our system is the detection of motion. The problem of estimating where a feature A moves from one frame to the next has many interesting facets that include objects undergoing partial or total occlusion or are being subject to complex appearance transformations. Lucas and Kanade (1981) propose a strategy for additive image alignment based on a Newton-Raphson type of formulation. In their report, the translation of a feature between frames is computed with a steepest descent minimization strategy. In principle, a more general transformation including affine wrapping and translation could be sought. However, in practice, Shi and Tomasi (1994) showed that this procedure can be numerically unstable. The procedure uses the optical flow constancy constraint, assuming that the feature reflected light intensity remains equal from frame to frame. That is, let $J_{i+1}(\mathbf{x})$ and $J_i(\mathbf{x})$ be two consecutive images, it has been shown (Lucas and Kanade 1981, Tomasi and Shi 1994) that the displacement $\mathbf{d}$ of a feature F can be computed using the recursive equation

$$\mathbf{d}_{j+1}=\mathbf{d}_j+Z^{-1}\mathbf{e}, \tag{2}$$

Where $Z=\sum_{x\in F}\begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}$ is the structural tensor, and $\mathbf{e}=\sum_{x\in F}(J_{i+1}(\mathbf{x})-J_i(\mathbf{x}))\mathbf{g}$ is a scaled version of $\mathbf{g}=(g_x,g_y)^T=\nabla J_i(\mathbf{x})$, the gradient.

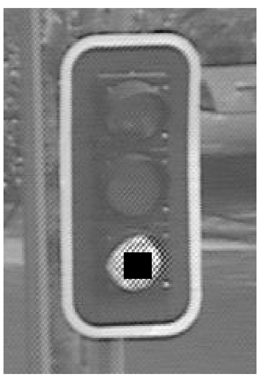
3 Synchronization

The semaphore green light status is used as the prime signal for synchronization. When the green light goes *on*, a synchronization signal is generated. Two continuous signals of this type span the time cycle T, whose value is an indicator of how long each individual light lasted in the previous light cycle at the crossroads. In what follows, we describe the synchronization signal, useful for a number of tasks that include vehicles running on red light.

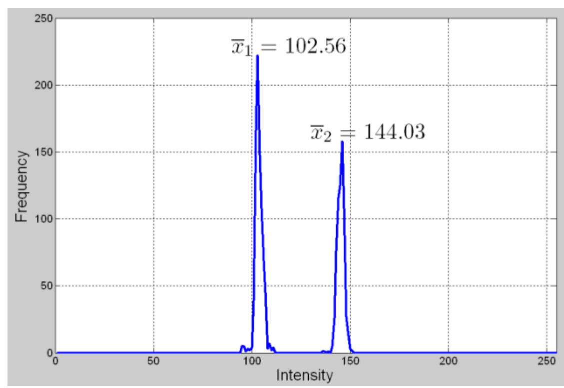
3.1 Training

At any given moment i, the semaphore state $S(i)$ can be either on or off, corresponding to classes ω_1 and ω_2, respectively. To characterize the semaphore state, we define a region of interest (ROI) inside the green light. Then, the ROI is sampled to find out its mean intensity value. Let $R_i(\mathbf{u})$ be the image ROI of dimensions $r \times s$ at instant i. The patch integer mean value can be computed as $\bar{x}_i = \lfloor 1/(rs) \sum_{u \in W} R_i(\mathbf{u}) \rfloor$, for the neighborhood W where the ROI is defined. After a considerable number n of frames has been processed, there will be a collection of observed intensity mean values $\mathbf{x} = \{\bar{x}_1, .., \bar{x}_n\}$. A histogram $h(a)$ can be used to describe the frequency at which a certain integer mean intensity value a appears. Next, the threshold φ that maximizes inter-class variance between both classes ω_1 and ω_2 is computed using Otsu's algorithm (Otsu 1978). It is assumed that the intensity distribution within the classes ω_1 and ω_2 follows a Gaussian distribution with mean values $\bar{x}_1$ and $\bar{x}_2$, and standard deviation s_1 and s_2, respectively. A new observation x_k will be assigned to the class ω_i with the minimum weighted Euclidian value $(\bar{x}_i - x_k)^2 / s_i^2$.

During learning initial mean class values $\bar{x}_1$ and $\bar{x}_2$ are acquired. The learning light module is only activated when the system starts operating. The results for a 120-s training period are shown in Fig. 2.



(a) Semaphore typical view



(b) The mean values $\bar{x}_1$ and $\bar{x}_2$ describe classes ω_1 and ω_2, respectively

Fig. 2. A camera is used to monitor the semaphore green light. In (a) the dark rectangle shows the area established to estimate the intensity value for the ROI inside the green light. In (b) a 120-hour observation period was used to generate the histogram presented here and to acquire the initial mean values

3.2 Operation

During operation, each mean class value is dynamically updated using the Estimation-Maximization procedure described in Stauffer and Grimson (2000). Let Q be a logical predicate that identifies the instant τ when the semaphore green light state has undergone a transition from off to on. That is

$$Q(\tau) = \begin{cases} true & (\mathrm{S_g}(\tau) \equiv on) \wedge (\mathrm{S_g}(\tau - 1) \equiv off), \\ false & otherwise. \end{cases} \tag{3}$$

The time difference $\tau_{j+1} - \tau_j$ between two consecutive times when transitional events take place, defines the time cycle T_{j+1}.

However, the semaphore green light blinks when it is transiting from *on* to *off* . This phase lasts approximately 0.5 s. This period is used to warn drivers that the yellow light is about to start. So, the condition given in (3) had to be adjusted to verify that at least its state was *off* in two frames $(\mathrm{S_g}(\tau-6) \equiv off) \wedge (\mathrm{S_g}(\tau-5) \equiv off)$ and then *on* in the next five frames $(\mathrm{S_g}(\tau-4) \equiv on) \wedge (\mathrm{S_g}(\tau-3) \equiv on) \wedge (\mathrm{S_g}(\tau-2) \equiv on) \wedge (\mathrm{S_g}(\tau-1) \equiv on) \wedge \ldots$ $(\mathrm{S_g}(\tau) \equiv on)$. This change warrants that the transition from *off* to *on* is not part of the blinking phase.

This module is initiated once the mean class values have been acquired or the system is out of synchrony. That is, if the number of unusual events occurring in one cycle is greater than a previously established value, $T_{unusuals}$, the beginning of the state *on* of the green light is searched and the time cycle T_{j+1} is determined. In this way, the system can operate for long periods of time and the detections of unusual events are accurate most of the time.

The traffic light control at a crossroads may be seen as a deterministic machine that cycles around a number of states $S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_n \rightarrow S_1$. When the time cycle T_{j+1} is known, the duration of every state S_i is also known because, in our case, there is a direct correspondence with the individual light time and the rest of the lights at the crossroads. In other words, the synchronization module provides a way to let the usual activity detection module know that a new state has arrived.

4 Unusual Activity Detection

Our unusual activity detection algorithm relies on what happens at pixel level. That is, during training, a background model is constructed to detect, via subtraction, the moving objects. Then, the moving objects are tracked down along their trajectory. Unusual activity is detected when there is a departure from the normal activity model that is learnt during training.

4.1 Training

In our model, the visual perception system consists of a double layer background structure (Salas, Jiménez, et al. 2007). In the first layer, the appearance of moving objects is obtained by subtracting the image of the static scenario to the current image in the sequence. In the second one, a probabilistic model of the orientations that moving objects could follow at each individual pixel, is constructed. In this manner, unusual events are described as foreground objects with outstanding regularity in the direction of motion.

4.2 Appearance Model

An important processing stage includes how to arrive to the initial background model (Gutchess, et al. 2001). In our system, we compute the mean value within a certain number of images, as in (Tai and Song 2004). For a given pixel at $\mathbf{x}$, its intensity values $J_k(\mathbf{x})$ in a certain time span are recorded in a histogram. The mean value is then used as an estimator of the appearance of the background model. This strategy works well whenever the vehicles are moving, which is assumed to be the case for the crossroads at the ROI.

Background updating is the problem of keeping an accurate representation of what remains fixed on the scene, despite variations in illumination conditions. Even in the situations where the image ROI is composed mainly of moving objects, it has been shown that a multiple layer background representation gives better results (Stauffer and Grimson 2000). In our case, a MOG model, as defined in Eq. (1) was used to update the background.

4.1.2 Motion Model

A second layer of the background is made out from the regular trajectories that moving objects describe on the scene, so the principal directions of motion are modeled with a MOG at every pixel in the ROI. In our case, the objects are assumed to be rigid and hence, although there are some effects due to perspective and scene location, the transformations observed involve primarily rotations and translations. Furthermore, we are assuming that a sufficiently high frame processing rate can be achieved, so that the appearance of the vehicles is effectively similar from frame to frame.

During training, the observed blob displacement is spread to all the pixels that compose it. Occlusion seems to be a crucial problem for robust tracking. Strategies to deal with it include the use of sub-features (Beymer, et al. 1997), high-level reasoning modules (Veeraraghavan, Masoud and Papanikolopoulos 2003), bounding box models (Atev, et al. 2005), temporal templates produced with interframe differences (Medioni, et al. 2001), active models (Johnson and Hogg 1996), and multiple hypotheses (Stauffer and Grimson 2000). In this study, we do not deal explicitly with occlusion because in the practice we have made two observations. Overall, occlusion accounts for a small portion of the cases, and second, it is common that unusual maneuvers are performed by isolated vehicles. When the latter is not the case, the event is likely to be detected as an unusual activity for all the vehicles in the group. This assumption reduces the computing burden without affecting the system effectiveness.

Each state S_k of the crossroads defines a usual activity space $U_k(\mathbf{x})$ which represents the description of the normal directions of motion present at each pixel location $\mathbf{x}$. Let us suppose that a blob is detected in the neighborhood A. When the blob is tracked down from frame to frame, an estimation of its current direction of motion θ is obtained. The MOGs corresponding to all the pixels $\mathbf{x} \in A$ are updated with the measured value of θ. This process is followed for a large number of frames corresponding to the same state S_k, thus creating in the process its respective usual space description U_k. At each pixel position, we have a MOG describing the usual directions of motions present in the training sequence.

4.2 Operation

During operation, observations are matched with the double layer background structure, Salas *et al.* (2007), constructed during training.

4.2.1 Appearance Model

During operation, each pixel of the ROI is compared against a background MOG model. If it fits, it is considered as a background pixel and is dynamically updated using the Estimation-Maximization procedure described in Stauffer and Grimson (2000). If not, it is treated as a moving pixel. The image of blobs obtained with all the moving pixels is analyzed in the second layer described in the next subsection.

4.2.2 Motion Model

During operation, the displacement computed when tracking a vehicle gives the direction of motion that is compared against the MOG of the pixel at $\mathbf{x}$, the pixel centroid. Thus, to a particular observation, a probabilistic measure that assesses its likelihood is assigned. Unlikely observations are called unusual events. This is different to other approaches, such as those of Chan *et al.* (2004) and Johnson and Hogg (1996), where the whole trajectory of the vehicle is needed before a decision can be taken. Since there is no need to build a high level representation of the situation, decisions can be taken faster and there are no problems associated with the comparison of curves or segments of curves.

Let $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be the ordered set of pixel points in the vehicle trajectory. The probability of observing this particular trajectory is $p(\mathbf{x}_1,...,\mathbf{x}_n) = p(\mathbf{x}_n \mid \mathbf{x}_{n-1},...,\mathbf{x}_1)\, p(\mathbf{x}_{n-1} \mid \mathbf{x}_{n-2},...,\mathbf{x}_1) ... p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{x}_1)$. By assuming a Markovian condition, where each observation depends solely on the previous one, the expression can be rewritten as $p(\mathbf{x}_1,...,\mathbf{x}_n) = p(\mathbf{x}_n \mid \mathbf{x}_{n-1})\, p(\mathbf{x}_{n-1} \mid \mathbf{x}_{n-2}) ... p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{x}_1)$. Since, $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$ are dependent, that is, the new position is the previous position plus a displacement. Then, $\mathbf{x}_i = \mathbf{x}_{i-1} + a_{i-1}\mathbf{u}_{i-1}$, where a is a constant related to the vehicle speed, and $\mathbf{u}_{i-1}$ a unitary vector describing the direction of motion, hence

$p(\mathbf{x}_i \mid \mathbf{x}_{i-1})$ can be written as $p(\mathbf{x}_i \mid \mathbf{x}_{i-1}) = p(a_{i-1}\mathbf{u}_{i-1} \mid \mathbf{x}_{i-1})$. Consequently, a possible measure for the likelihood of the trajectory X can be expressed by

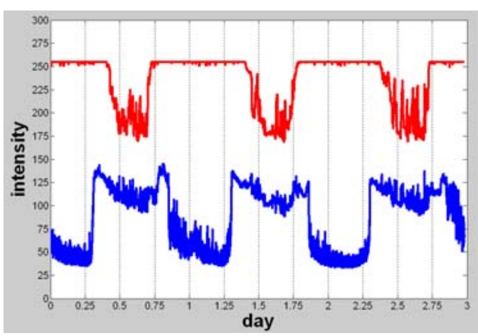
$$L(\mathbf{x}_1,...,\mathbf{x}_n) = \prod_{i=1}^{n-1} p(\mathbf{u}_i \mid \mathbf{x}_i). \tag{4}$$

The previous condition expresses temporal and spatial coherence of motion and can be part of the information carried out by the blob being tracked. When the score drops below a certain threshold for a number of consecutive frames, an unusual event declaration can be emitted.

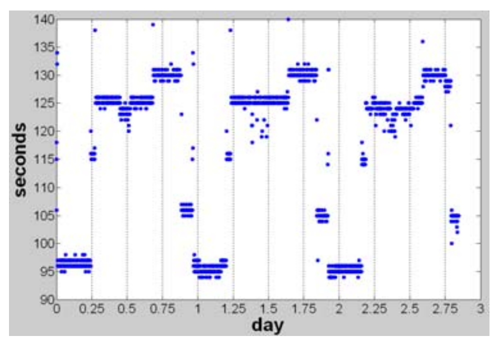
At each specific state S_k of the crossroads, certain routes may be considered normal. Skipping a red light or making a forbidden turn may be considered abnormal because either it is happening in the wrong moment or because it is infrequent. The synchronization vision system described in the previous section provides a way to let the usual activity detection module know which state is active.

5 Experimental Results

For experimentation, we set up a system to monitor a heavily transited crossroads in the city of Querétaro, México. The system consists of two cameras and a desktop computer. One of the cameras was placed on top of a tower, about 28 m above ground level. The second one was located approximately 6 m above ground level constantly viewing a semaphore located 42 m away. The computer had an Intel processor running at 3GHz, 2GB of internal RAM, and a Matrox Corona II frame grabber. The computer programs were written in Visual C++ to process gray scale images with a resolution of 320 columns times 240 rows.



(a) The upper and lower lines show the green light descriptor *on* and *off*, respectively



(b) Traffic light controller time cycle

Fig. 3. A camera is used to monitor the semaphore to synchronize the detection of unusual activities. A 72-hour observation period was used to generate the graphs shown in (a) and (b)

To set the system, a small area was sampled and its mean intensity value was computed to determine whether the light was *on* or *off*. Due to vibrations, there was a perceptible change in the position of the semaphore relative to the camera. We therefore defined a highly distinctive area T in the semaphore that was tracked down during the system operation. Relative to it, we sampled another area L that would tell us whether the light was *on* or *off*. As might be predicted, reduced activity during nighttime made the semaphore steadier. The observation started on a Saturday at 22:53:12h and ended on the next Tuesday at 23:31:08h, about 72h later. Overall, the horizontal and vertical displacements did not exceed the six pixels relative to the original feature position.

The green light status was monitored to determine whether it was *on* or *off*. During training, we took images for about 120 s. The sampling and posterior characterization created a bimodal histogram; the distinction between the *on*/*off* status was done using Otsu's thresholding algorithm (Otsu 1978). Subsequently, the class descriptor was updated following an online Estimation-Maximization criteria. Fig. 3(a) shows the behavior of the two descriptors for extended periods of time. During nighttime, the *on* class presented a large intensity value (almost the maximum illumination level, 255), while the *off* class remained low. On the other hand, the *off* class had a noticeable increase in value at around 8h while an abrupt decrease was detected around 18h. In between, both classes showed a parabolic-like behavior with a minimum at around 14:30h. In any case, a large distance between both values was measured during the experiment which made observations easier to classify. The semaphore was monitored during three consecutive days and the transition from *off* to *on* was recorded. The result is plotted on Fig. 3(b). During daytime, the vision system worked well. It extracted uniform descriptors that clustered around 95 s, 110 s, 130 s and 135 s. The current implementation achieved a peak performance of about 20 frames a second.

Once the time cycle was characterized, it was possible to start detecting unusual events such as forbidden turns or vehicles running on red light. The results reported in this article, are based on a sequence of 113,195 images taken in an interval that started on a Tuesday at 14:33:25 h and ended on the next Friday at 10:00:00 h. One must bear in mind that the traffic light control is composed of three states and the sequence has more than 870 complete cycles around these states. Previously, 10 cycles were used for training and the complete sequence for testing. Each training cycle sequence is formed of subsequences corresponding to the three different states. The subsequences of the same state were processed to obtain the usual event space for each particular state. As a result of the training phase, we obtained (a) a region of interest, (b) an initial model of the background, and (c) a description of the normal event space for each of the individual states that are part of the traffic light controller cycle. Fig. 4 shows the ROI obtained and an example of foreground objects.

A preliminary 120-s period, in both the training and testing sequence, was used for background initialization. The most frequent gray level for each pixel in the image was computed. Then, a MOG model was used to interpret the variations observed along the sequence. When the variations could be interpreted by a particular Gaussian model, the sample was used for learning. Otherwise, it was assumed that a foreground object was occluding the background. During operation, the event space was switched to the image corresponding with a state change. The appropriate event space was loaded and the execution continued. So the observed events were compared with respect to what was considered normal for that particular state. The probabilities along the trajectory were evaluated and those with low values were considered unusual events. Fig. 5 shows some examples of unusual activity detected with our system. The current implementation performs about nine frames a second. The numerical results are summarized in Table 1. Two kind of unusual event are reported (a) running on red light and (b) forbidden movement that refers to vehicles performing movements not allowed in the current state. The last two columns are false positives (Table 1,c), that means false detections. Most of the time, they are caused by irregular movements of the vehicle or by blob splitting. In all cases, the percentage (%), was calculated considering the total of the images analyzed each day.

Table 1. Statistics of the performed experiment. Four days are considered in the interval. The fourth column shows the number (#) and percentage (%) of vehicles running on red light. The fifth column shows the number (#) and percentage (%) of vehicles doing forbidden movements. The last pair of columns shows the number (#) and percentage (%) of false unusual detected events

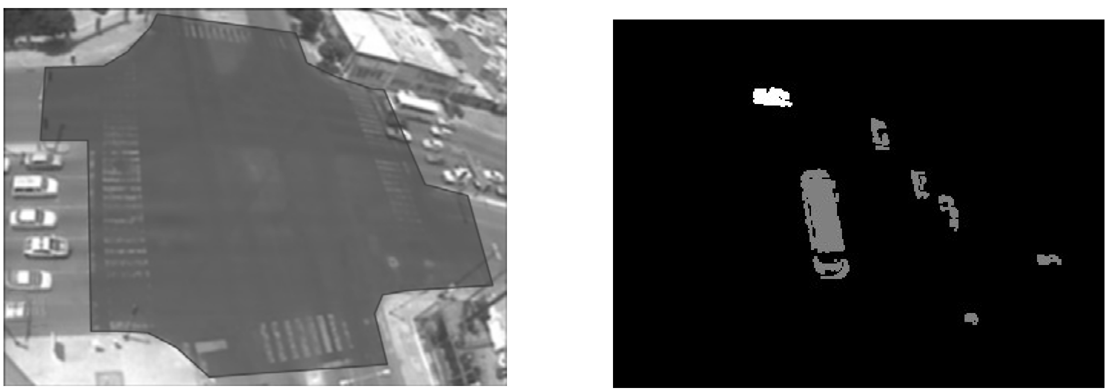
Day	Hours	Images	Red light (a)		Forbidden movement (b)		False positives (c)	
		#	#	%	#	%	#	%
1	4.5	15,995	125	0.781	110	0.688	463	2.895
2	12.0	43,200	161	0.372	354	0.819	888	2.056
3	12.0	43,200	151	0.349	261	0.604	662	1.532
4	3.0	10,800	40	0.370	100	0.926	322	2.981
Total	31.5	113,195	477	0.421	825	0.729	2,335	2.062

Conclusion

In this document, we present a machine-vision system suitable for the detection, in real-time, of unusual events occurring at a variable time cycle, traffic-light controlled, vehicular intersection. The method is based on the local processing (at pixel level) of the detected direction of motion at a crossroads. The use of a double layer background model, which combines appearance and motion, resulted in a model suitable for online processing. Under our perspective, unusual events were readily identified as outstanding objects, in the usual space description, either because they did not happen at the right time or because there were not enough samples of them during training.

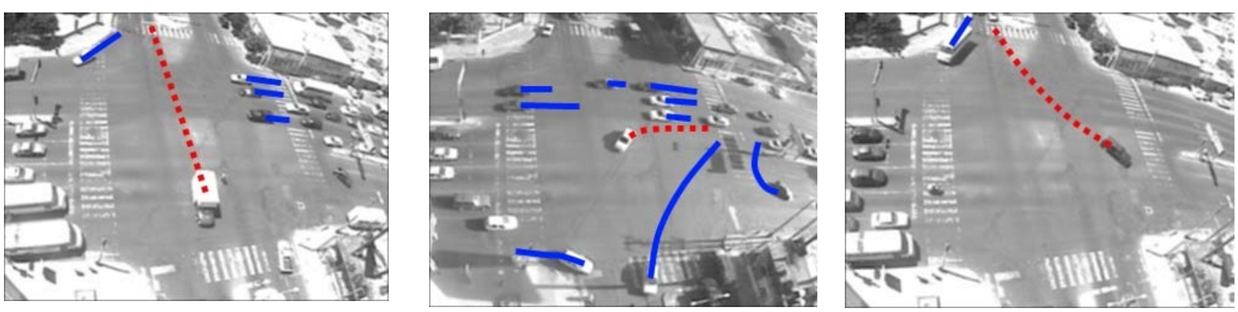
Since the time cycle is variable and the traffic light controller electronics is not accessible, a vision system was developed to monitor the green light status. The current approach works fine during daytime. With the system, we were able to discover the predominant pattern for light switching. Our strategy for discovering the offset between the computer and the crossroads controller time could be used to devise intelligent strategies to monitor both the synchronization factor and other image analysis tasks related to it, such as vehicular flow.

The current implementation provides an excellent test bed for studies on pattern learning that occur at different time spans, such as days or weeks. We aim to develop algorithms to understand what happens at different time scales in order to implement an automated deduction of a normal/abnormal situation with respect to a given time frame.



(a) ROI (b) Foreground objects

Fig. 4. Moving objects can be detected by subtracting the current image from the background model. The result is segmented into groups of connected pixels. This procedure is useful both to detect moving objects in a region of interest (ROI) and to update the background model considering only those regions in which variations are small



(a) Vehicle is running on red light. (b) Vehicle is making a too-wide turn. (c) Vehicle is making a forbidden turn.

Fig. 5. Some unusual events detected with our method. The dotted line shows the unusual trajectory

References

1 **Angel, A., M. Hickman, P. Mirchandani, and D. Chandnani.** "Methods of Analyzing Traffic Imagery Collected from Aerial Platforms." *IEEE Transactions on Intelligent Transportation Systems* 4, no. 2 (2003): 99 - 107.

2 **Atev, S., H. Arumugam, O. Masoud, R. Janardan, and N. P. Papanikolopoulos.** "A Vision-Based Approach to Collision Prediction at Traffic Intersections." *IEEE Transactions on Intelligent Transportation Systems* 6, no. 4 (2005): 416–423.

3 **Beymer, D., P. McLauchlan, B. Coifman, and J. Malik.** "A Real Time Computer Vision System for Measuring Traffic Parameters." *Computer Vision and Pattern Recognition.* San Juan, Puerto Rico: IEEE Computer Society, 1997. 495–501.

4 **Boiman, O., and M. Irani.** "Detecting Irregularities in Images and in Video." *International Conference on Computer Vision.* Beijing, China: IEEE Computer Society, 2005. 462–469.

5 **Brand, M., N. Oliver, and A. Pentland.** "Coupled Hidden Markov Models for Complex Action Recognition." *Computer Vision and Pattern Recognition.* Los Alamitos, CA, USA: IEEE Computer Society, 1997. 994-999.

6 **Chan, M. T., A. Hoogs, J. Schmiederer, and M. Petersen.** "Detecting Rare Events in Video using Semantic Primitives with HMM." *International Conference on Pattern Recognition.* Cambridge, UK: IEEE Computer Society, 2004. 150–154.

7 **Cucchiara, R., M. Piccardi, and P. Mello.** "Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System." *Transactions on Intelligent Transportation Systems.* Beijing, China: IEEE Computer Society, 2000. 119 – 130.

8 **Dailey, D. J., F. W. Cathey, and S. Pumrin.** "An Algorithm to Estimate Mean Traffic Speed using Uncalibrated Cameras." *IEEE Transactions on Intelligent Transportation Systems* 1, no. 2 (2000): 119-130.

9 **Dee, H., and D. Hogg.** "Detecting Inexplicable Behaviour." *British Machine Vision Conference.* Kingston, UK, 2004. 477–486.

10 **Duda, R.O., P.E. Hart, and D.G. Stork.** *Pattern Classification.* Wiley, 2001.

11 **Fuerstenberg, K. C., and B. Roessler.** "Advanced Intersection Safety - The EC project INTERSAFE." *Intelligent Vehicles Symposium.* 2006. 89 – 93.

12 **Gutchess, D., M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. K. Jain.** "A Background Model Initialization Algorithm for Video Surveillance." *International Conference on Computer Vision.* Vancouver, Canada: IEEE Computer Society, 2001. 733 – 740.

13 **Johnson, N., and D. C. Hogg.** "Learning the Distribution of Object Trajectories for Event Recognition." *Image and Vision Computing* 14, no. 8 (1996): 609–615.

14 **Jung, Y.K., K.W. Lee, and Y.S. Ho.** "Content-Based Event Retrieval using Semantic Scene Interpretation for Automated Traffic Surveillance." *IEEE Transactions on Intelligent Transportation Systems* 2, no. 3 (2001): 151 – 163.

15 **Kamijo, S., Y. Matsushita, K. Ikeuchi, and M. Sakauchi.** "Traffic Monitoring and Accident Detection at Intersections." *IEEE Transactions on Intelligent Transportation Systems* 1, no. 2 (2000): 108–118.

16 **Kato, J., T. Watanabe, S. Joga, J. Rittscher, and A. Blake.** "An HMM-Based Segmentation Method for Traffic Monitoring Movies." *IEEE Pattern Analysis and Machine Intelligence* 24, no. 9 (2002): 1291 – 1296.

17 **Kato, J., T. Watanabe, S. Joga, Y. Liu, and H. Hase.** "An HMM/MRF-Based Stochastic Framework for Robust Vehicle Tracking." *IEEE Transactions on Intelligent Transportation Systems* 5, no. 3 (2004): 142 – 154.

18 **Lucas, B. D., and T. Kanade.** "An Iterative Image Registration Technique with an Application to Stereo Vision." *Image Understanding Workshop.* Washington, DC, USA: Defense Advanced Research Projects Agency, 1981. 121–130.

19 **Medioni, G. G., I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia.** "Event Detection and Analysis from Video Streams." *IEEE Pattern Analysis and Machine Intelligence* 23, no. 8 (2001): 873–889.

20 **Otsu, N. "A Threshold Selection Method from Gray Level Histograms."** *IEEE Transactions on Systems, Man and Cybernetics* VIII (1978): 62–66.

21 **Salas, J., H. Jiménez, J. González, J. Hurtado, and S. Canchola.** "A Double Layer Background Model to

Detect Unusual Events." Edited by Springer Berlin / Heidelberg. *Lecture Notes in Computer Science* 4678 (2007): 406-416.

22 **Schoepflin, T., and D. Dailey.** "Dynamic Camera Calibration of Roadside Traffic Management Cameras for Vehicle Speed Estimation." *IEEE Transactions on Intelligent Transportation Systems* 4, no. 2 (2003): 90 – 98.

23 **Stauffer, C., and W. E. L. Grimson.** "Adaptive Background Mixture Models for Real-Time Tracking." *Computer Vision and Pattern Recognition.* Fort Colins, CO., USA: IEEE Computer Society, 1999. 246–252.

24 **Stauffer, C., and W. E. L. Grimson.** "Learning Patterns of Activity Using Real-Time Tracking." *IEEE Pattern Analysis and Machine Intelligence* 22, no. 8 (2000): 747–757.

25 **Tai, J.C., and K.T. Song.** "Background Segmentation and its Application to Traffic Monitoring using Modified Histogram." *Int. Conf. on Networking, Sensing and Control.* Taipei, Taiwan, 2004. 13 – 18.

26 **Tomasi, C., and J. Shi.** "Good Features to Track." *Computer Vision and Pattern Recognition* . Seattle, Washington, USA: IEEE Computer Society, 1994. 593–600.

27 **Veeraraghavan, H., O. Masoud, and N. P. Papanikolopoulos.** "Computer Vision Algorithms for Intersection Monitoring." *IEEE Transactions on Intelligent Transportation Systems* 4, no. 2 (2003): 78–89.

28 **Xiang, T., and S. Gong.** "Video Behaviour Profiling and Abnormality Detection without Manual Labelling." *International Conference on Computer Vision.* Beijing, China: IEEE Computer Society, 2005. 1238–1245.

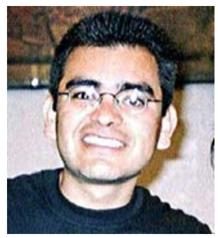
29 **Yamada, K., and M. Soga.** "A Compact Integrated Visual Motion Sensor for its Applications." *IEEE Transactions on Intelligent Transportation Systems* 4, no. 1 (2003): 35 – 42.

30 **Zhong, H., J. Shi, and M. Visontai.** "Detecting Unusual Activity in Video." *Computer Vision and Pattern Recognition* . Washington, DC., USA: IEEE Computer Society, 2004. 819–826.

***Sandra Luz Canchola Magdaleno** obtained her Master degree in Distributed Software Engineering from UAQ in 2004. She is currently a PhD. candidate from CICATA Querétaro-IPN. Her main interests are Image processing, Mathematical Morphology and Automatic Surveillance.*

***Joaquín Salas Rodríguez** obtained his Ph.D. degree from ITESM in 1996. He has worked at IPN since 1997. He has been visiting scholar at the Universitat Autonoma de Barcelona, PARC (formely Xerox PARC), Oregon State University, Stanford University, and the Ecole Nationale Superiore des Telecommunications at Brest. He has developed image analysis projects for companies including TAMSA, NHUMO, and IFE. He has published 30 papers on journals and international congresses. Since 1995, he is member of the Mexican National Researchers System.*

***Hugo Jiménez Hernández**, Engineering in Computer Science (2000) at Regional Technological of Querétaro, Master in Computer Science (2004) at Center of Research in Computer Sciences IPN, México, D.F., Nowadays he is a PhD. candidate from CICATA Querétaro-IPN. His research include Automatic Activities Detection, Associative Memories, and Time Series Analysis.*



***José Joel González Barbosa** was born in Guanajuato, Mexico, in 1974. He received the M.S. degree in Electrical Engineering from the University of Guanajuato, Mexico, and Ph.D. degree in Computer Science and Telecommunications from National Polytechnic Institute of Toulouse, France, in 1998 and 2004, respectively. He is an Associate Professor at the CICATA Querétaro-IPN, Mexico, where he teaches courses in Computer Vision, Image Processing and Pattern Classification. His current research interests include perception and mobile robotics.*

***Juan B. Hurtado Ramos.** He received the B.S. degree in Communications and Electronics Engineering in 1989 by the University of Guadalajara. He obtained his Ph.D. degree from the Research Center in Optics of the University of Guanajuato in 1999. He is currently a Profesor-Researcher of the CICATA Querétaro-IPN, Mexico within the Image Analysis group. His personal interests are mainly in the field of Metrology using optical techniques. Since 1998, he is member of the Mexican National Researchers System.*