Context-Independent Real-Time Event Recognition: Application to Key-Image Extraction

Aishy Amer [a] Eric Dubois [b] Amar Mitiche [a]

[a] INRS-Télécommunications; Montréal, Qc, H5A 1K6 Canada
[b] University of Ottawa; Ottawa, On, K1N 6N5 Canada
Email: *amer@inrs-telecom.uquebec.ca**

Abstract

The significant increase of video data in various domains requires effective ways to represent video by its semantic content. This paper proposes a system to extract useful events defined by approximate but efficient world models, independently of context. Changes and the behavior of low-level features of the scene's objects are continually monitored. When certain conditions are met, events related to these conditions are detected. Proposed events are sufficiently broad to assist surveillance and retrieval of video shots. Extensive experimentations on more than 10 indoor and outdoor video shots containing a total of 6371 images including sequences with noise and coding artifacts have demonstrated the reliability and the real-time response (up to 10 frames per second) of the proposed system.

1. Motivation

The significant increase of video data in various domains, such as surveillance and video retrieval, requires effective, efficient, and automatic ways to extract high-level video features and represent video shots. A video shot consists, in general, of moving objects [4], their low and high-level features within a given environment and context. Studies have shown that low-level features are not sufficient for effective video representation and that objects need to be assigned high-level features as well [5, 6].

High-level object features are generally related to the movement of object and are divided into context independent and context dependent features. Context independent features include object movement, activity, action [2], and related events. High-level features are generally applicable when they convey fixed meaning independently of context.

*This work was supported, in part, by the the Natural Sciences and Engineering Research Council of Canada under Strategic Grant SRT224122 and Research Grant OGP0004234.

An event expresses a particular behavior of a finite set of objects in a sequence of a small number of consecutive images of a video shot. An event consists of context-dependent and context-independent components associated with a time and location. For example, a *deposit* event has a fixed semantic interpretation (an object is added to the scene) common to all applications but *deposit of an object* can have a variable meaning in different contexts. In the simplest case, an event is the appearance of a new object into the scene or the exit of an object from the scene. In more complex cases, an event starts when the behavior of objects changes.

An event detection technique should automatically and efficiently provide generally useful events. In the remainder of this paper, Section 2 discusses related work, Section 3 describes a real-time system to detect useful events based on motion and other object features, Section 4 presents experimental results, and Section 5 concludes the paper.

2. Related work

There has been little work on context-*independent* event detection. The system in [3] is based on motion detection and tracking using prediction and nearest-neighbor matching. The system is able to detect basic events such as *deposit*. It can operate in simple environments where one human is tracked and translational motion is assumed. It is limited to applications of indoor environments, cannot deal with occlusion, and is noise sensitive. Moreover, the definition of events is not widely applicable.

The event detection system for indoor surveillance applications in [7] consists of object extraction and event detection modules. The event detection module classifies objects using a neural network. The classification includes: *abandoned object*, *person*, and *object*. The system is limited to one *abandoned object* event in unattended environments. The definition of *abandoned object* is specific to a given application. The system cannot associate abandoned objects and the person who deposited them.

1051-4651/02 $17.00 © 2002 IEEE

3. Proposed approach

In [1] we developed a computational framework to extract low and high-level features to represent video shots. The framework in [1] consisted of integrated modules for object segmentation, motion estimation, object tracking, and region merging. The purpose of this study is to use this framework to detect a number of useful events, independently of context. Event detection is performed by integrating object and motion features, i.e., combining trajectory information with spatial features, such as size and location. Objects and their features are represented in temporally linked lists. These lists are monitored to detect events as they occur. Here following are the definition of the events that the current system detects automatically. Let $I(n)$ be an image in a video shot and O_i a segmented object in $I(n)$ [1].

Enter An object, O_i, *enters* the scene at time n if all the following conditions are met:

- $O_i \in I(n)$,
- $O_i \notin I(n-1)$, i.e, zero match $M_0 :\dashv O_i$ meaning O_i cannot be matched to any object in $I(n-1)$, and
- c_i is at the border in $I(n)$. c_i is the centroid of O_i.

Enter is detected when a portion of object becomes visible.

Appear An object, O_i, *appears* at time n if

- $O_i \in I(n)$,
- $O_i \notin I(n-1)$, i.e., zero match $M_0 :\dashv O_i$, and
- c_i is *not* at the border in $I(n)$.

Exit (leave) An object, O_p, *exits* at time n if

- $O_p \in I(n-1)$,
- $O_p \notin I(n)$, i.e., zero match $M_0 : O_p \dashv$,
- c_p is at the image *border* in $I(n-1)$, and
- $g_p > t_g$ where g_p is the age of O_p and t_g a threshold.

Disappear An object, O_p, *disappears* at time n if

- $O_p \in I(n-1)$,
- $O_p \notin I(n)$, i.e., zero match $M_0 : O_p \dashv$,
- c_p is *not* at the border in $I(n-1)$, and
- $g_p > t_g$.

Move An object, O_i, *moves* at time n if

- $O_i \in I(n)$,
- $M_i : O_p \rightarrow O_i$ (a function assigning O_p at time $n-1$ an object O_i at time n) where $O_p \in I(n-1)$, and
- the median of the motion magnitudes of O_i in the last k images is larger than a threshold t_m. Typical values of k are three to five and t_m is one. There is no delay to detect this event because motion data at previous images are available.

Stop An object, O_i, *stops* at time n if

- $O_i \in I(n)$,
- $M_i : O_p \rightarrow O_i$ where $O_p \in I(n-1)$,
- the median of the motion magnitudes of O_i in the last k images is less than a threshold t_{m_s}.

Occlude/occluded Let $O_{p_1}, O_{p_2} \in I(n-1)$, $M_i : O_{p_1} \rightarrow O_i$ where O_i results from the occlusion of O_{p_1} and O_{p_2} in $I(n)$, $d_{p_{12}}$ the distance of the centroids of O_{p_1} and of O_{p_2}, $w = (w_x, w_y)$ the current displacement of O_{p_1}, i.e., between $I(n-2)$ and $I(n-1)$, and $d_{r_{\max}}$ ($d_{r_{\min}}$) is the vertical displacement of the lower (upper) row, $d_{c_{\max}}$ ($d_{c_{\min}}$) is the horizontal displacement of the right (left) column of O_{p_1}. Object occlusion is declared if

$$\begin{array}{l} ((|w_y - d_{r_{\max}}| > t_1) \wedge (d_{r_{\max}} > 0) \wedge (d_{i_{12}} < t_2)) \quad \vee \\ ((|w_y - d_{r_{\min}}| > t_1) \wedge (d_{r_{\min}} > 0) \wedge (d_{i_{12}} < t_2)) \quad \vee \\ ((|w_x - d_{c_{\max}}| > t_1) \wedge (d_{c_{\max}} > 0) \wedge (d_{i_{12}} < t_2)) \quad \vee \\ ((|w_x - d_{c_{\min}}| > t_1) \wedge (d_{c_{\min}} > 0) \wedge (d_{i_{12}} < t_2)) \end{array} \tag{1}$$

t_1 and t_2 are thresholds. With occlusion, at least two objects are involved where one is moving. With two objects, the object with the larger area is defined as the *occluding object*, the other the *occluded object*.

Remover/removed Let $O_i \in I(n)$ and $O_p, O_q \in I(n-1)$ with $M_i : O_p \rightarrow O_i$. O_p *removes* O_q if

- O_p and O_q were occluded in $I(n-1)$,
- $O_q \notin I(n)$, i.e., zero match $M_0 : O_q \dashv$, and
- the area A_q of O_q is smaller than that of O_i, i.e., $\frac{A_q}{A_i} < t_a$, $t_a < 1$ being a threshold.

Removal is detected after occlusion. When occlusion is detected the tracking technique [1] predicts the occluded objects. In case of removal, the features of the removed object can change significantly and the tracking system may not be able to track the removed objects. Conditions for removal are checked and if they are met, removal is declared. The object with the larger area is the remover, the other is the removed object.

Depositor/deposited Let $O_p \in I(n-1)$ and $O_i, O_j \in I(n)$ with $M_i : O_p \rightarrow O_i$. O_i *deposits* O_j if

- O_i has entered or appeared,
- $O_j \notin I(n-1)$, i.e., zero match $M_0 :\dashv O_j$,
- $\frac{A_j}{A_i} < t_a$, $t_a < 1$ being a threshold,
- $A_i + A_j \simeq A_p \quad \wedge [(H_i + H_j \simeq H_p) \vee (W_i + W_j \simeq W_p)]$, where A_i, H_i, and W_i are area, height, and width of an object O_i,
- O_j is close to a side, s, of the minimum bounding box (MBB) of O_i $s \in \{r_{\min_i}, r_{\max_i}, c_{\min_i}, c_{\max_i}\}$ (in order to be declared as deposited object). Let d_{is} be the distance between the MBB-side s and O_j. O_j is close to the MBB-side s if $t_{c_{\min}} < d_{is} < t_{c_{\max}}$ with thresholds $t_{c_{\min}}$ and $t_{c_{\max}}$, and
- O_i changes in height or width between $I(n-1)$ and $I(n)$ at the MBB-side s.

If the distance between the MBB-side s and O_j is less than $t_{c_{\min}}$, the deposited object is assumed have split from O_i and is merged to O_i. Only if this distance is large is *deposit* considered. This is so because in the real world, a depositor moves away from the deposited object and the deposit detection declares the event after the distance between the two objects is large. To reduce false alarms, *deposit* is declared if the deposited object remains in the scene for some time. Note that the system is able to differentiate between deposit events and segmentation error due to splitting of objects. It can also differentiate between stopping objects (e.g., seated person or stopped car) and deposited objects. Finally, it can differentiate between split objects and deposit objects.

Abnormal movements

i) *Stays long*: an object, O_i, *stays long* in the scene if

- $g_i > t_{g_{\max}}$, i.e., O_i does not leave the scene after a given time. $t_{g_{\max}}$ is a function of the frame-rate and the minimal allowable speed, and
- $d_i < t_{d_{\min}}$, i.e, the distance, d_i, between the current position of O_i in $I(n)$ and its past position in $I(l)$, with $l < n$ is less than a threshold $t_{d_{\min}}$ which is a function of the frame-rate, the motion, and the image size.

ii) *Moves fast/slow*: an object, O_i, *moves (too) fast* (or *moves (too) slow*) if the object speed in the last k (for example, five) images is larger (smaller) than a threshold.

Dominant object A *dominant object* 1) is related to a significant event, 2) has the largest size, 3) has the largest speed, or 4) has the largest age.

Other events and composite events can be easily extracted based on our detection strategy. For example, *approach a restricted site* can be easily extracted when the location of the restricted site is known. Other possible events include: stand, sit, walk, object lost, and found. Application-specific conditions can be easily integrated.

The thresholds used in the rules proposed, e.g., t_{m_s} in *stops*, were computed experimentally. The same values wered taken for all shot simulations.

4. An application: key-image extraction

The effectiveness of the proposed technique has been shown by tests conducted on more than 10 indoor and outdoor video shots containing a total of 6371 images with noise and coding artifacts. This effectiveness is demonstrated here on the problem of key-image extraction. Key-images, which are images of important events, are crucial to applications such as video surveillance or retrieval. Figs. 3, 1 and, 4 show samples of our results. They are key-images extracted automatically from shots. Each image is annotated (upper left corner) with its number, object ID, minimum bounding box, age, and related events. Only objects performing events are annotated in these figures. The good performance of the system is a result of special considerations to process inaccuracies and errors of the multi-level approach and to handle false alarms. For example, the system is able to differentiate between deposited objects, split objects, and objects at obstacle. It also rejects false alarms of entering or disappearing due to segmentation error.

5. Conclusion

This paper builds on the computational framework developed in [1]. Its purpose is to develop a system to extract useful events from video shots. Several context independent events have been rigorously defined and automatically detected using features extracted following segmentation, motion estimation and object tracking, as developed in [1]. The proposed events are sufficiently broad to assist video surveillance (e.g., send information to operator) and retrieval (e.g., event-based search). Applications include monitoring 1) of removal/deposit of objects, e.g., computing devices, 2) of traffic objects, and 3) behaviors of customers, e.g., in stores. The reliability of the proposed system has been demonstrated by extensive experimentations on more than 10 indoor and outdoor video shots containing a total of 6371 images including sequences with noise and coding artifacts. The proposed system provides a response in real-time for surveillance applications with a rate of up to 10 frames per second on a shared computing machine. Further research is planed in classification of motion as with purpose (vehicle or people) and without purpose (trees).

References

[1] A. Amer. *Object and Event Extraction for Video Processing and Representation in On-Line Video Applications*. PhD thesis, INRS-Télécommunications, Univ. Québec, Dec. 2001.

[2] A. Bobick. Movement, activity, and action: the role of knowledge in the perception of motion. Technical Report 413, M.I.T. Media Laboratory, 1997.

[3] J. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognit.*, 30(4):607–625, 1997.

[4] R. Jain, A. Pentland, and D. Petkovic. Workshop report. In *Proc. NSF-ARPA Workshop on Visual Information Management Systems*, Cambridge, MA, June 1995.

[5] A. Lippman, N. Vasconcelos, and G. Iyengar. Human interfaces to video. In *Proc. 32nd Asilomar Conf. on Signals, Systems, and Computers*, Asilomar, CA, Nov. 1998. Invited Paper.

[6] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval: the end of the early years. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(12):1349–1380, Dec. 2000.

[7] E. Stringa and C. Regazzoni. Content-based retrieval and real time detection from video sequences acquired by surveillance systems. In *Proc. IEEE Int. Conf. Image Processing*, pages 138–142, Chicago, IL, Oct. 1998.

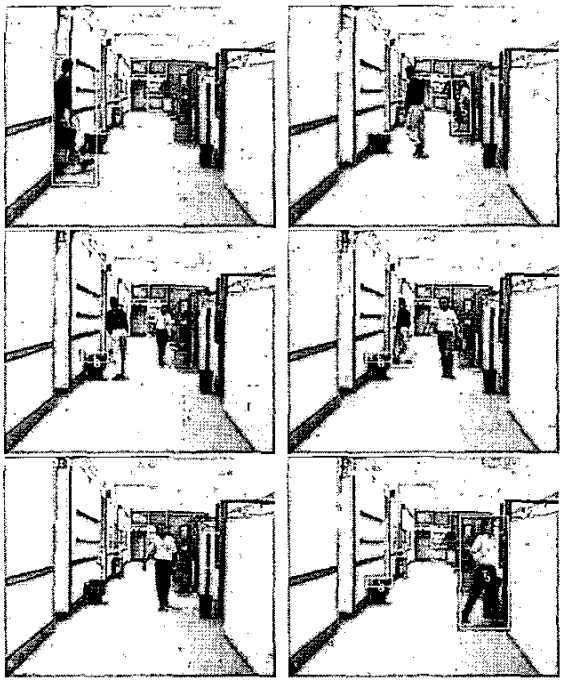

Figure 1. Key-images of the 'Hall' sequence (300 images). Image are annotated with events (upper left corner) and objects performing events with their MBB and ID. E.g., O_1 appears in the 1^{st} and O_6 is *deposited* by O_1 in last key image.

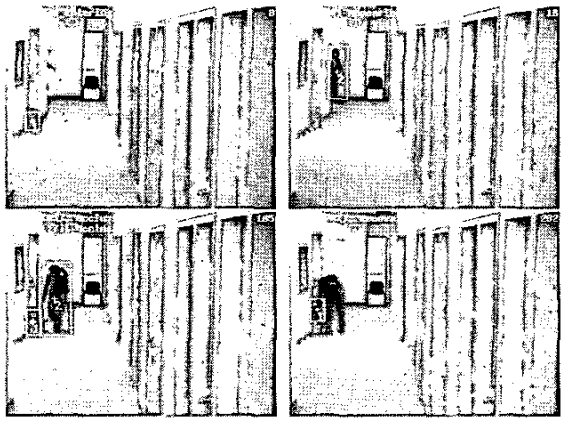

Figure 2. Key-event-images of the 'FloorT' sequence (636 images), e.g., O_2 *removes* O_1 in the 3^d key image.

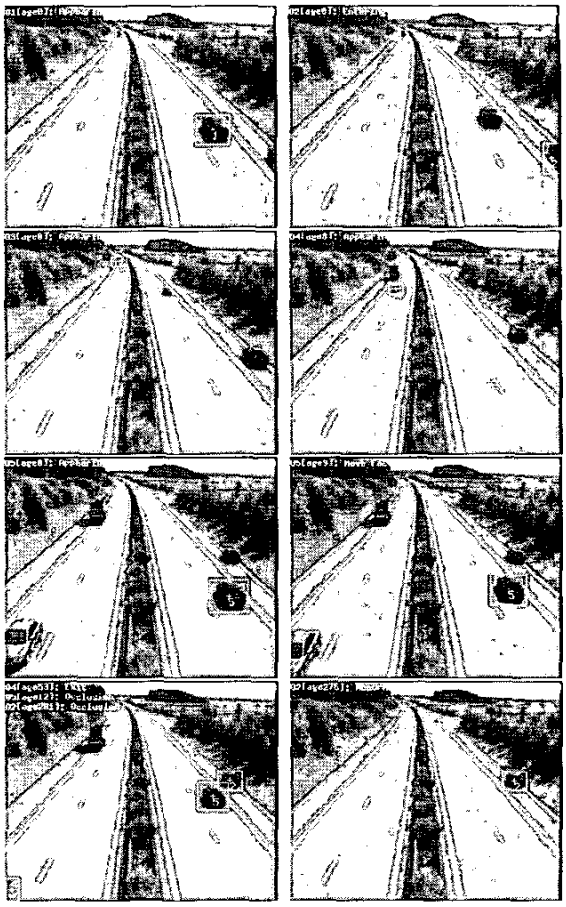

Figure 3. Key-images of the 'Highway' sequence (300 images), e.g., O_5 *moves fast* in 6^{th} and O_2 *stops for long* in the last key image.

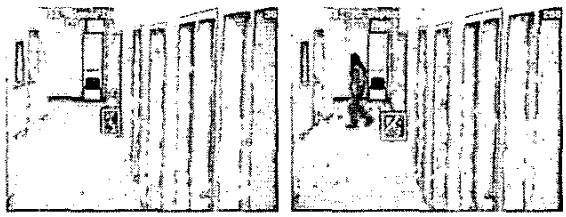

Figure 4. Key-event-images of the 'Floor' sequence (826 images), e.g., O_1 *appears* and *deposits* O_3.