A Real-Time System for High-Level Video Representation: Application to Video Surveillance

Aishy Amer[a], Eric Dubois[b], and Amar Mitiche[c]

[a]Concordia University, Electrical and Computer Engineering, Montréal, Québec, Canada
[b]School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada
[c]INRS-Télécommunications, Université du Québec, Montréal, Québec, Canada

ABSTRACT

The steadily increasing need for video content accessibility necessitates the development of stable systems to represent video sequences based on their high-level (semantic) content. The core of such systems is the automatic extraction of video content. In this paper, a computational layered framework to effectively extract multiple high-level features of a video shot is presented. The objective with this framework is to extract rich high-level video descriptions of real world scenes.

In our framework, high-level descriptions are related to moving objects which are represented by their spatio-temporal low-level features. High-level features are represented by generic high-level object features such as events. To achieve higher applicability, descriptions are extracted independently of the video context.

Our framework is based on four interacting video processing layers: *enhancement* to estimate and reduce noise, *stabilization* to compensate for global changes, *analysis* to extract meaningful objects, and *interpretation* to extract context-independent semantic features. The effectiveness and real-time response of the our framework are demonstrated by extensive experimentation on indoor and outdoor video shots in the presence of multi-object occlusion, noise, and artifacts.

Keywords: Content-based video shot representation, video abstraction, video indexing, high-level content, semantic features, video objects, events, object extraction, video interpretation, video surveillance

1. INTRODUCTION

Because of the ever-increasing needs for video content accessibility, developing automated and effective frameworks for *content-oriented* video representation have become an active field of research. Developing effective video representation systems requires the resolution of two key issues: defining what are the most *important* and most *common* video contents and what *level* of features are suitable to represent these contents. What are important video content? A video displays, in general, low and high-level features of objects within a given environment and context; an important observation is that the subject of the majority of video is related to moving objects, in particular people, that perform activities and interact creating object meaning such as events.[3, 4]

When viewing a video, the human visual system (HVS) is, in general, attracted to moving objects and their features; the HVS focuses first on the high-level object features (e.g., meaning) and then on the low-level features (e.g., shape). The HVS is able to search a video by quickly scanning ("flipping") it for activities and interesting events. In addition, studies have shown that low-level features are not sufficient for effective video representation and that objects must be assigned high-level features as well.[4, 5]

High-level intentional descriptions such as *what a person is thinking* can help solving video representation issues but extracting such information with the current state-of-the-art in video analysis is difficult and alternatives have to be found. Also extracting the context of a video data is difficult and may not be necessary to extract useful content as our investigation show. 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.

Send correspondence to Aishy Amer, E-mail: amer@ece.concordia.ca, Tel.: 1 514 848-4081, Fax. 514 848-2802, Address: Concordia University, 1455 de Maisonneuve, West, H-961; Montréal, Québec H3G 1M8 Canada

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,[4] and related events. High-level features are generally applicable when they convey fixed meaning independently of context.

High-level video representations can be structural and conceptual. Structural representations are based on objects using spatial, temporal, and relational features while conceptual representations are based on movement-related features. In this paper, we present a computational framework to conceptually represent an input video in real time based on its moving objects and related semantic features independent of context. Such high-level representation aims at assisting users of advanced applications, in particular video surveillance. To effectively represent video, our framework consist of four video processing layers (cf. Fig. 1): video enhancement to reduce noise and artifacts, video stabilization to compensate for global image changes, video analysis to extract low-level video features, and video interpretation to describe content in semantic-related terms.

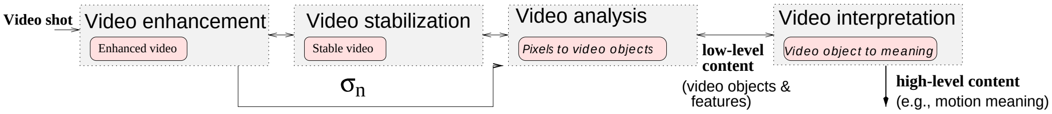


Figure 1. Our computational framework for video representation. σ_n represents the video noise.[15]

Typically, a video is a set of stories, scenes, and shots. To facilitate automated content-oriented video representation, a video has to be first segmented into shots (see, e.g.,[1, 2]). A shot displays, in general, multiple *objects*, their *semantic interpretation* (i.e., objects' meaning), their *dynamics* (i.e., objects' movement, activities, action, or related events), and their *syntax* (i.e., the way objects are spatially and temporally related).(In the remainder of this paper, the term video refers to a video shot.)

The paper is organized into six additional sections. Section 2 discusses related work, Section 3 describes our real-time framework for high-level video representation, Section 4 presents our method for analyzing video content to extract video objects and their low-level features, Section 5 presents our sub-framework for interpreting low-level spatio-temporal object features to extract context-independent semantic features, Section 6 presents experimental results, and Section 7 summarizes this paper.

2. RELATED WORK

In recent years, research interest in content-based video representation has shifted from motion and tracking towards detection and recognition of activities,[3, 4, 6] actions and events. Much work on video representation deals with the development of a generally applicable solution; however, there are few tests in the presence of noise and other artifacts. Most representation frameworks use mainly low-level features. Most high-level video representation frameworks are developed for narrow applications[4] and little work on context-*independent* representation exist. Two basic video processing levels are required to represent high-level video content: analysis level to extract low-level content and interpretation level to describe content in semantic-related terms.

2.1. Video analysis

The VideoQ system[7] uses video analysis based on optical flow, color, and edge features. Such a system has difficulties in the presence of large motion and occlusion. The AVI system[8] is based on motion detection using a background image and tracking using prediction and nearest-neighbor matching. The motion detection used is sensitive to noise and artifacts. The system is limited to indoor applications and cannot deal with occlusion. Recently, a new video analysis scheme, the COST-AM, has been introduced.[9] This method is based on motion detection and color segmentation and gives good object masks. Difficulties arise when the combination of motion and color fails and strong artifacts are introduced (see Sec. 6). In addition, the method tends to lose objects which is critical for subsequent processing.

2.2. Video interpretation

Most high-level video representation are developed for narrow applications. Narrow-domain systems recognize events and actions, for example, in hand sign applications or in Smart-Cameras based cooking (see the special section in[10] and[4, 6, 11]). In these systems, prior knowledge is, usually, inserted in the event recognition inference system and the focus is on recognition and logical formulation of events and actions.

Little work on context-*independent* or end-to-end video representation exist. The system in[8] 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[12] 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.

The system in[13] is designed for off-line processing applications and uses domain knowledge to facilitate extraction of events (wildlife hunt events). The system in[14] tracks several people simultaneously and uses appearance-based models to identify people. It determines whether a person is carrying an object and can segment the object from the person. It also tracks body parts such as head or hands. The system imposes, however, restrictions on the object movements. Objects are assumed to move upright and with little occlusion. Moreover, it can only detect a limited set of events.

3. OVERVIEW OF OUR FRAMEWORK

Our framework is oriented to three requirements: 1) flexible object representations that are easily cooperatively searched for video applications such as surveillance, abstraction, indexing, and manipulation, 2) reliable, stable processing of video that foregoes the need for precision, and 3) low computational cost. The proposed framework is designed to balance demands for effectiveness (solution quality) and efficiency (computational cost). Without real-time consideration, a representation can lose its applicability. On the other hand, framework stability is important for successful use.

The objective of this paper is to develop a *low-complexity automatic* framework for *stable* representation of video shots of real environments such as those with occlusions and coding artifacts. To achieve these requirements, our multi-layered framework 1) is divided into simple but effective tasks avoiding complex operations, 2) takes video noise level into account, and 3) corrects or compensates for estimation errors at the various processing steps. This framework involves four interacting processing layers: video enhancement (including noise estimation as in[15] and noise reduction as in[16]), video stabilization, video analysis (Sec. 4), and video interpretation (Sec. 5). The input to the video enhancement module is the original video and its output is an enhanced version of it. This enhanced video is then processed by the video analysis module which outputs low-level descriptions of the enhanced video. The video interpretation module takes these low-level descriptions and produces high-level descriptions of the original video.

Fig. 2 displays a block diagram of our framework where $R(n)$ represents a background image of the shot and σ_n is the estimated noise standard deviation. (Implemented modules are underlaid with gray boxes.) Our framework can be viewed as a framework of methods and algorithms to build automatic dynamic scene interpretation and representation. Such interpretation and representation can be used in various video applications. Besides applications such as video surveillance and retrieval, outputs of the proposed framework can be used in a video understanding or a symbolic reasoning framework. Our implementation of the video analysis and interpretation layers needs on between 0.1107 and 0.3507 seconds to extract content of two images on a SUN-SPARC-5 360 MHz.

In our framework, high-level features are related to moving objects and their semantic features. Moving objects are represented using quantitative and qualitative low-level features. Semantic features are represented

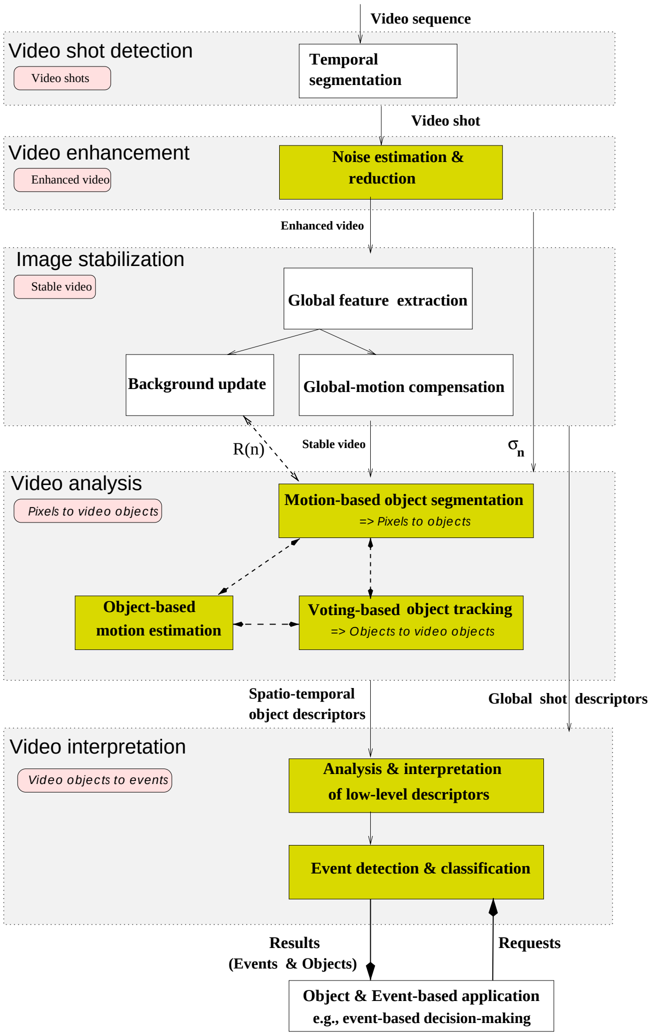

Video sequence
Video shot detection
Video shots
Temporal segmentation
Video shot
Video enhancement
Enhanced video
Noise estimation & reduction
Enhanced video
Image stabilization
Stable video
Global feature extraction
Background update
Global-motion compensation
R(n)
Stable video
σn
Video analysis
Pixels to video objects
Motion-based object segmentation
=> Pixels to objects
Object-based motion estimation
Voting-based object tracking
=> Objects to video objects
Spatio-temporal object descriptors
Global shot descriptors
Video interpretation
Video objects to events
Analysis & interpretation of low-level descriptors
Event detection & classification
Results (Events & Objects)
Requests
Object & Event-based application
e.g., event-based decision-making

Figure 2. A block diagram of our framework for video shot representation. σ_n represents the video noise.[15]

by generic motion-related high-level features such as events. Semantic features are defined by approximate but efficient world models. This is done by continually monitoring changes and behavior of low-level features of the scene's objects. When certain conditions are met, high-level semantic features such as events are detected. To achieve higher applicability, descriptions are extracted independently of the context of the video. Several context independent events are rigorously defined and automatically detected using features extracted following segmentation, motion estimation and object tracking.

Our framework outputs at each instant n a list of objects with their *identity* throughout the shot, *low-level features* (location, shape, size, motion), *trajectory*, *life span or age*, *event descriptions*, and *spatio-temporal relationship*. This can be used in applications where an interpretation ("what is this video shot about?") is needed. For example, in video surveillance, event-oriented alarms can be activated.

A key contribution of our framework is the layer interaction for correction and compensation of processing errors at higher layers where more information is available. This includes 1) processing inaccuracies and false alarms– for example, the framework is able to differentiate between deposited objects, split objects, and objects at an obstacle and 2) data update– for example, low-level object segments are used for tracking and tracking can correct and merge these segments if needed; tracking together with merging are then used to detect events. Such interaction allow balanced processing as missing information can prevent complete information.

4. OBJECT-ORIENTED VIDEO ANALYSIS

The proposed video analysis consists of: 1) motion-detection based object segmentation, 2) object-based motion estimation,[17] 3) region merging, and 4) feature-voting based object tracking. Video analysis may produce inaccuracies and much research has been done to enhance their performance. We propose to compensate for errors of low-level steps at higher levels when more reliable information is available. The critical tasks of the video analysis are the segmentation and the tracking steps.

4.1. Object segmentation

Our object segmentation trades precise segmentation at object boundaries for speed of execution and reliability in varying image conditions. The object segmentation consists of four steps: motion detection, morphological edge detection,[18] contour analysis,[18] and object labeling.

Motion detection The core of the segmentation method is the motion detection which must remain reliable throughout video shots. Typical difficulties of motion detection based on differencing are 1) it does not distinguish between object motion and other changes, for example, due to illumination changes and 2) it does not account for changes occurring throughout a long video shot. Usually a fixed threshold is used for all images of the shot to decide on moving and non-moving image parts. A fixed threshold method fails, e.g., when the amount of moving regions changes significantly. To answer these difficulties, we propose a three step thresholding method as follows: **1) Adaptation to noise:** First, a spatial threshold, T_g, is estimated using a robust thresholding method.[19] The threshold T_g is adapted to the amount of image noise as follows $T_n = T_g + c \cdot \sigma_n^2, \quad c < 1$, where σ_n is the noise standard deviation estimated as in.[15] **2) Quantization:** To further stabilize thresholding, T_n is quantized to T_q into m values to compensate for background and local illumination changes. In our implementation, m was set to 3. **3) Adding memory:** To adapt detection to temporal changes throughout a video shot the following memory function is used:

$$T(n) = \begin{cases} T_{\min} & : \quad T_q \leq T_{\min} \\ T(n-1) & : \quad T_q < T(n-1) \\ T_q & : \quad \text{otherwise} \end{cases} \tag{1}$$

4.2. Simultaneous tracking of multiple objects

Objects are tracked based on the similarity of their features in successive images. This is done in three steps: spatio-temporal feature extraction,[17, 18] object matching, and feature monitoring and correction. In the first step, object segmentation and motion estimation segment and extract objects and computes their spatial and temporal features. In the second step, using a voting-based feature integration, *each* object O_p of the previous image $I(n-1)$ is matched with an object O_i of the current image $I(n)$ creating a unique correspondence $M_i : O_p \rightarrow O_i$. This means that all objects in $I(n-1)$ are matched with objects in $I(n)$. In this step, each tracked object is assigned an *identity* throughout the image sequence. M_i provides a temporal link between objects to determine the trajectory of each object throughout the video and allows a semantic-based interpretation of the input video. In the third step, feature monitoring and correction, object segmentation errors, such as when object occlude or are split, are detected and corrected. These new data are then used to *update* the results of previous steps, i.e., object segmentation and motion estimation. For example, the error correction may produce new objects after detecting occlusion. Feature extraction and tracking need to be redone for these new objects.

4.2.1. Feature integration by voting

In this step, we combine spatial and temporal features using a non-linear voting scheme consisting of two steps: voting for object features of two objects (object voting) and voting for features of two correspondences in the case one object is matched to two objects (correspondence voting).

In the object voting step, three main feature votes are used: shape, size, and motion vote. The use of multiple votes aims at avoiding cases where one feature fails and the tracking module loses the object (especially in the case of occlusion). Two objects, O_p and O_i, match if, where t_m is a threshold,

$$\begin{array}{lcl} M_i & : & (d_i < t_r) \wedge (w_{x_i} < w_{x_{\max}}) \wedge (w_{y_i} < w_{y_{\max}}) \wedge (\zeta > t_m) \\ \bar{M}_i & : & \text{otherwise} \end{array} \tag{2}$$

with the vote confidence $\zeta = \frac{s}{d}$. M_i is accepted if O_i lays within a search area of O_p, its displacements is not larger than a maximal displacement, and if both objects are similar, i.e., $\frac{s}{d} > t_m$.

In the correspondence voting step, five voting functions are applied based on the features distance, confidence, size, shape, and motion (direction and displacement) to solve multiple matches. Note that in the object voting step, *all* objects $I(n-1)$ are matched against *all* objects of $I(n)$. Each object $O_p \in I(n-1)$ is matched to each object $O_i \in I(n)$. This may result in multiple matches for one object. In this case correspondence voting is applied as follows: Let s_i (s_j) be the variable that describes if M_i (M_j) is the better correspondence. Then

$$\begin{aligned} M_i &: (s_i > s_j) \wedge (\zeta_i > \zeta_j) \\ M_j &: s_i \leq s_j \end{aligned} \tag{3}$$

4.2.2. Feature monitoring and correction

A good tracking technique must account for errors of previous steps. Object segmentation, for example, is likely to output erroneous object masks and features. Our method corrects or compensates effects of such errors. This is done based on plausibility rules and predictions strategies to filter faulty object features, to monitor occlusion, and to merge divided objects. The new information is then used to *update* the previous analysis steps. The detection of segmentation errors is done based on an analysis of displacements of the four minimum bounding box (MBB) sides. This is shown in this paper by detection and correction of object occlusion.

Detecting object occlusion Feature monitoring is shown in this paper by detecting object occlusion. Define 1) $O_{p_1}, O_{p_2} \in I(n-1)$, 2) $M_i : O_{p_1} \rightarrow O_i$ where O_i results from the occlusion of O_{p_1} and O_{p_2} in $I(n)$, 3) $d_{p_{12}}$ the distance of the centroids of O_{p_1} and of O_{p_2}, 4) $w = (w_x, w_y)$ the current displacement of O_{p_1}, i.e., between $I(n-2)$ and $I(n-1)$, and recall 5) $d_{r_{\max}}$ ($d_{r_{\min}}$) is the vertical displacement of the lower (upper) row and 6) $d_{c_{\max}}$ ($d_{c_{\min}}$) is the horizontal displacement of the right (left) column of O_{p_1}.

Object occlusion is declared if

$$\begin{array}{l} ((|w_y - d_{r_{\max}}| > t_1) \wedge (d_{r_{\max}} > 0) \wedge (d_{i_{12}} < t_2)) \vee ((|w_y - d_{r_{\min}}| > t_1) \wedge (d_{r_{\min}} > 0) \wedge (d_{i_{12}} < t_2)) \vee \\ ((|w_x - d_{c_{\max}}| > t_1) \wedge (d_{c_{\max}} > 0) \wedge (d_{i_{12}} < t_2)) \vee ((|w_x - d_{c_{\min}}| > t_1) \wedge (d_{c_{\min}} > 0) \wedge (d_{i_{12}} < t_2)) \end{array} \tag{4}$$

where t_1 and t_2 are thresholds. If occlusion is detected then both the occluding and occluded objects are labeled with a special flag. This labeling enable our system to continue tracking both objects in following images even if they are completely non visible. Tracking non visible objects is important since they might reappear. The labeling is further important to help detect occlusion even if the occlusion conditions in Eq. 4 are not met.

Correction of occlusion by object prediction: If occlusion is detected, the occluded object O_i is split into two objects. This is done by predicting both object O_{p_2} and O_{p_1} onto $I(n)$ using the following displacement estimate:

$$\begin{aligned} d_{p_1} &= (\mathrm{MED}(d^1_{x_c}, d^1_{x_p}, d^1_{u_x}), \mathrm{MED}(d^1_{y_c}, d^1_{y_p}, d^1_{u_y})) \\ d_{p_2} &= (\mathrm{MED}(d^2_{x_c}, d^2_{x_p}, d^2_{u_x}), \mathrm{MED}(d^2_{y_c}, d^2_{y_p}, d^2_{u_y})) \end{aligned} \tag{5}$$

with MED represent a 3-tap median filter, $d^1_{x_c}(d^1_{y_c})$, $d^1_{x_p}(d^1_{y_p})$, $d^1_{u_x}(d^1_{u_y})$ as the current, previous and past-previous horizontal (vertical) displacement of O_{p_1} and $d^2_{x_c}(d^2_{y_c})$, $d^2_{x_p}(d^2_{y_p})$,$d^2_{u_x}(d^2_{u_y})$ as the current, previous and past-previous horizontal (vertical) displacement of O_{p_2}. After splitting occluded and occluding objects, the list of objects of $I(n)$ is *updated*, for example, by adding O_{p_2}. Then a feedback loop estimate the correspondences in case new objects are added.

5. CONTEXT-INDEPENDENT VIDEO INTERPRETATION

5.1. Overview

Content-oriented video applications such as surveillance, require the development of automatic and real-time systems to extract high-level video features. In this section, we propose a video interpretation module (Fig. 2) that defines and extract semantic features based on approximate but efficient world models. We propose perceptual descriptions of semantic feature that are common for a wide range of applications. Semantic feature detection is not based on geometry of objects but on their features and relations over time. This is done by continually monitoring changes and behavior of low-level features of the scene's objects. When certain conditions are met, high-level semantic features such as events are detected. Our context-independent interpretation module is applied here to detects events related to moving objects.

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 (or fixed meaning) 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 our system, behavior monitoring is done on-line, i.e., object data is analyzed *as it arrives* and events are detected as they occur.

The following are samples of events detected automatically by our framework (due to space constraints only selected events are defined here). The thresholds used in the following rules are adapted to object features. For example, the threshold, $t_{d_{\min}}$, when detecting *abnormal movements* is a function of the frame-rate, the motion, and the image size. Some thresholds are computed experimentally. However, the same values were taken for all shots used in simulations. In the following, let $I(n)$ be an image in a video shot and O_i a segmented object in $I(n)$.

5.2. Basic events

Basic events are related to trivial behavior of objects. Examples are *enter*, *appear*, *exit*, *disappear*, *move*, and *stops.* Following are definitions of some of the basic events our system is able to detect. For example, an object, $O_i \in I(n)$, *moves* in image $I(n)$ if 1) $M_i : O_p \to O_i$ (a function assigning O_p at time $n-1$ to an object O_i at time n) where $O_p \in I(n-1)$, and 2) the median of the motion magnitudes of O_i in the last k images is larger than a threshold.

5.3. Intra-object events

An intra-object event is related to non-trivial behavior of an object. Examples are *abnormal movements* and *dominant movements.* An abnormal movement is defined when, for example, an object *stays long* or *moves (too) fast/slow.* A dominant movement is given when an object, for examples, 1) performs a significant event, 2) has the largest size, 3) has the largest speed, 4) or has the largest age. Following are definitions of some of the intra-object events our system is able to detect. For example, an object, O_i, *stays long* in the scene if 1) $g_i > t_{g_{\max}}$, 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 2) $d_i < t_{d_{\min}}$, 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.

5.4. Inter-object events

An inter-object event is related to non-trivial behavior of two or more objects. Examples include objects *at an obstacle*, *occlude/occluded*, *expose/exposed*, *deposit/deposited*, and *remove/removed.* Following are definitions of some of these events.

Occlusion: Object occlusion is declared as defined in Eq. 4 (see Page 6). 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.* Note that *exposure* is detected when occlusion ends.

Removal: Let $O_i \in I(n)$ and $O_p, O_q \in I(n-1)$ with $M_i : O_p \to O_i$. 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 predicts the occluded objects. In case of removal, the features of the removed object can change significantly and the tracking framework 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.

Deposit: O_i *deposits* O_j (or O_j is *deposited* by O_i) if

- $O_p \in I(n-1)$, $O_i, O_j \in I(n)$, and M_i,
- $O_j \notin I(n-1)$, i.e., no 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 O_i,
- O_j is close to a side, s, of the minimum bounding box (MBB) of O_i. $s \in \{r_{\min_i}, r_{\max_i}, c_{\min_i}, c_{\max_i}\}$. Let d_{is} be the distance between the MBB-side s and O_j. O_j is close to 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.

Note that only if the distance between the deposited object and depositor is large is the event *deposit* considered (in the real world, a depositor moves away from the deposited object). Otherwise O_j is assumed have split from O_i and is merged to O_i. To reduce false alarms, *deposit* is declared if the deposited object remains in the scene for some time. Thus the framework is able to differentiate between deposit and segmentation errors (e.g., object split). It can also differentiate between stopping objects (e.g., seated person or stopped car) and deposited objects.

5.5. Extensions

Other events can be easily extracted based on our interpretation strategy. Examples include 1) *Standing* and *sitting* are characterized by continuous change in height and width of the object MBB; 2) The event *walk* can be easily detected as continuous moderate movements of a person. 3) The event *approaching a restricted site* is straightforward to detect when the location of a restricted site is given. For example, by monitoring the direction of an object's motion and distance to the site; 4) *Object lost/found*, at a time instant n, an object is declared lost if it has no corresponding object in the current image and occlusion was previously reported (but no removal). It is similar to the event *disappear*. Some applications require the search for lost objects even if they are not in the scene.

6. RESULTS AND APPLICATIONS

To test the performance of our framework, we conducted extensive experiments on shots containing over 6000 images with multi-object occlusion, noise, and artifacts of indoor and outdoor real environments. The presented framework works in real time for video shots with a rate of up to 10 frames/second. In this section, we first introduce sample results of the proposed video analysis module and then show how the video interpretation module can be used for different applications.

6.1. Results of the video analysis module

The current implementation of the proposed video analysis requires on average between 0.11 and 0.35 seconds to analyze the content of two images on a SUN-SPARC-5 360 MHz. For comparison, the current version (v.4x) of the reference method, COST-AM,[9] takes on average 175 seconds to segment objects of an image.

Samples of our results are shown in this section. Both objective and subjective evaluations, and comparisons to other methods show the robustness of the proposed methods while being of reduced complexity. Our method is objective and subjectively compared to the current version (4.x) of the COST-AM method.[9] For example,

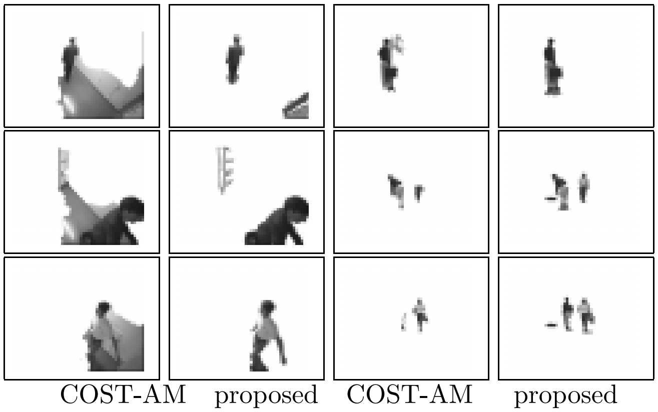
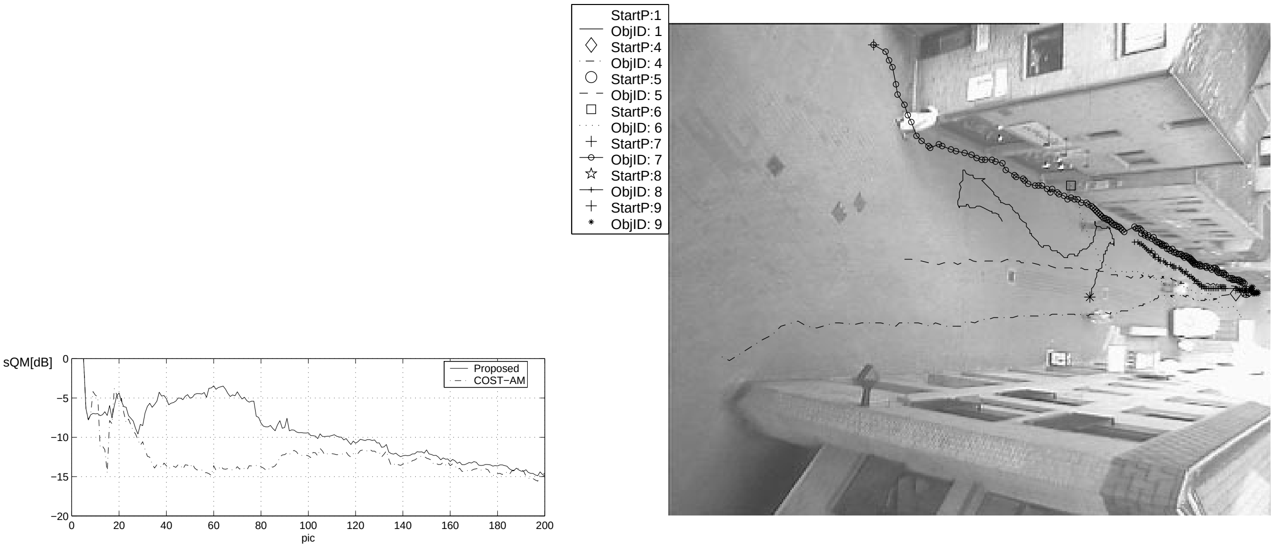


Figure 3. Subjective comparison of object segmentation for the 'Stair' (left) and 'Hall' (right) shots.



(a) Spatial accuracy, $sQM(dB)$, comparison: our method has better accuracy throughout the shot. Average gain $\simeq$ 3.5 dB.

(b) Estimated trajectories of the objects in the shot 'Urbicande'. 'StartP' is start position.

Figure 4. Performance of the video analysis module.

Fig. 6.1 gives objective comparison results based on the criteria given in.[21] The proposed segmentation is better with respect to all three criteria, especially it yields higher spatial accuracy. Fig. 3 subjectively confirms the good performance of the proposed segmentation compared to the reference, COST-AM, method. COST-Am method loses some objects and its spatial accuracy is poor. The proposed method remains robust to variable object size and is spatially more accurate. Robustness is further confirmed in the presence of MPEG-2 artifacts (25dB) and noise (30dB). The proposed segmentation algorithm needs a maximum of 0.15 seconds on a SUN-SPARC-5. Fig. 6.1 shows sample of our simulations of trajectory estimation using our tracking method.

6.2. Applications of the video interpretation module

Applications of our video representation framework include key-image extraction based on events, high-level video abstraction and summarization, and event-related alerts for surveillance applications.

6.2.1. Event-based video summary

The performance of our video interpretation is illustrated here by a summary of the shot ‘floor’ (Fig. 5).

```
 ‘floor’ Shot Summary based on Objects and Events; StartPic 1/EndPic 826
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
|Pic| Event               | Obj| Age| Status | Position           | Motion   | Size         |
|   |                     |    |    |        | start/present      | present  | start/present|
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|36 | Appear              | 1  | 8  | Move   | (126,140)/(123,135)| (0  ,-1 )| 320   /320   |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|268| is Deposit by Obj 1| 3  | 8  | Stop   | (121,140)/(121,140)| (0  ,0  )| 539   /539   |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|405| Occlusion           | 3  | 145| Stop   | (121,140)/(120,141)| (0  ,0  )| 541   /555   |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|405| Occlusion Obj 3     | 1  | 377| Move   | (126,140)/(83 ,109)| (1  ,0  )| 840   /2422  |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|411| Removal by Obj 1    | 3  | 150| Removal| (121,140)/(104,132)| (0  ,0  )| 541   /1451  |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|787| Appear              | 18 | 8  | Move   | (105,68 )/(108,86 )| (0  ,0  )| 91    /91    |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
|825| Exit                | 1  | 796| Exit   | (126,140)/(9  ,230)| (-10,7  )| 840   /247   |
|---| -------------------| ---| ---| -------| -------------------| ---------| -------------|
```

6.2.2. Event-based key-image extraction

In a surveillance environment, important events may occur after a long time has passed. During this time, the attention of human operators decreases and significant events may be missed. The proposed system for event detection identifies events of interest as they occur and human operators can focus their attention on moving objects and their related events.

This section presents automatic extracted key-images from video shots using our framework. Key-images are the subset of images which best represent the content of a video sequence in an abstract manner. Key-image video abstraction transforms an entire video shot into a small number of representative images. This way important content is maintained while redundancies are removed. Key-images based on events are appropriate when the system must report on specific events as soon as they happen.

Fig. 5 shows a sample of our results, images of key events extracted automatically. Only objects performing events are annotated (ID and MBB) in these figures. The good performance of the framework is a result of special considerations to handle inaccuracies and false alarms. For example, the framework is able to differentiate between deposited, split, and obstacle objects.

7. CONCLUSION

This paper proposed a computational framework to automatically and efficiently extract semantic content from video shots. Semantic content is defined as meaningful video objects and useful context-independent events. Several context independent events have been rigorously defined and automatically detected using features extracted following segmentation, motion estimation and object tracking. The proposed events are sufficiently broad to assist applications such as 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 framework has been demonstrated by extensive experimentations on indoor and outdoor shots containing over 6000 images with object occlusion, noise, and coding artifacts.

In our framework, special consideration is given to processing inaccuracies and false alarms. For example, the framework is able to differentiate between deposited objects, split objects, and objects at an obstacle. Errors are corrected or compensated at higher level level where more information is available. The proposed framework provides a response in real-time for surveillance applications with a rate of up to 10 frames per second.

Further research is planned in classification of motion as ‘with purpose’ and ‘without purpose’ (trees). In addition, the detection of background objects that move during the shot needs to be explicitly processed.

ACKNOWLEDGMENTS

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.

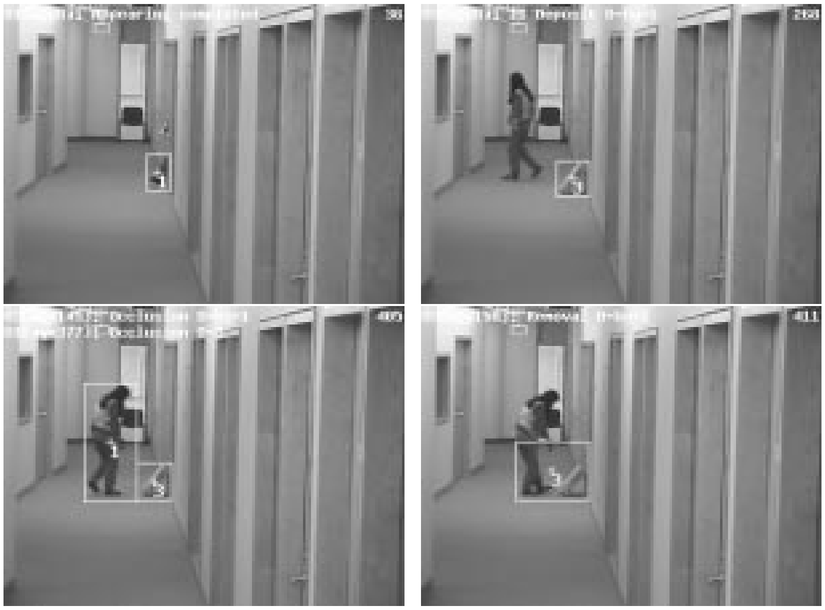


Figure 5. Key events of the 'Floor' sequence (826 images): important key events: O_1 deposits and then removes O_3. Both the *depositor/remover* and *deposited/removed* objects are detected.

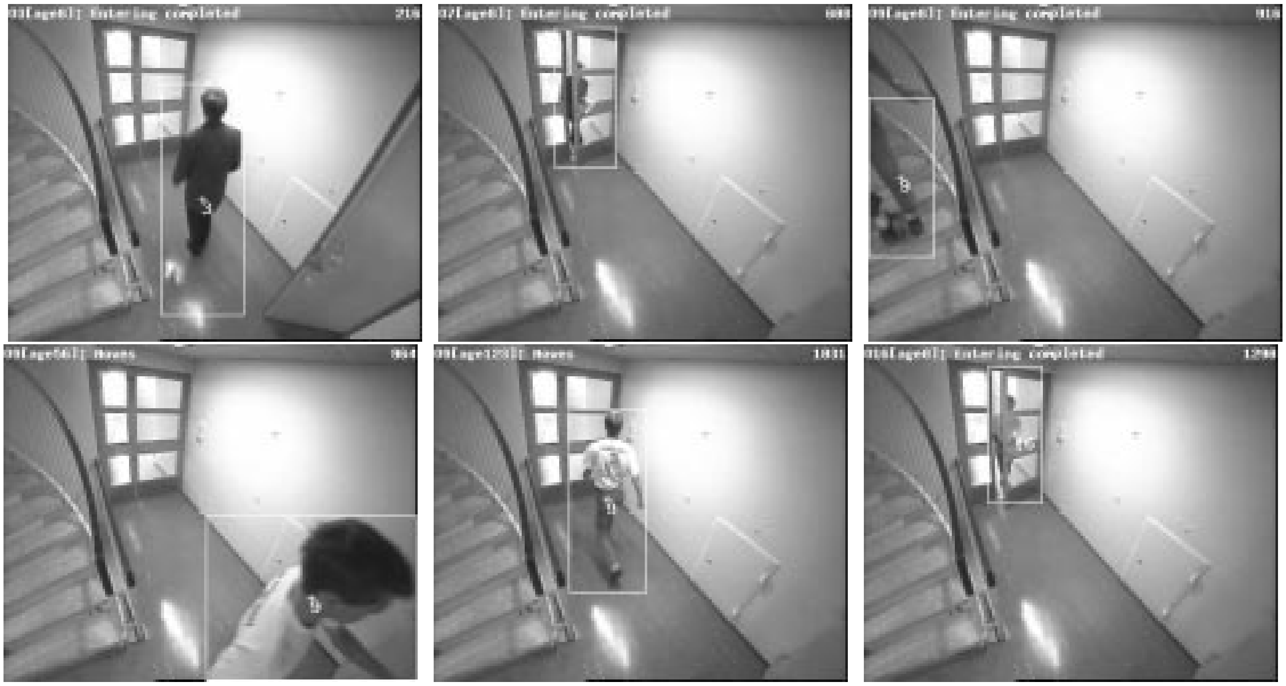


Figure 6. Key events of the 'Stair' sequence (1475 images). This sequence is typical for entrance surveillance application. The interesting feature of this application is that objects can enter from three different places, the two doors and the stairs. One of the doors is restricted. O_3 enters, moves, exits, re-enters, and exits. O_9 enters, tries to enter restricted door, exits, and re-enters.

REFERENCES

1. M. Naphade, R. Mehrotra, A. Ferman, J. Warnick, T. Huang, and A. Tekalp, “A high performance algorithm for shot boundary detection using multiple cues,” in *Proc. IEEE Int. Conf. Image Processing*, **2**, pp. 884–887, (Chicago, IL), 1998.
2. P. Bouthemy, M. Gelgon, and F. Ganansia, “A unified approach to shot change detection and camera motion characterization,” Tech. Rep. 1148, Institut National de Recherche en Informatique et en Automatique, Nov. 1997.
3. A. Pentland, “Looking at people: Sensing for ubiquitous and wearable computing,” *IEEE Trans. Pattern Anal. Machine Intell.* **22**, pp. 107–119, Jan. 2000.
4. A. Bobick, “Movement, activity, and action: the role of knowledge in the perception of motion,” Tech. Rep. 413, M.I.T. Media Laboratory, 1997.
5. 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**, pp. 1349–1380, Dec. 2000.
6. A. Lippman, N. Vasconcelos, and G. Iyengar, “Human interfaces to video,” in *Proc.* 32^{nd} *Asilomar Conf. on Signals, Systems, and Computers*, (Asilomar, CA), Nov. 1998. Invited Paper.
7. S. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, “A fully automatic content-based video search engine supporting multi-object spatio-temporal queries,” *IEEE Trans. Circuits Syst. Video Techn.* **8**(5), pp. 602–615, 1998. Special Issue.
8. J. Courtney, “Automatic video indexing via object motion analysis,” *Pattern Recognit.* **30**(4), pp. 607–625, 1997.
9. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tunceland, and T. Sikora, “Image sequence(1) analysis for emerging interactive multimedia services - the European COST 211 Framework,” *IEEE Trans. Circuits Syst. Video Technol.* **8**, pp. 802–813, Nov. 1998.
10. “Special section on video surveillance.” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, Aug. 2000.
11. N. Vasconcelos and A. Lippman, “Towards semantically meaningful feature spaces for the characterization of video content,” in *Proc. IEEE Int. Conf. Image Processing*, **1**, pp. 25–28, (Santa Barbara, CA), Oct. 1997.
12. 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*, pp. 138–142, (Chicago, IL), Oct. 1998.
13. N. Haering, R. Qian, and I. Sezan, “A semantic event detection approach and its application to detecting hunts in wildlife video,” *IEEE Trans. Circuits Syst. Video Techn.* **10**, pp. 857–868, Sept. 2000.
14. I. Haritaoglu, D. Harwood, and L. S. Davis, “W^4: Real-time surveillance of people and their activities,” *IEEE Trans. Pattern Anal. Machine Intell.* **22**, pp. 809–830, Aug. 2000.
15. A. Amer, A. Mitiche, and E. Dubois, “Reliable and fast structure-oriented video noise estimation,” in *Proc. IEEE Int. Conf. Image Processing*, **1**, pp. 840–843, (Rochester, USA), Sept. 2002.
16. A. Amer, *Object and Event Extraction for Video Processing and Representation in On-Line Video Applications.* PhD thesis, INRS-Télécommunications, Univ. du Québec, Dec. 2001.
17. A. Amer and E. Dubois, “Real-time motion estimation by object-matching for high-level video representation,” in *Proc. IAPR/CIPPRS Int. Conf. on Vision Interface*, pp. 31–38, (Calgary, Canada), May 2002.
18. A. Amer, “Memory-based spatio-temporal real-time object segmentation,” in *Proc. SPIE Int. Conf. on Real-Time Imaging*, (Santa Clara, California, USA), Jan. 2003. to appear.
19. A. Amer and E. Dubois, “Image segmentation by robust binarization and fast morphological edge detection,” in *Proc. IAPR/CIPPRS Int. Conf. on Vision Interface*, pp. 357–364, (Montréal, Canada), May 2000.
20. A. Amer, “Voting-based simultaneous tracking of multiple video objects,” in *Proc. SPIE Int. Conf. on Image and Video Communications and Processing*, (Santa Clara, California, USA), Jan. 2003. to appear.
21. P. Villegas, X. Marichal, and A. Salcedo, “Objective evaluation of segmentation masks in video sequences,” in *Proc. Workshop on Image Analysis for Multimedia Interactive Services*, pp. 85–88, (Berlin, Germany), May 1999.