Event Detection with Pan-Tilt Cameras

Diogo Vicente Jacinto C. Nascimento José Gaspar
Instituto Superior Técnico, Universidade Técnica de Lisboa
dpvicente@gmail.com, {jan, jag}@isr.ist.utl.pt

***Abstract*—Modern video surveillance systems are based on pan tilt and zoom cameras due to the intrinsic advantages of these devices with respect to fixed cameras. Even though it is patent that these equipments enclose great potential as surveillance instruments, it is not clear the best approach for its insertion into automatic surveillance systems. The experiments detailed hereafter present relevant results enabling the use of these devices. In particular it focus on: segmentation techniques, control modalities for the pan-tilt cameras and zoom impact on both types of algorithms. Furthermore the metrics applied to evaluate performance results are extended from the state of the art for fixed camera scenarios to a pan-tilt scenarios. The results attained in the scope of this work rely in datasets generated from a virtual reality framework which was implemented to simplify the test images generation. This approach also eases the replication of experiments allowing an proper understanding of the results.**

Fig. 1. Cube-based representation of the scene, not showing the top and front faces.

I. Introduction

Depending on the purpose, surveillance can be understood more qualitatively as *awareness to novel events*, or more precisely as *tracking the trajectories* of moving objects or people walking. The first case is a *configuration finding* problem [5], where typically one wants information every time the surveyed area changes its default pattern e.g. due to a new object in scene. In the second case the solution involves mainly an *identification or data association* problem, in order to successfully track different objects.

Metrics were already proposed for accessing the performance of the basilar components of the image-based surveillance systems namely the segmentation algorithms. These metrics evaluate correct or false detections and object-splits, object-merges or both [3]. Metrics were also proposed for the higher levels of *configuration finding* and *tracking* methodologies [5]. Although being quite mature the outcome of [3] and [5], it focus on fixed cameras and in particular does not consider the nowadays, constantly growing number of, video surveillance installations encompassing pan-tilt cameras.

Surveillance with pan-tilt cameras involves not only video processing but also controlling the pan and tilt angles. In this work we propose a novel metric tuned to assess the effectiveness control designs in scenarios considering pan-tilt cameras. Since the aim of this work is to contribute with useful results regarding the feasibility of applying pan-tilt cameras in automatic surveillance systems, the experiments detailed within this dissertation assess four main aspects: performance of segmentation and control modalities at different zoom levels; behavior of control modalities with random walk noise; applicability of the metrics proposed by [3] in a scenario considering pan-tilt cameras. This work focus on extending these metrics if required, for pan-tilt camera surveyed environments.

For this purpose, it is necessary to perform extensive testing and thus convenient to generate automatically ground truth information. It is therefore advantageous to build simulated setups. Various manners can be used to represent geometrically the background e.g. planar mosaic, a cylinder, a sphere or a cube [1], [4]. In particular we select the cube based representation as it can handle a complete spherical field-of-view (FOV), $360^o \times 360^o$, which is not possible in the planar or cylindric mosaics, and maps perspective images to/from the background using just homographies (as compared to using spherical mappings). Hence, a framework was implemented for testing purposes, where the scene can be characterized by cubic representation. In this scene, mobile objects were moved and surveyed combining video segmentation techniques with pan-tilt control modalities.

II. Event detection and scene representation

There is a large variety of segmentation algorithms, i.e. algorithms doing intrusion / event detection in static scenarios. Some examples are *Basic Background Subtraction* (BBS), *Who? When? Where? What?* (W4) and *Single Gaussian Model* (SGM) [3]. The BBS, as the name indicates, simply compares a current image with a learned background. The W4 learns two backgrounds, the maximum and minimum expected gray scale intensities for each pixel, and detects differences whenever pixels have values outside of the learned ranges. In SGM each pixel is described by a mean and covariance which are updated recursively per frame and, based on this values, foreground and background pixels can be then identified. Notably, all these algorithms rely on the existence of a background representation of the scene.

From the various manners for representing the background e.g. a planar mosaic; a cylinder; a sphere or a cube, we

have selected the cube based representation as it can handle a complete spherical field-of-view (FOV), $360^o \times 360^o$, which is not possible in the planar or cylindric mosaics, and maps perspective images to/from the background using just homographies (as compared to using spherical mappings). The choice of which representation to use depends in essence on the complete field-of-view of the camera. In particular, it is constrained by the field-of-view of the lens combined with the full extent of the pan and tilt motion. If the lens and the pan-tilt ranges cover a narrow field-of-view (horizontal and vertical) then all the referred representations are able to store the background information. When the field of view grows, some configurations stop being able to handle properly.

Mapping the images acquired by a rotating (pan-tilt) perspective camera to a planar mosaic is equivalent to applying homography transformations on the images [4]. More precisely, if one has a sequence of images acquired by a rotating camera, and has found the homographies linking each pair of consecutive images, then information in the last image can be mapped to the plane of the first image simply by cascading the various homographies in between.

The cascading of the homographies works well in the case of narrow fields-of-view. If the lens plus pan-tilt ranges total field-of-view becomes too large then the cascading a sequence of homographies can produce mappings of finite image points to infinite points.

One such example can be simply constructed with a single homography. Considering a camera, $P = K[I_{3\times3}\ 0_{3\times1}]$ that takes two images, separated by a rotation R, one has

$$\begin{cases} m_1 = K[I_{3\times3} \quad 0_{3\times1}]M \\ m_2 = K[R_{3\times3} \quad 0_{3\times1}]M \end{cases} \tag{1}$$

where M denotes one 3D point and (m_1, m_2) are corresponding image points, $m_2 = KRK^{-1}m_1$. Considering the camera normalized, $K = I_{3\times3}$, and that R is a rotation of θ degrees around the y-axis, then one has:

$$m_2 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} m_1. \tag{2}$$

If m_1 is a principal point, $m_1 = [0\ 0\ 1]^T$ then $m2 = [\sin\theta\ 0\ \cos\theta]^T$. Considering in addition that $\theta = 90^o$ then one finally obtains:

$$m2 = [1\ 0\ 0]^T$$

in other words, the principal point of one image is mapped to an infinite point on the other. This situation can be illustrated also graphically as in Fig.2(top).

Noting that one could also consider $\theta = -90^o$, then one concludes that the maximum field-of-view covered by cascading homographies is less than 180^o. In practice this maximum value is much smaller due to resolution issues. Considering for example images being mapped frontally to the planar mosaic in a one to one pixel relationship, then the planar mosaic would be artificially too large as the images being mapped close to 90^o would require a much larger area of the mosaic. If the mosaic resolution is selected according to the images acquired at $\pm90^o$, then the frontal ones would have a too small relevance. Hence, in order to obtain reasonable resolution compromises for mapping frontal and lateral images to the mosaic, the total field-of-view must be much less than the 180^o.

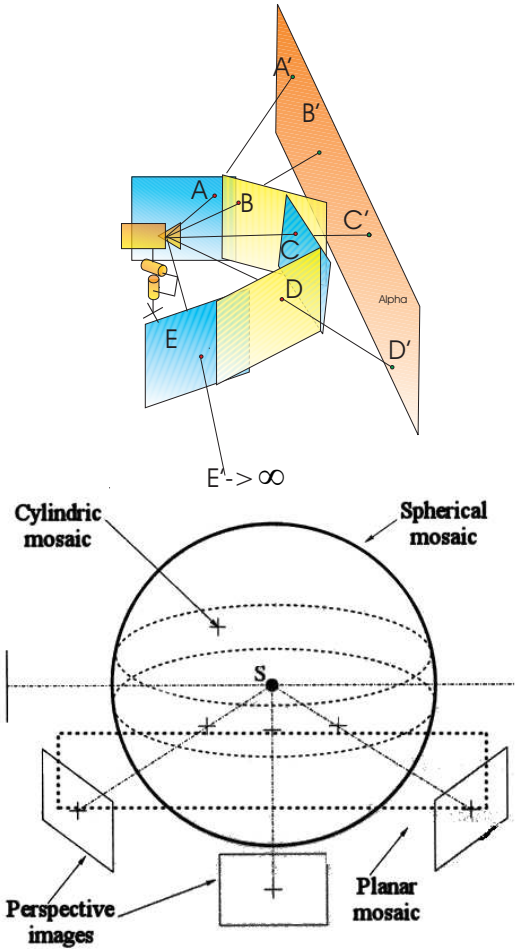


Fig. 2. Building a planar mosaic from various images. A, B, C, D and E represent pixels in the acquired images, while A', B', C', D', E' represent points is the planar mosaic α (top). Other representations, such as the cylindric or spherical mosaics, capable of holding images taken in various directions (bottom).

III. Building Experimental Setups

In order to assess surveillance methodologies, it is necessary to perform extensive testing and thus convenient to generate automatically ground truth information. It is therefore advantageous to build simulated setups. Figure 1 shows a cube based representation of a simulated scenario, having the pan-tilt camera in the center, which is able to survey objects moving within the 3D scene. In our experiments, this representation is maintained in two models, named the *operation model* which contains all the data, and the *background model* in which the mobile objects have been removed. The camera of the operation model captures images while surveying the test scenario, and the camera of the background model captures the corresponding images without the mobile objects, i.e. background images. This setup allows comparing test and background images as required by the segmentation algo-

rithms. The scenarios built to compute results were developed using Virtual Reality Modeling Language (VRML) [2].

Assessing surveillance methodologies implies performing extensive testing and in general one needs to have ground truth information, which can be obtained manually or automatically. Given the huge amounts of data, typically thousands of images, the manual segmentation of the foreground objects to detect is usually a too cumbersome task. Possibly generating automatically ground truth data is certainly convenient.

Simulated scenarios are therefore an important test and development tool. They allow generating video sequences, defining the number and trajectories of the foreground objects, controlling the camera intrinsic/extrinsic parameters and regulating the amounts of image noise. These data can be used repeatedly for exact comparisons among the various surveillance methodologies.

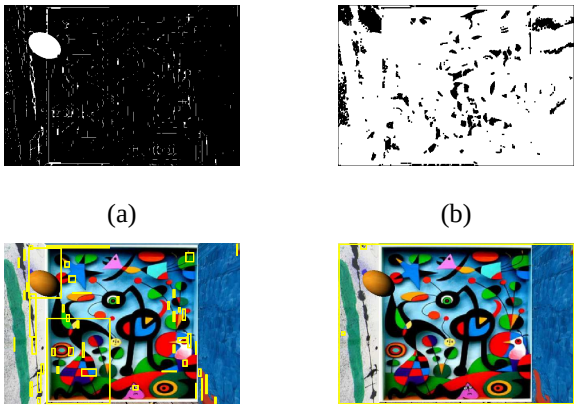
In order to further simplify the process, we have selected spherical objects. One spherical object has always a circular silhouette whatever its pose relatively to a pin-hole camera. Given the sphere radius r, its center location C_s, and the projection matrix P of the camera one obtains the projection of the silhouette following the next few steps:

1) Define a cone starting at the camera center, $C = -P_{1:3,1:3}^{-1} P_{1:3,4}$, having an axis passing through the sphere center C_s, and having its surface tangent to the sphere.
2) Define the 3D tangency circle of the cone (silhouette) and sample it to define a collection of points, $\{P_s\}$. Note that this circle has a radius smaller than the radius of the sphere, $r_2 = rl/d$ where $d = \overline{CC_s}$ denotes the distance between the camera and the sphere and $l = \sqrt{d^2 - r^2}$ is the hypotenuse of the triangle formed by C, C_s and a sphere point on the plane orthogonal to the line $[CC_s]$ and containing the point C_s. The center of the silhouette circle differs also from the sphere center, $C_{s2} = C_s + lr_2(C_s - C)/(rd)$. The line $[CC_s]$ is orthogonal to the silhouette circle, being therefore simple to define two directions spanning a plane containing the circle.
3) Finally, project the silhouette points, $\{P_s\}$ to the image plane using P.

The complete simulation process involves moving the object(s) and the camera, hence changing C_s and P. Upon projection of the silhouette of the object, one is able to compute bounding boxes usually required for correct/false/partial detection metrics.

IV. Control Modalities and Performance Metrics

In our work four modalities were considered for the control of the pan and tilt camera: *Random Search* (RaS), *Rotation Search* (RoS), *Local Search* (LoS) and *Local and Random Search* (LRS). In order to have a pan-tilt event detection algorithm this control strategies must be combined with the segmentation algorithms described in the previous section. All segmentation algorithms described rely in the existence of a background model for the test image, this fact has particular relevance when we move from a fixed camera scenario to a pan-tilt system where each acquisition can lead to a different background model. Figure 3 presents two examples where the incorrect background models were used for segmentation purposes. For small deviations, some errors in the segmented regions can occur but the object is detected, but in the case that the background model is totally unrelated with the acquired image, the results fail completely.



(a) (b) (c) (d)

Fig. 3. (a) presents the results of a thresholding operation where there is a small pan-tilt deviation between the test image and the background model. (b) presents the results of the same thresholding operation in the case that a big pan-tilt deviation exist between the test image and the background model. (c) and (d) represent respectively the overall segmentation results derived from (a) and (b). It can be easily identified that in the case of a small deviation, the object was still detected, but for a big deviation the object was missed. Note that, the yellow frames denote the events detected by the algorithms.

Both RaS and RoS represent open loop algorithms where the camera would acquire images independently from the segmentation results. In RaS the sensing device acquires images while it is moved randomly (uniform distribution) within the pan and tilt limits. Hence, RaS requires very high operating speeds, to jump everywhere at anytime, and has expectedly a limited performance, as when it finds an object it does not try to keep it in the FOV. It is however an interesting control modality because of its simple statistical characterization. In RoS the camera is rotated (pan angle) with a constant step during image acquisition process, and thus searches systematically the scene, similar to RaS after a long time of operation.

The rotational joints, pan and tilt, induce mobile coordinate systems (referential frames) describing the transformation of a 3D point seen by the camera, ${}^c\tilde{M} = [{}^cM^T \ 1]^T = [{}^cX \ {}^cY \ {}^cZ \ 1]^T$ to a 3D point seen in the world coordinate system, ${}^w\tilde{M} = [{}^wM^T \ 1]^T = [{}^wX \ {}^wY \ {}^wZ \ 1]^T$. Figure 4 lists those coordinate systems, starting from the world frame $\{W\}$, passing through the camera base $\{B\}$, the pan rotation $\{pan\}$, the tilt $\{tilt\}$ and finally ending in the camera frame, coincident with the pin-hole point, $\{C\}$. Using 4×4 transformation matrices, one has:

$$ {}^w\tilde{M} = {}^wT_c \cdot {}^c\tilde{M} = {}^wT_b \cdot {}^bT_{pan}(\alpha) \cdot {}^{pan}T_{tilt}(\beta) \cdot {}^{tilt}T_c \cdot {}^c\tilde{M} \quad (3) $$

where α and β denote the current pan and tilt values. Assuming zero offsets (translations) between all the coordinate

systems, then one has simply to consider the rotation matrices:

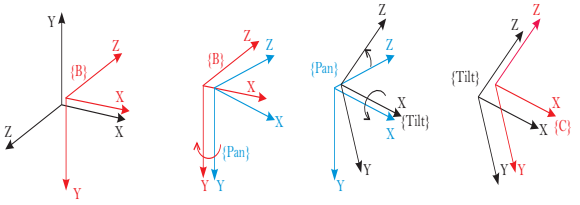
$$^{w}M = {}^{w}R_c \cdot {}^{c}M = {}^{w}R_b \cdot {}^{b}R_{pan}(\alpha) \cdot {}^{pan}R_{tilt}(\beta) \cdot {}^{tilt}R_c \cdot {}^{c}M. \quad (4)$$

The constant transformations have simple expressions, $^{w}R_b = diag(1,-1,-1)$ and $^{tilt}R_c$ is the identity. The other transformations are just rotations around canonical axis:

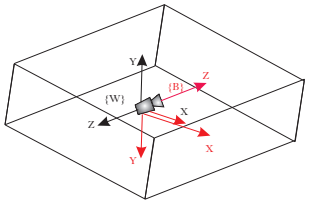
$$^{b}R_{pan} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}, \quad (5)$$

$$^{pan}R_{tilt} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix}. \quad (6)$$

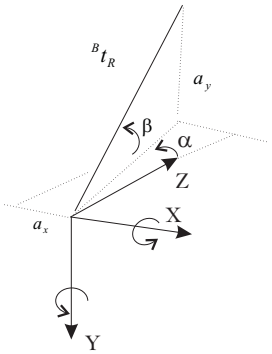
Using the transformations just presented, the camera projection matrix can be obtained as $P = K[^{c}R_w \ 0_{3\times1}]$, where $^{c}R_w$ corresponds to the inverse of $^{w}R_c$ detailed by Eq.4.



(a) detailed representation of the coordinate systems



(b) camera base in the world



(c) pan tilt angles

Fig. 4. Coordinate systems (frames) associated to the VRML world $\{W\}$, the camera base $\{B\}$, the pan and tilt rotation axis, $\{pan\}$ and $\{tilt\}$, and to the camera $\{C\}$.

Given the defined coordinate systems, the centering of an object in the image is stated simply as the regulation of the pan and tilt angles to move the image point marking the object center to the image center. Using the back-projection equation, one obtains one (valid candidate) 3D point, and then set the pan and tilt angles to place that 3D point at the origin. In formal terms:

$$^{w}D = {}^{w}R_b \cdot {}^{b}R_{pan}(\alpha+\delta\alpha) \cdot {}^{pan}R_{tilt}(\beta+\delta\beta) \cdot {}^{tilt}R_c \cdot {}^{c}D \quad (7)$$

where $\delta\alpha$ and $\delta\beta$ denote the incremental pan and tilt values to be found, $^{c}D = [0 \ 0 \ \|^{w}D\|]^{T}$, and ^{w}D comes from a back-projection equation:

$$^{w}D \doteq (KR)^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (8)$$

where R denotes the camera pose considering the current pan and tilt, (α, β) angles.

The inverse kinematics problem of finding the pan and tilt angles, as stated in Eq.7, has one single solution if one imposes $(\alpha+\delta\alpha) \in [0, 360^o[$ and $(\beta+\delta\beta) \in]-90^o, 90^o[$, i.e. excluding the poles or larger areas if imposed by the physical limits of the system.

Note however that the inverse kinematics problem can be further simplified considering that incrementing the pan angle is independent of incrementing the tilt angle. In other words, one can assume that the camera is at zero pan and tilt values ($R = I_{3\times3}$ in Eq.8), and the angles seen in Fig.4(c), correspond to the symmetric values of the increments to be performed:

$$\begin{aligned} \delta\alpha &= -atan(\frac{D_x}{D_z}) \\ \delta\beta &= -atan(\frac{D_y}{\sqrt{D_x^2 + D_z^2}}). \end{aligned} \quad (9)$$

The implementation of the pan and tilt control just has therefore to save the current values and compute the increments whenever necessary. In a real system it is important to add a dead zone into the calculations, so that the motors in the axis are not constantly shaking due to the (otherwise) noisy actuations.

LRS and LoS are closed loop algorithms, meaning that the segmentation results define the next orientation of the camera. Both algorithms try to center the object in the frame by computing and setting the pan-tilt angles when a detection is found. They differ when the object is lost, LRS starts commanding randomly the pan-tilt camera until an object is found, while LoS performs a local search around the last detection point before entering a random search mode. Figure 5 gives an overview of these control strategies.

An example of a very well known metric is the percentage of *Correct Detections* (%CD) in a sequence of N images [3]:

$$\%CD = 100 \times \frac{\sum_{i=1}^{N} CD(I_i)}{\sum_{i=1}^{N} GT(I_i)} \quad (10)$$

where $CD(I_i)$ denotes the number of correct detections (objects) found in the *i-th* image and $GT(I_i)$ is the ground truth number of objects in the image. In order to consider pan-tilt cameras, we propose instead using the percentage of *Events Found* (%EF):

$$\%EF = 100 \times \frac{\sum_{i=1}^{N} CD(I_i)}{\sum_{i=1}^{N} GT(I_i) + \sum_{i=1}^{N} GT(\bar{I}_i)} \quad (11)$$

where $\bar{I}_i$ is an image based representation of the scene observable by the pan-tilt, but not accounted in I_i. Hence, the denominator of the fraction represents now all objects moving

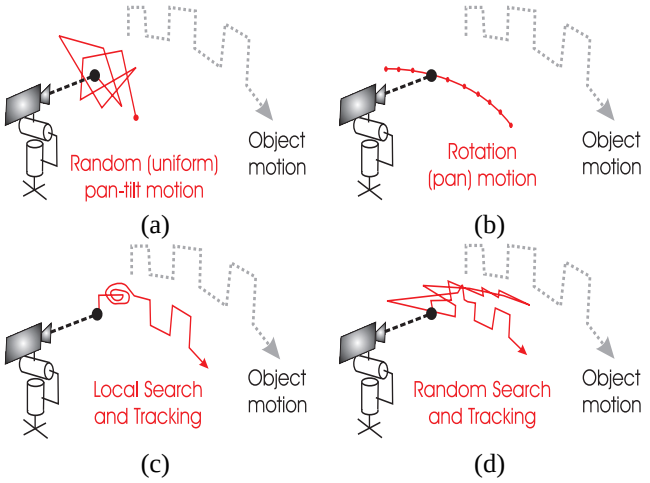


Fig. 5. Representation of the control modalities implemented:(a) RaS (b) RoS (c) LoS (d) LRS

in the complete field-of-view of the pan-tilt camera, i.e. the number of non-background objects that can be observed by sweeping the full pan and tilt angle ranges.

In particular the RaS control modality allows obtaining a simple expression for the probability of finding one object by randomly sampling the pan-tilt complete field-of-view, and thus estimating the %EF metric. Assuming a punctual object and no image noise[1], then the %EF metric can be theoretically estimated by a ratio of solid angles:

$$\%EF_{RaS} \approx 100 \times \frac{\Omega_{cam}}{\Omega_{pan\times tilt}} \tag{12}$$

where $\Omega_{cam} = 4\arcsin\left(\sin(\alpha/2)\sin(\beta/2)\right)$, is the solid angle of a perspective camera (pyramid) with $\alpha \times \beta$ (rad) FOV, and $\Omega_{pan\times tilt}$ is the solid angle corresponding to the complete FOV considering the full pan and tilt ranges. Therefore, $\Omega_{pan\times tilt}$ is the solid angle of a sphere minus the sphere caps not reached by the maximum tilting, $\Omega_{pan\times tilt} = 4\pi - 2\Omega_{cap}$. Each of the two non-reachable sphere caps can be represented by the solid angle of a cone with apex angle 2θ, i.e. $\Omega_{cap} = 2\pi\left(1 - \cos\theta\right)$ where θ is $pi/2$ minus the maximum tilt (τ_M) and minus half the vertical FOV, $\theta = pi/2 - \tau_M - \beta/2$. In the case that the maximum pan angle (ρ_M) is less than 2π, then $\Omega_{pan\times tilt} \leftarrow \rho_M/(2\pi) \times \Omega_{pan\times tilt}$.

V. Experiments

The first experiment performed for the setup described intended to reflect the capabilities of the different video segmentation techniques. To do so, one mobile object describing a growing spiral was surveyed at different zoom levels i.e. 9^o, 19^o, 34^o, 68^o and 90^o. Figure 6 gives an overview of the level of noise considered.

Two metrics were particular relevant for this assessment: percentage of correct detections and percentage of false alarms. The first characterizes the accuracy of the segmentation algorithm and the second the noise rejection robustness.

[1]Non punctual objects are considered by enlarging the camera FOV, and then the image noise can be mitigated by morphological processing.

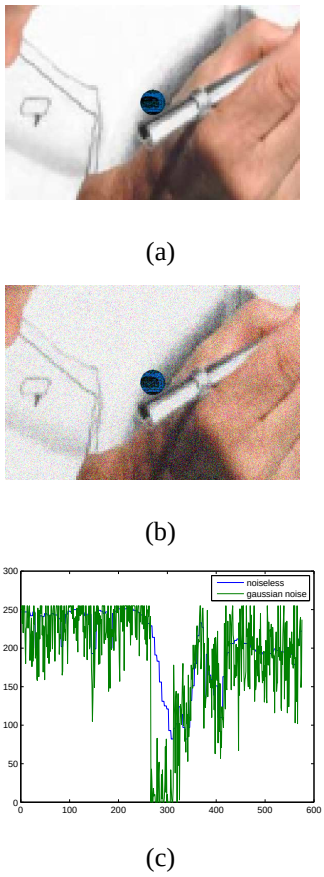


Fig. 6. Random noise added to the test images. Noise free test frame (a), test frame with noise (b) and random noise added to be added to the test frame (c).

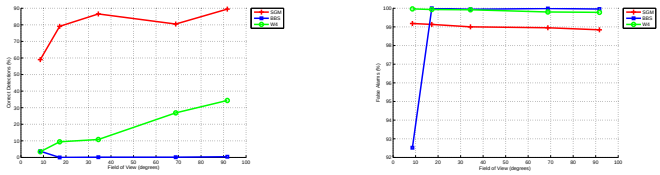


Fig. 7. Test results: As previously mentioned, correct detection and false alarms seem to be the most effective metrics to characterize the performance for the setup considered. The plots reflect the expected results where SGM seems to be the best candidate for segmentation purposes.

It is clear from Figure 7 that SGM is the most efficient segmentation algorithms for this setup, since it presents the higher percentage of correct detections and the less percentage of false alarms. W4 seems to have a more linear behavior than SGM leading to the idea that at a certain point, i.e. for a particular FOV it will have the same performance as SGM. BBS algorithm was completely overridden due to the noise. As detailed, more than having a very low percentage of correct detections it also presents the higher percentage of false alarms. This results were expected since BBS is based on a threshold operation which is not sufficient to clearly classify every pixel as foreground or background. W4 results were also expected, it had worst behavior than SGM since it only considers gray scale intensities to compute the thresholding operation. SGM uses 3 color channels and besides this level

of information, it also computes second order statistics i.e. median and covariance to describe each pixel which is more accurate that W4 thresholding operations.

A second experience was implemented to evaluate the control modalities in a multiple object environment. As previously detailed closed loop algorithms tend to follow an object depending on the size of the walk step. This reveals the inefficiency of closed loop algorithms in a multiple object scenario, since they would be focused on the first detected object, neglecting further events in scene. Considering a military scenario this could be critical since distracting techniques could be used to mislead the algorithms, furthermore it is clear that an open loop strategy would be more useful in this case. To compare/highlight the advantages of open loop control modalities with respect to fixed cameras a two objects experiment was implemented. In this way, two objects were moved in a noise free scenario with the trajectories represented in 8. This scenario was then surveyed using the previously considered pan and tilt control modalities, i.e. RaS, RoS, LoS and LRS combined with W4 segmentation algorithm.

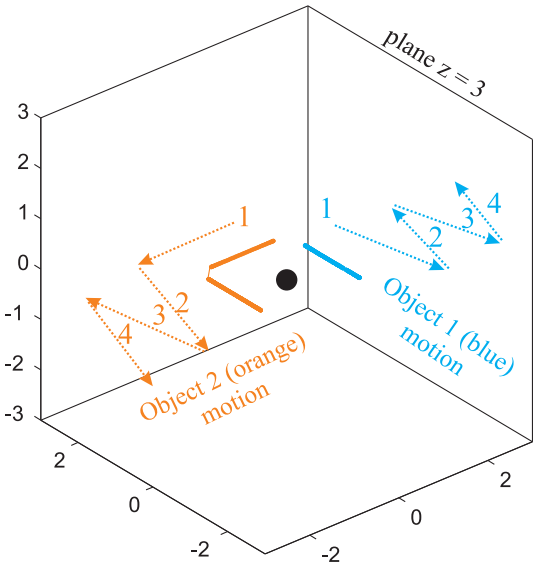


Fig. 8. Two objects, *blue* and *orange*, moving around the pan-tilt camera (black dot in the center of the cube). The camera is initially facing the plane $z = 3$. Both objects start in its field of view. Object *blue* always stays within the initial field-of-view while object *orange* goes behind it.

Figure 9 presents the pan and tilt behavior for all commanding techniques, as well as the number of events detected per frame. It should be highlighted that during this experiment only two mobile objects were used. It is clear from 9 that in terms of total number of events LoS and LRS will be the best algorithms. As previously mentioned this consideration is deeply dependent on the surveillance purpose. If the surveillance goal is to detect as much different objects as possible, possibly RaS or RoS might win.

In order to illustrate the information introduced by the %EF metric, the control modalities have been applied in the simulated setup described in Sec.II, and assessed by both the %CD and %EF metrics. The mobile object (ball) moves around the camera, spanning a space larger than the FOV of the camera. Predictably, the open loop control modalities fail

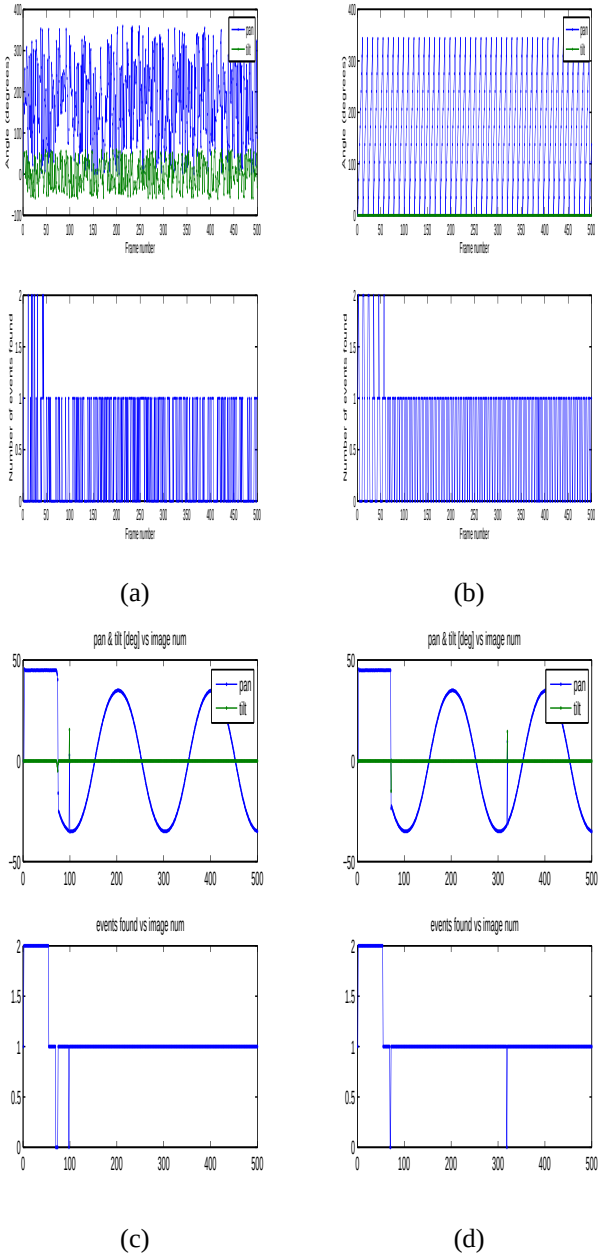


Fig. 9. Open versus closed-loop control modalities, pan-tilt signals and number of objects detected in each frame. It is clear that, once the first object is detected the closed loop algorithms will find at least one object per frame. In this way, it is not possible to state that the closed loop methods are the golden bullet standards for variety in terms of objects detected, but they are at least an efficient approach if the aim is detect everything possible.(a) RaS (b) RoS (c) LoS (d) LRS

more detections of the object since they do not track it after finding it. This aspect is expected to be less severe as the FOV of the camera increases.

Figure 10 shows the performance metrics, %CD and %EF, for five FOV configurations. Each control modality has been tested on 500 images long sequences, using the BBS event detection methodology as in our synthetic scenario the W4 and SGM methodologies yield similar results. Both metrics confirm that all the control modalities tend to detect more times the mobile object when the FOV of the camera increases.

Note however that the %CD metric does not show the expected clear distinction between open and closed loop control modalities. The interpretation is that the %CD metric does not count the objects that are out of the instantaneous FOV but, being in the vicinity of the camera, could be found (tracked) with a closed loop modality. The %EF metric effectively confirms the intuition that the closed loop control modalities are advantageous. Consistently, the theoretical prediction of the %EF for the RaS (labelled RaS-T in the plot) closely matches the experimentally observed results of the %EF and thus confirms the statistical significance of the realized number of experiments.

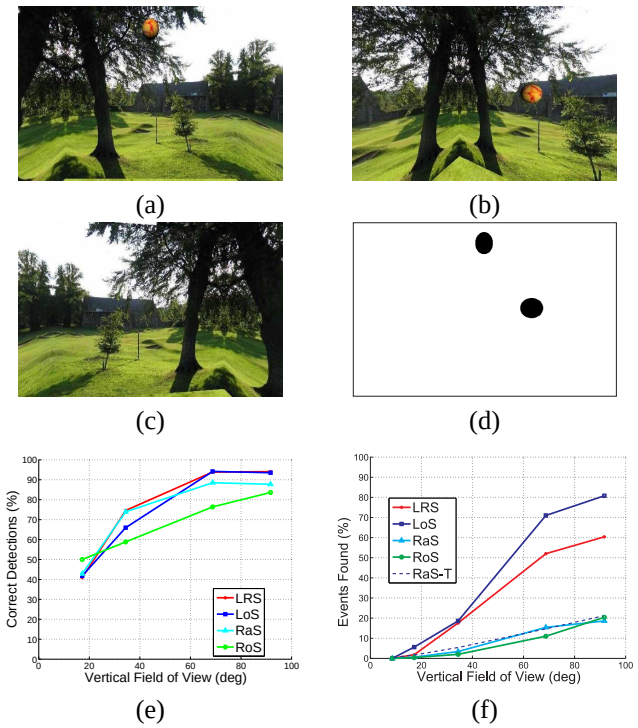


Fig. 10. Three sample images of the simulated scenario while the camera is panning to the left and the object is falling (a), (b) and (c); object detections superimposed on a single image (d). The %CD and %EF metrics versus the vertical FOV of the camera (e and f).

A fourth experiment was implemented in order understand the evolution of the commanding techniques with the random walk noise variation. For this purpose, one object was moved in accordance with the model:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} \partial x_t \\ \partial y_t \\ \partial z_t \end{bmatrix} \tag{13}$$

where $\partial x_t, \partial y_t, \partial z_t$ are Gaussian random variables, denoting the random steps of the random walk motion. In order to grant that the object would remain within the scenario, a saturation process detailed by equation 14 was used. In this equation, $M_t = [x_t, y_t, z_t]'$ should be considered the object's position and $d_{\min}$, $d_{\max}$ the scenario limits.

$$M_t = \begin{cases} d_{\min} \cdot \frac{M_t}{\|M_t\|}, & \|M_t\| < d_{\min} \\ d_{\max} \cdot \frac{M_t}{\|M_t\|}, & \|M_t\| > d_{\max} \\ M_t, & otherwise \end{cases} \tag{14}$$

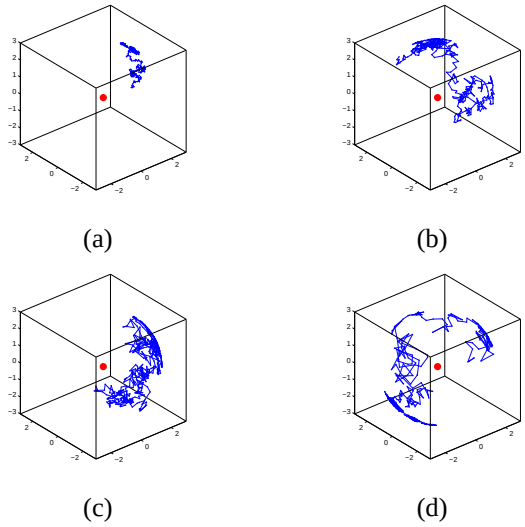


Fig. 11. Random Walk movement with different steps. The camera position is represented by the red sphere, while the object movement can be identified in blue. (a). Step = 0.2 m (b). Step = 0.6 m (c). Step = 1.0 m (d). Step = 1.4 m

This scene was then surveyed using four different pan-tilt surveillance sets (i.e. RaS, RoS, LoS and LRS) and the results evaluated using the metrics previously described. During this experiment, as detailed in figure 11 four random walk steps were evaluated: 0.4, 0.6, 1.0 and 1.4. The camera was configured to have a field of view of 34 degrees which represents a typical field of view for cameras used in surveillance systems. W4 was the algorithm selected for segmentation purposes since like SGM it presented the best performance for the setup considered.

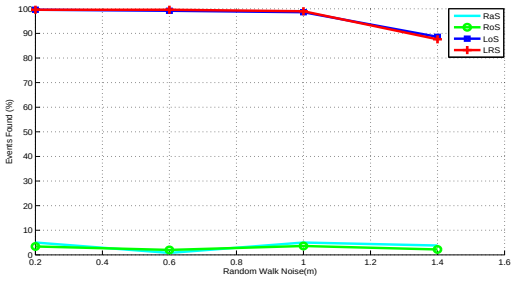


Fig. 12. Overview of results intended to clarify performance of the control algorithms with respect to random walk noise variation. It is clear from this plot the efficiency of both LRS and LoS with respect to RaS and RoS. The closed loop control strategies are able to follow the object for small steps, but when the noise step increases these strategies start losing performance since they are not able to follow the object properly. The open loop strategies are quite robust to step variation but lack in performance. The Percentage of Correct Detections for different walk noise is not very useful to evaluate the commanding algorithms performance, since it compares the detected regions by the algorithm with the ground truth image but it does not considers information regarding the possible available events. It gives average information regarding the segmentation but does not includes the results from the commanding information.

VI. Final Notes and Future Work

This article highlights the need of novel metrics for performance evaluation of surveillance systems encompassing pan

and tilt cameras. While previously proposed metrics considered already false detections, object splits, merges, and both, and time/space evolutions such as identifying configuration vs tracking problems, most of the research was concentrated on fixed cameras. When considering pan-tilt cameras, one has also the objective of designing control algorithms that give a sense of the events happening in the complete scenario, in other words one desires to build surveillance systems that are more omni-aware. This work proposed a metric adjusted to evaluate such designs and showed its theoretical estimation for a case of random searching. Future work will focus on designing novel control modalities.

REFERENCES

[1] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *Int. Journal of Comp. Vision*, 74(1):59–73, 2007.

[2] W3 Consortium. Virtual reality modeling language. http://www.w3.org/MarkUp/VRML/.

[3] Jacinto C. Nascimento and Jorge S. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4):761–774, 2006.

[4] S. N. Sinha and M. Pollefeys. Towards calibrating a pan-tilt-zoom camera network. In *Department of Computer Science, University of North Carolina at Chapel Hill*, pages 91–110, 2006.

[5] K. Smith, D. Gatica-Perez, and S. Ba J. Odobez. Evaluating multi-object tracking. In *Comp. Vision and Patt. Recogn. - Workshops*, 2005.