Trajectory-based Abnormality Categorization for Learning Route Patterns in Surveillance

Pau Baiget, Carles Fernández, Xavier Roca, and Jordi Gonzàlez

Abstract The recognition of abnormal behaviors in video sequences has raised as a hot topic in video understanding research. Particularly, an important challenge resides on automatically detecting abnormality. However, there is no convention about the types of anomalies that training data should derive. In surveillance, these are typically detected when new observations differ substantially from observed, previously learned behavior models, which represent normality. This paper focuses on properly defining anomalies within trajectory analysis: we propose a hierarchical representation conformed by Soft, Intermediate, and Hard Anomaly, which are identified from the extent and nature of deviation from learned models. Towards this end, a novel Gaussian Mixture Model representation of learned route patterns creates a probabilistic map of the image plane, which is applied to detect and classify anomalies in real-time. Our method overcomes limitations of similar existing approaches, and performs correctly even when the tracking is affected by different sources of noise. The reliability of our approach is demonstrated experimentally.

1 Introduction

Recognizing abnormal behaviors is a main concern for research on video understanding [7]. The challenge exists not by the difficulty of implementing anomaly detectors, but because it is unclear how to generally define *anomaly*. On the one hand, anomalies in video surveillance are usually related to suspicious or dangerous behaviors, i.e., those for which an alarm should be fired. Unfortunately, such a vague concept is difficult to learn automatically without prior explicit formal mod-

Pau Baiget · Carles Fernández · Xavier Roca · Jordi Gonzàlez
Dept. Ciències de la Computació & Computer Vision Center
Edifici O, Campus Universitat Autnoma de Barcelona, 08193 Bellaterra, Barcelona, Spain,
e-mail: pbaiget,perno,xavir,poal@cvc.uab.cat

els. Existing top-down approaches take advantage of this prior models to identify suspicious behavior [3] or to generate conceptual descriptions of detected behavior patterns [5]. However, these approaches are scenario oriented and are designed to recognize or describe a reduced set of specific behaviors.

On the other hand, from a statistical perspective, recent works in anomaly detection assume that normal behavior occurs more frequently than anomalous one, and define anomalies as deviations from what is considered to be normal [1, 8, 10, 11]. In surveillance, a standard learning procedure consists of extracting observations by motion tracking over continuous recordings to build scenario models that determine the normality or abnormality of new observations.

We define anomalies considering the types of deviations from a learned model. We distinguish among three semantic types, namely *Soft* (*SA*), *Intermediate* (*IA*), and *Hard* (*HA*) anomalies. In essence, a *SA* represents a slight deviation from the parameters of a typical pattern, e.g. a person running, stopping, walking backwards for a while, etc. Secondly, a *HA* occurs when an observed event occurs completely outside of the model parameters, e.g. a person appearing from an unnoticed entrance. These two types are currently recognized by different approaches, see Table 1, but there exists an important gap between them. The *IA* represents observations that deviate from learned patterns but still fit into the model, e.g., starting a typical path and changing to a different one.

To recognize these anomaly types, we present a novel unsupervised learning technique that generates a scenario model in terms of paths among observed entry and exit areas. We use Gaussian Mixture Models (GMM) to describe transition probabilities between pairs of entry and exit points. GMM model n-dimensional datasets, where each dimension is conditionally independent from the others. Thus, although our method currently uses only trajectory positions to fit GMM into the training set, it can be easily extended to features like bounding boxes, orientations, speed, etc. Additionally, our methodology identifies online previously introduced anomalies in new observed trajectories.

Recent surveillance techniques to learn motion patterns have been applied to anomaly detection, behavior description, semantic scene labeling, or tracking enhancement. Johnson and Hogg [9] and Fernyhough et al [6] use vector quantization to learn routes and semantic regions. The methods depends on target size. More recently, Makris et al [10] proposed a method to label semantic zones –entry, exit, stop area– and model typical routes. However, temporal consistency is not maintained, and speed variation anomalies are not covered. Hu et al [8] use spatiotemporal trajectory clustering to detect abnormality and predict paths. However, anomalies are only defined as low probable fittings. Piciarelli et al [11] proposed an online method to generate route models from key points. The order of apparition of the observations affects the model. Finally, a probabilistic method by Basharat et al [1] detects abnormality not only from trajectory positions but also from object sizes. However, this method is very dependent on a clean training dataset, classifying as anomaly slight variations of previously observed trajectories.

Next section describes the creation of scenario models from noisy trajectory training sets. Next, the hierarchy of anomalies is introduced. Subsequently, the per-

Approach	Model used	Robust to noise	Types of anomaly
Fernyhough et al [6]	Vector quantization	No	HA
Makris et al [10]	GMM and vector quantization	Yes	HA
Hu et al [8]	Fuzzy C-means and GMM	Yes	SA, HA
Piciarelli et al [11]	Vector quantization	No	HA
Basharat et al [1]	Transition vectors and GMM	No	SA, HA
Our approach	GMM with Splines	Yes	SA, IA, HA

Table 1 Comparison of previous approaches in terms of model, robustness and the types of anomalies detected.



(a) (b) (c)

Fig. 1 (a) Original dataset of 4250 trajectories. (b) Detected Entry areas. (c) Dataset after removing noise caused by tracking failure and semistationary motion [10].

formance of the proposed method is demonstrated. Last section concludes the paper and shows future lines of research.

2 Scenario model learning

Our method learns from large recording datasets without previous manual selection of trajectories. Therefore, the initial dataset can contain spurious data, see Fig. 1(a). The most relevant noise issues are due to:

1. Failure of the motion-tracking algorithm. Errors may appear in form of *false* trajectories, where the motion history of multiple targets are mixed, or *split* trajectories that represent only a portion of the history.
2. Semistationary motion noise, e.g., trees, curtains, or window reflections. Apparent activity is detected in the vicinity of the noise source.
3. Non-smooth trajectories caused by inaccurate tracking or severe scenario conditions, causing irrealistic representations of motion in the scene.

The two first problems are solved by applying a multistep learning algorithm of Entry and Exit zones [10]. Fig. 1(c) shows the detected entry areas of the scenario. The sets S and E allow to obtain a subset $T' \subseteq T$ of trajectories that start in some entry of S and end in some exit of E, see Fig. 1.(d):

$$T' = \{t \in T | \exists_{s \in S, e \in E} : begin(t,s) \ \wedge \ end(t,e)\} \quad (1)$$

The third problem is solved by representing each T' with a continuous function model that solves tracking inaccuracies. We use a sequence of cubic splines [4], denoting as $\mathrm{S}(t)$ the spline representation of trajectory t. The required number of cubic splines is automatically decided by computing the error between the original trajectory and the computed spline sequence:

$$error(t, \mathrm{S}(t)) = \sum_{(x,y) \in t} d((x,y), \mathrm{S}(t)) \quad (2)$$

Inner points are sampled with any required precision, maintaining the original temporal consistency. Moreover, it only requires storing intermediate control points and derivatives, thus reducing disk storage demand in large datasets.

2.1 Scenario model

Next we detail the creation of route models from T'. A *route* $R_{s,e}$ between areas $(s,e) \in S \times E$ is defined using GMM as a *normal* way to go from s to e. Due to scene constraints or speed variations, there could be more than one route assigned to a pair (s,e). A *path* $P_{s,e}$ contains the routes from s to e. The final scene model includes all possible paths for each pair (s,e).

$$R_{s,e} = \{G_1, \ldots, G_k\} \quad (3)$$

$$P_{s,e} = \{R^1_{s,e}, \ldots, R^U_{s,e}\} \quad (4)$$

$$M = \{P_{s,e} | s \in S, e \in E\} \quad (5)$$

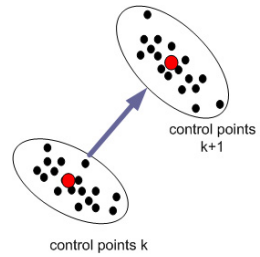
2.2 Learning algorithm

The following procedure is applied to each pair $(s,e) \in S \times E$. Let $T_{s,e} = \{t_1, \ldots, t_N\} \subseteq T'$ be the set of trajectories starting at s and ending at e. Each trajectory t_n is represented by a sequence $r(t_n)$ of K equally spaced control points, obtained by sampling from $s(t_n)$, $r(t_n) = \{p^n_1, \ldots, p^n_K\}$, where p^n_k corresponds to the sample point k/K of $s(t_n)$.

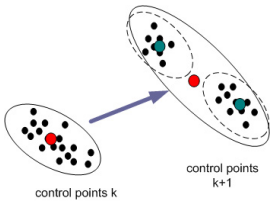
The learning algorithm is sketched in 2(a): the input is a matrix of $K * N$ points, where each row k is the list of $k-$th control points of each trajectory in $T_{s,e}$. The path $P_{s,e}$ is initialized considering that all trajectories follow a single route. The algorithm traverses the lists of control points from 1 to K. At each step, the algorithm fits one- and two-component GMMs into the current point list, formed by the k–th control point of each trajectory, $\{p^1_k, \ldots, p^n_k\}$. The representativity of the model is evaluated through a density criterion:

Initialize $P_{s,e}$ with a single route R^1
for $k = 1$ to K **do**
  $nc \leftarrow |P_{s,e}|$
  for $c = 1$ to nc **do**
    $list \leftarrow points(k,c)$
    $G \leftarrow GMM(list,1)$
    $(G_1,G_2) \leftarrow GMM(list,2)$
    if $\delta(G) > (\delta(G_1)+\delta(G_2)) * \alpha/2$ **then**
      create R^{c_1}, R^{c_2} from R^c
      add G_1 to R^{c_1}
      add G_2 to R^{c_2}
      split $points(k,c)$ according to G_1 and G_2
      substitute R^c with R^{c_1}, R^{c_2} in $P_{s,e}$
    else
      add G to R^c
    end if
  end for
end for

(a)



(b)



(c)

Fig. 2 (a) Route Modeling Algorithm for a pair $(s,e) \in S \times E$. It considers two possible situations: (b) the density of a single gaussian is enough to represent the k-th set of control points; or (c) two gaussians represent it better, so the route model splits.

$$\delta(G) = \frac{w}{\pi \cdot \sqrt{|\Sigma|}} \tag{6}$$

where w is the prior probability of G and Σ is its covariance matrix. If the mean density of the two-component GMM is higher than the single one, current route R^c splits into subroutes R^{c_1} and R^{c_2}, see Fig. 2(c). Once the algorithm has been applied to each pair (s,e), M contains the set of paths associated to *normality*. A trajectory deviating from M will be considered an anomaly. This result is enriched by considering the type of anomaly observed, as explained next.

3 Online anomaly detection

We distinguish three types of anomaly, see Fig. 3: (i) soft anomaly, **SA**: a modeled path is followed, i.e., the value of $p(P_{s,e}|\Theta,\omega)$ remains stable, but certain speed or orientation values differ from learned parameters; (ii) intermediate anomaly, **IA**: the most probable path $P_{s,e}$ changes during the development, for a whole window ω; and (iii) hard anomaly, **HA**: a completely unobserved path is followed, which can be caused by $e' \notin E$ or because the probability of having started from s is too low for the whole window ω.

This hierarchy provides degrees of deviations between new observations and the model. Indeed, a *SA* can be considered as a route deviation inside a path; a *IA* detects path deviations within M; and a *HA* complete deviates from M. The prior probability

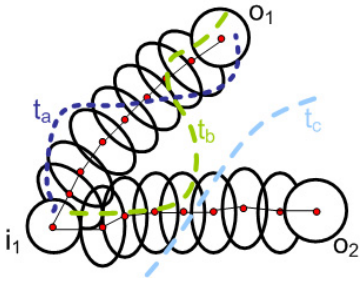


Fig. 3 Types of anomalies: t_a incurs in *SA* by partially falling out of a route. t_b shows *IA* because of changing paths. t_c deviates from all routes, thus producing *HA*.

that a pixel location (x,y) is a part of a modeled path, $p(P_{s,e}|(x,y))$, is obtained by computing a probabilistic map of the image plane. These probabilities are stored in a $h \times w \times |M|$ matrix Θ, where h,w are the dimensions of the image and $|M|$ is the number of paths:

$$\Theta_{x,y} = p(P_{s,e}|(x,y)) = max\,(p(G_j|(x,y)) \tag{7}$$

where G_j belongs to some route $R^u_{s,e} \in P_{s,e}$.

Anomaly detection is performed online as new trajectories are available. Given the current observation, we maintain a temporal window ω with the last $|\omega|$ observations. We compute the probability of being in a given path by assuming correlation with previous frame steps:

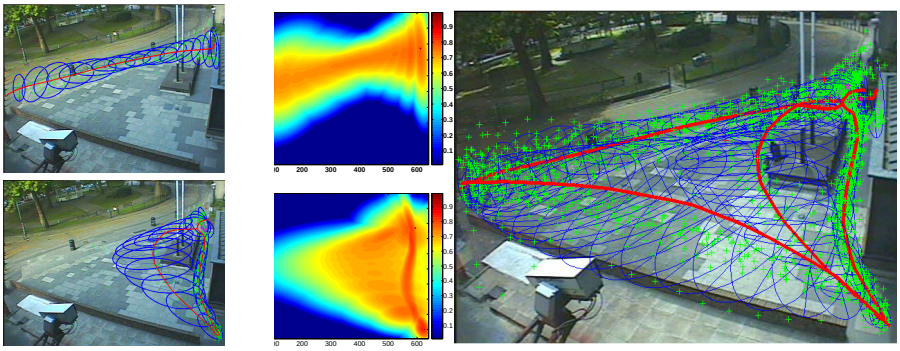
$$p(P_{s,e}|\Theta,\omega,(x,y)) = \alpha * \Theta_{x,y} + (1-\alpha) * \frac{\sum_i^{|w|} p(P_{s,e}|\Theta,w_i)}{|w|} \tag{8}$$

where w_i is the i–th observation in ω, and the update factor α is set to 0.5.

The anomaly detection algorithm labels trajectories as new tracking observations arrive. A new trajectory t is associated to one of the learned entries $s \in S$. If no entry is close enough, t is labeled as *HA*. Otherwise, the probabilities of being in each path containing s are computed using Eq. 8, for each new observation (x,y). In case of similar probabilities, t is not assumed to follow any concrete path.

When the probability of path $P_{s,e}$ is higher than the others, the trajectory is assumed to be following it for the next frames. If a different path becomes more probable t starts being labeled as *IA* until the initial path becomes the most probable again; if t finally ends in e', then the *IA* period is that in which $P_{s,e}$ was less probable. Finally, if all paths are improbable for a given ω, t is marked as *HA* until a probable path is assigned. Assuming a path $P_{s,e}$, we select the gaussian G^* that better represents the current observation:

$$j^* = \arg\max_j \; p(G_j|(x,y)) \tag{9}$$



(a) (b) (c)

Fig. 4 (a) Learned routes, (b) their Θ maps, and (c) the final model.

To know if path $P_{s,e}$ is being followed normally, we compute the sum of probability increments that ω is between Gaussians G_{j^*-1} and G_{j^*+1}, previous and next on the path. This is encoded into a *normality* factor $F(\omega)$:

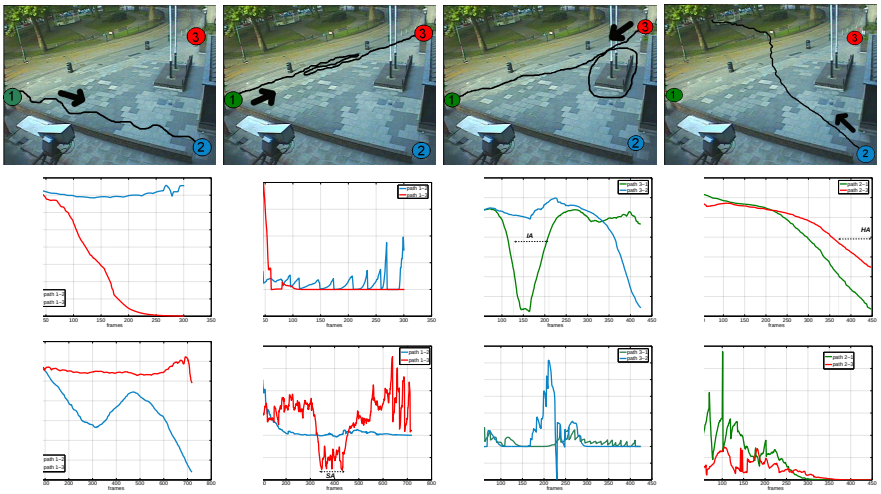
$$F(\omega) = \begin{cases} \sum_i^{|\omega|} -\Delta p(G_{j^*-1}|\omega_i) & \text{if } h(G_{j^*},\omega) > 0 \\ \sum_i^{|\omega|} \Delta p(G_{j^*+1}|\omega_i)) & \text{otherwise} \end{cases} \tag{10}$$

where $\Delta p(G|\omega_i) = p(G|\omega_i) - p(G|\omega_{i-1})$, and $h(G_{j^*},\omega) = p(G_{j^*}|\omega_{|\omega|}) - p(G_{j^*}|\omega_1)$ measures the direction of ω towards G_{j^*}. Positive values of $F(\omega)$ certify that the last ω observations are normal according to M. Otherwise, the trajectory deviates from M, so we annotate subsequent frames as *SA* until the deviation is over. Note that $F(\omega)$ is only representative when a path is followed, i.e., the path has been the most probable during ω. Thus, when labeling as *IA* or a *HA*, the value of $F(\omega)$ does not provide any extra information.

4 Results

We evaluate the method using a real database from [2]. The dataset was cleaned as detailed in the text, to avoid the three sources of noise described. A wide range of values were tested for K –number of control points per path– and ω –length of temporal window–. Here we use $K = 10$ and $|\omega| = 20$. Fig. 4 shows results for two paths, their probabilistic maps Θ, and the resulting model M.

Path P_1 is more probable than P_2 if $p(P_1|\Theta,\omega,(x,y)) > 1.5 * p(P_1|\Theta,\omega,(x,y))$. A trajectory is considered *HA* if all paths have probability below 0.5. Fig. 5 shows four results: the first row are trajectories in the image plane. The second and third rows show the temporal evolution of $p(P_{s,e}|\Theta,\omega,(x,y))$ and $F(\omega)$, respectively. Case (a) represents a normal trajectory –single probable path, $F(\omega) > 0$–. Case (b) also has a single assignment, but $F(\omega) < 0$ when the target goes backwards –interval marked



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

Fig. 5 Top: examples of (a) normal trajectory, (b) *SA*, (c) *IA*, and (d) *HA*.

as *SA*–. Case (c) shows an *IA*, since the assignment changes and stabilizes; it is unclear which part is the anomalous one. Finally, case (d) is a *HA*, since the trajectory differs from any path having entry (2).

5 Conclusions

We provided a trajectory-based definition of anomaly in video surveillance. Our hierarchy classifies anomalies regarding the extent and nature of the deviations from a learned model. A novel GMM representation of observed common routes creates probabilistic maps that typify abnormality in real-time. The characteristics of our method overcomes limitations of similar existing approaches, and performs well in spite of noisy tracking conditions. This technique has direct applications in the area of intelligent video surveillance, by providing a richer semantic explanation of observed abnormality than existing approaches. However, there is still a gap between the notion of anomaly in terms of attentional interest and its statistical definition; future work will refine the proposal with a conceptual layer exploiting common knowledge about suspicious behaviors.

Acknowledgments

This work was originally supported by EC grant IST-027110 HERMES project; and by the Spanish projects Avanza I+D ViCoMo (TSI-020400-2009-133), TIN2009-14501-C02 and CONSOLIDER-INGENIO 2010 MIPRCV (CSD2007-00018).

References

1. A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *CVPR*, Anchorage, USA, 2008.
2. J. Black, D. Makris, and T. Ellis. Hierarchical database for a multi-camera surveillance system. *Pattern Analysis and Applications*, 7(4):430–446, 2004.
3. F. Bremond, M. Thonnat, and M. Zuniga. Video understanding framework for automatic behavior recognition. *Behavior Research Methods*, 3(38):416–426, 2006.
4. C. de Boor. *A practical guide to splines*. Springer-Verlag, New York, USA, 1978.
5. C. Fernández, P Baiget, F.X Roca, and J Gonzàlez. Interpretation of Complex Situations in a Semantic-Based Surveillance Framework. Signal Processing: Image Communication, volume 23, issue 7, pp. 554-569, August, 2008.
6. J. H. Fernyhough, A. G. Cohn, and D. Hogg. Generation of semantic regions from image sequences. In *ECCV'96*, pages 475–484, London, UK, 1996. Springer-Verlag.
7. J. Gonzàlez, D. Rowe, J. Varona, and F. Roca. Understanding dynamic scenes based on human sequence evaluation. *IMAVIS*, 27(10):1433-1444, September 2009
8. W. Hu, X. Xiao, Z. Fu, and D. Xie. A system for learning statistical motion patterns. *IEEE TPAMI*, 28(9):1450–1464, 2006.
9. N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *BMVC'95*, pages 583–592, Surrey, UK, UK, 1995. BMVA Press.
10. D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE TSMC–Part B*, 35(3):397–408, June 2005.
11. C. Piciarelli and G. L. Foresti. On-line trajectory clustering for anomalous events detection. *PRL*, 27(15):1835–1842, 2006.