TRACKING GROUPS OF PEOPLE FOR VIDEO SURVEILLANCE

Frédéric Cupillard, François Brémond and Monique Thonnat

INRIA Sophia Antipolis, ORION group
2004, route des Lucioles, BP93 - 06902 Sophia Antipolis Cedex, France
{Frederic.Cupillard, Francois.Bremond, Monique.Thonnat}@sophia.inria.fr

ABSTRACT

We propose in this paper a method for tracking groups of people in a metro scene to recognise abnormal behaviours such as violence or vandalism. After presenting the overall system, we describe the tracking algorithm for groups of people. Finally we present results illustrating the algorithm.

1. INTRODUCTION

In this paper, we present a method for tracking groups of people in a metro scene to recognise abnormal behaviours such as violence or vandalism [1]. Given a video sequence, our proposed algorithm is able to track real groups of people present in the scene. This algorithm is composed of three steps. First, from the current image of metro scene, a motion detector detects mobile objects in the scene, stores each of them in a *moving region* structure which is classified into different types, such as PERSON, GROUP or NOISE. Second, a tracking module detects and follows the real groups present in the scene by computing the trajectories of *moving regions* that can correspond to persons inside a real group. Third an interpretation module recognises the behaviour of the tracked groups. In this article, we focus on the second step, tracking groups of people. Contrary to traditional trackers [2, 3, 4, 5] that track each person individually, our algorithm tracks globally all the persons belonging to the same group and tracks these persons with a time delay to keep coherent the structure of the group during the entire sequence. We illustrate this algorithm with video sequences taken for the European project ADVISOR (http://www-sop.inria.fr/orion/ADVISOR/default.html).

2. MOTION DETECTOR

The goal of this step is to detect mobile objects in the scene and classify them into *moving regions* with a label corresponding to their type, such as PERSON. This task can be divided into three subtasks: detection of *moving regions*, extraction of features, classification of *moving regions*.

First for **detection of *moving regions***, we compute a difference image between a current image and a reference image [6]. Because the camera is fixed, the reference image is a still image representing the scene without mobile objects (also called background image). Second a thresholding of this difference image is realized, then a filtering and connected component analysis create all the *moving regions* (represented by their bounding boxes) that correspond to the mobile objects. The reference image is updated during the process to take into account illumination changes.

The **extraction of features** step consists in the computation for each *moving regions* of 8 parameters: centre of gravity, position, height and width all four defined both in 2D (in the image) and in 3D (in the scene).

The **classification of *moving regions*** step consists in labelling the *moving regions* into 8 semantic classes of objects: 4 classes for people (PERSON, OCCLUDED PERSON, GROUP, and CROWD) and 4 classes for other objects (METRO TRAIN, SCENE OBJECT, NOISE and UNKNOWN). An occluded person is for example a person partially occluded behind a pillar. A scene object is for instance a seat or a trash. The *moving regions* can be merged to improve the matching with a class. For instance, two *moving regions* corresponding to the head and the body of a person can be merged to form a new *moving region* belonging to the class PERSON.

3. GROUP TRACKING

3.1. Tracking Approach

A real group of people is defined as a set of persons who are close to each other. We detect real groups of people thanks to a structure that we call a *group*. This *group* structure is a set of *moving regions* characterised by four particularities:

- Size coherence: each *moving region* of a *group* has the dimensions of a person or bigger if several persons partially overlapp each other.
- Spatial coherence: all *moving regions* inside a *group* are close to each other.
- Temporal coherence: the speed of the *moving regions* inside a *group* cannot exceed the speed of a person.
- Structure coherence: The number and the size of *moving regions* inside a *group* should be stable. If a person goes away from the real group and come back several times, we would like that the *moving region* corresponding to this person stay inside the *group* structure in order to compute a stable behaviour analysis of the real group.

There are three main problems in tracking groups:

- Appearance/disappearance of real groups: a real group could appear and disappear anywhere in the observed scene.
- Dynamic of real groups: some persons or other groups could come and merge into an already existing group then leave it.
- Motion detector failure: persons are not always well detected by a *moving region* (head and body could be detected

in two different *moving regions*) and well labelled (a person could be labelled as NOISE). A class GROUP for *moving regions* exists but characterises a real group only if persons partially overlapp each other. For example, two persons walking together will be characterised by two *moving regions* labelled PERSON and not GROUP.

To handle these problems, the tracking algorithm is divided in three tasks. The first task consists in tracking *moving regions* from frame to frame and in obtaining a graph (shown on figure 1) representing all possible trajectories of all *moving regions*. In this graph, each connected part (sub-graph) can correspond to a group of persons crossing each other. The second task consists in computing inside the sub-graph all possible *paths* (trajectories of a *moving region*) than can correspond to a person inside the real group. In the third step, we compute the *group* structure that gathers all these paths.

3.2. Frame to Frame Tracker

The goal of the frame to frame tracker is to link from frame to frame all *moving regions* computed by the motion detector. The output of the frame to frame tracker is a graph $\Gamma(t) = (X, U)$, where X is a set of nodes containing the detected *moving regions* and U is a set of links between two *moving regions* at successive times. This graph provides all the possible trajectories that a *moving region* may have. The link between a new *moving region* M_{new} and an old one M_{old} is computed depending of three criteria: their 2D (in the image) and 3D (in the real world) distance and the similitude between their bounding boxe size. If the criteria are met, we note: $link(M_{old}; M_{new}) = 1$ else $link(M_{old}; M_{new}) = 0$. As there could be many errors of detection, we use non-strict criteria to allow a large number of links. Moreover, an old *moving region* could be linked to several new ones (split case) or several old *moving regions* could be linked to the same new one (merge case). We define the set of old *moving regions* $\mathcal{O}$ and new ones $\mathcal{N}$ as following:

$$\mathcal{O} = \{M_i(t_c - q) \mid 1 \le i \le nb\_old, 1 \le q \le 3\}$$

$\mathcal{O}$ contains old *moving regions*, all those detected at $t_c - 1$ and also those that did not get linked (disappear, miss detection) at the previous q frames.

$$\mathcal{N} = \{M_j(t_c) \mid 1 \le j \le nb\_new\}$$

$\mathcal{N}$ contains new *moving regions* detected in current frame processed at time t_c.

We define a function F (respectively G) to compute the links between $\mathcal{O}$ and $\mathcal{N}$ (respectively between $\mathcal{N}$ and $\mathcal{O}$).

$$F : \mathcal{O} \longrightarrow \mathcal{P}(\mathcal{N})$$

$$F(M_i(t_c - q)) = \{M_j(t_c)_{,1 \le j \le nb\_new} \mid link(M_i(t_c - q); M_j(t_c)) = 1\}$$

$\mathcal{P}(\mathcal{N})$ is defined as the set containing all the possible sub-sets of $\mathcal{N}$

$$G : \mathcal{N} \longrightarrow \mathcal{P}(\mathcal{O})$$

$$G(M_j(t_c)) = \{M_i(t_c - q)_{,\substack{1 \le i \le nb\_old \\ 1 \le q \le 3}} \mid link(M_i(t_c - q); M_j(t_c)) = 1\}$$

$\mathcal{P}(\mathcal{O})$ is defined as the set containing all the possible sub-sets of $\mathcal{O}$

3.3. Computing Paths

The purpose of this task is to select trajectories of *moving regions* (called *paths*) that can correspond to real persons inside a group during a temporal window. A *path* $P_k^{t_c}$ represents a temporal link in the graph $\Gamma(t)$ between *moving regions* during a temporal interval $[t_c - T; t_c]$, where t_c is the time of the last (current) processed image and T is the size of the temporal window used to analyse *groups*. $P_k^{t_c}$ is composed of a temporal sequence of *moving regions* $M_k(t), t \in [t_c - T; t_c]$.

A size coefficient $Size(P_k^{t_c})$ qualifies a path $P_k^{t_c}$:

$$Size(P_k^{t_c}) = \frac{\sum_{t \in [t_c - T; t_c]} Size_{3D}(M_k(t))}{T}$$

With $Size_{3D} = Height_{3D}.Width_{3D}$ of the bounding box of the *moving region*.

This coefficient characterises a *path*: if its size coefficient is bigger than the size of a person, then the *path* is likely to correspond to a real person inside a group. So this coefficient allows us to rank *paths*. This allows to distinguish the *paths* containing big *moving regions* (labelled as GROUP, PERSON) from the *paths* containing smaller ones (labelled as NOISE) and so less interesting.

3.3.1. Creation of Paths

The most common case to create a *path* is when a *moving region* has been classified as a GROUP by the motion detector. In this case, this *moving region* is likely to represent several persons inside a real group. In order not to miss any person inside a real group, we also decide to create a *path* each time a trajectory (in the graph $\Gamma(t)$) is crossing an other one. The common *moving region* of both trajectories can represent several persons inside a real group if the *moving region* is at least as big as a person.

More precisely, for each new processed image (at time t_c), *paths* are created if one of these three predicates is true:

- ***Moving region* labelled as a GROUP:**
  if a new *moving region* is classified as a GROUP a new *path* is created. More precisely:
  $\exists M_j(t_c),\ Class(M_j(t_c)) = GROUP,$
  $\nexists P_{old}^{t_c - 1} \mid M_j(t_c) \in P_{old}^{t_c - 1},$
  $\Rightarrow$ 1 *path* $P_{new}^{t_c}$ is created: $P_{new}^{t_c} = (M_j(t_c))$
- **Splitted *moving region*:**
  if an old *moving region* is linked to E new ones,
  E new *paths* are created. More precisely:
  $\exists M_i(t_c - q) \mid Size_{3D}(M_i(t_c - q)) \ge Size\_person,$
  $E = card\ F(M_i(t_c - q)) \ge 2,$
  $\nexists P_{old}^{t_c - 1} \mid M_i(t_c - q) \in P_{old}^{t_c - 1}$
  $\Rightarrow$ E *paths* $P_{new_l}^{t_c}, _{1 \le l \le E}$ are created:
  $\forall\, l_{,1 \le l \le E}\,, P_{new_l}^{t_c} = (M_i(t_c - q), M_j(t_c)),$
  $M_j(t_c) \in F(M_i(t_c - q))$
- **Merged *moving region*:**
  if a new *moving region* is linked to several old ones, a new *path* is created. More precisely:
  $\exists M_j(t_c) \mid Size_{3D}(M_j(t_c)) \ge Size\_person,$
  H= $card\ G(M_j(t_c)) \ge 2,$
  $\nexists P_{old}^{t_c - 1} \mid M_j(t_c) \in P_{old}^{t_c - 1}$
  $\Rightarrow$ 1 *path* $P_{new}^{t_c}$ is created: $P_{new}^{t_c} = (M_j(t_c))$

3.3.2. Update of Paths

For each new image processed at time t_c, each old *path* $P_k^{t_c-1}$ is updated in $P_k^{t_c}$. If M_{last} is the last *moving region* added in $P_k^{t_c-1}$ and is linked to the *moving region* M_{new} detected in the new frame, $P_k^{t_c-1}$ is duplicated in $P_k^{t_c}$ and extended with M_{new}. If M_{last} is not linked to any new *moving region*, the *path* $P_k^{t_c-1}$ is only duplicated. As a result, the size coefficient of a such *path* decreases.

A special case appears when several old *paths* that have the same *moving region* M_{last} could be extended with several new ones (M_{last} is linked to several $M_{new_j}, {}_{j\geq 2}$). To limit the number of possible extensions that could increase significantly because of the large number of links given by the frame to frame tracker, we only keep extensions of the most interesting *paths* (*path* with the biggest size coefficient). The better old *path* is extended with all the new *moving regions* $M_{new_j}, {}_{j\geq 2}$ whereas the other old ones are only extended with the new *moving region* that has the biggest $Size_{3D}$.

3.3.3. Removing Paths

A *path* P_i is removed if one of these two cases is verified:

- P_i is totally overlapping another *path* P_j and the size of P_j is bigger than the size of P_i.
- P_i does not belong anymore to a *group* (see below)

3.4. Group Structure

The *group* structure is a set of *paths* belonging to a connected sub-graph of $\Gamma(t)$ that corresponds to a real group of people. The problem is that the number of *paths* (trajectories that can correspond to persons inside groups) can be important and that some *paths* can correspond to detection errors. So the goal of the *groups* computing step is to select the *paths* of the connected sub-graph that best match with the trajectories of real persons. A *group* G_m is represented by its N *paths* $P_{m,k}, {}_{1\leq k\leq N}$.

If t_c is the time of the last (current) processed image, *groups* are computed at time $t_c - T$ with a delay T. This delay is used to obtain a stable *group* structure. The **delay** T constitutes a temporal window $[t_c - T; t_c]$. In this window, we first compute all possible future trajectories of *moving regions* detected at time $t_c - T$. This analysis allows us to select at time $t_c - T$ the *moving regions* that best match a real group that would be observed from time $t_c - T$ to t_c.

We characterise a *group* G_m by a quality coefficient Q_{G_m}:

$$Q_{G_m}(t) = \alpha Q_{G_m}(t-1) + (1-\alpha)Q'_{G_m}(t) \quad t > 0$$

$$Q_{G_m}(0) = Q'_{G_m}(0)$$

$$Q'_{G_m}(t) = \sum_{k=1}^{N_{paths}} Prox(P_{m,best}; P_{m,k}).Size(P_k^{t+T})$$

$Q'_{G_m}(t)$ characterises the new set of *paths* selected at time t and is called instantaneous quality coefficient.

α is an update coefficient $\in [0;1]$ to balance the previous quality coefficient against the instantaneous quality coefficient. α allows to obtain a stable quality coefficient over time.

$P_{m,best}$ is the *path* with the biggest size coefficient.

N_{paths} is the number of *paths* selected in the *group*,

We use the best *path* $P_{m,best}$ trajectory to give the notion of direction to the *group* G_m. Usually a *group* is formed by a dense part and a less compact part around. A dense part can be for example several persons overlapping each other and detected as only one *moving region* bigger than the other *moving regions* around. As we work in a temporal window, we approximate the direction of the *group* G_m by the direction of its biggest *path* $P_{m,best}$.

$Prox(P_{m,best}; P_{m,k})$ computes the proximity between $P_{m,best}$ and $P_{m,k}$ and is defined as following:

$$Prox(P_{m,best}; P_{m,k}) = e^{-dist(P_{m,best}; P_{m,k})}$$

This proximity is equal to 1 when both *paths* are connected (own a common *moving regions*) and decreases exponentially to zero when the distance $dist(P_{m,best}; P_{m,k})$ increases.

The distance $dist(P_{m,best}; P_{m,k})$ is defined as the minimum $3D$ distance between their respective *moving regions* detected at the same time:

$$dist(P_{m,best}; P_{m,k}) = \min_{t\in[t_c-T;t_c]}(3D\_distance(M_{P_k}(t); M_{P_{best}}(t)))$$

This proximity allows us to compute $Q'_{G_m}(t)$ by giving more weight to the paths which are close to the best path. By this way, $Q_{G_m}(t)$ characterises the density of the group over the time.

3.4.1. Creation of Groups

As we mentioned above, we wait a delay T between creating a *group* at time $t_c - T$ and detecting the corresponding *moving regions* at time t_c. The creation of a *group* G_m consists in selecting the *paths* (that do not yet belong to a *group*) with the biggest size coefficient among a tree of the graph $\Gamma(t)$. In this tree, the root is a *moving region* detected at time $t_c - T$ and corresponds to the appearence of the group in the scene. To limit memory space and processing time, we just keep the N_{max} *paths* with biggest size coefficient. To avoid the creation of *groups* that do not correspond to real groups of people in the scene, we check before the creation of the *group* if the quality coefficient is higher than a threshold. This threshold corresponds to the smallest size that a real group could have.

3.4.2. Update of Groups

During the update step of a *group* G_m at time $t_c - T$, the main task consists in adding, extending or removing the *paths* composing the *group*. The goal is to obtain the most dense (with the biggest $Q'_{G_m}(t)$) set of *paths* contained in the same connected sub-graph (connected before the time $t_c - T$) of the graph $\Gamma(t)$. We propose to approximate this goal by first removing all the *paths* $P_{m,i}$ too far from the best *path* $P_{m,best}$ ($dist(P_{m,best}; P_{m,i}) > threshold$) and second to select the remaining *paths* with the best size coefficients (e.g. on figure 1, *path 1* belongs to G_1 at time $t_c - T - 1$ but is not selected in G_1 at time $t_c - T$). Finally, the new best path $P_{m,best}$ is reselected and the quality coefficient recomputed.

There is a special case when several *groups* **merge**. Two *groups* G_r and G_s merge together at time $t_c - T$ if:
$\exists$ path $P_{r,i}^{t_c}, \in G_r$,
$\exists$ path $P_{s,j}^{t_c} \in G_s \mid M_{P_{r,i}}(t_c - T) = M_{P_{s,j}}(t_c - T)$

In this case, we first collect the *paths* from all the merged *groups* and select the best *path* with the biggest size coefficient (among the different best *paths*). And then we continue to update the merged *groups* as in the general case.

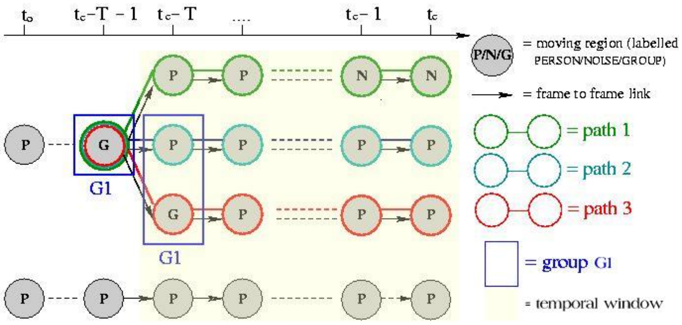

Fig. 1. The 3 *paths*: $path1$, $path2$, $path3$ are updated at time t_c whereas *group* G_1 is updated with a delay T

3.4.3. Removing groups

A *group* is removed if the quality coefficient (corresponding to the *group* density) is lower than a threshold (the same one used in the creation step). For instance, when persons inside a group split far away from each other, the quality coefficient of the *group* becomes too small. When a *group* is removed, all its *paths* are also removed.

4. RESULTS

This tracking module has been tested on several metro sequences. The longest sequence lasts 5 minutes (more than 6500 frames). In this section we are showing different image samples of the processed videos. In these images, a red box corresponds to a *moving region* classified as a PERSON, a green box corresponds to a *moving region* classified as a GROUP and a blue box corresponds to *moving regions* tracked globally as a group. The three images on figure 2 focus on the creation of a *group* in two situations : the first occurs when a *moving region* is labelled as a GROUP (left images) and the second situation occurs when an old *moving region* splits into two new *moving regions* (both right images). The three images on figure 3 illustrate the use of the temporal window to help the tracker to keep coherent the structure of a *group* during the tracking: a person first goes away from another one then comes back close to the second person. The next two images of figure 4 demonstrate that the algorithm is able to track a *group* during a long period of time. In this sequence, the group is tracked during more than 800 frames. Then, figure 5 shows an example of a *group* being removed when the persons inside the *group* split far away from each other. Currently, the main limitation of the system is an imperfect estimation of real group size due to errors in the motion detector. The system over estimates a group of persons when there are shadows or reflections strongly contrasted. The system under estimates a group of persons when the persons are occluded, overlapping each others or in case of miss detection (person has the same colour than the background). However in most cases, these imperfect estimations do not induce errors in the tracker thanks to the temporal delay: by default the tracker creates *paths* than can correspond to person trajectories in a group over a temporal window and only create the *group* (containing a selection of these *paths*) with a quality coefficient good enough.

5. CONCLUSIONS

We proposed in this paper an algorithm to track real groups of people in a metro scene. The originality of the apporach consists in tracking globally all the persons belonging to the same group and in using a delay to keep coherent the structure of this group. The algorithm allows to track efficiently groups of people in several video sequences, compensating detection errors. In metro scene, the persons inside a group cannot be tracked individually because they cannot be segmented when they cross each other. However our algorithm can track correctly groups of people from beginning to end. This is essential for the interpretation module to be able to recognise the behaviours of groups. Currently, we are planning to extend the model of group by computing relevant informations in order to keep track of groups of people in special cases. For example, if two persons in a group overlap each other for a long period of time, the dimension of the group becomes lower than the size that a real group should have and this group will not be tracked anymore. In this case, the knowledge of the number of persons inside the group could help the system to not lose this group. Future developments include computation of group trajectory, speed and events inside the group (for example: two persons fighting) in order to recognise abnormal behaviours such as violence or vandalism in a metro scene

6. REFERENCES

[1] M. Thonnat and N. Rota, “Image understanding for visual surveillance application,” in *Third international workshop on cooperative distributed vision CDV-WS’99*, Kyoto, Japan, Nov. 1999, pp. 51–82.

[2] F. Brémond and M. Thonnat, “Tracking multiple non-rigid objects in a cluttered scene,” in *proc. of the 10th Scandinavian Conference on Image Analysis (SCIA)*, Lappeenranta (Finland), June 1997.

[3] I. Cox and S. Hingorani, “An efficient implementation of reid’s Multiple Hypothesis Tracking algorithm and its evaluation for the purpose of visual tracking,” in *IEEE Transactions on pattern analysis and machine intelligence*, Sept. 1996, vol. 18.

[4] F. Meyer and P. Bouthemy, “Region-based tracking in an image sequence,” in *Proc. of European Conference on Computer Vision (ECCV)*, May 1992, pp. 476–484.

[5] Z. Zhang, “Token tracking in a cluttered scene,” in *Int’l J. of Image and Vision Computing*, Mar. 1994, vol. 12.

[6] N. Rota, R. Stahr, and M. Thonnat, “Tracking for visual surveillance in VSIS,” in *First IEEE International Workshop on Performance Evaluation of Tracking and Surveillance PETS2000*, Grenoble, France, Mar. 2000.

(a) (b1) (b2)

Fig. 2. Creation of a *group*: one *moving region* is detected and labelled as a GROUP in the top right on the left image (a), thus after checking the T following frames, a *group* is created. On the image (b1), two persons in close contact are detected as one *moving region* (miss labelled as a PERSON), then on the image (b2), these two persons are detected as two *moving regions*. Because the old *moving region* is splitting into two new ones and after checking the T following frames, a *group* is created even if no *moving region* is classified as a GROUP.

(a1) (a2) (a3)

Fig. 3. The three images show two persons inside a group (a1) where a first person goes away from the other person (a2) then comes back close to the second (a3). Even when they are far away from each other, the group containing both persons is not removed (a2). This illustrates the use of the temporal window to keep coherent the structure of the *group*.

(a1) A *group* is tracked at frame 661 (a2) The *group* continues to be tracked at frame 1303

Fig. 4. The algorithm of *group* tracking is able to track real groups of persons during a long period of time. In this case the group is tracked during more than 800 frames (32s).

(a1) Several *moving regions* compose a *group* (a2) These *moving regions* are not in a *group* anymore

Fig. 5. On the image (a1), several *moving regions* (close to each others) are composing a *group*. Then on the image (a2), as the persons split far away from each others, the density of the *group* becomes too low and the *group* is removed.