Graph Mining for Object Tracking in Videos

Fabien Diot[1,2], Elisa Fromont[1], Baptiste Jeudy[1]
Emmanuel Marilly[2], Olivier Martinot[2]

[1] Université de Lyon, Université Jean Monnet de Saint-Etienne
Laboratoire Hubert Curien, UMR CNRS 5516, 42000, Saint-Etienne, France
[2] Alcatel-Lucent Bell Labs, Centre de Villarceaux,
Route de Villejust, 91620, Nozay, France

Abstract. This paper shows a concrete example of the use of graph mining for tracking objects in videos with moving cameras and without any contextual information on the objects to track. To make the mining algorithm efficient, we benefit from a video representation based on dynamic (evolving through time) planar graphs. We then define a number of constraints to efficiently find our so-called spatio-temporal graph patterns. Those patterns are linked through an occurrences graph to allow us to tackle occlusion or graph features instability problems in the video. Experiments on synthetic and real videos show that our method is effective and allows us to find relevant patterns for our tracking application.

1 Introduction and Related Work

Object tracking in videos is a very popular research field in computer vision due to the numerous applications such as video-surveillance in very diverse environments (airports, cities, large public areas), pedestrian protection systems, automatic calibration methods using moving robots, tracking complicated surfaces, medical image applications etc. [11]. Most of the ongoing research [1, 11] makes strong assumptions about the objects to track (people, car, etc.) which are often modelled in advance, or about the tracking context (stable background, object moving in a single direction, stable lighting conditions, etc.) to perform an efficient tracking. These methods rely on two steps, the object detection in the frame and the tracking process. For detection, techniques are based on frame difference or the use of background subtraction [12], optical flow (detection of the relative motion between a static camera and the filmed objects) [15] or background information on the objects to track (skin color, shape etc.). For the tracking process, techniques consist in predicting the next region (or contour) of interest using probabilistic or deterministic methods [7] (and then possibly add another detection step). They use some discriminant features attached to the objects and/or use apriori learned models of the objects which can possibly be updated during the tracking step [13].

In this work, we would like to show how data mining and in particular graph mining can help to track multiple objects in a video in the specific case in which both the objects and the background are moving and when no supervised

information about the objects to track is known in advance (which could allow to learn some models apriori). We regard a video as a dynamic graph, whose evolution over time is represented by a series of *plane* graphs, one graph for each video frame. The graph representing each frame is a region adjacency graph (RAG) [6]. In RAGs, the barycenters of the different regions in a frame are the nodes of the graph, and an edge exists between two nodes if the regions are adjacent in the frame. By representing a video as a series of plane labelled graphs, subgraph patterns in this series may correspond to objects that frequently appear in a video, such as the planes in the frames of Fig. 4.1 and 4.1.

This paper is based on [14] where we have already assessed the interest of our plane graph mining algorithm called PLAGRAM compared to a generic graph mining algorithm such as GSPAN [16] on which it is based. PLAGRAM can efficiently mine a dynamic graph representing a video (i.e. a plane graphs database). Note that most existing algorithms which mine dynamic graphs (e.g., dynamic networks) consider graphs with only edges insertions or deletions i.e., the time series of graphs share the same set of nodes over time (see, e.g.,[3]), or in which nodes and edges are only added and never deleted (see, e.g., [2]). In [5, 17], the problem is to mine spatio-temporal relationships between moving objects (the mined relationships are restricted to some predefined graphs like cliques, star graphs or sequences). In our approach, however, there is no information about the correspondence between the nodes in one graph (video frame) and those in the others. In [14], some simple constraints were used in a post processing step to obtain some so-called spatio-temporal patterns. However, the definition of spatio-temporal patterns (and especially, of the distance) was not anti-monotonic which prevented the computation of spatio-temporal patterns during the mining step. Moreover, the spatio-temporal patterns obtained were quite small in practice which led to a low recall when using them for object tracking. In this article, we present an extended version of the plane graph mining algorithm called DYPLAGRAM_ST which can benefit from the spatio-temporal constraints to directly and thus more efficiently mine the spatio-temporal patterns. Besides, we propose a method based on a global occurrences graph to combine these patterns in order to build spatio-temporal paths that can be used to follow some objects in the videos. By allowing a pattern to change along a path, it is possible to take into account instability in the video or change of view point which improves the recall of the patterns.

The tracking methods presented at the beginning of this introduction typically do not consider moving objects in changing environments. When it is the case as in [4], multiple cameras are used to tackle object occlusions or features instability using stereo vision. The setting taken in [8] is close to the one we are interested in since they consider cameras embedded in surveillance cars but they rely on strong background information (here GPS position) to perform an effective tracking. Our method is also similar to [18] but they do not use the topological information provided by the subgraph patterns and they use a spatio-temporal Markov Chain Monte Carlo algorithm to sample the possible paths represented in our occurrences graph.

The outline of this paper is the following. In Section 2, we recall some important definitions and explain the proposed extensions to the DyPlagram algorithm proposed in [14]. In Section 3 we show how to compute the spatio-temporal paths used for object tracking. Section 4 shows a large set of experiments on a synthetic and on a real video to assess both the efficiency of our new algorithm DyPlagram_st but also the usefulness of the spatio-temporal paths to tackle the problem of object tracking in videos. We conclude in Section 5.

2 Spatio-Temporal Patterns Mining

2.1 Dynamic Plane Graphs

The definitions in this section are similar to those of [14]. As in [14], our algorithm mines 2-connected plane graphs that satisfy various spatio-temporal constraints. The restriction to 2-connected plane graphs was motivated by the use of plane graphs in our video data and because it allows to test subgraph isomorphism in polynomial time. Moreover, this restriction also dramatically decreases the branching factor of the search space and improves the efficiency (as already shown in [14]).

Definition 1 (Plane graph). *A plane graph is $G = (V, E, F, f_e, L)$ where V is a set of nodes, E is a set of edges, F is a set of faces and L is a labeling function on $V \cup E$. Exactly one of the faces $f_e \in F$ is called the external face, the other faces are the internal faces. The graph is 2-connected if each face is a simple cycle (the face does not use a node or an edge more than once).*

Our aim is to find 2-connected plane subgraphs which satisfy some constraints in a database of graphs.

Definition 2 (Plane subgraph isomorphism, occurrence). *Given two plane graphs $G = (V, E, F, f_e, L)$ and $G' = (V', E', F', f'_e, L')$, G' is a* plane subgraph *of G if there is an injective function f from V to V' which preserves the edges, the internal faces of G' and the labels. The function f is called an* occurrence *of G' in G.*

The frames in a video are ordered, and this order is taken into account when computing spatio-temporal patterns. We thus define a dynamic graph as an ordered set of graphs.

Definition 3 (Dynamic plane graph). *A dynamic plane graph $\mathcal{D}$ is an ordered set of plane graphs $\{G_1, G_2, .., G_n\}$. Each node of these graphs is associated to spatial coordinates (x, y).*

Example 1. In our video application, each plane graph G_i represents a video frame. Each node in a graph represents a segmented frame region, and is associated to the coordinates (x, y) of the barycenter of this region. The labels on nodes are built either by a discretization of the size of the regions or of the color.

We define an occurrence of a plane graph in a dynamic plane graph and its frequency.

Definition 4 (Occurrences of a plane graph in a dynamic graph). *Given a plane graph P and a dynamic graph $\mathcal{D} = \{G_1, ..., G_n\}$, the set of occurrences of P in $\mathcal{D}$ is defined as $Occ(P) = \{(i, f) \mid f$ is an occurrence of P in $G_i\}$.*

Definition 5 (Frequency of a plane graph in a dynamic graph). *The frequency $freq(P)$ of a plane graph P in a dynamic graph $\mathcal{D}$ is the number of graphs $G_i \in \mathcal{D}$ in which there is an occurrence of P, i.e., $|\{i \mid \exists f,\ (i, f) \in Occ(P)\}|$.*

2.2 Occurrences Graph and Spatio-Temporal Patterns

In typical subgraph mining problems, where the input collection of graphs does not represent a dynamic graph, the frequency $\text{freq}(P)$ of a pattern graph P is computed regardless of the fact that its occurrences may be far apart w.r.t. time and/or space. To define a frequency that takes into account spatio-temporal distance between the occurrences, we define in this section the notion of an occurrences graph in which occurrences of the same pattern that are close to one another are linked. Then, we define spatio-temporal patterns in this occurrences graph and the associated frequency (called freq_{st}).

The definitions in this section, although similar to the one of [14], have been changed to integrate the spatio-temporal patterns computation during the mining step instead of during a post-processing step. This offers more pruning opportunities.

Definition 6 (Distance between occurrences). *The distance between two occurrences $o = (i, f)$ and $o' = (i', f')$ of a plane graph $P = (V, E, F, f_e, L)$ in a dynamic graph $\mathcal{D}$ is defined as: $dist(o, o') = \max_{s \in V} d(f(s), f'(s))$, where d denote the Euclidean distance between the nodes.*

This distance has an anti-monotonic property:

Proposition 1. *For any patterns $P = (V, E, F, f_e, L)$ and $P' = (V', E', F', f'_e, L')$ such that P is a plane subgraph of P' and two occurrences $o_1 = (f_1, i)$, $o_2 = (f_2, i)$ of P and two occurrences $o'_1 = (f'_1, i)$, $o'_2 = (f'_2, i)$ of P' such that f_1 is a restriction of f'_1 (i.e., $f_1 = f'_1$ on V) and f'_2 is a restriction of f_2, then we have $dist(o_1, o_2) \leq dist(o'_1, o'_2)$.*

proof (sketch): the set from which the maximum is taken for $\text{dist}(o_1, o_2)$ is included in the set for which the maximum is taken for $\text{dist}(o'_1, o'_2)$.

The depth first traversal of the search space by our mining algorithm define a parent relationship on patterns:

Definition 7 (Parent of a pattern and of an occurrence). *Given a pattern P with $n \geq 2$ internal faces, the pattern $p(P)$ with $n-1$ faces from which P was built is called the parent of P. And given an occurrence $o = (f, i)$ of P, we call the* parent of o *the occurrence $p(o) = (f', i)$ such that f' is the restriction of f to the nodes of $p(P)$.*

The definition of the parent of an occurrence is then used to define the occurrences graph. The nodes of the occurrences graph are the occurrences of a pattern and the edges connect "close" occurrences. This graph is constructed for each pattern in the mining algorithm.

Definition 8 (Occurrences graph and Spatio-temporal pattern). *Given a spatial threshold ϵ, a temporal threshold τ, a plane graph $P = (V, E, F, f_e, L)$ and a dynamic graph $\mathcal{D}$, we define the* occurrences graph of P *as an oriented graph whose set of nodes is $Occ(P)$.*

- *If P has only one face, then there is an edge from (f, i) to (g, j) if $0 < j - i \leq \tau$ and $dist(f, g) \leq \epsilon.(j - i)$ and there is no (h, k) with $i < k < j$ and $dist(f, h) \leq \epsilon.(k - i)$.*
- *If P has more than one face, then there is an edge from $o = (f, i)$ to $o' = (g, j)$ if there is an edge $(p(o), p(o'))$ in the occurrences graph of $p(P)$ and $dist(f, g) \leq \epsilon.(j - i)$.*

A spatio-temporal pattern S based on P *is a connected component of the occurrences graph of P.*

This definition is such that the occurrences graph of a pattern P is always a subgraph of the occurrences graph of its parent pattern $p(P)$ (if we identify the node o of the occurrences graph of P with the node $p(o)$ of the occurrences graph of $p(P)$). This ensures that the spatio-temporal patterns based on P get "smaller" as the pattern P grows, and this ensures that the frequency of a spatio-temporal pattern defined below has the anti-monotonicity property.

Definition 9 (Frequency of a spatio-temporal pattern). *The frequency of a spatio-temporal pattern S based on a graph pattern P in a dynamic graph $\mathcal{D}$ is $freq_{st}(S) = |\{i \mid \exists f,\ (i, f) \in S\}|$.*

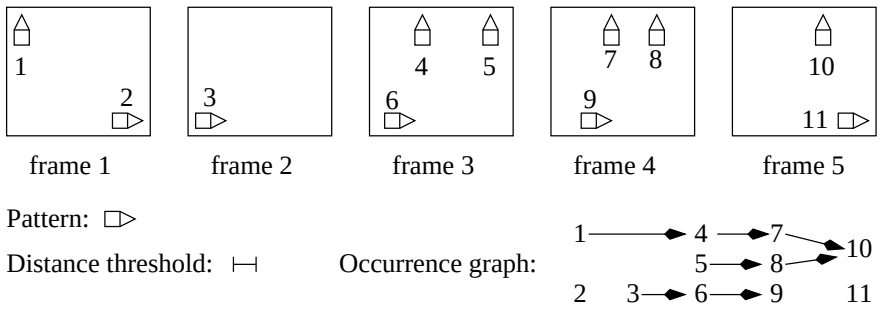

Fig. 1. Occurrences of a pattern and occurrences graph of this pattern (temporal threshold $\tau = 2$ and distance threshold ϵ).

Example 2. Fig. 1 shows 11 occurrences of a pattern P in a video with five frames. $\text{freq}(P) = 5$. Since occurrences 1 and 4 are close to each other, i.e., their spatial distance is lower than 2ϵ and their temporal distance is $2 \leq \tau$, there is an edge $(1, 4)$ in the occurrences graph of P. Conversely, the edges $(3, 5)$ or $(2, 11)$ do not exist in the occurrences graph, as the spatial distance between 3 and 5 or the temporal distance between 2 and 11 are too large. There are 4 spatio-temporal patterns $S_1 = \{1, 4, 5, 7, 8, 10\}$, $S_2 = \{3, 6, 9\}$, $S_3 = \{2\}$ and $S_4 = \{11\}$. The frequencies of these patterns are: $\text{freq}_{st}(S_1) = 4$, $\text{freq}_{st}(S_2) = 3$, and $\text{freq}_{st}(S_3) = \text{freq}_{st}(S_4) = 1$.

Proposition 2. *Given a pattern P with more than one face, and given a spatio-temporal pattern S based on P then there is a spatio-temporal pattern S' based on the parent $p(P)$ of P with a larger freq_{st}, i.e., $\text{freq}_{st}(S) \leq \text{freq}_{st}(S')$.*

This proposition shows that, given a minimum threshold minfreq_{st} on freq_{st}, if a pattern does not have a frequent spatio-temporal pattern then any super-pattern does not either. This allows to prune the search space.

2.3 DyPlagram_st Algorithm

Given a frequency threshold minfreq (also called minimum support), a minimum threshold minfreq_{st} for freq_{st} a spatial threshold ϵ and a temporal threshold τ, the proposed algorithm DyPlagram_st computes all spatio-temporal patterns with $\text{freq}_{st} \geq \text{minfreq}_{st}$ based on patterns with $\text{freq} \geq \text{minfreq}$ (the thresholds ϵ and τ are used in the construction of the occurrences graph, see Def. 8).

The proposed algorithm DyPlagram_st is based on DyPlagram [14] which itself is based on gSpan. Its main characteristics are :

- a recursive depth first exploration of the search space;
- the use of canonical codes to avoid considering the same graph several times;
- at each level, patterns are extended by adding a whole face to the current pattern.

The new definition of the freq_{st} is now anti-monotonic, and we can use it in the DyPlagram_st algorithm. However, this frequency is not defined on patterns but on spatio-temporal patterns. We must therefore also build the occurrences graph and the spatio-temporal patterns in the algorithm.

Given an occurrence $o = (f, i)$ of a pattern P, an extension E of P is a set of edges such that $P \cup E$ has exactly one more face than P and there is an occurrence $o' = (f', i)$ of $P \cup E$ that extends o, i.e., such that f is the restriction of f' to P.

As its predecessors, DyPlagram_st uses canonical codes to represents patterns and extensions. This allows to efficiently enumerate only the so called valid extensions of a pattern. Informally, a valid extension of a pattern is an extension that lead to a pattern not already considered by the algorithm. This is a very efficient way to avoid considering several times the same pattern. We do not detail here how these codes are built, the interested reader can refer to [14].

The DyPlagram_st algorithm first builds all frequent one face patterns and then calls the following recursive function `mine` for all of them.

mine$(P, \text{minfreq}, \text{minfreq}_{st}, \tau, \epsilon, \mathcal{D})$

```
1   occurrences_graph(P) = empty_graph
2   for each occurrence of P in D do
3       Add this occurrence to occurrences_graph(P)
4       Computes all valid extensions of this occurrence
5   Computes the edges of occurrences_graph(P) (using ε and τ)
6   Computes all spatio-temporal patterns based on P
7   for each spatio-temporal pattern S based on P do
8       if freq_st(S) ≥ minfreq_st then output(S)
9   if there is no frequent spatio-temporal pattern then return
10  else
11      for each extension E of P do
12          if the code of E ∪ P is canonical and freq(E ∪ P) ≥ minfreq then
13              mine(P ∪ E, minfreq, minfreq_st, τ, ε, D)
14      return
```

In this algorithm, lines 1, 3, 5, 6, 7, 8, and 9 were not in DyPlagram [14].

Thanks to Prop. 2, this algorithm is correct and output exactly the spatio-temporal patterns whose freq_{st} is above the user defined threshold σ.

3 Spatio Temporal Path

When tracking an object in a real video, we cannot expect that the object is represented by the same graph pattern during the whole video (e.g., due to changes in view point or instability of the segmentation). Thus, if we want to track it using spatio-temporal patterns, we propose to build a path in the union of all occurrences graphs. To allow this path to "jump" from a spatio-temporal pattern to another, similarity edges are added between overlapping occurrences of different patterns. Weights are also added on the edges so that minimum weight paths can then be computed in this global occurrences graph.

Definition 10 (Similarity of two occurrences). *Let $o = (i, f)$ and $o' = (i, f')$ be two occurrences of two different patterns $P = (V, E, F, f_e, L)$ and $P' = (V', E', F', f'_e, L')$. The similarity between these occurrences is defined as $\sigma(o, o') = \frac{|f(V) \cap f'(V')|}{|f(V)|}$.*

This similarity is not symmetric and it is used to weight the edges in the global occurrences graph.

Definition 11 (Global occurrences graph). *Given a set of patterns $\mathcal{P}$, temporal and spatial thresholds τ and ε, a similarity threshold σ, the global occurrences graph is a weighted oriented graph: its node set is $V = \cup_{P \in \mathcal{P}} Occ(P)$ and its edge set is $E = E_{\mathcal{P}} \cup E_{sim}$ where :*

- *$E_{\mathcal{P}}$ is the union of the edge sets of all patterns occurrences graphs. The weight of an edge $((i,f),(i',f'))$ is $w = \frac{(i'-i-1)}{\tau}$.*
- *$E_{sim} = \{(o,o',w)|\ o=(i,f), o'=(i,f'), \sigma(o,o') < \sigma\}$ is the set of similarity edges with*

$$w = \begin{cases} 0 & \text{if } |V| < |V'| \\ \frac{1}{2}(\frac{1-\sigma(o,o')}{1-\sigma} + \frac{d}{\epsilon}) & \text{otherwise.} \end{cases}$$

where V and V' are the node sets of the patterns corresponding resp. to occurrences o and o', and d is the distance between the barycenters of o and o'.

A spatio-temporal path *is a path in the global occurrences graph.*

The edges in $E_{\mathcal{P}}$ are edges between 2 occurrences of the same pattern that are not in the same frame. If these two occurrences are in consecutive frame, the weight is 0 (when $i' = i+1$) otherwise the weight increases with the number of frames between them (normalized by the temporal threshold τ).

The edges in E_{sim} are *similarity edges* between 2 occurrences of different patterns that are in the same frame and whose similarity is below σ. We want to favor paths that use large patterns, thus the weight of an edge from an occurrence of a small pattern to a larger one is 0. The weight of an edge from an occurrence of a large pattern to a smaller one increases as the similarity decreases and the spatial distance increases.

4 Experiments

Some experiments in [14] have already assessed the efficiency of the plane graph mining algorithm called DyPlagram compared to a generic graph mining algorithm such as gSpan [16]. The introduction of this plane graph mining algorithm was necessary to effectively mine the graphs extracted from videos. Experiments on a very simple video (one object, moving background, no occlusion, no disappearance) showed promising results for object tracking but the interest of the spatio-temporal patterns for more complex videos was not thoroughly evaluated. Besides, in [14], we did not present an effective way to use spatio-temporal constraints in DyPlagram nor a systematic method to combine the spatio-temporal patterns into spatio-temporal paths to track object in videos. Our proposed experiments aim to answer three main questions:

1. Are the spatio-temporal constraints well exploited by the new DyPlagram_st algorithm compared to the process presented in [14]?
2. How are the results of the DyPlagram_st algorithm where the spatial and the temporal constraints are pushed directly into the mining process compared to the post-processing experiments described in [14]?
3. How meaningful (in terms of precision and recall) are the spatio-temporal paths to track objects in a synthetic and in a real video?

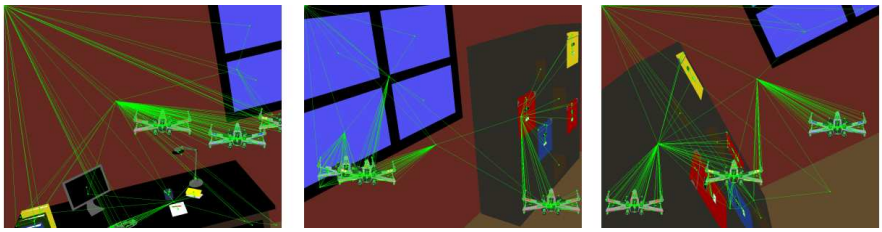

Fig. 2. Example of RAGs obtained from the synthetic video

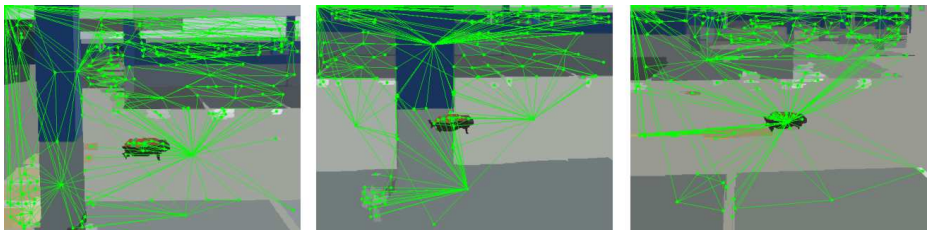

Fig. 3. Example of RAGs obtained from the real video

4.1 Video Datasets

We used 2 datasets for these experiments. One was created from a synthetic video which allows us to avoid the possible segmentation problems. The second comes from a real (but simple) video with its possible segmentation issues.

For both videos, we used two possible labels on the nodes of the RAGs. The first possible one comes from a *discretization of the size* of the segmented regions (in pixels). The discretization uses 10 bins of equal size that were computed using all the possible region sizes (sorted for the discretization) for a given video. The second is a *color discretization* of the mean color of the segmented regions. We divided each of the 3 RGB channels in 3 parts, resulting in 27 bins of equal range.

The synthetic video has 721 frames in total. In average the RAGs are composed of 240.7 nodes with an average degree of 3.9. Three identical objects (X-wings) are moving in the video such that they may overlap or even get (partially) out of the field of view (this helped us to evaluate how well spatio-temporal patterns can be used to represent the trajectory of the X-wings individually). The 3 X-wings have different colors but this feature is not always used in the experiments. Fig. 4.1 show three examples of RAGs we obtained for this dataset.

The real video is composed of 950 frames (25 frames per second), each RAG has on average 194.5 nodes with an average degree of 5.35. This video shows a drone flying across a covered parking lot. Before building the RAGs, we segmented each frame of the video independently using the algorithm presented in

[10] and available on the web[3]. This algorithm has 3 parameters for which we used standard values. This algorithm helps the merging of small regions which may result in an unstable segmentation when objects are getting close to or moving away from the camera. In order to prevent this behavior, we modified the code of this algorithm to make its second parameter independent from the size of the regions. Fig. 4.1 show three examples of RAGs we obtained for this video.

4.2 Evaluation of the Patterns

To evaluate our spatio-temporal patterns, we use some ground truth. For both the real and the synthetic videos, we have tagged the positions of the plane(s) (objects o) in each frame of the video.

We introduce two measures which assess how precisely a spatio-temporal pattern p corresponds to a given target object o in the video frames. These measures are adaptations of the popular *precision* and *recall* measures as described below:

- **precision**: fraction of the occurrences of p (in the target graphs) of which every node maps to o in the corresponding video frames. The intuition behind this measure is to evaluate the *purity* of p, that is, p has the maximum precision if it maps only to o and nothing else.
- **recall**: Let n be the number of frames in which o is present. The recall is defined as the fraction of n in which there exists at least one occurrence of p where every node maps to o. Here, the intuition is to evaluate the *completeness* of p. More precisely, the idea is to check whether the occurrences of p map to all occurrences of o in the set of video frames.

Since our algorithm is exhaustive, that is, it mines for all frequent spatio-temporal patterns in the graph database without supervision, the mining result may consist of different spatio-temporal patterns corresponding to different objects, or even to no specific one (w.r.t. the proposed measures). To be able to evaluate the precision and recall of our spatio-temporal patterns for all 3 different planes (in the synthetic video) and for the drone (in the real video), we have considered that the spatio-temporal patterns starting in every frames of the video have been tagged according to the object it belongs to. In other words, we evaluate the precision and recall of each spatio-temporal patterns knowing in advance what the first occurrence of the pattern in each occurrences graph maps to.

4.3 Spatio-Temporal Paths for Object Tracking

To assess the effectiveness of the spatio-temporal paths for object tracking, we apply the following strategy. We first build the occurrences graph and then, for each target object, we select the occurrences matching them in the first frame and

[3] http://www.cs.brown.edu/∼pff/segment/

Table 1. Evaluation of the connected components (CC) issued from all patterns with minfreq = 721 and $\tau = 1$ for DyPlagram and for DyPlagram_st. The labels are created from the size of the region.

	DyPlagram with post-processing			DyPlagram_st		
$\epsilon = 10$, $\text{minfreq}_{st} = 10$						
	Precision(%)	Recall(%)	Number of CCs	Precision(%)	Recall(%)	Number of CCs
plane 1	78	7	151	78	7	**114**
plane 2	72	3	129	**95**	**3**	71
plane 3	87	2	**131**	88	2	**84**
$\epsilon = 20$, $\text{minfreq}_{st} = 50$						
	Precision(%)	Recall(%)	Number of CCs	Precision(%)	Recall(%)	Number of CCs
plane 1	77	15	73	82	17	65
plane 2	93	26	43	**100**	29	**39**
plane 3	100	10	60	100	10	60
$\epsilon = 170$, $\text{minfreq}_{st} = 50$						
	Precision(%)	Recall(%)	Number of CCs	Precision(%)	Recall(%)	Number of CCs
plane 1	45	38	27	**51**	42	24
plane 2	51	10	15	49	8	17
plane 3	60	12	21	**69**	13	19

compute the path of lowest cost starting from those occurrences and reaching the last frame using Dijkstra's shortest path algorithm. In all experiments reported here we use a similarity of 2/3 ($\sigma = 0.65$).

In practice the minimum support threshold minfreq can be set, for example, to 1/5 of the total number of frames (to make sure that the patterns occur enough and help the mining process). By default, it will be equal to the minfreq_{st} threshold. minfreq_{st} should be set as low as possible (depending on available memory). The τ should, in general, be set as high as possible (as will be shown in the experiments). The ϵ constraint depends on the motion speed of the target object and on the resolution of the video. We most of the time use 20 pixels.

4.4 DyPlagram_st vs DyPlagram

The experiments showed in Table 1 allow us to compare the DyPlagram algorithm presented in [14] with our new upgraded algorithm, DyPlagram_st, which uses the spatial and the temporal constraints directly in the mining process. These experiments are made with the same synthetic video as in [14] and the same discretization procedure which uses only the size of the region. The same minimum support (minfreq = 721) has been used as well as the same *gap* constraint $\tau = 1$ as in [14]. The minfreq_{seq} threshold used in DyPlagram to prune part of the search space is not used by DyPlagram_st which uses a different minfreq_{st} threshold (explained in Sec. 2.2). However, in these experiments, we set the same threshold for freq_{seq} and freq_{st}. Note that the results for DyPlagram are not exactly the ones reported in [14] because we found that the strategy proposed in [14] was overly optimistic as far as precision was concerned. Indeed, the chosen spatio-temporal patterns (i.e., the connected components (CC)) were the ones for which the first occurrence matched a pattern that was selected in the first frame of the video. This means that a spatio-temporal pattern that also matched a chosen object but for which the first occurrence belongs to a pattern that was not selected in the first frame would not be taken into account to com-

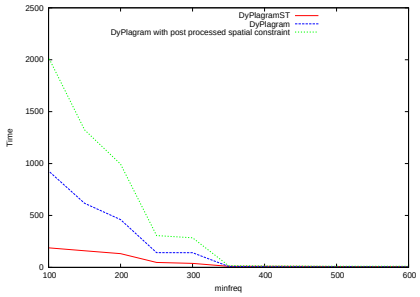

Fig. 4. Time(s) taken by both versions of the DyPlagram algorithm, with $\tau = 1$, $\text{minfreq}_{st} = 50$ and $\epsilon = 20$ to generate all the occurrences (red vs blue line) and to generate the occurrences graph ((red vs green line)

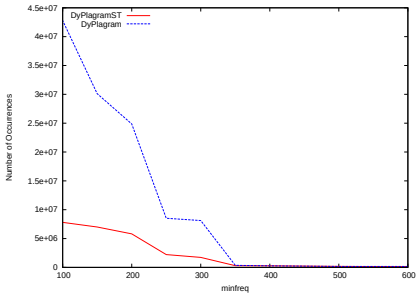

Fig. 5. Number of occurrences generated by DyPlagram while pushing the spatial constraint (red plain line) or not (blue dashed line)

pute the precision of this object. Here we compute the precision and recall for all the CC whose first occurrence matches an object of interest. We expect the precision/recall results to be comparable for both algorithms, although the CC computed by DyPlagram are expected to be more numerous than the ones computed by DyPlagram_st.

As can be seen in Table 1, the connected component obtained with DyPlagram_st are in general less numerous, more precise and have a better recall than the ones obtained with DyPlagram. As already discussed in [14], the distance threshold ϵ has an important impact on the obtained results. Indeed, if it is set too low (to 10 pixels, in our example), we obtain spatio-temporal patterns with high average precision for each X-wing as different occurrences of patterns which map to different X-wing are very well distinguished. However, this leads to a low average recall: since only very close occurrences of the same pattern are linked, the spatio-temporal patterns tend to be short (i.e., have low freq_{st}). When using a distance threshold $\epsilon = 10$, no spatio-temporal patterns with $\text{freq}_{st} \geq 50$ were found for X-wing2 for DyPlagram ([14]), which explains why we used a minfreq_{st} of 10 in this case. Conversely, for a higher ϵ of 170 pixels, the average precision drops as the different X-wings are not well distinguished anymore. For example, it was possible to obtain spatio-temporal patterns with higher recall for the plane 1 (when comparing to the other experiments), but, they had low average precision. Since the plane 1 gets partially out of the video frames around 6 times, a higher number of spatio-temporal patterns were derived for this X-wing for $\text{minfreq}_{st} = 50$ and ϵ of at least 20, which represent the different time intervals where this X-wing is visible through the video. As another example, the plane 2 is hidden only twice by the plane 3 (during around 15 frames) and never goes out of the video frames. This explains the lower number of patterns found for this object, also for $\text{minfreq}_{st} = 50$ and $\epsilon \geq 20$.

Fig. 4 and 5 show efficiency results comparing DyPlagram [14] and DyPlagram_st. As expected, pushing the spatial constraints during the mining

Table 2. Evaluation of the spatio-temporal path with minfreq = 250, minfreq_{st} = 150, $\sigma = 0.65$, $\epsilon = 20$. The numbers between parenthesis correspond to the best precision and recall of the best path in term of recall, and the emphasized results are the best results for each plane

		Size Discretization			Color Discretization		
	τ	Precision(%)	Recall(%)	Paths	Precision(%)	Recall(%)	Paths
plane 1	10	**98.32** (99.72)	**97.50** (99.30)	34	93.92 (99.74)	93.60 (99.86)	21
plane 2		**99.63** (99.73)	**97.26** (98.19)	24	98.65 (100)	**96.82** (99.02)	17
plane 3		**9.49** (16.64)	**8.70** (15.39)	4	- (-)	- (-)	0
plane 1	25	95.79 (100)	94.59 (99.02)	38	**99.17** (99.73)	**98.40** (100)	21
plane 2		65.66 (99.61)	64.61 (98.05)	32	98.54 (100)	96.34 (99.02)	20
plane 3		2.93 (9.09)	2.50 (8.59)	29	31.95 (31.95)	29.54 (29.54)	2
plane 1	100	79.05 (100)	74.37 (94.31)	42	97.76 (100)	95.36 (99.30)	29
plane 2		72.57 (97.53)	67.05 (93.62)	35	**98.87** (100)	96.30 (99.02)	39
plane 3		5.42 (18.46)	4.82 (16.36)	31	**86.27** (90.52)	**75.92** (82.80)	23

step allow us to generate less occurrences (especially for support < 350) in a lower time.

4.5 Evaluation of the Spatio-Temporal Path for Object Tracking

For both datasets, we report the precision and recall results for the spatio-temporal patterns (which have a first occurrence on the object of interest anywhere in the video) and for the spatio-temporal paths (which have a first occurrence on the object of interest in the first frame of the video). The spatio-temporal patterns or connected components (CC) correspond to the global occurrences graph without the similarity edges.

Synthetic Video The experiments reported in Table 2 show the precision and recall results for the paths obtained on the synthetic video when varying the gap between 10 and 100. Results for the CC are similar to the ones reported in Table 1.

Because of the nature of the video, we use a global minimum support minfreq of 250 in order to prune the number of frequent patterns. Indeed, since the synthetic video has been especially made to produce stable graphs, DyPlagram_st returns a lot of frequent patterns on this dataset which leads to a huge global occurrences graph that possibly does not fit into memory for processing. To be able to perform various experiments, especially with the size discretization which does not permit to distinguish the three planes at the mining step, we set the minfreq_{st} to 150 (although as already discussed, it is better to set it as low as possible).

Overall, we obtain very good results for the first two planes (precision and recall close to 100%). We can clearly see the lack of discriminative power of the size discretization when the gap increases. Indeed the paths start to follow different planes, reducing their precision and their recall. For those two planes the color discretization always shows good results, with average precisions and recalls close to the ones of the best paths (values in brackets). Since the 3rd plane moves back and forth horizontally across the field of view (getting almost completely out every 120 frames), only few paths starting on the plane manage to reach

Table 3. Precision, recall and coverage recall computed for the connected components computed and for the real video with minfreq = minfreq_{st}, and $\sigma = 0.65$

		$\epsilon = 10$			$\epsilon = 20$		
τ	minfreq_{st}	Precision(%)	Recall(%)	CC	Precision(%)	Recall(%)	CC
10	100	**100**	26.18	10	**92.48**	22.97	13
	50	93.55	17.40	20	91.35	15.44	25
	10	89.78	2.87	294	89.70	2.72	334
25	100	91.28	35.34	11	89.02	30.03	14
	50	90.28	25.12	18	83.79	20.14	24
	10	88.90	3.18	307	89.47	2.94	358
100	100	89.52	**38.21**	14	89.02	**31.03**	19
	50	92.27	24.38	27	90.30	22.45	30
	10	89.01	4.03	258	89.88	3.63	302

the end of the video when we use a low gap. The paths which uniquely follow this plane are thus more expensive than other paths on which the algorithm can "jump" using the similarity edges decreasing the precision and recall. As we can see, increasing the gap allows to overcome this problem with the color discretization while keeping good results for the other two planes.

Real Video The experiments reported in Table 3 and 4 were made without using a global minimum support threshold (which is equivalent to set minfreq = minfreq_{st}). Because of the segmentation, this dataset is a lot less stable than the synthetic one resulting in less frequent patterns. For this one, so far, only the color discretization gave good precision/recall results (we also tried the size and some other color discretization).

Connected Component (CC) The results for the connected components are presented in Table 3. Those experiments have been obtained for $\epsilon = 10$ and $\epsilon = 20$, above that the precision started to drop significantly (which is expected for large ϵ values if other distracting objects are frequent).

As expected, the precision is a little higher with $\epsilon = 10$ (100% for $\epsilon = 10$ when $\tau = 10$ and $\text{minfreq}_{st} = 100$ against 92.48% for $\epsilon = 20$). The fact that the average recall also decreases with a higher distance is more surprising at first glance. This is explained by the fact that most of the time, $\epsilon = 10$ is enough to follow the drone, but sometimes the drone or the camera movement accelerates. In those cases a higher distance might give longer and better CC but also might introduce some noisy ones which would decrease the average recall and precision.

The average recall also lowers when we lower minfreq_{st}. This is due to the fact that when using a low minfreq_{st} DyPlagram_st outputs short spatio-temporal patterns that necessarily have a low recall. Lowering minfreq_{st} slightly reduces the precision of the connected components but increases their number.

As also expected, higher gaps lead to better recall (38.21% for $\tau = 100$ when $\epsilon = 10$ and $\text{minfreq}_{st} = 100$ against 26.18% for $\tau = 10$) as well as improve the coverage of the spatio-temporal patterns in the whole video. The precision doesn't seem to be influenced by τ when we allow small spatio-temporal patterns (i.e., a low minfreq_{st}).

Table 4. Precision and recall computed for the spatio-temporal paths for the real video with minfreq = minfreq_{st} and $\sigma = 0.65$.

		$\epsilon = 10$			$\epsilon = 20$		
τ	minfreq_{st}	Precision(%)	Recall(%)	Paths	Precision(%)	Recall(%)	Paths
10	100	96.30 (96.30)	67.89 (67.89)	1	98.23 (100)	80.94 (82)	2
	50	98.25 (100)	70.00 (71.26)	2	26.16 (38.96)	24.03 (36.21)	3
	10	91.93 (93.34)	69.60 (70.63)	8	18.75 (36.09)	17.88 (34.73)	8
25	100	98.43 (100)	68.89 (70)	6	98.51 (100)	78.68 (79.68)	6
	50	98.66 (100)	69.05 (70)	7	98.72 (100)	78.82 (79.68)	7
	10	99.06 (100)	69.36 (70.21)	10	**99.03 (100)**	**80.63 (81.36)**	10
100	100	100 (100)	67.42 (67.78)	8	100 (100)	77.52 (79.68)	9
	50	100 (100)	67.36 (67.68)	9	100 (100)	77.54 (79.68)	9
	10	100 (100)	67.21 (67.78)	10	99.26 (100)	79.17 (79.78)	10

Spatio-Temporal Paths Table 4 shows the results for the CC on the real dataset for the color discretization.

A distance ϵ equal to 20 gives the best results in most cases with high precision and good recall (99.03 for precision and 80.63 for the recall for $\epsilon = 20$, $\tau = 25$ and $\text{minfreq}_{st} = 10$ for example). However, the values for $\tau = 10$ show the limits of the use of the shortest path algorithm to tackle our problem. Similarly to what was happening with the third plane in the synthetic video, the shortest path might not always be following the object we want to track if elements in the background or other objects offer better stability than the object we want to track and are close enough to "jump" on them.

The results with our preferred setting (low $\text{minfreq}_{st} = 10$, high $\tau = 100$ and a distance $\epsilon = 20$) show that the spatio-temporal paths can indeed be used to follow an object in the video. The similarity edges introduced are very useful to increase the recall of the patterns and experiments with a higher similarity constraint (for example with $\sigma = 0.8$) show worst results. This shows the importance of this "inexact" matching phase in the process. On the downside, the choice of the labels on the node (here it is a color information) seems to play a very important role to get interesting spatio-temporal patterns although it is difficult to evaluate in an unsupervised setting what could be the best ones. One solution could be to attach more diverse information on the labels of the nodes to overcome this problem.

5 Conclusion

We have presented an unsupervised method based on graph mining to track objects in videos. More precisely, we have used paths computed in an occurrences graph of these frequent graph patterns. The graph is created by linking through spatial, temporal and similarity constraints the frequent patterns to follow one or multiple objects simultaneously in a video. The results on a synthetic and on a real video show that this method is effective to tackle our tracking problem. However, it strongly relies on the labels of the nodes (discretization and chosen features). This problem could be tackled by taking into account multiple and diverse ordered information on the nodes to automatically select the best features depending on the video. Some future work could also be done on the computation of the best paths in the video as the current shortest path algorithm assumes that

our objects of interest are followable from the first to the last frame of the video. Although still naive, we believe that our method could be useful to tackle the difficult problem of tracking multiple objects in the specific case in which both the objects and the background are moving and when no supervised information about the objects to track is known in advance. The proposed method could also benefit from the very recent work which uses supporters (points or objects moving in a correlated way with the tracked objects) or distracters (objects which should not be confused with the objects to track) for example presented in [9] as these would typically represent correlated frequent subgraphs.

References

1. A. Yilmaz, O.J., Mubarak, S.: Object tracking: A survey. ACM Computing Surveys 38(4), 13+ (2006)
2. Berlingerio, M., Bonchi, F., Bringmann, B., Gionis, A.: Mining graph evolution rules. In: Proceedings ECML-PKDD. pp. 115–130 (2009)
3. Borgwardt, K.M., Kriegel, H.P., Wackersreuther, P.: Pattern mining in frequent dynamic subgraphs. In: Proceedings ICDM. pp. 818–822 (2006)
4. Cai, L., He, L., Xu, Y., Zhao, Y., Yang, X.: Multi-object detection and tracking by stereo vision. Pattern Recogn. 43(12), 4028–4041 (Dec 2010)
5. Celik, M., Shekhar, S., Rogers, J.P., Shine, J.A.: Mixed-drove spatiotemporal co-occurrence pattern mining. IEEE TKDE 20(10), 1322–1335 (Oct 2008)
6. Chang, R.F., Chen, C.J., Liao, C.H.: Region-based image retrieval using edgeflow segmentation and region adjacency graph. In: IEEE ICME. pp. 1883–1886 (2004)
7. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24(5), 603–619 (2002)
8. Diego, F., Evangelidis, G., Serrat, J.: Night-time outdoor surveillance by mobile cameras. In: ICPRAM (2012)
9. Dinh, T.B., Vo, N., Medioni, G.G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1177–1184 (2011)
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vision 59(2), 167–181 (Sep 2004)
11. Goszczynska, H.: Object Tracking. InTech (2011)
12. Kim, Z.: Real time object tracking based on dynamic feature grouping with background subtraction. In: IEEE CVPR. (2008)
13. Kuo, C.H., Huang, C., Nevatia, R.: Multi-target tracking by on-line learned discriminative appearance models. In: IEEE CVPR. pp. 685–692 (2010)
14. Prado, A., Jeudy, B., Fromont, E., Diot, F.: Mining spatiotemporal patterns in dynamic plane graphs. IDA Journal 17(1), to appear (2013)
15. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: IEEE CVPR. pp. 2432–2439 (2010)
16. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: IEEE ICDM. pp. 721–724 (2002)
17. Yang, H., Parthasarathy, S., Mehta, S.: A generalized framework for mining spatiotemporal patterns in scientific data. In: ACM SIGKDD. pp. 716–721 (2005)
18. Yu, Q., Medioni, G.: Multiple-target tracking by spatiotemporal monte carlo markov chain data association. IEEE Trans. Pattern Anal. Mach. Intell. 31(12), 2196–2210 (Dec 2009)