

## Contribution

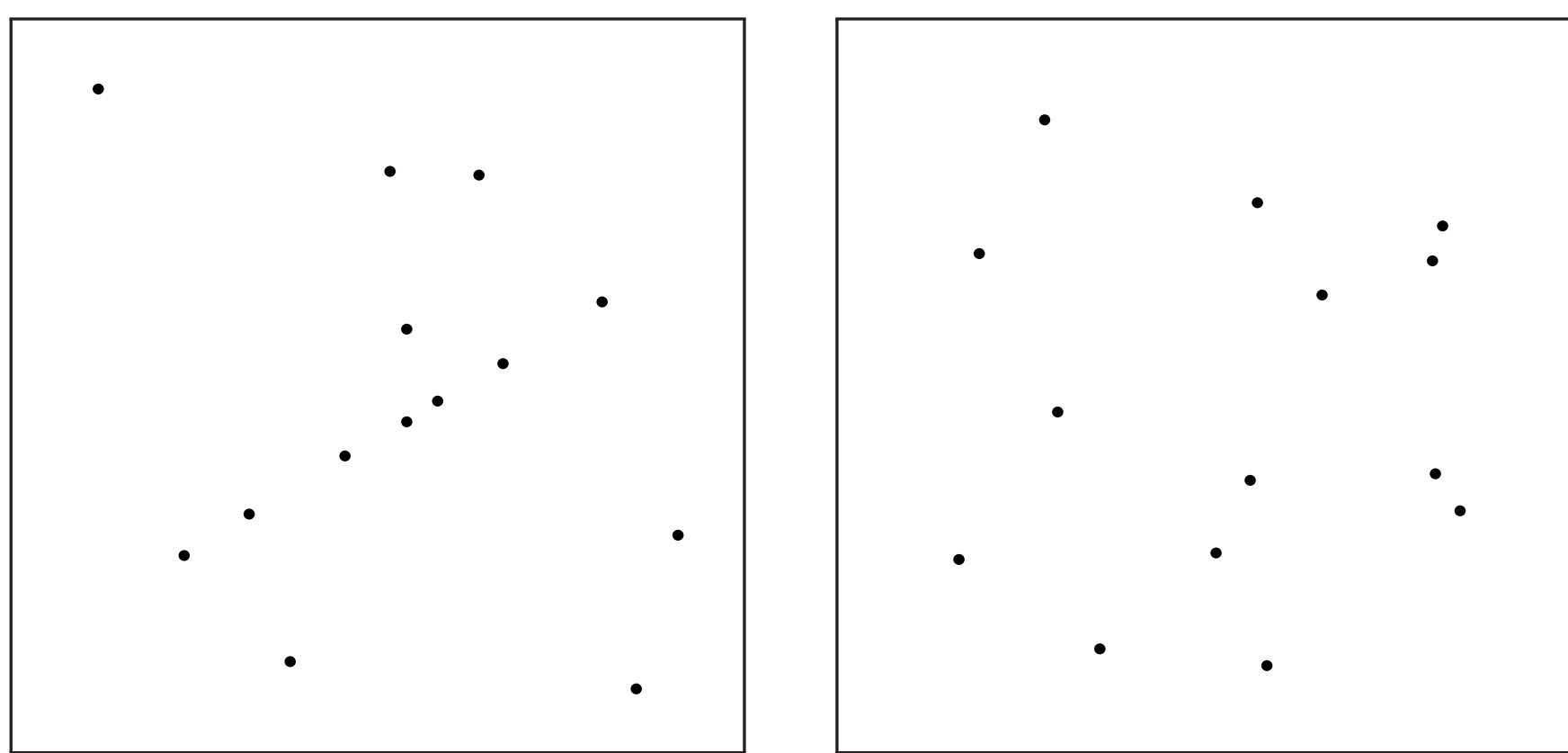
We present a straight continuation of the work initiated by M. Primet *et. al.* in [1] which resulted in the algorithm ASTRE (A-contrario Smooth TRajec-tory Extraction). This algorithm is dedicated to the **detection of smooth trajectories in a (noisy) point set sequence**.

Thanks to the a contrario framework, ASTRE realizes some optimality conditions, however its **main drawback** is its **quadratic complexity** with respect to the number of frames, that may be prohibitive for many practical applications.

We here propose **a variant named CUTASTRE** that cuts the input sequence into overlapping temporal chunks that are processed in a sequential (but non-independent) way, which results in a **linear complexity** with respect to the number of frames.

## The a-contrario framework

The trajectory detection method developed in [1] relies on the a contrario methodology introduced by Desolneux, Moisan and Morel in [2]. The idea underlying its development (dubbed "**Helmoltz Principle**") is that the human visual system detects structures in an image as coincidences which could not appear by chance in a random setting.



**Illustration of the Helmtoltz principle [1, 2, 3]:** Why can't we help seeing an alignment of dots on the left image? According to the Helmtoltz principle, we *a priori* assume that the dots should have been uniformly and independently drawn, and we perceive a structure (aligned dots) because such an alignment is very unlikely to happen by chance. Alignments of several dots can be found in the right image, but they do not pop out, because they are likely to happen by chance considering the total number of points.

**Mathematical formalization of this principle.** The a-contrario methodology is based on two main ingredients:

- A **naïve model**  $\mathcal{H}_0$ : describes what could be pure noise data.
- A **measurement function**  $\mathcal{M}$ : characterizes the kind of structures looked for, e.g.  $\mathcal{M}(x)$  is small iff it is unlikely to observe the realization  $X = x$  for a (random) structure  $X$  in  $\mathcal{H}_0$ .

The amount of surprise when observing an actual structure  $x$  with measurement  $\delta := \mathcal{M}(x)$  is then quantified by  $\mathbb{P}_{\mathcal{H}_0}(\mathcal{M}(X) \leq \delta)$ .

### Number of False Alarms.

Let  $\{X_i\}_{i=1, \dots, N_s}$  the set of (random) structures, a **number of false alarms** is a family  $\{\text{NFA}_i\}_i$  of functions that verify the so-called **NFA property**:

$$\forall \varepsilon > 0, \quad \mathbb{E}_{\mathcal{H}_0}[\#\{i, \text{NFA}_i(X_i) \leq \varepsilon\}] \leq \varepsilon,$$

thanks to Lemma 1 in [3], given a realization  $\{X_i = x_i\}_i$  and a set of positive weights  $\{w_i\}_i$ , one can set

$$\text{NFA}_i(x_i) = w_i \cdot \mathbb{P}_{\mathcal{H}_0}(\mathcal{M}(X_i) \leq \mathcal{M}(x_i))$$

to define a NFA as soon as  $\sum_{i=1}^N \frac{1}{w_i} \leq 1$ .

## From ASTRE to CUTASTRE

### ASTRE

**Data description:** the algorithm is designed to perform trajectory detection over a sequence of **K frames** (with domain  $\Omega$ ), each **containing N points**,

we denote by  $X_i^k$  the  $i$ -th point of frame  $k$ .

**Naive model  $\mathcal{H}_0$ :** a **uniform** and **iid** draw of  $N$  points in each of the  $K$  frames.

**Measurement function:** given a (random) trajectory

$$T_\ell = X_{i_1}^{k_0} \rightarrow X_{i_2}^{k_0+1} \rightarrow \dots \rightarrow X_{i_\ell}^{k_0+\ell-1}$$

with length  $\ell$ , we compute its **acceleration**

$$a(T_\ell) = \max_{p=3, \dots, \ell} \left\| X_{i_p}^{k_0+p-1} - 2X_{i_{p-1}}^{k_0+p-2} + X_{i_{p-2}}^{k_0+p-3} \right\|.$$

**NFA construction:** the **amount of surprise** when observing a trajectory  $t$  with length  $\ell$  and acceleration  $\delta := a(t)$  is estimated by the upper bound

$$\mathbb{P}_{\mathcal{H}_0}(a(T_\ell) \leq \delta) \leq (\pi \delta^2 / |\Omega|)^{\ell-2},$$

allowing the construction of a NFA thanks to [3].

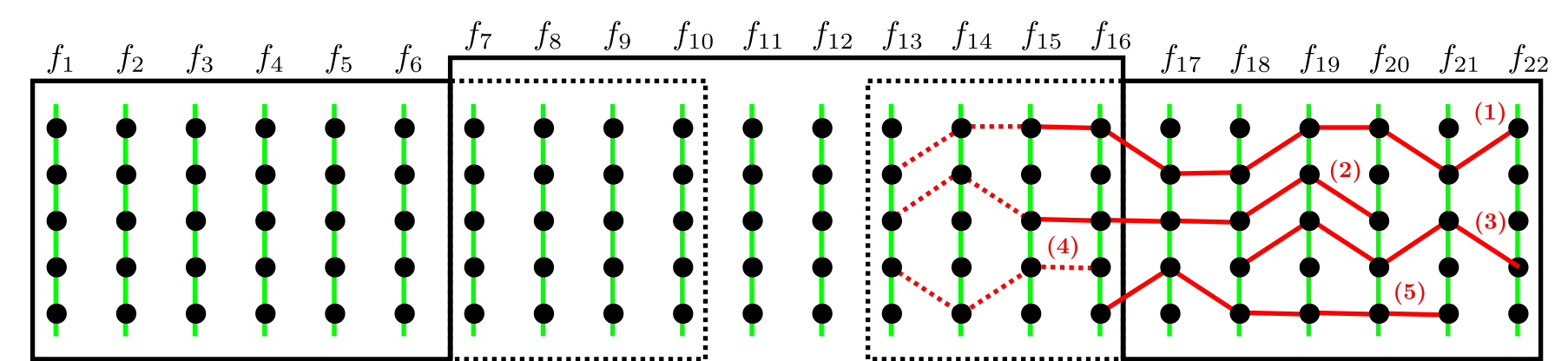
### CUTASTRE

Our approach consists in grouping the frames of the full sequence into **overlapping chunks**. An algorithm similar to ASTRE is applied to each chunk, but enables **trajectory extension** from a chunk to its predecessor.

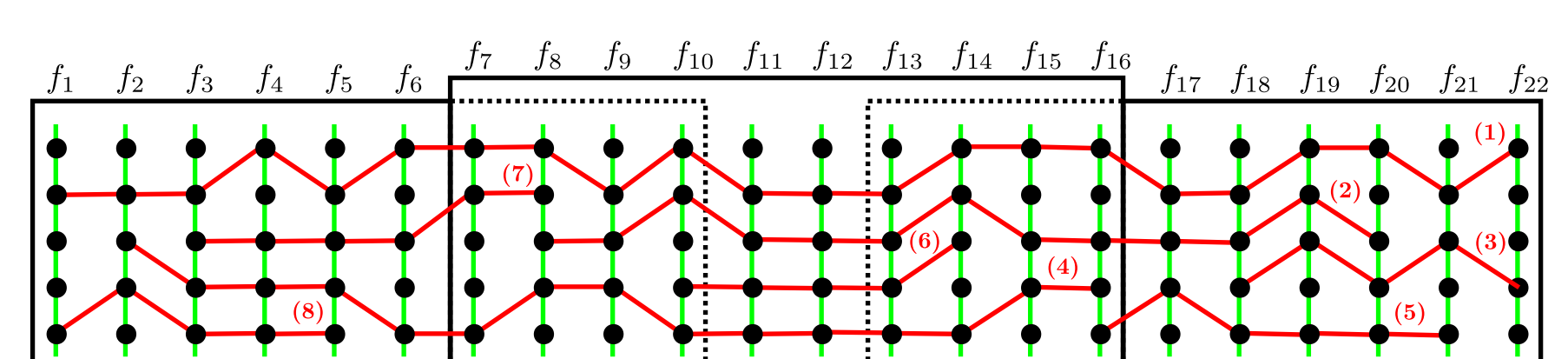
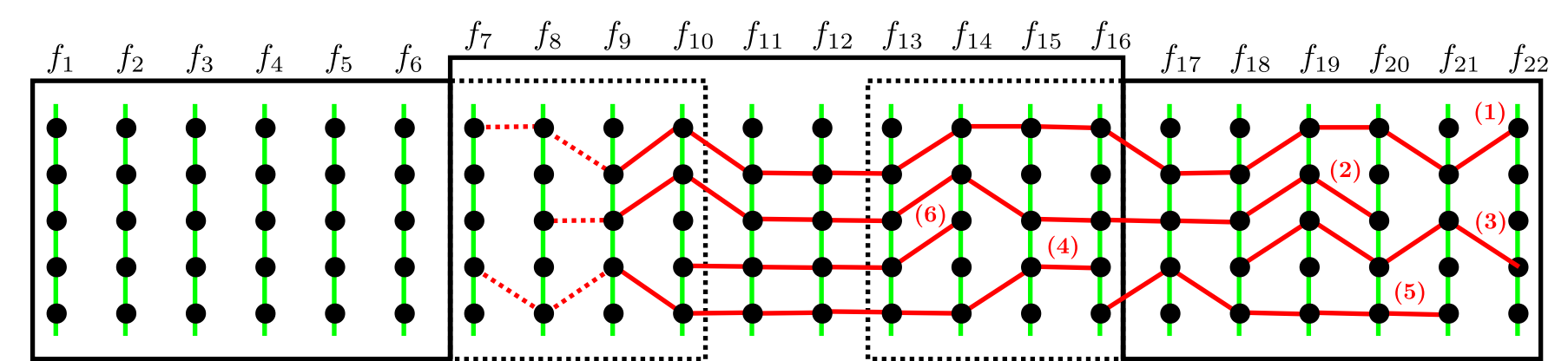
1. In the first chunk, trajectories are detected using ASTRE.
2. Links of the overlap area are then removed, except the last one.
3. Next chunk is then processed, the extension strategy roughly consists in a **modification of  $\mathcal{G}(x, y, \ell)$**  when  $x$  or  $y$  belongs to a previously detected trajectory.
4. The process is repeated (from step 2.) until the last chunk.

**CUTASTRE parameters:** The ASTRE algorithm has the NFA threshold  $\varepsilon$  as unique parameter, which is easy to set ( $\varepsilon$  is a simple bound on the average number of detections made in the naive model). With CUTASTRE we introduced **two new parameters** (the chunks and overlap sizes), **fortunately tuning these parameters appears to be quite simple**, according to the experiments performed on synthetic and real-life data.

### CUTASTRE processes chunks sequentially, ...



### ... allows trajectory extension, ...



### ... leading to a linear complexity $\mathcal{O}(K)$ .

## Main results

### Drastic improvement of the temporal complexity.

CUTASTRE **breaks the prohibitive quadratic complexity** of ASTRE with respect to the number of frames  $K$ .

Typical execution times (s) on PSMG sequences [4].

K	no speed threshold		$\mathcal{S}_{\text{hpc}} = 150$	
	ASTRE	CUTASTRE	ASTRE	CUTASTRE
200	30	1.4	1.2	0.09
500	270	3.6	11	0.26
1000	2160	7.6	80	0.51
3000	-	24.8	1230	1.64
5000	-	39.7	-	2.76

Each sequence contains  $K/10$  trajectories with lengths  $100 \leq \ell \leq 200$  and 10 spurious points per frame.

### Conservation of the detection performances.

Surprisingly, the **performances are not affected** by our acceleration strategy, and are in general **even slightly above** those of the original algorithm.

## Perspectives

### Extend this variant to handle missing points.

This functionality is already available with ASTRE (trajectories "with holes") but with a  **$\mathcal{O}(K^5)$  complexity**. Work is in progress.

### Introduce some features.

According to the nature of the data, we might introduce some new features, for instance adding an **orientation** to the points of the input sequence would be very convenient for applications involving **microtubules** or **small insects**.

## References

- [1] M. Primet and L. Moisan, "Point tracking: an a-contrario approach", *preprint MAP5*, 2012.
- [2] A. Desolneux, L. Moisan and J.-M. Morel, *From Gestalt Theory to Image Analysis. A Probabilistic Approach*, Springer-Verlag, collection Interdisciplinary Applied Mathematics, 2008.
- [3] B. Grosjean and L. Moisan, "A-contrario detectability of spots in textured backgrounds", *Journal of Mathematical Imaging and Vision*, vol. 33:3, pp. 313-337, 2009.
- [4] J. Vesteróy and D. Chetverikov, "Experimental comparative evaluation of feature point tracking algorithms", *Performance Characterization in Computer Vision*, pp. 167-178, Springer, 2000.