Sparse analysis for mesoscale convective systems tracking

Jean-Baptiste Courbot, Vincent Duval, Bernard Legras

Sparse analysis for mesoscale convective systems tracking

Jean-Baptiste Courbot[a,b,*], Vincent Duval[a], Bernard Legras[b]

[a]*Inria Paris, MOKAPLAN team, France*
[b]*LMD, UMR 8539, PSL-ENS/CNRS/Sorbonne Université/Ecole Polytechnique, Paris, France.*

Abstract

In this paper, we study the tracking of deformable shapes in sequences of images. Our target application is the tracking of clouds in satellite images. We propose to use a recent state-of-the-art method for off-the-grid sparse analysis to describe clouds in image as mixtures of 2D atoms. Then, we introduce a method to handle the tracking with its specificities: apparition or disappearance of objects, merging, and splitting. Numerically, this method corroborates the magnitude of the results provided by recent state-of-the-art alternatives. Unlike its counterparts, the choice or regularization and correlation parameters allows additional flexibility regarding the interpretation of clouds' life cycles. Finally, it also provides additional information on the cloud temperature during its life cycle, which seem in accordance with the underlying physical processes.

Keywords: Remote Sensing Image processing, Shape tracking, gridless sparse analysis

1. Introduction

1.1. Problem formulation

In this paper, we study the tracking of deformable shapes in sequences of images. Our target application is the tracking of organized cloud systems, called Mesoscale Convective Systems (MCS) in infrared satellite images.

The MCS are formed when clouds generated by convective instability cluster as a common cloud system with a large upper deck of thick opaque ice clouds accompanied by precipitations over a large area [16]. Such systems are the major part of the high cloud cover in the tropical region where they can reach sizes of up to 1000 km in diameter. Under some favorable circumstances, they can eventually organize as tropical cyclones. The behavior of aggregation and of the MCS life cycle is complex and only partly understood [5, 15]. Collecting large amounts of information about MCS is a required step towards a finer understanding of their behavior. MCS are mostly observed through satellite imaging, notably from geostationary imagers that chart the full tropical cloud cover at a resolution of a few kilometers every ten or fifteen minutes. The high clouds are particularly visible in the infrared window channel, which provides a measure of the emitting surface brightness (temperature, in Kelvin degrees). An example of infrared observation of tropical MCS is given in Figure 1, and more details on this kind of observation can be found in [36]. In this work, we focus on tropical MCS observed through a single channel. Due to the complexity and the huge amount of available information, there is a need for an *automatic* and *robust* tool able to detect and keep track of MCS in time series of these images. Especially, the tool must be robust with respect to changes in image calibration, so that applying it on different dataset does not require tuning algorithms.

This problem is challenging in two aspects. First, we do not know in advance when, where, and in which quantity MCS may appear in the images. In addition, a MCS detection and tracking technique can introduce splitting and merging effects [24]. Secondly, to extract quantitative information about MCS, one must be at least able to describe in an appropriate fashion its shape and its temperature. Hence, the descriptors to extract must contain this information.

1.2. Previous works

MCS Cloud tracking in infrared satellite images. There is no unique practical definition of a MCS applicable to satellite data [17]. In the literature, several works address specifically the problem of MCS tracking in infrared satellite image. In [24], the authors propose a method based on a thresholding step followed by an area-overlap linking step to perform the tracking. The final product, used for measurements, is set as a fit of major and minor axis to the contours. This method coarsely describes shapes, and does not account for local brightness temperature minima in MCS (see Fig. 1), so close MCS components are indistinguishable.

This aspect of the problem has been handled in [14], in which the authors propose an algorithm, named TOOCAN, similar to region growing techniques [1]. The method consists in iteratively dilating "seeds" from lower to higher brightness temperature, under a maximum contrast constraint. The method generates MCS as patches grown around a skeleton made of the time/space trajectories of local brightness temperature minima. It does not allow for merging or splitting and tends to separate large multi-center clusters into individual components. Other methods, such

*Corresponding author: jean-baptiste.courbot@uha.fr
JBC is now with Institut de Recherche en Informatique, Mathématiques, Automatique et Signal (IRIMAS EA7499), Université de Haute-Alsace, Mulhouse, France.
This work was completed within the French Idex with reference number ANR-10-IDEX-0001-02 PSL

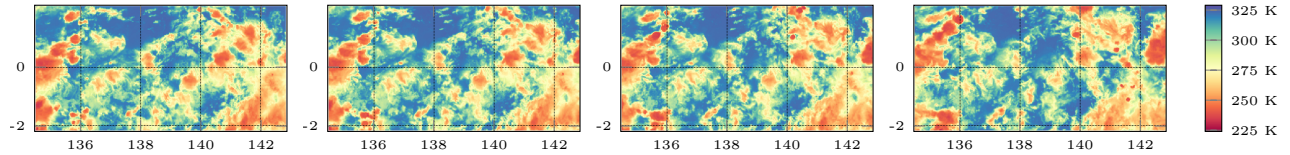


Figure 1: Example of four consecutive infrared images (10.45 μm channel) provided by the Himawari satellite [3], with 2×2 km at the sub-satellite point (140.7°E, 0°N). In this figure and in the following, the colors encode brightness temperature within the same range.

as [30], use level sets as MCS descriptors. All these methods use pixel-based information to infer the properties of MCS.

Alternatively, the authors of [23] propose to use a Bayesian tracking method, based on the Multiple Hypothesis Tracking (MHT) framework [4, 25]. In that paper, the authors first threshold images and fit ellipses to the resulting connected components. These ellipses are the input of the MHT method, which explicitly handles the birth, death, split and merge events. However, the output of this method consists in ellipse contour only, and does not provide information about the cloud's surface brightness.

To our knowledge, there is no method in the literature directly accounting for both brightness temperature and shape information for MCS tracking. Our approach towards this goal is the modeling of MCS as mixtures of 2D atoms, such as Gaussian ellipsoids (detailed below).

Object and shape tracking. Beside MCS tracking, there is a large corpus of literature addressing the problems of object tracking. For instance, Kalman filtering was used, in a long-term fashion, to track MCS in [17]. However the assumptions of the Kalman filter (normal distributions for movements) limit this kind of approach. One of the main improvements in this regard are the Bayesian particle filtering method [20] and its developments [21]. This approach relies on the sampling of a given parameter space, weighted by an appropriate likelihood/data fit term. Regarding the apparition and disappearance of multiple objects in time sequences, the Multi-Bernoulli Particle Filter (MBPF) [22] and its improvements [26, 34, 35] seem relevant to our problem. In many cases, particle filters are applied in a localization context: objects to be tracked are characterized by a few parameters (*e.g.* position and size). In some works, such as [33], the parameters are used for a contour tracking framework.

Nevertheless, in the context of this work the assumption of fixed and low-dimension description does not hold: for each mixture its cardinal should be sampled as well as its parameters. This leverages convergence, speed, and prior distribution fitting issues. Hence, sampling-based methods such as particle filtering are not well-suited to our problem.

Sparse recovery. On the other hand, tools from the compressed-sensing and the sparse recovery theories are specifically designed for data decomposition into simple elements. The topic of sparse recovery and optimization under sparse constraint is well studied (see [37] for a survey). The sparsity prior may be leveraged using a greedy approach such as the OMP algorithm [32], or in a variational approach involving a sparsity-promoting functional (*e.g.* the ℓ_1 norm) as in the LASSO [31] or Basis Pursuit [10]. However, these methods are discretized: the atoms to search for are sampled on a pre-established grid, commonly called dictionary. Choosing the dictionary for a given problem has important consequences in the result quality, as well as in the performance of the considered algorithm.

In recent years, several methods have been proposed to overcome the issues introduced by discretization. In [13], the Continuous Basis Pursuit interpolates between parameters defined on a grid so as to reach subgrid accuracy. Another approach, which follows from [7, 9, 11], is to work in a fully continuous setting, replacing the ℓ_1 norm with its continuous counterpart, the total variation of measures. Following [2], we call this method the BLASSO.

To numerically solve the BLASSO problem, several approaches have been proposed. In [9, 29], the problem is recast as a semi-definite program, whereas the ADCG solver proposed in [6, 7] relies on an alternating gradient based method which progressively adds Dirac masses. Recently, a variant of the ADCG called Sliding Frank-Wolfe (SFW) appeared in [12], which is guaranteed to converge in a finite number of steps under suitable assumptions.

1.3. Contributions

The objective of this paper is to automatically produce information on MCS dynamic to help understand their behavior, which is not fully known. To do so, we introduce a shape tracking algorithm based on recent off-the-grid sparse analysis algorithm. Our method allows the decomposition of MCS into simple components (2D ellipsoid Gaussians) and their tracking in time. This representation allows to directly use pixel intensities, so that the output (several mixtures of parametrized Gaussians) reflects the actual behavior of the observed MCS, instead of using an arbitrary infrared threshold as a proxy for MCS detection. With this model, information regarding shape (size and eccentricity), position (latitude/longitude of components), and brightness temperature can be directly obtained from the image sequence. To our knowledge, the current state-of-the art methods do not allow such direct analysis but only as a byproduct. Besides, our method provides two parameters to modulate its output with respect to the desired interpretation of MCS in infrared satellite images.

The outline of this paper is the following. In Section 2, we describe the observation model to invert, and we present how the SFW algorithm can be used to analyze single mixtures in single images.Then, we present our method to track shapes in images using sparse analysis in Section 3. This method is based on the principles of SFW for individual object analysis, and extends it to manage the specificities of tracking: apparition

and disappearance of objects, as well as splitting and merging. Finally, Section 4 presents the experimental results on images extracted from the Himawari satellite full-disc image (with a focus on tropical area), as well as a numerical comparison with two state-of-the-art methods [14, 23].

In this paper, we track *objects* (MCS) represented by *mixtures*, so we use the two terms without distinction. The mixtures contain *atoms* that are weighted *Dirac masses*, so here also the two terms will be used without distinctions. For convenience, we use in this paper the convention that high intensities in the images correspond to low surface brightness temperature at 10.45μm, and images will be considered in their vector form. Vectors and multi-valued mathematical objects will be represented in bold.

2. Observation model and sparse shape analysis with SFW

2.1. Observation Model

We assume that each MCS is described by a mixture of Gaussians. This assumption does not reflect the physical behavior of MCS, which is complex and partly unknown. Instead, it is chosen so as to capture at best the intensity variations in the image, while being flexible (enabled by the mixture) and numerically tractable (enabled by the sparsity assumption).

We observe a sequence $\{\mathbf{y}_1, \dots, \mathbf{y}_T\}$ of T images, indexed by time. In our observation model, each image $\mathbf{y}_t$, $1 \leq t \leq T$, is built from an imaging operator applied to a measure:

$$\mathbf{y}_t = \mathbf{\Phi}\mu_{\boldsymbol{w}_t,\boldsymbol{\theta}_t} + \boldsymbol{\epsilon_t} \tag{1}$$

$\boldsymbol{\epsilon_t}$ is an error term accounting for the model misfit. In other words, $\mathbf{y}_t$ is a mixture of several atoms. In this paper, we consider the case of 2D Gaussian ellipsoids, parametrized by position, scale, and eccentricity. Here and in the following, we consider for the measures μ a positive weighted Dirac mass sum of the form:

$$\mu_{\boldsymbol{w}_t,\boldsymbol{\theta}_t} = \sum_{n=1}^{N} w_{t,n}\delta_{\boldsymbol{\theta}_{t,n}} \tag{2}$$

with the weight vector $\boldsymbol{w}_t = \{w_{t,1}, \dots, w_{t,N}\}$, such that $w_{t,n} > 0 \ \ \forall n \in \{1, \dots, N\}$[1]. We denote by $\mathcal{D}$ the bounded domain of interest for the parameter set, such that $\boldsymbol{\theta}_{t,n} \in \mathcal{D}$. The imaging operator $\mathbf{\Phi}$ is defined by $\mathbf{\Phi}\mu = \int_{\mathcal{D}} \phi \, \mathrm{d}\mu_{\boldsymbol{w},\boldsymbol{\theta}}$ where $\phi : \mathcal{D} \to \mathbb{R}$ is, in our case, a convolution by a 2D Gaussian ellipsoid kernel.

Gathering the Dirac masses into labeled object mixtures (the MCS), we may rewrite:

$$\mathbf{y}_t = \sum_{l=1}^{L} \mathbf{\Phi}\mu^{(l)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t} + \boldsymbol{\epsilon_t}. \tag{3}$$

where l denotes one label among the L objects appearing in the sequence.

Moreover, between $t = 1$ and $t = T$ an object l may appear (at $t_{l,\text{start}}$) or disappear (at $t_{l,\text{stop}}$). Accounting for these phenomena, the complete observation model is:

$$\mathbf{y}_t = \sum_{l=1}^{L} \mathbf{1}_{\{t_{l,\text{start}} \leq t \leq t_{l,\text{end}}\}} \mathbf{\Phi}\mu^{(l)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t} + \boldsymbol{\epsilon_t} \tag{4}$$

The parameter of each Gaussian are the position (x, y), the big axis a, the eccentricity e and the orientation α. Hence, given the time t, a label l, and an atom index n, we have $\boldsymbol{\theta}^{(l)}_{t,n} = \{x, y, a, e, \alpha\}^{(l)}_{t,n}$. This parametrization allows to directly control ellipsoid eccentricity in order to avoid degenerate cases.

We also consider the following phenomena:

- *merging*: several mixtures, separated at time t, may be grouped at time $t + 1$.
- *splitting*: a mixture at time t may be split in several mixtures at time $t + 1$.
- *exchange*: atoms belonging to a label at time t may switch to another label at time $t + 1$.

The purpose of this paper is to invert the observation model (4) for the parameters of the measure, knowing only each $\mathbf{y}_t$ and $\mathbf{\Phi}$, while accounting for the split and merge phenomena. Let us notice, in particular, that for each time frame t, we do not know the number L of objects involved, nor the number of Dirac masses N in each mixture.

We seek the inversion of the observation model (4) within a continuous framework: the parameters are real-valued within the domain $\mathcal{D}$, which is a compact defined by lower and upper bounds for each parameter component.

2.2. Retrieving sparse shapes with Sliding Frank-Wolfe

Before handling the inversion of (4), we consider the following sub-problem: *how can we retrieve a single mixture from an image ?* This amounts to invert the simplified observation model (1) at a given time t, assuming there is only one object in the image.

To retrieve $\mu_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ in a sparse fashion, we minimize among all non-negative measures μ the following criterion:

$$C(\mathbf{y}_t, \mu, \lambda) = \frac{1}{2} \|\mathbf{y}_t - \mathbf{\Phi}\mu\|_2^2 + \lambda\mu(\mathcal{D}) \tag{5}$$

where $\lambda > 0$ is the regularization parameter, and $\mu(\mathcal{D})$ denotes the total mass of the non-negative measure μ. In the case (2) of a sum of Dirac masses, $\mu_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}(\mathcal{D}) = \sum_{n=1}^{N} w_{t,n}$. Such a minimization problem is challenging, since, contrary to standard ℓ_1 problems, not only the weights $\{w_{t,n}\}$ of the masses, but also their positions $\{\boldsymbol{\theta}_{t,n}\}$ in the parameter space are unknown, since they are not assumed to belong to a predefined a grid.

To optimize (5), we use the SFW, a variant of [6, 7] proposed recently in [12]. This greedy algorithm iteratively adds new Dirac masses to the current set, then optimizes only the weights $\boldsymbol{w}_t$ in a classical LASSO setting, and then optimizes locally both $\boldsymbol{w}_t$ and $\boldsymbol{\theta}_t$ using a quasi-Newton method such as BFGS [28].

The key elements in this algorithm are the choice of new Dirac masses and the stopping criterion. Both rely on a certificate η

[1] Formally, in Eq. (2) N depends on the time frame t. Since there is no ambiguity when N is used, this dependence is not made explicit.

Algorithm 1 Sliding Frank-Wolfe

Input: $\mathbf{y}_t$, λ
Output: Minimizer $\hat{\mu}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$
Initialization: $\mu_{\boldsymbol{w}_t^{(0)},\boldsymbol{\theta}_t^{(0)}} = 0$.
while $\max_{\mathcal{D}} \eta^{(k)} > 1$ (see Eq. (6)) :

1. Augment the support:

$$\boldsymbol{\theta}_t^{(k)} = \boldsymbol{\theta}_t^{(k-1)} \cup \arg\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{(k)}$$

2. Adjust weights only (LASSO):

$$\tilde{\boldsymbol{w}}_t^{(k)} = \arg\min_{\boldsymbol{w}\in\mathbb{R}_+^k} C(\mathbf{y}_t, \mu_{\boldsymbol{w},\boldsymbol{\theta}_t^{(k)}}, \lambda)$$

3. Local descent on all parameters, using BFGS and starting at $\mu_{\tilde{\boldsymbol{w}}_t^{(k)},\boldsymbol{\theta}_t^{(k)}}$:

$$\boldsymbol{w}_t^{(k)}, \boldsymbol{\theta}_t^{(k)} = \text{local descent of } C(\mathbf{y}_t, \mu_{\boldsymbol{w},\boldsymbol{\theta}}, \lambda) \underset{\boldsymbol{w}\in\mathbb{R}_+^k, \boldsymbol{\theta}\in\mathcal{D}^k}{}$$

4. Measure update: $\mu_{\boldsymbol{w}_t^{(k)},\boldsymbol{\theta}_t^{(k)}} = \sum_{n=1}^k w_{t,n}^{(k)} \delta_{\boldsymbol{\theta}_{t,n}^{(k)}}$

written, at step k, as:

$$\eta_t^{(k)} \overset{\text{def.}}{=} \frac{1}{\lambda}\boldsymbol{\Phi}^\top\left(\mathbf{y}_t - \boldsymbol{\Phi}\mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}}\right); \tag{6}$$

where $\boldsymbol{w}_t^{(k-1)}$ and $\boldsymbol{\theta}_t^{(k-1)}$ are estimated at the previous step. This certificate is a continuous function defined over the compact set $\mathcal{D}$. It is high-valued in the locations of $\mathbf{y}_t$ that are not well explained by $\mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}}$. Besides, $\mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}}$ is a solution to (5) if and only if $\max_{\mathcal{D}} \eta^{(k)} \leq 1$. Indeed, that condition is equivalent to standard optimality conditions for constrained convex problems, as explained (informally) in Appendix A. Hence, this condition is a natural stopping criterion for the SFW algorithm.

The SFW procedure is described in Algorithm 1, and Fig. 2 provides a stepwise illustration of the algorithm. Details on its implementation are given in Appendix B. We refer the reader to [12] for further detail on the SFW algorithm.

3. Proposed method

3.1. Method outline

In this section, we introduce our method to track several deforming shapes in image sequences, based on the sparse image decomposition presented in the previous section.

Our approach requires to consider several aspects of the problem, listed below and detailed in the next subsections:

- *Initialization* refers to the estimation of the mixtures number and parameters in the initial image $\mathbf{y}_0$.
- *Propagation* is the task of recursive mixture estimation, from time $t-1$ to time t. This step models the time evolution of MCS.

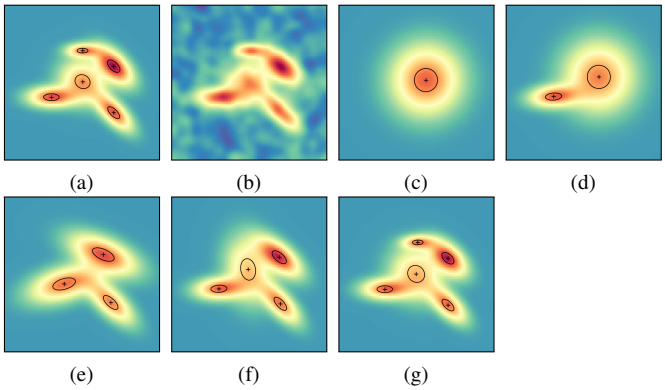


Figure 2: Illustration of SFW steps on a 100×100 px synthetic image (cf. Alg. 1). (a) synthetic image generated with the true $\boldsymbol{\theta}_t$ containing 5 atoms and a 2D ellipsoid kernel ϕ. (b) is $\mathbf{y}_t$, generated from (a) with an additive blurred (Gaussian filter with 3 px scale) Gaussian noise. (c)–(g) are images generated with $\Phi\mu_{\boldsymbol{w}_t^{(k)},\boldsymbol{\theta}_t^{(k)}}$ from $k = 1$ to $k = 5$ within the SFW algorithm, on which the atom are superimposed to illustrate the parameters (positions, scales, orientation).

- *Creation* manages the apparition of mixtures at time t that did not exist at time $t-1$. This step handles the apparition of a new MCS in the image sequence.
- *Merging* and *splitting* handle the interaction between mixtures over time. This step manages the aggregation of MCS in the time sequence.

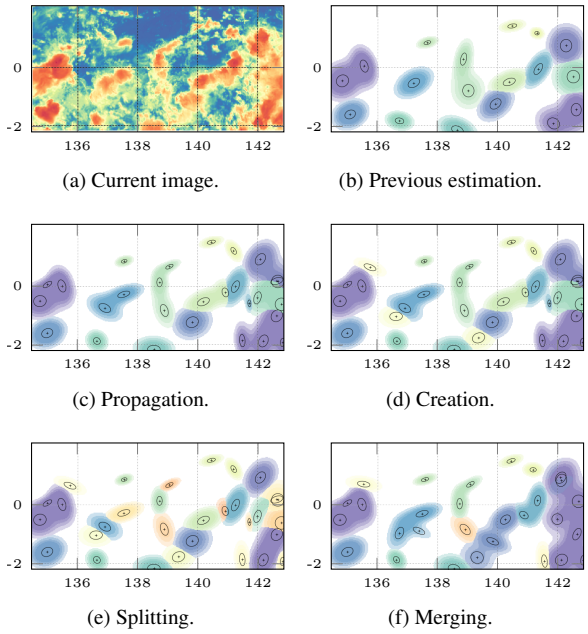
The method we propose is based on the four steps described previously, and is called SAST for Sparse Analysis for Shape Tracking. The algorithm iteratively provides estimators $\hat{\mu}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ from $t = 1$ to $t = T$, with, at each step, a recursive adaptation of the estimator. It is described in Alg. 2 and a step-by-step illustration of the algorithm is provided in Fig. 3.

Algorithm 2 SAST algorithm

Input: Image sequence, correlation threshold C_{thr}, regularization parameter λ
Output: Estimator $\hat{\mu}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$, $t \in \{1,\ldots,T\}$
for t = 1 to t = T :
  if t = 1 :
    Initialize: Alg. 3 applied to $\mathbf{y}_0$
  else:
    Propagate: Alg. 4 with $\hat{\mu}_{\boldsymbol{w}_{t-1},\boldsymbol{\theta}_{t-1}}$
    Create: Alg. 3 applied to $\mathbf{r}_t = \mathbf{y}_t - \boldsymbol{\Phi}\hat{\mu}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$
  Split: Alg. 6 using C_{thr}.
  Merge: Alg. 5 using C_{thr}.
  Local variations: apply SFW (Alg. 1) on the modified mixtures starting from their current values.
  Save the current value of $\hat{\mu}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ as a result.

Note that we introduced an additional local adjustment step after splitting and merging. Its purpose is to provide further flexibility after the split and merge steps, so that newly formed mixtures can be adapted to the image. To do so, we process each mixture with the SFW algorithm, using the current mixtures

values as initial guess. The following subsections detail each step of Alg. 2.



(a) Current image. (b) Previous estimation.

(c) Propagation. (d) Creation.

(e) Splitting. (f) Merging.

Figure 3: Step-by-step illustration of the SAST algorithm. (a) has the same intensity range as in Fig. 1 and (b)-(f) represents outputs by color (labels) with an opacity defined by the set (235K, 245K, 255K, 265K) of brightness temperature thresholds. Individual atoms are represented in overlay.

3.2. Initialization

At this step, the problem is the retrieval of several shapes in the image $\mathbf{y}_0$. This is the conjunction of two problems:

1. finding the number of objects,
2. estimating the parameter of each objects.

These problems are handled through a two-step SFW procedure. The first step consists in applying SFW to an under-resolved version of $\mathbf{y}_0$, to produce an estimation of the number of mixtures as well as a coarse estimation of their parameters. This step yields several atoms, that are for now considered as separate objects: they may be merged or split later (see Subsection 3.5). We observed that this procedure provides the same mixture number than when applying SFW on the fully-resolved image, while being notably faster.

The second step processes each of the resulting atoms separately. It consists in applying SFW using the fully resolved image, in order to obtain several atoms per object.

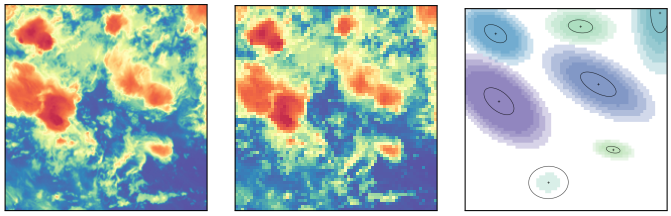
The initialization procedure is summarized in Alg. 3 and its first step is illustrated in Fig. 4.

3.3. Propagation and disappearance

At this step, we handle the problem of retrieving, at time t, a mixture l that already exists at time $t-1$. This amounts to finding μ_{w_t,θ_t} minimizing $C(\mathbf{y}_t, \mu_{w_t,\theta_t}, \lambda)$, knowing already $\hat{\mu}_{w_{t-1},\theta_{t-1}}$. We assume that the variation in mixture number and parameters values are small between $t-1$ and t. Hence, the propagation

Algorithm 3 Initialization: coarse-to-fine SFW

Input: Image $\mathbf{y}_0$
Output: Estimation $\hat{\mu}_{\boldsymbol{w}_0,\boldsymbol{\theta}_0}$
Step 1.
Let $\tilde{\mathbf{y}}_0$ be the under-resolved version of $\mathbf{y}_0$.
Apply SFW (Alg. 1) to $\tilde{\mathbf{y}}_0$.
Step 2.
for each resulting Dirac mass :
  Apply SFW on $\mathbf{y}_0$ starting at the coarse estimation from step 1.



(a) $\mathbf{y}_0$ (b) Under-resolved $\tilde{\mathbf{y}}_0$ (c) SFW on $\tilde{\mathbf{y}}_0$.

Figure 4: Illustration of step 1 of the initialization step. In (c), labels are represented by color and the opacity corresponds to brightness temperature thresholds (see Fig. 3).

step consists in applying SFW to $\mathbf{y}_t$, using $\hat{\mu}_{w_{t-1},\theta_{t-1}}$ as an initial guess. This starting measure is modified as follow:

- we need to allow for pruning, to handle decreasing number of atoms in mixtures. This is done by a LASSO optimization on the weights only, followed by a removal of the zero-valued outputs.
- we account for the average trend of mixtures in $\mathcal{D}$, as the difference of the parameter averages between $t-1$ and $t-2$. This corresponds to speed in the parameter space $\mathcal{D}$.

After this step, an object may have no Dirac mass left. In this case, the object is assumed to have disappeared, so an estimation of $t_{l,\text{end}}$ is known and the object is not proceed further. The propagation step is summarized in Alg. 4.

Algorithm 4 Propagation using SFW

Input: Estimation $\hat{\mu}_{w_{t-1},\theta_{t-1}}$, $\mathbf{y}_t$
Output: Estimation $\hat{\mu}_{w_t,\theta_t}$
if $t > 2$:
  Estimate the trend as $\hat{\boldsymbol{w}}_s = \overline{\boldsymbol{w}}_{t-1} - \overline{\boldsymbol{w}}_{t-2}$ and $\hat{\boldsymbol{\theta}}_s = \overline{\boldsymbol{\theta}}_{t-1} - \overline{\boldsymbol{\theta}}_{t-2}$.
  Initial guess: $\hat{\mu}_{w_{t-1}+\hat{w}_s,\theta_{t-1}+\hat{\theta}_s}$.
else:
  Initial guess: $\hat{\mu}_{w_{t-1},\theta_{t-1}}$.
Pruning, adjusting weights only (LASSO).
Run the SFW iterations.

3.4. Apparition

We address the problem of retrieving, at a given time t, mixtures that did *not* exist at time $t-1$. The goal is to detect where

new mixtures appear, and how many. This problem is close to the problem handled at initialization, excepting that some parts of the image are already explained by mixtures.

Let us consider the current residual, noted $\mathbf{r}_t$:

$$\mathbf{r}_t = \mathbf{y}_t - \mathbf{\Phi}\hat{\mu}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t} \tag{7}$$

This residual is obtained with the current estimator, *i.e.* after the propagation step. Reusing the methods from the initialization step, the creation of new object at time t is done by the application of Alg. 3 to the residual $\mathbf{r}_t$.

3.5. Merging mixtures

When considering the merging of two mixtures, we have to consider the following cases:

- two or more mixtures overlap entirely: they both describe the same phenomenon and must be merged under one label;
- two or more mixtures overlap partly: some Dirac masses in these mixtures describe the same phenomena, and they must be moved under the same label.

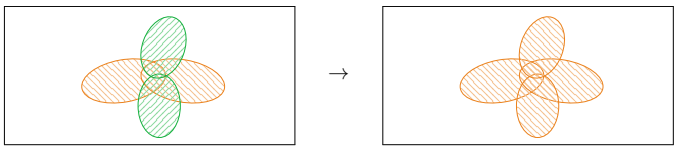
An illustration of these two cases is given in Fig. 5.

The merging procedure we propose account for the two situations. We measure correlation coefficients between mixture images $\mathbf{\Phi}\mu^{(l)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$, corresponding to each label l at the fixed time t.

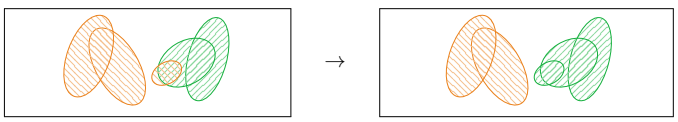
Let l_1 be a given "reference" mixture, and l_2 an other mixture that may be merged:

- if $\mathbf{\Phi}\mu^{(l_1)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ and $\mathbf{\Phi}\mu^{(l_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ are too much correlated, they are merged.
- if the image of a Dirac mass $\mu^{(l_2,n_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ from the mixture l_2 is too much correlated to the image of another Dirac mass $\mu^{(l_1,n_1)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ from l_1, it is merged to l_1 and removed from l_2.

This approach requires the definition of a correlation threshold C_{thr}, above which the merging is performed. In practice, this threshold defines the tolerance to overlap in the resulting image. Alg. 5 summarizes the merging step.



(a) Mixture merging.



(b) Atom transfer.

Figure 5: Illustration of the two cases considered for merging. In case (a) the two mixtures are merged, while in case (b) the smaller atom is merged to the right-hand mixture and removed from the left-hand mixture.

Algorithm 5 Correlation-based merging

Input: Correlation threshold C_{thr}, current measure $\mu_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$
Output: Merging of correlated mixtures.

Let $\mathcal{L}$ be the mixture list, sorted by total mass in decreasing order, and $\mathcal{R}$ be the reference mixture list, empty at start.
for $l_1 \in \mathcal{L}$: ▷ *reference mixture*
  Append l_1 to $\mathcal{R}$.
  for $l_2 \in \mathcal{L}$ with $l_2 \notin \mathcal{R}$: ▷ *other mixture*
    Step 1. Single atom merging or transfer
    if $N_2 = 1$:
      $\forall n_1 \in \{1,\dots,N_1\}$, compute:
      $c_{n_1,l_2} = \text{Corr}(\mathbf{\Phi}\mu^{(l_1,n_1)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}, \mathbf{\Phi}\mu^{(l_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t})$
      if $\exists n_1 \in \{1,\dots,N_1\}$ s.t. $c_{n_1,l_2} > C_{\text{thr}}$:
        Merge l_1 and l_2. The label of the mixture with the larger total mass is maintained.
    else:
      for $n_2 \in \{1,\dots,N_2\}$: ▷ *within* l_2
        Let $\mu^{(l_2,\setminus n_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ be $\mu^{(l_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ without the n_2 atom.
        Compute $\begin{cases} c_{n_2,l_1} = \text{Corr}(\mathbf{\Phi}\mu^{(l_2,n_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}, \mathbf{\Phi}\mu^{(l_1)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}) \\ c_{n_2,l_2} = \text{Corr}(\mathbf{\Phi}\mu^{(l_2,n_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}, \mathbf{\Phi}\mu^{(l_2,\setminus n_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}) \end{cases}$
        if $c_{n_2,l_1} > c_{n_2,l_2}$:
          Move $\mu^{(l_2,n_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$ to the mixture l_1.
    Step 2. Mixture merging.
    if $\text{Corr}(\mathbf{\Phi}\mu^{(l_1)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}, \mathbf{\Phi}\mu^{(l_2)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}) > C_{\text{thr}}$:
      Merge l_1 and l_2. The label of the mixture with the larger total mass is maintained.

3.6. Splitting

At a given time t, a mixture may also be split into several parts. This situation arises, within a mixture, when two or more components are too much separated from one another, i.e. they do not overlap enough to represent the same object anymore.

To quantify this separation, we evaluate the correlation matrix of the mixture Dirac masses' images. This matrix is thresholded using C_{thr}, below which the Dirac masses are considered uncorrelated. The resulting binary matrix is used to build a undirected, unweighted graph Γ where nodes represent Dirac masses and edges represent correlations above the threshold. If the graph Γ has several disconnected component, then the mixture must be split along these separated components. The procedure is illustrated in Fig. 6 and reported in Alg. 6.

Let us remark that this step relies on the same principles as the merging step, so that using both in long time series allow the estimator to be balanced, *i.e.* there is no over-splitting or over-merging.

4. Results

This section presents the results of the SAST method for the tracking of MCS in infrared satellite images, as well as a comparison with the two state-of-the-art methods proposed in [14] and [23].

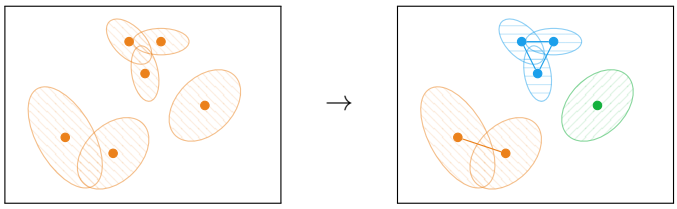


Figure 6: Illustration of the splitting procedure. Left: a single mixture containing 6 atoms. Right: graph of correlated atoms, colored by disconnected components. The corresponding atoms groups form the new mixtures.

Algorithm 6 Splitting

Input: Correlation threshold C_{thr}, current measure $\mu_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$

Output: Splitting of non-correlated Dirac masses within mixtures.

for Each mixture $l \in \{1,\dots,L\}$:
  Compute the correlation matrix $\mathbf{C}$ with $\boldsymbol{\Phi}\mu^{(l,n)}_{\boldsymbol{w}_t,\boldsymbol{\theta}_t}$, $1 \le n \le N$.
  Threshold this matrix with C_{thr}.
  Build the graph Γ from $\mathbf{C}$.
  if Γ has more than one connected component :
    Split l along the connected components.

Since the behavior of MCS is not fully known, creating realistic synthetic data on dynamic MCS is far from trivial. Since no ground truth is available, we experiment the three methods on real Himawari [3] images, and then compare them qualitatively and numerically.

4.1. Experimental settings

4.1.1. SAST algorithm

The SAST algorithm requires several choices regarding its implementation. The first one is the choice of the convolution kernel, which will describe the shape of individual atoms forming mixtures.

As mentioned previously, we have chosen to describe MCS as mixtures of 2D Gaussian, since MCS contour is complex and difficult to determine clearly without using a specific threshold in infrared satellite image at 10.45 μm.

$\mathcal{D}$ contains each parameters set $\boldsymbol{\theta}^{(l)}_{t,n}$, so its bounds must be specified.We choose to set the position bounds at the image border, and α between $-\pi$ and π. In order to avoid degenerate cases, we set the big axis between 2 and 20 pixels and the eccentricity between 0 and 0.9 to avoid degenerate cases. These choices were made to provide fast, yet efficient, results on the data presented hereafter. The output of the algorithm is only marginally affected by changes in these settings.

Two parameters affect the overall behavior of the SAST algorithm:

- the regularization parameter λ sets the sparsity compromise in (5). A larger λ yields fewer atoms, so the description of the clouds will be coarser and obtained more quickly. Conversely, a smaller λ will provide a finer fit to the image.
- the correlation threshold C_{thr} sets the limit for both splitting and merging. A larger C_{thr} implies a larger number of small mixtures, while a smaller value yields larger and fewer mixtures.

These two parameters must be set in accordance with the purpose of the analysis: fine *vs.* coarse, and atomic *vs.* grouped analysis respectively. We observed that a coarser analysis is relevant when used in conjunction with a more grouped aggregation, and conversely. As a consequence, we choose to study the three following cases :

A. $\lambda = 1.25$ and $C_{\text{thr}} = 0.5$,

B. $\lambda = 2$ and $C_{\text{thr}} = 0.25$,

C. $\lambda = 3$ and $C_{\text{thr}} = 0.125$.

These parameters were determined heuristically: a larger (resp. smaller) λ yields a too coarse (resp. overfit) image analysis, while a smaller (resp. larger) C_{thr} yields few (resp. too many) irrelevant groups with respect to the observation.

4.1.2. State-of-the-art methods

In the literature, two recent algorithms address the problem of tracking MCS in time sequences of images. The first one, coined TOOCAN [14], tracks binary shapes based on threshold, surface, and connectivity in the spatio-temporal domain. TOOCAN produces discrete-labeled map of pixels, so there is no first-order information regarding shape or intensity. We implemented TOOCAN strictly based on the description in [14], using the same parameter values.

The second method relies on a recursive Bayesian tracking framework, called MHT [23]. This method provides parametric estimation of ellipses, each one describing the shape of a cloud or cloud system. However, the tracking is not applied directly on images: instead, a binarized image (threshold at 233K) is produced, from which small surface regions are removed. The remaining contours are fit by ellipses, which form the observations considered by the model. Hence, the MHT method works in parametric contour space, without considering the images intensity. For the comparison, we used the implementation of MHT provided by the authors.

Here, we present comparative results on a real image sequence acquired by the Himawari satellite [3], which has a 2×2 km Sub-Satellite Point (SSP) resolution. Since both TOOCAN and MHT were applied initially applied to MSG-1 [27] infrared images (3×3 km SSP resolution), the parameters of both methods were adapted to the Himawari setting. Besides, in preliminary experiments we did not observe any issue related to the time sampling (15 minutes in MSG-1 and 20 minutes in our Himawari sequence) on the results of these two methods.

4.2. Comparative results

4.2.1. Qualitative results

In Fig. 7, we illustrate the three methods on a Himawari sub-satellite image in the infra-red channel at 10.45 μm covering *ca.* 800×400 km^2 over 50 time frames taken each 20 minutes. Several comments can be made regarding these results:

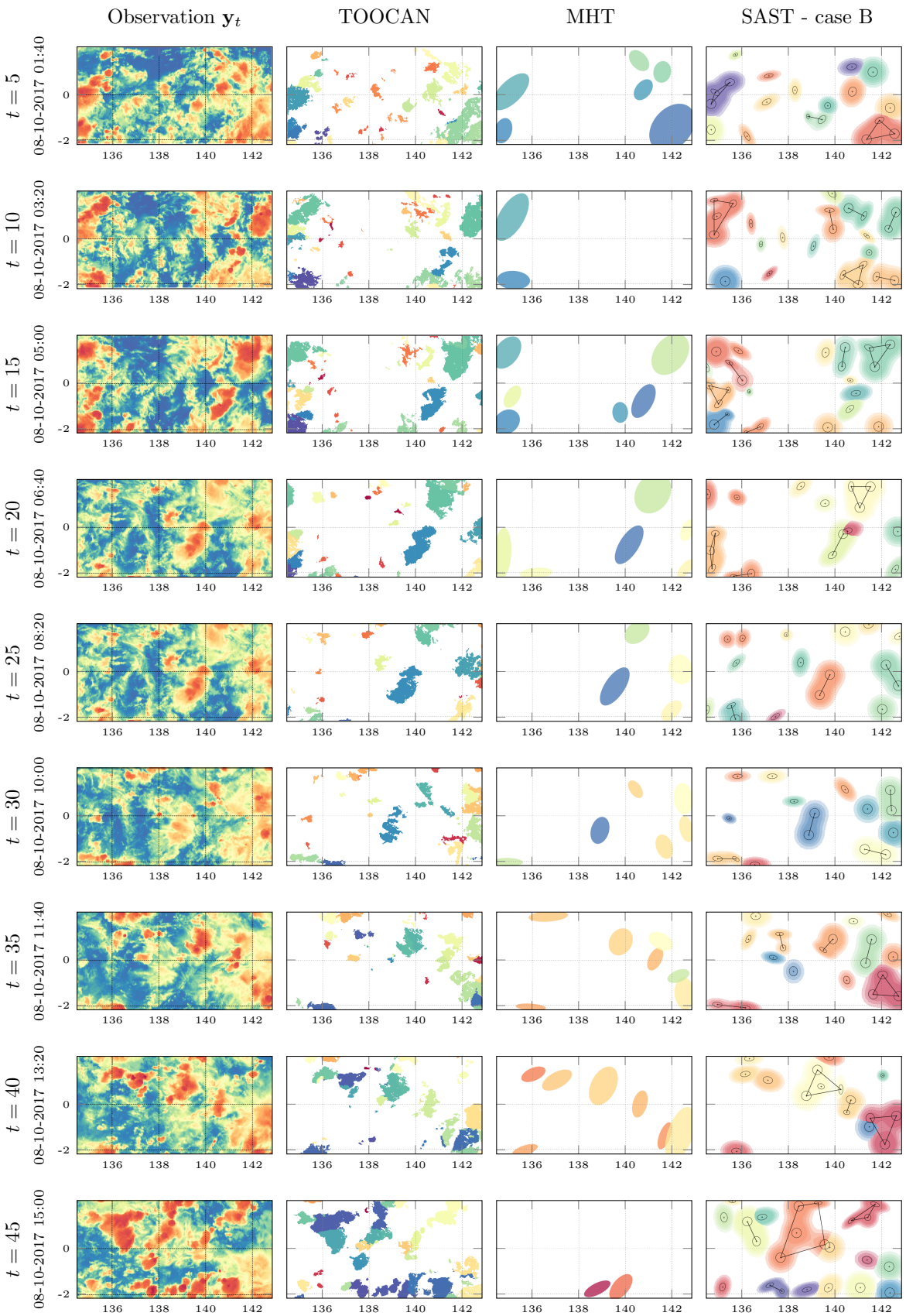


Figure 7: First column: sequence of images from Himawari taken August 10, 2017 from 01:40 to 15:00 UTC with longitude and latitude. Second column: results with the TOOCAN algorithm [14], each color representing a label. Third column: results with the MHT method [23], represented as labeled ellipses. Fourth column: results with the SAST method in case B, using the same representation as in Fig. 3. Mixtures are in addition delimited by the convex hull of their atom's positions.

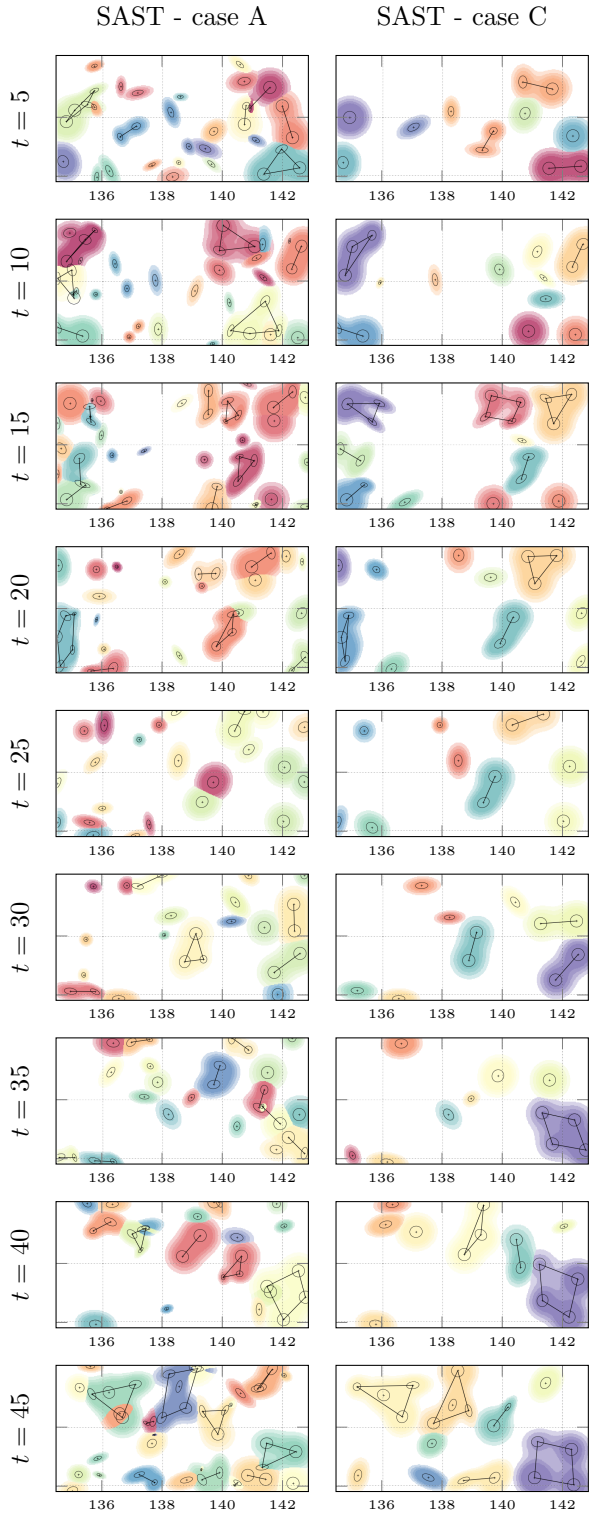


Figure 7 (Cont.): Output of the SAST method for cases A (left) and C (right).

- the TOOCAN algorithm yields very fine details, as its analysis is performed at the pixel level. However, resulting shapes and splitting of large clusters are sometimes questionable especially where the image intensity smoothly varies near the 235 K threshold value. This is one of the drawbacks of the threshold-based methods. Besides, TOOCAN may separate features that could be analyzed as propagation of MCS, for instance at $t = 45$ near (136°E, 1°N).

- unlike TOOCAN, the MHT method describes coarser details in the images. While it does capture the overall displacements, several informations are missing: some objects are undetected for some time period, such as the blue ellipse moving from 142°E at frame 5 to 139°E at frame 30. In consequence, sometimes large parts of the original image were missed as in frame 45.

- unlike TOOCAN and MHT, our method handles full range of temperatures, providing more complete object description. Despite being less precise than TOOCAN regarding the pixel-wise description, the output of SAST explicitly yields shape parameters (position, size, eccentricity, as well as the number of component per MCS) that can be used for the physical processes analysis (beyond the scope of this work).

The three cases studied for SAST provide additional insights:

- In case A, $\lambda = 1.25$ yields the finer details: SAST fits very well the image and 342 labels are found in the sequence of frames. Because C_{thr} is high, splitting occurs frequently and there is some overlap between mixtures.

- Alternatively, case C provides a coarse estimation of the mixtures because $\lambda = 3$. On the other hand, because C_{thr} is low, the merging are frequent. In consequence, 55 labels were found, and we observe that they are well preserved across the time sequence.

- Case B is an intermediate stage between cases A and C. It provides finer details than case C, but also less overlap between mixtures than in case A. In this case, 141 mixtures were found.

More generally, we observed that for a fixed λ, diminishing C_{thr} yields larger mixtures while, when C_{thr} is fixed, mixtures become more isolated when λ is augmented. Hence C_{thr} and λ cannot be chosen independently. Let us finally note that the accurate choice of these parameters must be made by an expert on a large MCS database. In this paper, we focus on the feasibility of the proposed method and its potential w.r.t existing works.

4.2.2. *Quantitative results*

As the three methods tested in the experiments differ in the type of results, quantitatively comparing them must be made carefully. The common ground between the outputs comes down to positions, scales, and labels. The minimal brightness a direct output of our method, and can also be computed in a second step of TOOCAN and MHT, taking the maximum value in $\mathbf{y}_t$ within the masks provided by each label.

Besides, the outputs differ in label number: on the sequence described above, MHT fount 32 labels, TOOCAN found 123 labels and SAST found 55 to 342 labels. In addition, the time scales of the output results vary between methods, and within

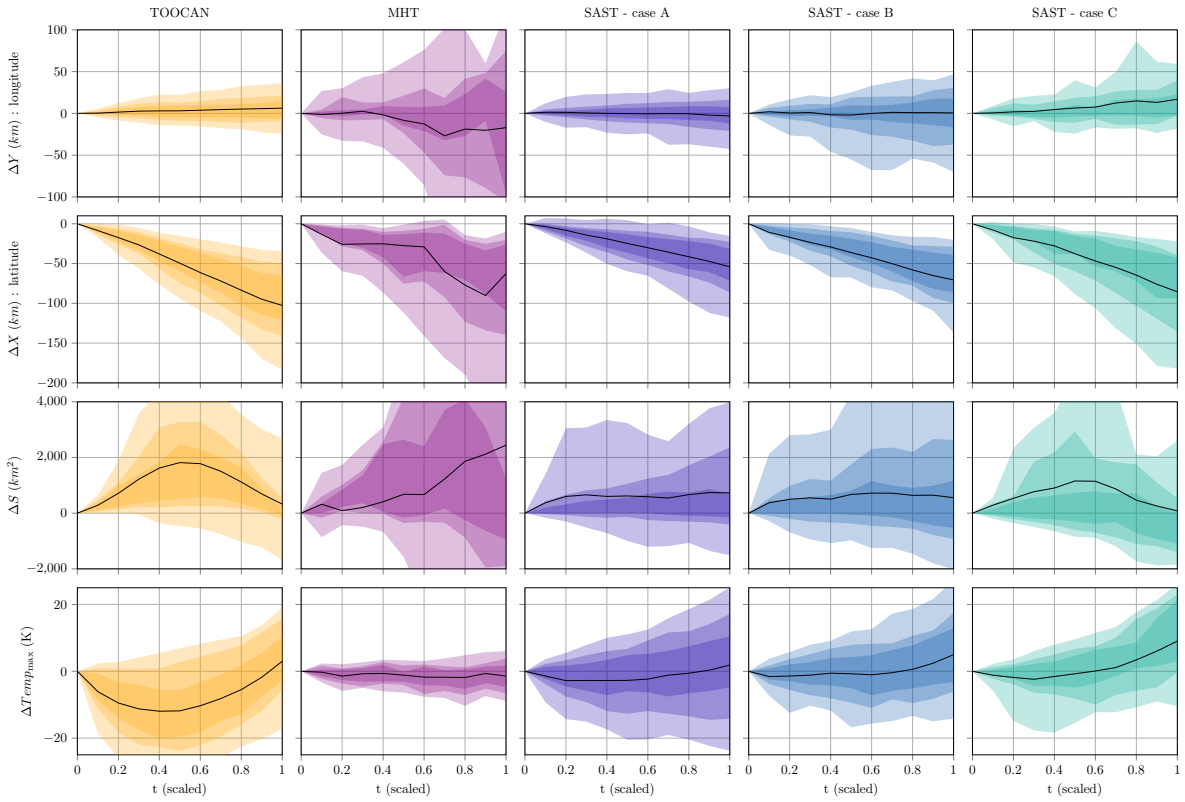


Figure 8: Time-scaled, relative variations of positions (first and second line), surface (third line) and temperature (fourth line) for the three evaluated methods. The average measure is in black, and the shade of color represent deciles number 1–9, 2–8 and 3–7 by order of opacity.

each output. As a consequence, the outputs cannot be paired for comparison. Here, we bypass these concerns by comparing the empirical distributions of the time-scaled values of the objects detected over 3 time frames or more.

In Fig. 8, we report the measure for relative variations for longitude and latitude of the objects, as well as their surface:

- The measurements provided by MHT seem more scattered than for other methods. This is partly due to the low number of labels (32), but also to the method itself, as the 44 remaining (existing over 3 or more frames) labels of SAST, case C induce a lower scattering.
- Regarding the position parameters, the three methods agree on the general displacement. Vertical (latitude) displacement is small and balanced, while horizontal (longitude) displacements are all similarly shifted, depicting well the east-to-west propagation of clouds visible in Fig. 7.
- The size parameters have also the same magnitude, despite the fact that they are related to different objects.

The temperature measurements given in the fourth line of Fig. 8 also give insights on the MCS life cycle. It can be seen that at the beginning of an object's life, on average temperatures first diminish and then rise. This is in accordance with the physics of the MCS, since the updraft first rises at high altitude at low temperatures, and then slowly disaggregate while diminishing their altitude, forming a large anvil and raising its temperature [19].

While the methods agree on the magnitude of the phenomena, their apparent behavior differs notably. Especially, the size measurements highlight an interpretation issue regarding the life cycle of a MCS: when a new updraft rises in the vicinity of an existing MCS, it may either be considered as the propagation of the same MCS, or as a new MCS.

In the first case, many objects have a short lifetime and disappear in a split or merge event, so the MCS size rarely decreases. This case corresponds to the measures provided by MHT, and SAST in cases A and B.

In the second case, there is less short-live objects, but MCS do not survive by in merging of new objects, hence their size decreases and they disappear. This corresponds to the TOOCAN output and with SAST in case C. Note that in this case, the latitude displacement are more important since lifetimes are longer.

Summing up, the SAST method agrees with TOOCAN and MHT on the overall measurements, while providing the ability to adapt to the desired interpretation of MCS lifecycle interpretation. By varying the parameter λ, we can analyze the images and the formation of aggregates at various scales.

4.2.3. Computation time

The implementation of SAST was made in Python 3.5, and will be made available online[2]. This implementation was not

[2]Repository web address: `github.com/courbot/sast`

specifically optimized for speed. On a work station with 12 Intel i7-8700 3.2GHz CPUs and a 12 Gb RAM, the 400×200 px $\times$ 50 frame images depicted in Fig. 7 were processed in 20 min, 10 min, and 6 min for cases A, B and C respectively. This time is mainly used for the large number of executions of the SFW algorithm. We believe that a C/C++ implementation, coupled with GPU computation, should significantly reduce this time. For comparison, our implementation of TOOCAN using Python/Scipy took 19 min to process the same dataset and MHT took 2 min to process the same sequence, mostly because it handles parameter sets instead of image sequences.

5. Conclusion

In this paper, we propose a novel method, named SAST, for the dynamic analysis of MCS in time sequences of images. This method successfully achieves the image fitting through a recent off-the-grid sparse analysis algorithm, and manages as well the apparition, disappearance, splitting and merging events. On real data, we show that the results obtained by SAST are consistent with two other state-of-the-art methods. Besides, we showed that SAST provides more information (position, shape, eccentricity, number of subcomponents) and flexibility with respect to the existing methods.

Several perspectives stem from this work, among which the the development of multi-spectral approaches, the implementation for large-scale applications, and the use of physical models (such as motion divergence) as priors.

Acknowledgements

This project has been supported by the IRIS program of PSL university. To process the Himawari data, this paper benefited from the IPSL mesocenter CICLAD facility hosted by Sorbonne Université, and the data were provided by the AERIS data and service center.

The authors would like to thank Alexandros Makris (Computational Vision and Robotics Laboratory, FORTH, Greece) for kindly providing the code for MHT tracking. They would also like to thank Jean-David Benamou (Inria Paris, France), François-Xavier Vialard (Université Paris-Est Marne-la-Vallée, France), Caroline Muller, Genevieve Seze (LMD, UMR 8539, Paris, France) and Pasquale Selitto (LISA, CNRS UMR 7583, Paris, France) for fruitful discussions regarding MCS tracking.

The authors would also like to thank the anonymous reviewers for their helpful and valuable comments.

References

[1] Adams, R., Bischof, L., 1994. Seeded region growing. IEEE Transactions on pattern analysis and machine intelligence 16, 641–647.

[2] Azais, J.M., De Castro, Y., Gamboa, F., 2015. Spike detection from inaccurate samplings. Applied and Computational Harmonic Analysis 38, 177–195.

[3] Bessho, K., Date, K., Hayashi, M., Ikeda, A., Imai, T., Inoue, H., Kumagai, Y., Miyakawa, T., Murata, H., Ohno, T., et al., 2016. An introduction to himawari-8/9 – japan's new-generation geostationary meteorological satellites. Journal of the Meteorological Society of Japan. Ser. II 94, 151–183.

[4] Blackman, S.S., 2004. Multiple hypothesis tracking for multiple target tracking. IEEE Aerospace and Electronic Systems Magazine 19, 5–18.

[5] Bouniol, D., Roca, R., Fiolleau, T., Poan, D.E., 2016. Macrophysical, Microphysical, and Radiative Properties of Tropical Mesoscale Convective Systems over Their Life Cycle. Journal of Climate 29, 3353–3371. doi:10.1175/JCLI-D-15-0551.1.

[6] Boyd, N., Schiebinger, G., Recht, B., 2017. The alternating descent conditional gradient method for sparse inverse problems. SIAM Journal on Optimization 27, 616–639.

[7] Bredies, K., Pikkarainen, H.K., 2013. Inverse problems in spaces of measures. ESAIM: Control, Optimisation and Calculus of Variations 19, 190–218.

[8] Byrd, R.H., Lu, P., Nocedal, J., Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing 16, 1190–1208.

[9] Candès, E.J., Fernandez-Granda, C., 2014. Towards a mathematical theory of super-resolution. Communications on Pure and Applied Mathematics 67, 906–956.

[10] Chen, S.S., Donoho, D.L., Saunders, M.A., 2001. Atomic decomposition by basis pursuit. SIAM review 43, 129–159.

[11] De Castro, Y., Gamboa, F., 2012. Exact reconstruction using Beurling minimal extrapolation. Journal of Mathematical Analysis and applications 395, 336–354.

[12] Denoyelle, Q., Duval, V., Peyré, G., Soubies, E., 2018. The Sliding Frank-Wolfe Algorithm and its Application to Super-Resolution Microscopy. URL: `https://hal.archives-ouvertes.fr/hal-01921604`. working paper or preprint.

[13] Ekanadham, C., Tranchina, D., Simoncelli, E.P., 2011. Recovery of sparse translation-invariant signals with continuous basis pursuit. IEEE transactions on signal processing 59, 4735–4744.

[14] Fiolleau, T., Roca, R., 2013. An algorithm for the detection and tracking of tropical mesoscale convective systems using infrared images from geostationary satellite. IEEE transactions on Geoscience and Remote Sensing 51, 4302–4315.

[15] Holloway, C.E., Wing, A.A., Bony, S., Muller, C., Masunaga, H., L'Ecuyer, T.S., Turner, D.D., Zuidema, P., 2017. Observing Convective Aggregation. Surveys in Geophysics doi:10.1007/s10712-017-9419-1.

[16] Houze, R.A., 2004. Mesoscale convective systems. Reviews of Geophysics 42. doi:10.1029/2004RG000150.

[17] Huang, X., Hu, C., Huang, X., Chu, Y., Tseng, Y.h., Zhang, G.J., Lin, Y., 2018. A long-term tropical mesoscale convective systems dataset based on a novel objective automatic tracking algorithm. Climate dynamics 51, 3145–3159.

[18] Jaggi, M., 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization., in: ICML (1), pp. 427–435.

[19] Luo, Z., Liu, G.Y., Stephens, G.L., 2008. CloudSat adding new insight into tropical penetrating convection. Geophysical Research Letters 35, L19819–L19819. doi:10.1029/2008GL035330.

[20] Mahler, R.P., 2003. Multitarget Bayes filtering via first-order multitarget moments. IEEE Transactions on Aerospace and Electronic systems 39, 1152–1178.

[21] Mahler, R.P., 2007a. PHD filters of higher order in target number. IEEE Transactions on Aerospace and Electronic Systems 43.

[22] Mahler, R.P., 2007b. Statistical multisource-multitarget information fusion. Artech House, Inc.

[23] Makris, A., Prieur, C., 2014. Bayesian multiple-hypothesis tracking of merging and splitting targets. IEEE Transactions on Geoscience and Remote Sensing 52, 7684–7694.

[24] Mathon, V., Laurent, H., 2001. Life cycle of Sahelian mesoscale convective cloud systems. Quarterly Journal of the Royal Meteorological Society 127, 377–406.

[25] Reid, D., et al., 1979. An algorithm for tracking multiple targets. IEEE transactions on Automatic Control 24, 843–854.

[26] Reuter, S., Vo, B.T., Vo, B.N., Dietmayer, K., 2014. The labeled multi-Bernoulli filter. IEEE Transactions on Signal Processing 62, 3246–3260.

[27] Schmetz, J., Pili, P., Tjemkes, S., Just, D., Kerkmann, J., Rota, S., Ratier, A., 2002. An Introduction to Meteosat Second Generation (MSG). Bulletin of the American Meteorological Society 83, 977–992. doi:10.1175/1520-0477(2002)083¡0977:AITMSG¿2.3.CO;2.

[28] Shanno, D.F., 1970. Conditioning of quasi-Newton methods for function minimization. Mathematics of computation 24, 647–656.

[29] Tang, G., Bhaskar, B.N., Shah, P., Recht, B., 2013. Compressed sensing off the grid. IEEE transactions on information theory 59, 7465–7490.
[30] Thomas, C., Corpetti, T., Mémin, E., 2010. Data assimilation for convective-cell tracking on meteorological image sequences. IEEE Transactions on Geoscience and Remote Sensing 48, 3162–3177.
[31] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) , 267–288.
[32] Tropp, J.A., Gilbert, A.C., 2007. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on information theory 53, 4655–4666.
[33] Vaswani, N., Rathi, Y., Yezzi, A., Tannenbaum, A., 2010. Deform PF-MT: particle filter with mode tracker for tracking nonaffine contour deformations. IEEE Transactions on Image Processing 19, 841–857.
[34] Vo, B.N., Vo, B.T., Pham, N.T., Suter, D., 2010. Joint detection and estimation of multiple objects from image observations. IEEE Transactions on Signal Processing 58, 5129–5141.
[35] Vo, B.T., Vo, B.N., Cantoni, A., 2009. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. IEEE Transactions on Signal Processing 57, 409–423.
[36] Yuan, J., Houze Jr, R.A., 2010. Global variability of mesoscale convective system anvil structure from a-train satellite data. Journal of Climate 23, 5864–5888.
[37] Zhang, Z., Xu, Y., Yang, J., Li, X., Zhang, D., 2015. A survey of sparse representation: algorithms and applications. IEEE access 3, 490–530.

Appendix A. SFW stop criterion and optimality conditions

In this Appendix, we explain informally (ignoring the functional analysis issues raised by the infinite-dimensionality of the space of measures) how the condition $\max_{\mathcal{D}} \eta^{(k)} \leq 1$ in Alg. 1 is equivalent to $\mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}}$ minimizing (5) under non-negativity constraints.

For the sake of simplicity, we drop the dependency on $\mathbf{y}_t$ and λ, and we write that minimization problem as

$$\min_{\mu \in K} C(\mu) \tag{A.1}$$

where K is the closed convex set of non-negative measures, and $C(\mu) := C(\mathbf{y}_t, \mu, \lambda)$. The optimality condition for that problem is that $\nabla C(\mu)$ is normal to K, pointing "outwards", *i.e.*

$$\langle \nabla C(\mu), \nu - \mu \rangle \leq 0, \tag{A.2}$$

for all $\nu \in K$. When μ is of the form $\sum_{n=1}^{k-1} w_n \delta_{\theta_n}$, choosing ν of the form $\alpha\delta_\theta$ and letting $\alpha \to +\infty$, then choosing ν of the form $\mu \pm \varepsilon\delta_{\theta_n}$ for $\varepsilon > 0$ small enough, we see that (A.2) implies (and is in fact equivalent to)

$$\forall \theta \in \mathcal{D}, \quad \nabla C(\mu)(\theta) \leq 0, \tag{A.3}$$

$$\forall n \in \{1, \dots, k-1\}, \quad \nabla C(\mu)(\theta_n) = 0. \tag{A.4}$$

In other words, $\nabla C(\mu)$ is a non-negative continuous function which vanishes at every θ_n.

Now, from (5), one may observe that

$$\nabla C(\mu) = \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\mu - \mathbf{y}_t) + \lambda \tag{A.5}$$

so that $\nabla C(\mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}})(\theta) = \lambda\left(-\eta^{(k)}(\theta) + 1\right)$.

It can be checked that the optimality condition of Step 4 in Alg. 1 implies that $\eta^{(k)}(\boldsymbol{\theta}_{t,n}^{(k-1)}) = 1$ holds for every n at every iteration k. As a result, Eq. (A.4) always holds for $\mu = \mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}}$. Eventually, the condition $\max_{\mathcal{D}} \eta^{(k)} \leq 1$ is equivalent to (A.3), hence to the optimality of $\mu_{\boldsymbol{w}_t^{(k-1)},\boldsymbol{\theta}_t^{(k-1)}}$.

Appendix B. Implementation of the SFW algorithm

In this Appendix, we provide several implementation details for the SFW algorithm detailed in Alg. 1.

Step 1: computation of $\eta^{(k)}$. This certificate is valued on the compact $\mathcal{D}$. In practice, we grid $\mathcal{D}$ and evaluate $\eta^{(k)}$ in each grid node.

Note that computing exactly $\arg\max_{\mathcal{D}} \eta^{(k)}$ is not necessary: it can be proved (see [18, Th. 1]) that the convergence of the algorithm still holds when using an approximation with controlled error. Moreover, in practice, even if the locations of the argmax are coarsely estimated in Step 1, the local descent performed in Step 3 corrects the estimation of $\theta^{(k)}$.

Stop condition: searching for $\max_{\mathcal{D}}(\eta^{(k)})$. This values is approached, because of the previous griding This does not prevent the algorithm to perform correctly, as the sub-grid errors are compensated in the gridless step 4 (see also [7]).

Step 3: weight adjusting. At this step, we allow $\boldsymbol{w}$ to vary but not $\boldsymbol{\theta}$. This amounts to a classical LASSO problem with a fixed dictionary, and we solve it with coordinate descent.

Step 4. Local variations. As suggested in [12], we use a limited-memory bounded BFGS algorithm [8] to restrict the search space to $\mathcal{D}$.