# Air Traffic Representation and Analysis Through Local Covariance

Georges Mykoniatis, Florence Nicol, Stéphane Puechmorel

# Air Traffic Representation and Analysis Through Local Covariance

Georges Mykoniatis†, Florence Nicol‡, Stephane Puechmorel (*)

Ecole Nationale de l'Aviation Civile
Toulouse France
Email: †georges.mykoniatis@enac.fr, ‡florence.nicol@enac.fr, (*)stephane.puechmore@enac.fr

***Abstract*—Air traffic is generally characterized by simple indicators like the number of aircraft flying over a given area or the total distance flown during a time window. As an example, these values may be used for estimating a rough number of air traffic controllers needed in a given control center or for performing economic studies. However, this approach is not adapted to more complex situations such as those encountered in airspace comparison or air traffic controllers training or for adapting dynamically the airspace configurations to the traffic conditions. An innovative representation of the traffic data, relying on a sound theoretical framework, is introduced in this work. It will pave the way to a number of tools dedicated to traffic analysis. Based on an extraction of local covariance, a grid with values in the space of symmetric positive definite matrices is obtained. It can serve as a basis of comparison or be subject to filtering and selection to obtain a digest of a traffic situation suitable for efficient complexity assessment.**

***Keywords*—*Air traffic complexity; spatial data; manifold valued images; covariance function estimation; non-parametric estimation.***

## I. Introduction

This article is an extension of a methodology introduced in [1] for the complexity. It introduces a way of representing the air traffic such that salient features are intrinsically taken into acccount. The theoretical material underlying this new approach to traffic characterization is a Gaussian field model. Namely, the speed vectors of the aircraft at given positions are assumed to be sampled from an unknown underlying Gaussian vector field. Letting the point of observation fixed, the mean value of the field is also the most probable direction of the aircraft, while the variance is an indicator of the local disorder: a fully organized traffic in parallel tracks will have zero covariance, thus a very low complexity while at the other range an isotropic one will be the hardest to predict and control.

Complexity is not the only feature of the traffic that can be captured by a Gaussian field model. Considering once again the mean and covariance at a point, one can infer a geometrical characterization of the area around the reference point. When the covariance matrix is close to singular, it indicates a situation with a very low probability of crossings, and thus a well defined major flow for the aircraft. Computing at several points of the airspace allows the extraction of the most probable routes. When the rank of the covariance matrix increases and eventually become full, a route crossing occurs with high probability. Looking at the neighboring points the geometry of the crossing can be reconstructed. Applied to a whole airspace, the above procedure may be used to produce a graph of flight paths, that is representative of the traffic geometry at a given time. The resulting data structure is a decorated graph, with the vertices corresponding to the crossings along with the covariance information while the edges are described only using the mean value of the field.

In all cases, the most efficient way of gathering the relevant information from the Gaussian field model is to sample the mean and variance of the field on a evenly spaced grid of points covering the airspace of interest. It gives a representation of traffic situations as images whose pixels are covariance matrices, and if needed mean value. This is a work in progress that will ultimately allow the use of deep learning on such pseudo-images in conjunction with a database already analyzed by experts, to produce a complexity metric with low tuning requirements. A by product is the ability to compute distances between traffic situations, allowing for efficient indexing in dedicated databases. The graph structure described above is also a direction of implementation for the traffic data, since several tools exist for dealing with such objects [2].

The rest of the paper is structured as follows. In Section II, the traffic is modeled after a Gaussian random field, whose covariance function is estimated on two dimensional grid. In Section III, tools dedicated to the processing of such grids of symmetric positive definite matrices are introduced. Finally, in Section IV, a conclusion is drawn, introducing the next generation of algorithms able to exploit this novel representation.

## II. State of the art

Key performance indicators (KPI) are of common use in air transportation. However, they are designed mainly to address global aspects of the systems and cannot address problems, where it is mandatory to be able to distinguish between traffic situations based on the structure of the trajectories. As an example, the training of air traffic controllers relies

on carefully selected traffic patterns that are presented to the trainees in an order of increasing perceived complexity. Creating such scenarios is quite a lengthy process, involving hundreds of hours of works by experienced controllers. On the other hand, it is easy to start from real situations, with known flight plans, and to use a traffic simulator to put the trainees in a nearly operational setting. The drawback of this approach is the need to evaluate the traffic patterns in order to assess a complexity value for each of them. It has to be done automatically, to avoid having to resort to human experts.

In a more operational context, nearly the same question arises when trying to find the right number of controllers needed at a give time to take care of the incoming flights in their assigned airspace. Too many controllers induces an extra cost and too few put a high pressure on the operators, with possible detrimental effects on flight safety. Assessing the right level of complexity of the expected traffic may greatly improve over the current state of the art that simply estimates the number of aircraft that will be present. Once again, it is mainly a matter of finding an adequate traffic complexity indicator [3] [4].

In the Air traffic flow management context, it is important to identify areas where the complexity of the traffic does not allow to split the work between two controllers. These areas will then be used as not sequable (Sector Building Blocks) in order to be grouped with others not contrained (Sharable Airspace Modules) for building sectors of control adapted to the traffic situation, This approach is used in the dynamic and evolutive airspace configurations.

A lot of work was dedicated to the issue of air traffic complexity measurement. Unfortunately, no really satisfactory solution exists, as the problem itself is ill posed: depending on the point of view, the complexity may be a concept roughly equivalent to the cognitive workload or, on the contrary, be based on purely structural features, without any reference to the way it will be used. One of the most widely used complexity measures is the dynamic density [5], that combines several potential indicators, like number of maneuvering aircraft, number of level changes, convergence and so on. All these values are used as inputs of a multivariate linear model, or in recent implementations, of a neural network. The tuning of the free parameters of the predictors is made using examples coming from an expertized database of traffic situations. While being quite efficient for assessing complexity values in an operational context, the method has two important drawbacks:

- The tuning procedure requires a sufficient number of expertized samples. A costly experiment involving several air traffic controllers must be set up.
- The indicator is valid only in a specific area of the airspace. Adaptation to other countries or even control centers requires a re-tuning that is almost as complicated as the first one.

The last point is a severe flaw if one wants to use dynamic complexity in the context of air traffic databases, as a world covering has to be obtained first. Even for country sized databases, some geographical tuning has to be added.

Another way to deal with complexity is through purely geometrical indicators [6] [7]. Within this frame, there is no reference to a perceived complexity but only to structural features. An obvious benefit is that the same metric may be used everywhere, without needing a specific tuning. It is also the weak point of the method as the relation with the controllers workload is not direct.

## III. Traffic data

In many situation, the structure of the data available is dictated by the technology used to collect them and only in a few cases the converse will be true. In the field of air traffic control (ATC), not only technological breakthroughs impact the data available but they also change the way operators are perceiving and using them in their work. It is thus mandatory to understand the operational practices of air traffic controllers (ATCOs) in order to find a sound model for the traffic data.

### *A. A brief introduction to air traffic control*

In the early days of ATC, separation between aircraft was ensured using procedures: pilots were bound to follow their filled flight plan, that is be at a given report position at a given time. Assuming a constant speed. a simple linear interpolation between two report points allowed then controllers to estimate the intermediate positions at any time and check that no pairs of aircraft come to close to each other. In oceanic areas, this mean of controlling the traffic is still in use, although the accuracy of on-board inertial measurement units (IMUs) makes the estimation of aircraft positions much more accurate. A major breakthrough was the introduction of the radar in civil aviation: ATC switched then from procedure based to surveillance based control. It is the current ATC framework for which ATCOs are trained. Surveillance based control makes it natural to represent the air traffic in an given airspace as a set of plots, that are dated positions and speed samples, since it is the way radars collect the information. Aircraft are identified by a four digit code, sent by the on-board transponder as a response to the radar beam. It is non-unique, assigned by the controller in charge of the flight within a control sector and may change along the aircraft trajectory. Correlation with the flight plan is thus not automatic and must be done prior to any further processing.

Finally, future air systems are undergoing a paradigm shift with the introduction of trajectory based operations (TBO), that are going to succeed the current surveillance based operations. It contrast with the current air traffic management (ATM) system, the trajectory information will be the primary data available to ATCOs. This paradigm shift is planned to occur within 20 years, so that the model chosen to represent the traffic has to be TBO compliant.

When dealing with a traffic situation, ATCOs have to forecast the aircraft positions some minutes ahead in order

to detect encounters below separation norms (such a situation is termed as a conflict in ATC) and give appropriate headings to avoid them. Only in rare occasions altitude changes are needed. Conflict resolution by velocity increase or decrease it not uncommon in the US airspace where traffic is organized in miles in trails, but almost never used in Europe. In all cases, a very influential factor on complexity is the organization level of the traffic, that impacts also the ability to forecast its evolution through time. It is intrinsic to the traffic situation and does not depend on a particular airspace structure. Please note that even in the context of future TBO, it remains a valid indicator as near random situation is highly sensitive to uncertainties and thus requires large enough safety margins, even for an automated trajectory planner.

### B. Data sources

Air traffic data come from various sources with different confidence levels. Highest quality is reached by surveillance data gathered by air services navigation providers (ANSPs), which are generally not publicly available. In the mid-range, ADSB data is easy to obtain from both commercial or free providers. Low to medium accuracy data can be generated by traffic simulators, using filled or estimated flight plans. The present work makes use of raw radar data collected by the French civil aviation authority, converted from the binary format ASTERIX [8] to a textual form. Each day of traffic is saved in a compressed file, for a total duration of one month. ADSB data was not considered in the study, but will be used in ongoing work to address wider airspaces.

Due to the physics underlying the radar measurements and the low level preprocessing algorithms, some errors are present in the original dataset:

- Noisy observations. They appear as small oscillations around a mean trajectory and can be attenuated by using either a Kalman filter or a local linear model. However, the covariance estimation procedure that will be described later tends to smooth out the noise, so that its effect is very limited, and is too low to be noticed in practice.
- Bad correlation between trajectories and measurements. In such a case, there is a mixing of two or more flight paths, resulting in several jumps across them. This kind of error has no impact on the computation of local covariance matrices since the trajectory information is not used.

Finally, some atypical flights are related to radio electric navigation means calibration, or to training or experiments. They are easily identified and can be discarded automatically from the dataset. It is the only preprocessing stage that was applied to the raw dataset to get the working dataset.

### C. Database setup

Raw radar data are organized as records separated by new-line delimiters in compressed text files. Each record comprises the following fields, separated by the delimiter '—':

- A Unique identifier of type int32. It identifies the flight and is assigned by a low-level tracking algorithm.
- A time stamp, formatted as a posix date time.
- Three real numbers representing respectively the x,y position in the so-called CAUTRA coordinate system and the altitude in feet.
- Three real numbers representing respectively the velocity in knots, the heading in degrees and the vertical speed in feet per minute.
- The name of the flight, as displayed on the controller's screen.
- The aircraft type.
- The departure and arrival airport codes.

The last four fields are uniquely associated with a flight identifier. The coordinate system in which the position is expressed is specific to the French ATC system: horizontal positions are in nautical miles, obtained by applying a stereographic projection centered at 0 deg. longitude, 45 deg. latitude. Altitude is expressed in flight levels.

A conversion to a more generic unit system must be done prior to storage in a database. The following transformations are thus applied:

- Horizontal positions are converted to longitude and latitude in the WGS84 ellipsoid and expressed in degrees. The transformation is done using the open source proj4 software [9].
- Vertical position is converted to meters.
- All velocities are converted to meters per second.

The sampling rate is 4 seconds, resulting in approximately 6-7 million records per day, yielding a grand total of roughly 200 million records for the complete dataset. Based on that, the expected dataset size for a year of worldly traffic will be in the order of $310^{10}$ records. Since the final goal of this work is to release a tool able to characterize the complexity of the traffic in any area and for at least ten years, the database must accommodate $310^{11}$ records. The storage needed for tha raw radar data can be thus estimated roughly to 15T-20To, which is a very tractable value for modern hardware.

Given the above remarks, the main point that remains is to choose between a SQL (Postgresql) or a document-oriented (MongoDB) database. For raw radar data coming from the French civil aviation authority, both solutions are fine. However, keeping in mind that different sources will be added in the future, the MongoDB implementation was selected as it offers an extra degree of flexibility. Please note that this does not preclude the use of a SQL database for serving specific queries: in such a case, the document-oriented database acts as data pool feeding the SQL one.

A dedicated tool has been developed to automatically read incoming raw radar files from a directory and store the samples into a mongoDB database after unit and coordinate system conversion . A complementary tool receiving online ADSB traffic from the Opensky network [10] is also available but not yet in operation.

## IV. TRAFFIC MODELING

### A. A Gaussian field approach

An mentioned above, all samples are assumed to be plots $(t, x, v)$ where $t$ is the sampling time, $x \in \mathbb{R}^3$ is the sample position and $v \in \mathbb{R}^3$ the aircraft speed. As a consequence, a traffic dataset is a sequence $(t_i, x_i, v_i)_{i=1\dots N}$ of plots collected in a given time interval and spatial area. Please note that plots do not incorporate trajectory information, so that different flight patterns may generate exactly the same dataset. However, if the sampling rate is high enough, and if $(t_0, x_0, v_0)$ and $(t_1, x_1, v_1)$ are successive samples belonging to the same trajectory, the distance between $x_1$ and $x_0 + (t_1 - t_0)v_0$ is small, so that no confusion between trajectories can be made unless a true encounter occurs, which is of course unlikely to occur on real traffic. This remark also suggests a representation of the traffic as a dynamical system, in which the aircraft trajectories are integral lines of a time varying vector field. To put it in a more formal way, an explicit association between the plots and the trajectories must be assumes, namely the observations come as a sequence of labeled plots $(t_{ij}, x_{ij}, v_{ij})_{i=1\dots M, j=1\dots N_i}$, where $i$ is the trajectory number and $j$ the index of the plot on the trajectory. Please note that the number of points sampled on each trajectory is generally varying. A continuous model of the traffic is then defined as a smooth mapping $X\colon [t_0, t_1] \times \Omega \to \mathbb{R}^3$ such that its integral lines:

$$\begin{aligned} &\gamma_i\colon t \in [t_{i1}, t_{iN_i}] \mapsto \\ &x_{i1} + \int_{t_{i1}}^{t} X\left(u, X(u, \gamma_i(u))\, du,\ i = 1 \dots M\right. \end{aligned} \tag{1}$$

satisfy the interpolation condition:

$$\forall i = 1 \dots M, \forall j = 1 \dots N_i, \gamma_i(t_{ij}) = x_{ij}, \gamma_i'(t_{ij}) = v_{ij} \tag{2}$$

or equivalently:

$$\forall i = 1 \dots M, \forall j = 1 \dots N_i, \gamma_i(t_{ij}) = x_{ij}, X(t_{ij}, x_{ij}) = v_{ij} \tag{3}$$

As is, the problem is ill-posed as infinitely many such $X$ may be found. Adding a smoothness penalty term allows a unique solution to be found. A possible choice is:

$$E(X) = \int_{t_0}^{t_1} \left( \int_{\Omega} \lambda \| \frac{\partial X(t,x)}{\partial t} \|^2 + \| \Delta_x X(t,x) \|^2 dx \right) dt \tag{4}$$

where the partial derivative in time accounts for time variation and the Laplacian quantifies for the departure from an organized solution where the speed value at one point is the mean value of its neighbors speed.$\lambda > 0$ is a free parameter that tunes the relative importance of both terms. In the sequel, it will be set to 1, but using an arbitrary value is just a matter of scaling the time coordinate. The original problem 3 can be reformulated as:

$$\begin{cases} \text{argmin } E(X) \\ \gamma_i(t_{ij}) = x_{ij}, X(t_{ij}, x_{ij}) = v_{ij}, \\ i = 1 \dots M, j = 1 \dots N_i \end{cases} \tag{5}$$

The penalized problem gives rise to a solution that is known as a spline interpolation. In its special instance 1, the solution can be obtained in closed form.

**Proposition 1.** *1 The problem has a unique solution $X$, that can be expressed for the $t \neq t_{ij}, i = 1 \dots M, j = \dots N_i$ as a sum:*

$$\begin{aligned} X(t,x) = &\sum_{i=1}^{M} \sum_{j=1}^{N_i} a_{ij} \frac{1}{\sqrt{|t - t_{ij}|}} erfc\left( \frac{\|x - x_{ij}\|^2}{|t - t_{ij}|} \right) \\ &+ btx + cx + dt + e \end{aligned} \tag{6}$$

with erfc the complementary error function.

*Proof.* The complete proof is quite long, a simplified version is given here. The first step is to reformulate 1, using the extra assumption that the sought after field must be vanishing at infinity, both in $x$ and $t$. Using an integration by parts, the criterion $E(X)$ given in 4 can rewritten as:

$$E(X) = \int_{\mathbb{R}} \int_{\mathbb{R}^3} \langle LX(t,x), X(t,x) \rangle \, dxdt$$

with $L$ the differential operator:

$$L = -\partial_{tt} + \Delta_x^2$$

The space of fields $X$ such that $LX$ is defined in weak sense and $E(X) < +\infty$ is a Sobolev space $\mathcal{H}$, with inner product:

$$\langle X, Y \rangle_{\mathcal{H}} = \int_{\mathbb{R}} \int_{\mathbb{R}^3} \langle LX(t,x), Y(t,x) \rangle \, dxdt$$

Let $G$ be the Green's function of $L$. The interpolation conditions can be written as:

$$\begin{aligned} &\int_{\mathbb{R}} \int_{\mathbb{R}^3} \langle LG(t_{ij}, x_{ij}, t, x), X(t,x) \rangle \, dxdt = v_{ij} \\ &i = 1 \dots M, j = 1 \dots N_i \end{aligned}$$

They define an affine subspace $\mathcal{A}$ of $\mathcal{H}$, so that the field $X$ solving 1 is the orthogonal projection of the origin onto $\mathcal{A}$. Since the subspace of $\mathcal{H}$ orthogonal to $\mathcal{A}$ is generated by the $G(t_{ij}, x_{ij}, t, x), i = 1 \dots M, j = 1 \dots N_i$, the optimal field as the form:

$$X(t,x) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} a_{ij} G(t_{ij}, x_{ij}, t, x) + K(t,x)$$

where $K$ is an element of the kernel of $L$, determined by the boundary conditions. The second step of the proof is to find the Green's function of $L$, that can be done using a partial fourier transform in $x$, in the sense of distributions. After some computations, one can show that it is proportional to:

$$\frac{1}{\sqrt{|t - t_{ij}|}} \text{erfc}\left( \frac{\|x - x_{ij}\|^2}{|t - t_{ij}|} \right)$$

thus completing the proof. □

Looking at the expression of $X$, it appears to be undefined at the sampling times $t_{ij}$, even if it remains integrable. In fact, the field involves delta distributions at the sample times.This is not

an uncommon situation, as it is encountered when computing an harmonic field generated by a finite set of point charges. Nevertheless, it is a severe flaw for the air traffic application and needs to be corrected. A first approach is to replace the exact interpolation conditions by a a mean value condition. A possible choice is:

$$\int_{\mathbb{R}^3} X(t_{ij},x)\frac{1}{(2\pi)^{3/2}\sigma}\exp-\frac{\|x-x_{ij}\|^2}{2\sigma^2}dx = v_{ij}$$

yielding a simple expression for $X$:

$$X(t,x) = \sum_{i=1}^{M}\sum_{j=1}^{N_i} a_{ij}\frac{1}{\sqrt{\sigma+|t-t_{ij}|}}\text{erfc}\left(\frac{\|x-x_{ij}\|^2}{\sigma+|t-t_{ij}|}\right) + btx + cx + dt + e$$

The parameter $\sigma$ must be tuned according to the typical spatial scale of variation of the field, in the order of $10NM$ for air traffic.

Another way of looking at the same problem is to assume that the sampling positions and times are no completely deterministic, but come from an underlying stochastic process. Randomness in the spatial component, may represent measurement errors, or uncertainties if the position is only estimated. In the time component, it is mainly related to delays (positive or negative) due to some unknown parameters, like the wind, that affect the ground speed of the aircraft. In a more formal way, the sampling process at point $(t,x)$ is assumed to be of the form $(t,x)+U$ with $U$ a random variable taking its values in $R\times\mathbb{R}^3$ and having a density $p(t,x)$. The observed field at $(t,x)$ becomes now a random variable of the form $X((t,x)+U)$. Assuming a deterministic field of the form 6 and looking only at the part corresponding to the Green's functions, the expectation of $X$ at position $(t,x)$ is given by:

$$\sum_{i=1}^{M}\sum_{j=1}^{N_i} a_{ij}\int_{\mathbb{R}}\int_{\mathbb{R}^3} G(t_{ij}+\eta,x_{ij}+\xi,t,x)p(\eta,\xi)d\eta d\xi$$

which is non longer degenerate. Letting the expected values be the new constraints of the problem yields an expression with Green's function $G$ replaced by the kernel:

$$\tilde{G}(t,x,u,v) = \int_{\mathbb{R}}\int_{\mathbb{R}^3} G(t+\eta,x+\xi,u,v)p(\eta,\xi)d\eta d\xi$$

The two approaches are mainly equivalent, the second one introducing a more probabilistic view.

Finding the coefficients $a_{ij}, a, b, c, d, e$ is done by solving a linear system, and represents quite a huge amount of computational power. Furthermore, the probabilistic or mean value approaches do not allow a simple characterization of the traffic. However, the optimality of the reconstructed field with respect to a smoothness criterion is an extremely valuable property, that worth to be preserved.

The main idea underlying the new representation that will be introduced now is the fact that kernel functions, like $\tilde{G}$, occur in the field expression $X(t,x)$ as weighting factors, while the coefficients $a_{ij}$ are linear functions of the observations, independent of $t,x$. For the sake of simplicity, the explicit time dependence will be dropped when possible in the sequel, so that sampling positions will be in $\mathbb{R}^4$. Doing so, a double indexing in $i$ and $j$ is no longer needed, and a single one will be used.

Collected samples may be viewed as realizations of an underlying stochastic process $X$ with domain $\mathbb{R}^4$ and taking its values in $\mathbb{R}^3$. Such a process is called a Gaussian vector field when for any collection of points $(x_1,\ldots,x_p)$, the joint distribution of the random variables $(X(x_1),\ldots,X(x_p))$ is Gaussian. Such a process is characterized by its mean and covariance functions:

$$\mu(x) = E[X(x)] \quad (7)$$

$$C(x,y) = E\left[(X(x)-\mu(x))(X(y)-\mu(y))^t\right] \quad (8)$$

In practice, $\mu$ and $C$ must be estimated from a dataset of couples $(x_i,v_i)_{i=1\ldots N}$ where $v_i$ is the observed vector value at position $x_i$. Available methods fall into two categories: parametric and non parametric. In the parametric approach, $\mu$ and $C$ are approximated by members of a family of functions depending on a finite number of free parameters that are tuned to best match the observations. Considering the above discussion about optimal field reconstruction, its is natural to approach $\mu$ by an expansion of the form:

$$\mu(x) = \sum_{i=1}^{M}\sum_{j=1}^{N_i} a_{ij}\frac{1}{\sqrt{\sigma+|x_t-t_{ij}|}}\text{erfc}\left(\frac{\|x_s-x_{ij}\|^2}{\sigma+|x_t-t_{ij}|}\right) + V(x)$$

where $x_t$ (resp. $x_s$) denotes the time (resp. spatial) component of $x$, and $V$ accounts for the bilinear term. The covariance function $C$ may be represented pretty much the same way, using tensor products of $\tilde{G}$ kernels. Due to the covariance part, estimating the free coefficients is even more expensive than the deterministic problem and is intractable in practical applications.

In the non parametric approach, a different methodology is used: the samples themselves act as coefficients of an expansion involving the kernel functions $\tilde{G}$. Apart from the obvious benefit of avoiding a costly least square procedure, the complementary error function appearing in the kernel expression decays very fast at infinity: in practice $\tilde{G}$ can be assumed to be compactly supported, so that evaluating an approximate mean and covariance at a given location requires far less terms that the number of samples. Due to its simplicity and the previous property, a non parametric estimation was selected to process the traffic.

### *B. Mean and covariance matrix estimation*

A kernel estimator of the covariance function $C$ of a stationary stochastic process $X$ can be found in [11]. Using a straightforward extension, a kernel estimator for the correlation function of a locally stationary random field is given in [12]. Finally, a weighed maximum likelihood approach is taken in [13], for computing at any location $x$ the mean $\mu(x)$

and variance $C(x,x) = \Sigma(x)$ of a Gaussian random field. This last work can easily be generalized to yield an estimate for the covariance function, under normality assumption for the random field $X$. Given a dataset $(x_i, v_i)_{i=1\ldots N}$, where the sampling positions $x_i$ are assumed to be distinct, the weighted joint log likelihood of the couples $(x_i, v_i), (x_j, v_j), j \neq i$ at locations $x, y$ is given, for a fixed kernel bandwidth $h$, by:

$$
\begin{aligned}
&L(x,y) = \\
&-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{ij}^t \Sigma^{-1}(x,y) V_{ij} K_h\left(x_i - x\right) K_h\left(x_j - y\right) \\
&+\frac{1}{2}\log\left(\left|\Sigma^{-1}\right|\right)\left(\sum_{i=1}^{N}\sum_{j=1^N} K_h\left(x_i - x\right) K_h\left(x_j - y\right)\right) \\
&+A
\end{aligned}
$$

where:

$$V_{ij} = (v_i - m(x), v_j - m(y))$$

$m(x)$ is the mean of $X(x)$ and $\Sigma(x,y)$ is the variance matrix of the Gaussian vector $(X(x), X(y))$. The term $A$ accounts for the $\log$ of the normalizing constant occurring in the expression of the multidimensional normal distribution and will play no role in the sequel. The kernel function $K_h$, with $h > 0$ its bandwidth parameter, is obtained from $\tilde{G}$ and is defined to be:

$$K_h(x) = \frac{1}{\sqrt{h + |x_t|}}\operatorname{erfc}\left(\frac{\|x_s\|^2}{h + |x_t|}\right)$$

The differential of the log likelihood with respect to the mean value can be computed as:

$$\sum_{i,j} \operatorname{tr}\left(V_{ij}^t \Sigma^{-1}(x,y)\right) K_h\left(x_i - x\right) K_h\left(x_j - y\right) \tag{9}$$

The first order optimality condition yields for the non-parametric estimate for the mean function:

$$\hat{m}(x) = \frac{\sum_{i=1}^{N} v_i K_h\left(x - x_i\right)}{\sum_{i=1}^{N} K_h\left(x - x_i\right)} \tag{10}$$

A similar derivation can be made to obtain the differential with respect to the $\Sigma$ matrix, using the two identities below:

$$d\Sigma^{-1} = -\Sigma^{-1} d\Sigma \Sigma^{-1} \tag{11}$$

$$d\log\left(\left|\Sigma^{-1}\right|\right) = -\operatorname{tr} d\Sigma\Sigma^{-1} \tag{12}$$

The non-parametric estimator for $\Sigma(x,y)$ is then:

$$\hat{\Sigma}(x,y) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{C}_{ij}(x,y) K_h\left(x_i - x\right) K_h\left(x_j - y\right)}{\sum_{i=1}^{N}\sum_{j=1}^{N} K_h\left(x_i - x\right) K_h\left(x_j - y\right)} \tag{13}$$

with:

$$\hat{C}_{ij}(x,y) = \begin{pmatrix} v_i - \hat{m}(x) \\ v_j - \hat{m}(y) \end{pmatrix}\begin{pmatrix} v_i - \hat{m}(x) \\ v_j - \hat{m}(y) \end{pmatrix}^t$$

Using the definition 10 of $\hat{m}$, it appears that $\hat{\Sigma}(x,y)$ is block diagonal:

$$\begin{pmatrix} \hat{\Sigma(x)} & 0 \\ 0 & \hat{\Sigma(y)} \end{pmatrix} \tag{14}$$

with:

$$\Sigma(x) = \frac{\sum_{i=1}^{N} \hat{C}_{ii}(x) K_h\left(x_i - x\right)}{\sum_{i=1}^{N} K_h\left(x_i - x\right)} \tag{15}$$

This estimator is similar to the one in [12], and is of Nadaraya-Watson [14] type. It enjoys asymptotic normality. The reason for the vanishing of the off diagonal blocks is a consequence of the special shape of the kernel that implicitly approximates the joint distribution of $X(x)$ and $X(y)$ by product laws.

### C. Computation of the mean and covariance functions

To simplify the computation, the altitude was not explicitly used except for selecting the flight levels of interest. All the samples will thus have a $z$ component equal to the mean altitude. The same was done for the time. In order to allow further treatments, mean and covariance functions will be evaluated only at points of an evenly spaced two dimensional grid whose points are located at coordinates:

$$
\begin{aligned}
&p_{nm} = (x_0 + n\delta_x, y_0 + m\delta_y) \\
&n \in \{-L, \ldots, L\},\, m \in \{-M, \ldots, M\}
\end{aligned}
$$

where $\delta_x, \delta_y$ are respective step sizes along $x$ and $y$ axis. In the expressions 10,15 of the mean and covariance functions evaluated at grid point $p_{nm}$, the kernel appears as $K_h\left(x_i - p_{nm}\right)$. If the grid is fine enough, one can approximate it by $K_h\left(p_{kl} - p_{nm}\right)$ where $p_{kl}$ is the grid point closest to $x_i$.

Values depends only on the difference between the grid points indices and are thus independent on the location $p_{kl}$. Furthermore, since $K_h$ is assumed to have compact support, $K_h\left(p_{kl} - p_{nm}\right)$ will vanish when indices differences exceed a given threshold. Gathering things together, all non-zero values of the kernel can be tabulated on a grid of size $(2P+1)\times(2Q+1)$ if the support of the kernel $K_h$ is contained in the square $[-P\delta_x, P\delta_x] \times [-Q\delta_y, Q\delta_y]$:

$$K_h(i,j) = K_h\left(i\delta_x, j\delta_y\right) \tag{16}$$

$$n \in \{-P, \ldots, P\}, m \in \{-Q, \ldots, Q\} \tag{17}$$

All the entries in the equation 16 can be scaled so that they sum to 1: this saves the division by the sum of kernel values. Simultaneous evaluation of the mean at all grid points can then be made in an efficient manner using Algorithm 1. Once the mean has been computed, the covariance is estimated the same way (see Algorithm 2).

When the grid is not fine enough to replace the true sample location by a grid point, a trick based on bilinear interpolation can be used. Using again the equation 16 and the closest grid point $p_{k_1 l_1}$ to $x_i$, the true sample position $x_i$ will be located within a cell as indicated in Figure 1. The kernel value can be

**Algorithm 1** Mean kernel estimate

**for** $i \leftarrow 0, 2L; j \leftarrow 0, 2 * M$ **do**
  $m(i,j) \leftarrow 0$
**end for**
**for** $k \leftarrow 0, N-1$ **do**
  $(k,l) \leftarrow$ ClosestGridPoint$(x_i)$
  **for** $i \leftarrow -P, P; j \leftarrow -Q, Q$ **do**
    **if** $k+i \geq 0 \wedge k+i \leq 2L$ **then**
      **if** $l+j \geq 0 \wedge l+j \leq 2M$ **then**
        $m(k+i, l+j) \leftarrow m(k+i, l+j) + K_h(i,j)v_i/N$
      **end if**
    **end if**
  **end for**
**end for**

**Algorithm 2** Covariance kernel estimate

**for** $i \leftarrow 0, 2L; j \leftarrow 0, 2 * M$ **do**
  $C(i,j) \leftarrow 0$
**end for**
**for** $k \leftarrow 0, N-1$ **do**
  $(k,l) \leftarrow$ ClosestGridPoint$(x_i)$
  **for** $i \leftarrow -P, P; j \leftarrow -Q, Q$ **do**
    **if** $k+i \geq 0 \wedge k+i \leq 2L$ **then**
      **if** $l+j \geq 0 \wedge l+j \leq 2M$ **then**
        $A \leftarrow (v_i - m(k,l))(v_i - m(k,l))^t$
        $C(k+i, l+j) \leftarrow C(k+i, l+j) + K_h(i,j).A/N$
      **end if**
    **end if**
  **end for**
**end for**

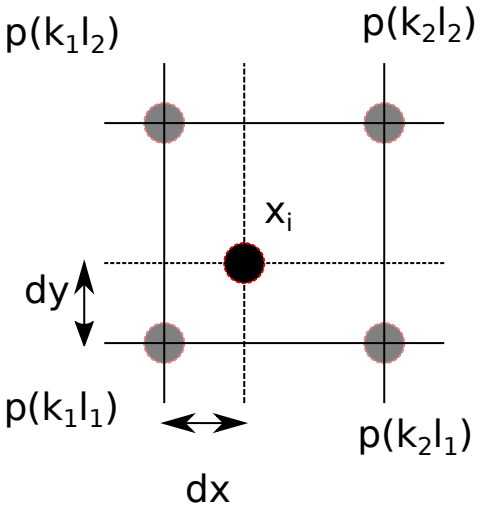


Fig. 1. Bilinear interpolation

approximated as:

$$K_h(k_1,l_1) + \frac{dx}{\delta_x}a + \frac{dy}{\delta_y}b + \frac{dx}{\delta_x}\frac{dy}{\delta_y}c \tag{18}$$

with:

$$a = K_h(k_2,l_1) - K_h(k_1,l1)$$
$$b = K_h(k_1,l_2) - K_h(k_1,l1)$$
$$c = K_h(k_2,l_2) + K_h(k_1,l1) - K_h(k_2,l_1) - K_h(k_1,l_2)$$

Gathering terms by tabulated values yields a kernel value:

$$K_h(k_1,l_1)\,(1 - s_x - s_y + s_x s_y) \tag{19}$$
$$+ K_h(k_2,l_1)\,(s_x - s_x s_y) \tag{20}$$
$$+ K_h(k_1,l_2)\,(s_y - s_x s_y) \tag{21}$$
$$+ K_h(k_2,l_2) s_x s_y \tag{22}$$

where:

$$sx = \frac{dx}{\delta_x},\ sy = \frac{dy}{\delta_y}$$

It is thus possible to compute the mean and covariance functions on a coarser grid using Algorithms 1 and 2 by applying them on the four locations $(k_1,l_1), (k_2,l_1), (k_1,l_2), (k_2,l_2)$, with an observed value multiplied by their respective coefficients $(1 - s_x - s_y + s_x s_y)$, $(s_x - s_x s_y)$,$(s_y - s_x s_y)$,$K_h(k_2,l_2) s_x s_y$.

The overall complexity of the algorithm is linear in the number of grid points and in the number of samples. It is similar to filtering an image and can be implemented the same way on modern Graphics Processing Units (GPU). Please note also that for kernels with large supports, a fast Fourier transform may be used at the expense of a slight increase in the complexity order that will be balance by the constant term due to the support size.

## V. Processing tools

The preceding phase allows the computation of a traffic pattern digest as a two dimensional grid of Symmetric Positive Definite (SPD) matrices. It may be used as is for building an index in a database, using the same procedure as for images. However, the geometry underlying the space of $2 \times 2$ positive definite matrices is not euclidean, but hyperbolic. The proposed index is an adaptation of images distances, using hyperbolic geometry.

### *A. The Riemannian structure of symmetric positive definite matrices*

The purpose of this part is to introduce at a basic level the tools used to build the index. Results are given without proofs, the interested reader may refer to [15] for a more in-depth exposition.

**Proposition 2.** *The space of $n \times n$ SPD matrices, denoted by $SPD(n)$, may be endowed with a Riemannian manifold structure with metric at point $A$ given by the differential:*

$$ds^2 = \mathrm{tr}\left(A^{-1}dA\right) \tag{23}$$

**Proposition 3.** *Let $A, B$ be SPD matrices. It exists a unique minimizing geodesic joining them in $SPD(n)$. It is given in parametrized form by:*

$$\gamma : t \in [0,1] \mapsto A^{1/2}\left(\exp t \log\left(A^{-1/2}BA^{-1/2}\right)\right)A^{1/2} \tag{24}$$

Proposition 3 yields the geodesic distance between any two matrices $A, B$ from $SPD(n)$ as $d(A,B) = \sqrt{\mathrm{tr}\log^2(A^{-1}B}$. It can be expressed as $d(A,B) = \sqrt{\sum_{i=1}^{n}\log^2\lambda_i}$ with $\lambda_i, i = 1\dots n$ the eigenvalues of $A^{-1}B$.

The geodesic distance between matrices from $SPD(3)$ may be used to compute a distance between grids produced by the traffic processing phase in a very simple way, as indicated in Algorithm 3.

**Algorithm 3** Distance between grids

1: $A, B$ are $P \times Q$ grids of $SPD(3)$ matrices.
2: $dsq = 0$
3: **for** $i \leftarrow 0, P-1; j \leftarrow 0, Q-1$ **do**
4: $\quad dsq \leftarrow dsq + \mathrm{tr}\log^2(A(i,j)^{-1}B(i,j)$
5: **end for**
6: $d(A,B) = \sqrt{dsq}$

Please note that this distance is based on a point-wise comparison and is very similar to the $L^2$ distance used for images. It has a higher cost of evaluation due to the distance computation in $SPD(3)$ that involves an matrix inverse, product and logarithm. However, grid distance computation is easily parallelized on modern graphics hardware since it involves independent operations on small matrices. As an example, computing the distance between two grids of size $100 \times 100$ on a TitanX pascal card from Nvidia takes around $100\mu$s.

### *B. Grid filtering*

In the traffic processing phase, grids have sizes ranging from $100 \times 100$ to $300 \times 300$. Due to the processing cost incurred by the $SPD(3)$ setting, it is advisable in many cases, and especially if one wants to use the grids as index in a traffic database, to reduce the size of grids to more tractable dimensions, say $10\times10$ to $50\times50$. This has to be done without wiping out the salients features of the traffic captured by the original grid. In the spirit of what is done in the first layers of an image processing deep network, it is proposed to apply in sequence a filtering and a selection process on the original grid.

**Definition 1.** *Let $A_i, i = 1\dots n$ be a sequence of elements of $SPD(n)$, $w_1, \dots w_n$ be a sequence of real numbers and $B$ be an element of $SPD(n)$. The log-euclidean weighted combination (LWC) at $B$ of the $(A_i)_{i=1\dots n}$ with weights $(w_i)_{i=1\dots n}$ is the matrix:*

$$B^{1/2}\exp\left(\sum_{i=1}^{n} w_i \log\left(B^{-1/2}A_iB^{-1/2}\right)\right)B^{1/2} \tag{25}$$

The LWC may be used to compute a filtered version of a grid using the same procedure as for an image. The process is given in Algorithm 4 that yields the filtered grid as $B$.

**Algorithm 4** Grid filtering

1: $A$ is a $P \times Q$ grid of $SPD(2)$ matrices.
2: $w_i, i = 1\dots9$ is a sequence of real numbers
3: **for** $i \leftarrow 0, P-1; j \leftarrow 0, Q-1$ **do**
4: $\quad (C_1, \dots C_9)$ are the adjacent cells to $A(i,j)$ and itself.
5: $\quad B(i,j) \leftarrow LWC(C_1, \dots, C_9)$ with weight $w_i, i = 1\dots9$ at $A(i,j)$.
6: **end for**

The filtering process on $SPD(3)$ grids behaves roughly like in image processing: when the weights are real numbers in the interval $[0,1]$ that sum to 1, then a weighted mean is produced. It tends to smooth out the grid, making spatially close matrices more similar. On the opposite, when weights sum to 0, the equivalent of a high pass filter is produced, that emphases sharp variations. Please note that the size of the grids after filtering is unaltered.

The second processing phase is simplification to reduce grid size. The main idea is to replace a block of grid cells by a single one using a digest. An obvious approach is to replace a block by its mean, that can be obtained from LWC by using equal positive weights $1/n$ if $n$ is the number of cells in the block. A major drawback is that the important information tends to be lost, with matrices going close to multiples of the identity in many cases. Another way of dealing with the problem is to introduce an order on $SPD(3)$ and to select the largest (resp. lowest) element in the block. This procedure has two benefits:

- The selected matrix is an element of the original grid.
- As in deep learning networks, it will select the most representative elements.

After some experiments on simulated matrix images, the order chosen is a lexicographic one, the first comparison being made on the determinant of the matrices and the second on the trace. After the selection phase, the size of the grid is reduced by the ratio of the number of elements considered in a block. In the current implementation, it is $3\times3$, thus shrinking the grid by a factor 3 in each dimension. The filtering/selection phases may be chained in order to get smaller grids. As for the distance computation, it is quite easy to implement the process on a GPU, all operations being independent.

## VI. Results

The covariance representation of the traffic has served as an input to a clustering algorithm. It gives a simple mean of

evaluating its pertinence by identifying hot spots of complexity within a given airspace. Some examples are given in figure 2, with the complexity level ranging from green (lowest) to red (highest). Points with a complex crossing patterns are correctly identified as hot spots, while non convergent tracks have the lowest complexity.

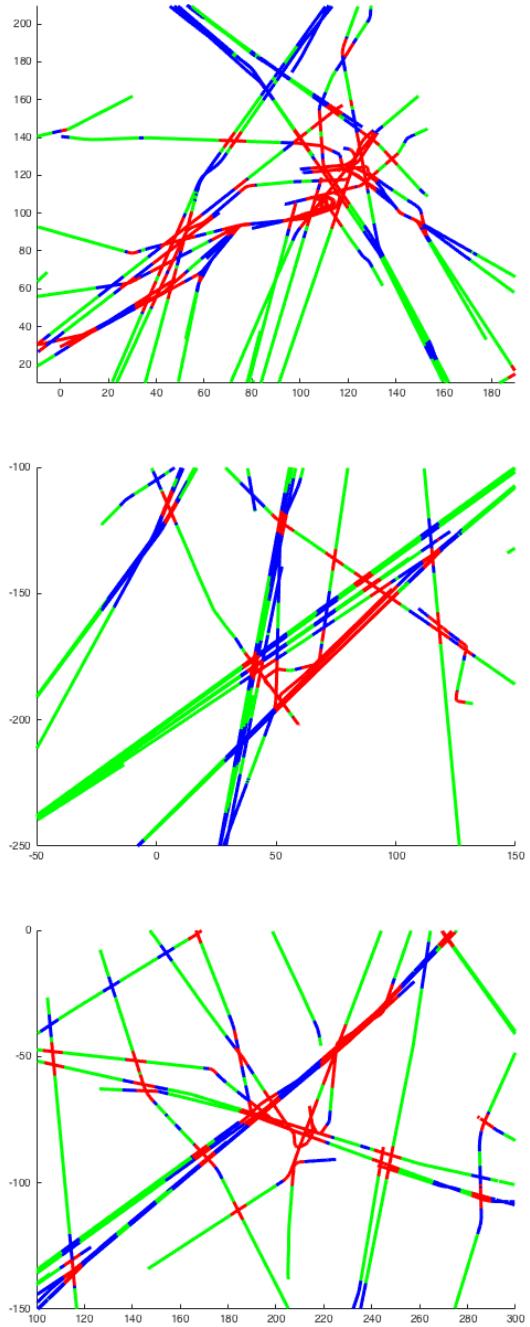

Fig. 2. Clustering of the airspaces over Paris (top), Toulouse (middle) and Lyon (bottom).

## VII. Adding density information

In the previous computations, the number of aircraft or the density (i.e. number of flights per unit area) are not taken in consideration, as the Gaussian field model encodes only the geometry of the traffic. In applications however, especially those involving complexity, theses values have a highly influential affect. As for controller's workload, it is of the same order of importance as the geometry. The raw number of aircraft is not really relevant as soon as the airspace considered exceed the size of a control sector. The density is a better indicator and includes the aircraft number just be integrating over the area of interest. It is easily estimated with kernels. Given a set of sampled flight positions $(x_i)_{i=1^N}$ at a sampling time $T$, the average density is given at point $x$ by:

$$d_T(x) = \sum_{i=1}^{N} \frac{1}{h} K\left(\frac{\|x - x_i\|}{h}\right) \tag{26}$$

where $h > 0$ defines the size of the kernel and $K : \mathbb{R}^+ \to \mathbb{R}^+$ has the following properties:

$$\begin{cases} t > 1 \Rightarrow K(t) = 0 \\ \int_{\mathbb{R}^+} K(t)dt = 1 \\ \forall t > 0 \in \mathbb{R}, K(0) > K(t) \\ K \in C^1(\mathbb{R}) \end{cases} \tag{27}$$

Please note that in contrast with probability density non-parametric estimation, the value obtained after summation is not divided by the number of samples, neither by the size of the time window. Since the function $K$ integrates to 1, the overall number of aircraft present at time $T$ can be recovered by integration. The first assumption is not strictly needed, but simplifies the computations by avoiding taking into account flights that are too far to be influential. The effective support of the kernel is determined by the parameter $h$. In an operational context, taking 5 to 10 horizontal separation norms, i.e. 25 NM to 50 NM is enough. The third assumption ensures that flights away from the current position cannot be as influential as one located there. It is a very mild assumption as all the usual kernels are strictly decreasing (whith the exception of the rectangular one, but that does not satisfies the final smoothness assumption). Finally, the function $K$ must be at least $C^1$.

In contrast with the covariance based representation, the density is easily computed by a simple trick related to Fourier transform.

**Proposition 4.** *The Fourier transform of the density is given by:*

$$\hat{d}(\xi) = \Theta(h\xi) \sum_{i=1}^{N} e^{i\langle \xi, x_i \rangle}$$

*where, if $d$ is the dimension of the space (i.e. 2 or 3):*

$$\Theta(\xi) = \|\xi\|^{-(d-2)/2} \int_{\mathbb{R}^+} K(t) t^{d/2} J_{(d-2)/2}\left(\|\xi\| t\right) dt$$

*Proof.* This is just a standard application of the well known relation between translations and Fourier transforms:

$$\mathcal{F}(f(x - x_0))(\xi) = e^{i\langle \xi, x_0 \rangle} \mathcal{F}(f)(\xi)$$

The function $\Theta$ has a special form, since it is the fourier transform of a radial basis function [16]. □

The power of the Fourier transform approach lies in the fact that $\Theta$ does not depend on the samples and can be pre-computed. The only part to estimate is the sum of exponentials. Here again, a computational trick allows a very efficient implementation. First of all, the problem reduces to a one dimensional case since:

$$e^{\langle \xi, x \rangle} = \prod_{j=1}^{d} e^{\xi_j x_j}$$

Assuming that points of evaluation in the Fourier domain are located on an evenly spaced grid, then any local interpolation formula, like Lagrange polynomial interpolation, express the value of the function $e^{\xi_j x_j}, j = 1 \dots d$ as a linear combination of the function values on grid points. The summation problem thus further reduces to a summation of weighted complex exponentials over a regular grid. Provided the grid size is selected to be a power of 2, this can be accomplished through a Fast Fourier Transform (FFT). Many efficient implementations exist for modern hardware, including GPU. Due to the ability to compute efficiently in the Fourier domain, the density estimator adds only a marginal cost to the overall computation.

Taking into account the density in the overall process of distance computation between traffic images is not straightforward. Two main approaches can be retained:

- Consider the density as an extra information in each generalized pixels that will thus be made of a positive real value (the density) and a symmetric definite positive matrix.
- Use the density as a multiplicative weighting factor. In this case, the distance between two symmetric positive definite matrices will be weighted by the distance between the densities at the same points.

Both have pro and cons, and the final choice may depend on the target application. If complexity is considered to be the most important features of the traffic, then the additive procedure suffers from a severe flaw: since it only adds to the overall distance, areas with little traffic will contribute nearly at the same level as crowded ones if the geometry of flight paths are very different. However, for an ATCO, the perceived complexity may be reversed just because of the number of aircraft that must be managed may exceed his monitoring capacities. The weighting approach does not exhibit such a drawback as density difference acts by product. Another benefit of the multiplicative approach is that areas with simultaneous low density will almost not contribute. However, a flaw still exists since areas with similar densities and very different traffic configurations will be ignored.

Based on the previous remarks, an hybrid approach was retained:

- The density is added to he generalized pixels of the traffic image and contributes in an additive manner to the overall distance. The distance between densities is modeled after the Hellinger distance [17]. At the same location $x$, the squared distance between two densities $d_1(x), d_2(x)$ is taken to be:
$$\left(\sqrt{d_1(x)} - \sqrt{d_2(s)}\right)^2$$
- The supremum of the densities $\sup(d_1(x), d_2(x))$ acts as a weighting factor for the distance between covariance matrices at point $x$.

A comparison between perceived workload on the Reims control center in France and the above indicator is currently under progress. The results are expected to be delivered in the first quarter of 2019.

## VIII. Conclusion and future work

The work presented here is still under development. However, primary results on complexity assessment are promising. Based on the experience gathered on the topic, it is expected that the new approach presented here will outperform the current state-of-the-art metrics. Furthermore, thanks to the ability to compute the distance between two grids of $SPD(3)$ elements, it offers the unique opportunity to derive a database index for traffic situations that will be an invaluable tool for practitioners in the field of air traffic management. One of the main possible usage of such metrics is to feed algorithms for Dynamic Airspace Configuration function. The metric should allow to identify areas where the complexity is too high for sharing the work between two different controllers. These areas will then be used as Blocks to gather with other volumes less constrained for creating new sectors. These will allow a dynamic airspace configuration adapted to the traffic situation. A possible future work is the validation of this approach of dynamic sectorisation by comparing on a given airspace, the areas identified by controllers as critical and those identified by an algorithm using the complexity metric based on local covariance.

From a computational point of view, some work must be done on code optimization in order to speed up the process and allow a full spatio-temporal estimation. Parallel computing is also under investigation.

## References

[1] G. Mykoniatis, F. Nicol, and S. Puechmorel, "A new representation of air traffic data adapted to complexity assessment," *ALLDATA 2018 April 22 2018 - Athens, Greece*, 2018.

[2] N. inc., "Neo4j, a native graph database," http://neo4j.com/, 2018.

[3] M. Prandini, L. Piroddi, S. Puechmorel, and S. Brazdilova, "Toward air traffic complexity assessment in new generation air traffic management systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 809–818, Sept 2011.

[4] A. Cook, H. A. Blom, F. Lillo, R. N. Mantegna, S. Miccich, D. Rivas, R. Vzquez, and M. Zanin, "Applying complexity science to air traffic management," *Journal of Air Transport Management*, vol. 42, pp. 149 – 158, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969699714001331

[5] L. I. S. Shelden, R. Branstrom, and C. Brasil, "Dynamic density: An air traffic management metric," NASA, Tech. Rep. NASA/TM-1998-112226, 1998.

[6] K. Lee, F. E., and A. Prichett, "Air traffic complexity : An input-output approach," in *Proceedings of the US Europe ATM Seminar*. Eurocontrol-FAA, 2007, pp. 2–9.

[7] D. Delahaye and P. S., "Air traffic complexity based on dynamical systems." in *Proceedings of the 49th CDC conference*. IEEE, 2010.

[8] EUROCONTROL, "All-purpose structured eurocontrol surveillance information exchange (asterix)," https://www.eurocontrol.int/services/asterix.

[9] PROJ contributors, *PROJ coordinate transformation software library*, Open Source Geospatial Foundation, 2018. [Online]. Available: https://proj4.org/

[10] T. opensky network, "A community-base receiver network," http://opensky-network.org/.

[11] P. Hall, N. I. Fisher, and B. Hoffmann, "On the nonparametric estimation of covariance functions," *Ann. Statist.*, vol. 22, no. 4, pp. 2115–2134, 12 1994. [Online]. Available: https://doi.org/10.1214/aos/1176325774

[12] Y. L., N. W., M. H., N. D. Turner, J. R. Lupton, and R. J. Carroll, "Nonparametric estimation of correlation functions in longitudinal and spatial data, with application to colon carcinogenesis experiments," *Ann. Statist.*, vol. 35, no. 4, pp. 1608–1643, 08 2007. [Online]. Available: https://doi.org/10.1214/009053607000000082

[13] J. Y., Z. G., R. L., and H. W., "Nonparametric covariance model," *Statistica Sinica*, vol. 20, no. 1, pp. 469–479, 2010. [Online]. Available: http://www.jstor.org/stable/24309002

[14] E. A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964. [Online]. Available: https://doi.org/10.1137/1109020

[15] F. Nielsen and R. Bhatia, *Matrix Information Geometry*. Springer Berlin Heidelberg, 2012. [Online]. Available: https://books.google.fr/books?id=MAhygTspBU8C

[16] G. Fasshauer, *Meshfree Approximation Methods with MATLAB*, ser. Interdisciplinary mathematical sciences. World Scientific, 2007. [Online]. Available: https://books.google.fr/books?id=gtqBdMEqryEC

[17] A. van der Vaart, *Asymptotic Statistics*, ser. Asymptotic Statistics. Cambridge University Press, 2000. [Online]. Available: https://books.google.fr/books?id=UEuQEM5RjWgC