Parallel Complexity Copmputation Based on Dynamical Systems

Tambet Treimuth , ENAC,7, Ave Ed Belin 31055 Toulouse France (treimuth@recherche.enac.fr)
Daniel Delahaye, ENAC, 7, Ave Ed Belin 31055 Toulouse France (delahaye@recherche.enac.fr)
Sandra Ulrich Ngueveu, LAAS, 7 Avenue du Colonel Roche 31400 Toulouse France (sungueve@laas.fr)
Stéphane Puechmorel , ENAC, 7, Ave Ed Belin 31055 Toulouse France (puechmor@recherche.enac.fr)

Abstract

This paper describes a parallel approach for computing an air traffic complexity metric that is based on dynamical systems. It is important in order to quantify the complexity and geometric features of traffic. Lyapunov exponents are used for understanding the geometry of current air traffic in the airspace. It is computed on different points on a spatial grid. Those values can indicate areas where air traffic controllers must pay more attention because the complexity is higher at those points.

A parallel implementation of the air traffic complexity metric computation is introduced in this paper. Serial implementation of this problem is too slow for real world problems. This process is parallelizable because Lyapunov exponent values can be computed simultaneously for each grid point in the airspace.

Intdouction

The number of aircraft in the airspace that air traffic controllers have to direct is increasing but there exists a maximum number of aircraft that one air traffic controller can handle. It is not enough if we just take into account the maximum number of aircraft that are allowed at each moment in a sector of air traffic controller. There are situations when they can accept more aircraft. It depends on the geometry of the traffic. Geometric features of the traffic have to be included to quantify the complexity. This method for evaluating complexity of air traffic takes into account the intrinsic traffic disorder. ATC controllers can have different workload even if the number of aircraft present in the sector is the same. This complexity computation method is based on dynamical systems. There exists many different methods for measuring the complexity of air traffic. The first complexity indicator incorporating structural considerations along with the simple number of aircraft is the "Dynamic Density" of Laudeman et al. from NASA [8]. The "Dynamic Density" is a weighted sum of the traffic density (number of aircraft), the number of heading changes ($>$ 15 degrees), the number of speed changes ($>$0.02 Mach), the number of altitude changes ($>$750 ft), the number of aircraft with 3-D Euclidean distance between 0-25 nautical miles, the number of conflicts predicted in 25-40 nautical miles. These factors are summed together using weighting factors that were determined by showing different traffic scenarios to several controllers. B.Sridhar from NASA [10], has developed a model to predict the evolution of such a metric in the near future. Efforts to define "Dynamic Density" have identified the importance of a wide range of potential complexity factors, including structural considerations. However, the instantaneous position and speeds of the traffic itself does not appear to be enough to describe the total complexity associated with an airspace. A few previous studies have attempted to include structural consideration in complexity metrics, but have done so only to a restricted degree. For example, the Wyndemere Corporation proposed a metric that included a term based on the relationship between aircraft headings and dominant geometric axis in a sector [6]. The importance of including structural consideration has been explicitly identified in work at Eurocontrol. In a study to identify complexity factors using judgment analysis, Airspace Design was identified as the second most important factor behind traffic volume [7]. Histon, et, al. [4], [5] investigated how this structure can be used to support structure-based abstractions that controllers appear to use to simplify traffic situations. The previous models do not take into account the intrinsic traffic disorder which is related to the complexity. The first efforts related with disorder can be found in [2]. This paper introduces two classes of metrics which measure the

disorder of a traffic pattern.G.Aigoin has extended and refined the geometrical class by using a cluster based analysis [1]. All the previous metrics capture only one feature of the complexity and are not able to produce an aggregate metric which can capture all the possible situations (high-low density, how-low convergence, translation organization, round about organization).

Dynamical Systems

The first part of this algorithm consists of finding a model of dynamical system that represent current traffic situation at the airspace. Later this model can be used for computing complexity values for different points in the airspace. A dynamical system model describes the traffic situation in the airspace. This model consists of linear and non-linear part. This model is based on aircraft positions and velocities at each time step. Based on this input data we can get coefficients of model of dynamical system. The same process is repeated for each time step when the complexity metric is computed.

Linear Model

For this approach a set of aircraft that are flying in the airspace is considered as a dynamical system. A model of dynamical system is computed for each time step. A dynamical system can be described by the following equation:

$$\dot{\vec{X}} = \mathbf{A}.\vec{X} + \vec{B} \tag{1}$$

where $\vec{X}$ is the state vector of the system :

$$\vec{X} = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{2}$$

This equation describes relationship between aircraft velocities $\dot{\vec{X}}$ to aircraft positions $\vec{X}$. This model can be used for understanding the behaviour of the traffic in the airspace. Eigenvalues of the matrix A can be used for understanding the evolution of the traffic. See Figure 1

Those values of A and B can be computed with Least Square Minimization method. This way A and B are found that can minimize the error E. For each aircraft i, it is supposed that position $\vec{X}_i = [x_i, y_i, z_i]^T$ and speed vector $\vec{V}_i = [vx_i, vy_i, vz_i]^T$ are part of input data. An error criterion between the dynamical system
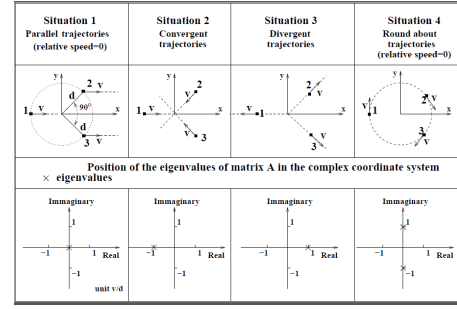

Fig. 1. Eigenvalues describe the behaviour of dynamical systems

model and the observations is computed :

$$E = \sum_{i=1}^{i=N} \left\| \vec{V}_i - \left(A.\vec{X}_i + \vec{B} \right) \right\|^2$$

The position matrix and the velocity matrix have following structure:

$$X = \begin{bmatrix} x_1 & ... & x_N \\ y_1 & ... & y_N \\ z_1 & ... & z_N \\ 1 & ... & 1 \end{bmatrix} \quad V = \begin{bmatrix} vx_1 & ... & vx_N \\ vy_1 & ... & vy_N \\ vz_1 & ... & vz_N \end{bmatrix}$$

Non Linear Extensions in Space

A non linear dynamical system is summarized by the following equation [3]:

$$\dot{\vec{X}}(t) = \vec{f}(\vec{X}) \tag{3}$$

where $\vec{X}$ is the state vector of the system ($\vec{X} = [x, y, z]^T$) and $\vec{f} : C^2$ vector field, describe systems which integral curves may fit the observed trajectories. This equation associates a vector speed $\dot{\vec{X}}$ to a position in the space coordinate $\vec{X}$ and then synthesis a particular vector field. Based on the observations of the aircraft (positions, speed vectors), the dynamical system has to be adjusted with the minimum error.This fitting is done with a Least Square Minimization (LMS) method for which the following criterion is used :

$$E_1 = \sum_{i=1}^{i=N} \|\vec{V}_i - \vec{f}(\vec{X}_i)\|^2 \tag{4}$$

where N is number of observations.

If we consider criterion E_1 only, it can be shown that there is an infinite number of vector fields $\vec{f}$ which can be adjusted to the observations. In order to keep the smoothest one, another criterion is added

which has to be minimized, the so-called "div-curl" criterion :

$$E_2 = \int_{R^3} \alpha \|\nabla div \vec{f}(\vec{X})\|^2 + \beta \|\nabla curl \vec{f}(\vec{X})\|^2 d\vec{X} \quad (5)$$

with α, β positive weights controlling the smoothness of the approximation by focusing on constant divergence or constant curl. In the following, we will consider $\alpha = \beta = 0.5$; in such case :

$$E_2 = \|\Delta \vec{f}(\vec{X})\|^2$$

where $\Delta \vec{f}(\vec{X})$ is the Laplacian of the vector field $\vec{f}$.

The joint minimization of E_1 and E_2 induces a unique optimum vector field:

$$\vec{f}(\vec{X}) = \sum_{i=1}^{N} \phi(\|\vec{X} - \vec{X_i}\|).\vec{a_i} + \mathbf{A}.\vec{X} + \vec{B}$$

with $\vec{a_i}$ parameter vectors (one for each observation),

$$\phi(\|\vec{X} - \vec{X_i}\|) = \mathbf{Q}(\|\vec{X} - \vec{X_i}\|^3)$$

and

$$Q = \begin{bmatrix} \gamma & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & \gamma \end{bmatrix}$$

with $\gamma = \partial_{xx}^2 + \partial_{yy}^2 + \partial_{zz}^2$

The resulting adjustment is done without error ($\Rightarrow minE_1 = 0$).

When $\alpha = \beta = 0.5$, the vector spline function ϕ has the following structure :

$$\phi(\|\vec{X} - \vec{X_i}\|) = 12.\|\vec{X} - \vec{X_i}\|$$

($r_i = \|\vec{X_i} - \vec{X}\|$). It must be noticed, that farthest observations has more weight in such computation. In order to compute the smoothest vector field which fit exactly the measures, all observations have to be taken into account in the computation. This velocity field is used for computing the Lyapunov exponents.They are computed at different point of this velocity field. All steps of the algorithms are described in the following section.

Algortihm

The algorithm for computing Lyapunov exponents can be divided into three parts:

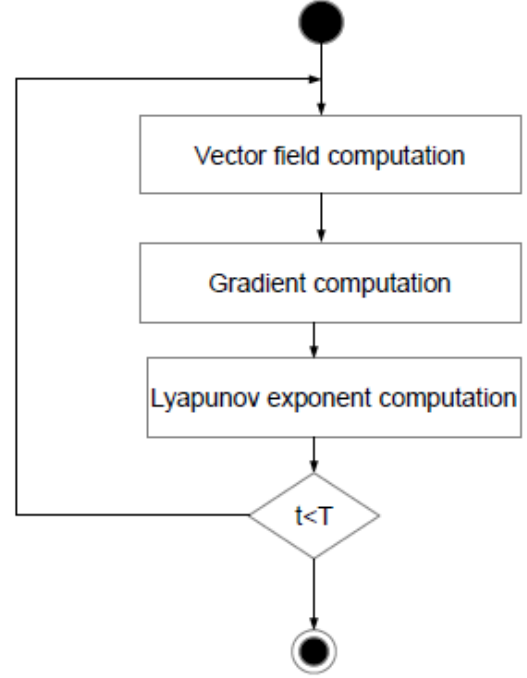- Computation of Dynamical Regression
- Computation of gradients

Fig. 2.   Three main steps of complexity computation

- Computation of Lyapunov exponents

Those three steps are repeated for each time step t until the final step T.

Dynamical Regression

The first part of this method consists of finding the model of dynamical system that correspond to the current traffic in the airspace. This part was described in the previous section. The goal of this step is to find the matrix A and the vector B.That were described previously in the linear model part. Those coefficients describe a velocity field that can be used for computing Lyapunov exponents.

Gradient Computation

The Lyapunov exponents are computed at each gridpoint. Lyapunov exponents values can be computed only if gradients are known at those points. The gradient of the velocity field are computed based on the dynamical system model that was computed at the previous point. Those values can be used as an input data for the for the Runge Kutta method. Those
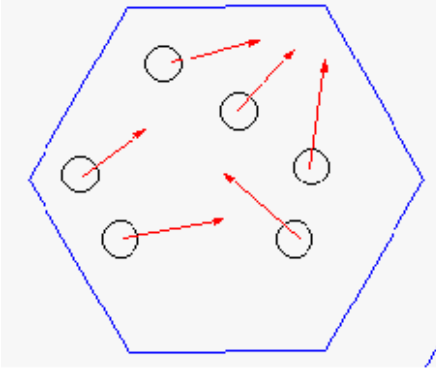
Fig. 3. High Lyapunov exponent value with six aircraft

Fig. 4. Low Lyapunov Exponent value with six aircraft

computations for each grid point are independent and can be computed simultaneously.

Lyapunov Exponent Computation

Lyapunov exponent measures the complexity for each grid point where it is computed. Lyapunov exponent measures the divergence of two trajectories that have nearby initial points. Runge- Kutta method is a method for numerical integration.

Lyapunov Exponents

Air traffic controllers can accept more aircraft if aircraft are well structured. The first example shows the situation with six aircraft. There are many potential conflicts. It causes high workload for air traffic controller. See Figure 3 The second example has six aircraft also but all aircraft have same direction. See Figure 4. This is much easier for air traffic controller. Those differences can be analysed with Lyapunov exponents.

This metric for complexity relies on a measure of sensitivity to initial conditions of the underlying dynamical system called Lyapunov exponents. In order to figure out what Lyapunov exponents are, let consider a point and look at its evolution when transported by the dynamical system. Let $\vec{x}_0$ be fixed (initial point) and let γ be a point trajectory of the dynamical system associated to the vector field $\vec{f}$ given by :

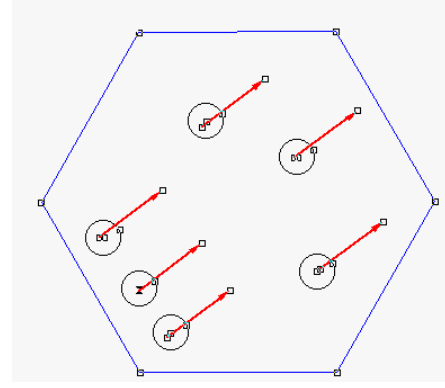$$\gamma(t,\vec{x}_0) = \vec{x}_0 + \int_0^t \vec{f}(u,\gamma(u,\vec{x}))du \qquad (6)$$

Assume now that trajectory is disturbed by a small perturbation $\vec{\varepsilon}$, we have :

$$\gamma(t,\vec{x}_0+\vec{\varepsilon}) = \gamma(t,\vec{x}_0) + \nabla_{\vec{x}}\vec{f}(\gamma(t,\vec{x})).\vec{\varepsilon} + o(\|\vec{\varepsilon}\|)$$

where $\nabla_{\vec{x}}\vec{f}(t,\gamma(t,\vec{x}))$ is the differential of the vector field $\vec{f}$ at $\vec{x}$. Divergence to nominal trajectory with respect to time is thus $\|\gamma(t,\vec{x}_0) - \gamma(t,\vec{x})\| = D(t,s)$ (see figure).

$\gamma(t,\vec{x})$ being defined as a flow :

$$\frac{\partial\gamma(t,\vec{x})}{\partial t} = \vec{f}(t,\gamma(t,\vec{x})) \quad \gamma(0,\vec{x}) = \vec{x}$$

with $\vec{f}$ a smooth vector field, it is possible to show that $D(t)$ satisfies a differential equation also. Given a nominal trajectory $\gamma(t,\vec{x}_0)$, then divergence of nearby trajectories can be found up to order one in $\|\vec{x}-\vec{x}_0\|$ by solving :

$$\frac{\partial D(t,\vec{x})}{\partial t} = \nabla_{\vec{x}}\vec{f}(t,\gamma(t,\vec{x})).D(t,\vec{x}) \quad D(0,\vec{x}) = \|\vec{x}-\vec{x}_0\|$$

If the three space dimensions are considered (x,y,z), and since the previous equation is linear, it can be extended to the matrix form :

$$\frac{dM(t)}{dt} = \nabla_{\vec{x}}\vec{f}(t,\gamma(t,\vec{x})).M(t) \quad M(0) = Id$$

Where each column of the M matrix corresponds to the divergence associated to the principal coordinate axis. This equation is called the variational equation of the flow. The variational equation describes in some sense a linear dynamical system "tangent" to the original one. Let $U^t(t)\Sigma(t)V(t) = M(t)$ be the singular value decomposition of $M(t)$. The Lyapunov
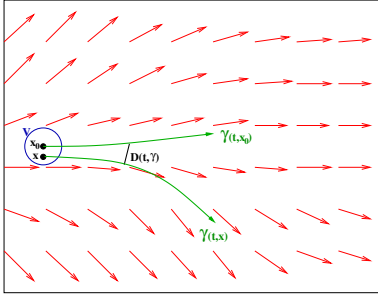
Fig. 5. Time evolution of a reference trajectory and a perturbed trajectory

exponents are mean values of the logarithms of the diagonal elements of $\Sigma(t)$:

$$\kappa(s) = -\frac{1}{T} \int_0^T \log(\Sigma_{ii}(t)) dt \ \forall \Sigma_{ii}(t) \le 1$$

When Lyapunov exponents are high, the trajectory of a point under the action of the dynamical system is very sensitive to initial conditions (or to the parameters on which the vector field may depend), so that, situation in the future is unpredictable.On the other hand, small values of the Lyapunov exponents mean that the future is highly predictable (expected to be comfortable for a controller).The Lyapunov exponent measures the distance between two trajectories. See figure 5).

So, the Lyapunov exponent map determines the area where the underlying dynamical system is organized. It identifies the places where the relative distances between aircraft do not change with time (low real value) and the ones where such distance change a lot (high real value).

Those Lyapunov exponent values can be computed with Runge Kutta algorithm. It adds all te deviation until the last time step. This total sum is equal to the Lyapunov exponent. This value describes the potential workload to the air traffic controller.

Graphical Processing Unit (GPU)

Parallel Compuation

A parallel computing platform and programming model CUDA was used for computing complexity.

CUDA programming is used to solve problems that can be expressed as data parallel computations. Applications that must process big data sets can use this model to speed up computations. Our goal for this project was to distribute data across threads in the way that each thread working on a part of the data. A sequential implementation of algorithm was not fast enough if complexity values for larger problems. For example, if complexity values for the whole country must be computed. A parallel complexity computation algorithm is proposed instead of preceding sequential algorithm. Complexity is evaluated on points evenly spaced on a spatial grid. This velocity field is used to computed Lyapunov exponent values for each grid point. Since velocity field computation and Lyapunov exponent computation was time-consuming with sequential computation, we had to find way to parallelize this part of computation program. Velocity field values computation can be parallelized because previous implementation consisted of many nested loops. Those loops were unrolled because there are not dependencies between different iterations of loops. Velocity field values must be computed for many points of airspace simultaneously also.

Parallel Computation Patterns

There exists many patterns that can be used for parallelizing serial computation. Reduce pattern was used for improving computation time[9]. Reduce pattern can be uses if there exists a set of completely independent operation. It combines all the elements in a collection into a single element using some associative combiner operator. The usefulness of reduce is easy to understand. The serial implementation of serial addition. Four variables are added sequentially.See Figure 6 One simple example is summation, which shows up frequently in mathematical computations. In the reduce pattern, a combine function is used to combine all the elements of a collection pairwise and create a summary value. It is assumed that pair of elements can be combine in different orders.See Figure 7

Paralllel implementation
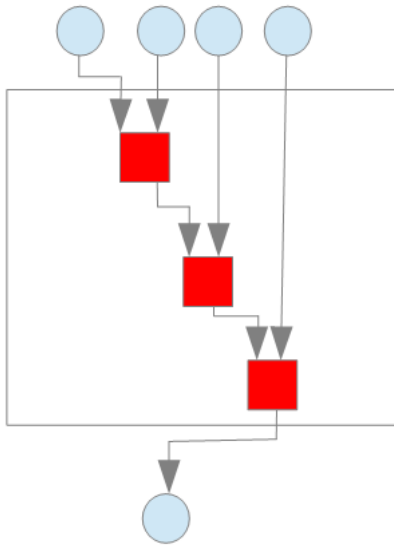
Velocity Field Computation
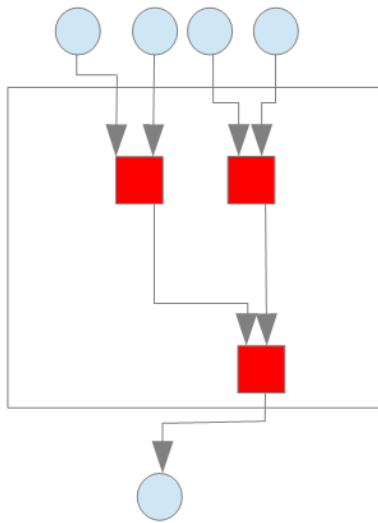
Fig. 6. Serial implementation

Fig. 7. Tree implementation

Lyapunov exponent values are used to computed complexity metrics. It can be computed only if we know gradient values that describe velocity field in the airspace.See figure Those gradient values can be computed from the velocity field values that were found at the previous step. Gradient matrices are needed for each grid point in the airspace. For each time step the grid gradient values must be computed.

The same computation must be repeated for each time step. Those gradient values must be used for Lyapunov exponent computation. It is used as input values for Runge Kutta algorithm that The airspace for the big problem could be divided into millions of evenly distributed grid points. So, we have to compute those gradient matrices for millions of times. For parallel solution nested loops can be again unrolled. There are no dependencies between gradient matrices.

Runge Kutta Computation

Lyapunov exponents provide a quantitative measure of the divergence or convergence of nearby trajectories for a dynamical system. Runge-Kutta algorithm can be used to compute Lyapunov exponent value for each grid point. Runge-Kutta algorithm can be executed in parallel for each point in the airspace because there are different gradient values for each grid point. The result of the Runge-Kutta function is the value that describes the complexity of the airspace at the point where it was computed. Previously described steps must be repeated for each time step because the complexity of the airspace changes frequently. Complexities for different time steps can be computed simultaneously because those velocities fields are independent. Previously described computed can be divided into four parts.

If the GPU does not have enough data the computation time could be even slower than with single threaded CPU.This is the reason why parallel computation algorithm must be designed carefully. The matrix A and the vector B that were part of the Equation 1 are computed for each time step. For this computation Intel MKL library is used. These values are computed only once at each time step. The GPU computation is not needed for this part the Intel MKL provided fast enough computation time.

Those A and B describe the traffic situation at that time step. Those values can be used for computed velocity field valued at each point in the airspace. If only linear model is used then those values are not precise enough. All the situations cannot be described with A and B. That is reason why the non linear extension is used. This gives more precise results. This computation is function of the number of aircraft square.

.

Results

Air traffic data from the French airsapce was used for the complexity computation. A GeForce GTX 660 GPU Ti with Compute Capability 3.0 was used for computation. The radar data was used for computing Lyapunov Exponents. The radar data describes trajectories after the conflicts between aircraft were solved. It does not give real complexity values. The complexity values are much lower. The trajectory projections were used for more precise complexity results. See Figure 8. The parallel computation of Lyapunov exponents provided remarkable speed up of computation. See Table I. First column describes number of grid points where Lyapunov exponents were computed. Second column gives information about Lyapunov exponents computation with CPU. The third column describes the computation time with GPU.If the complexity is analysed for the French airspace then the complexity is computed at 200000 points. The complexity of air traffic can be used for measuring the workload. Red color indicates the higher complexity of air traffic and higher workload for air traffic controllers. See figure 9. If the complexity of the air traffic is not so high then air traffic controller could accept more aircraft. See Figure 10. There is not red color on the complexity map that represent high complexity of air traffic. Lyapunov exponent values can be divided into two groups: high and low Lyapunov expoenent values. High Lyapunov exponents represent high sensitivity to initial conditions:

- The future is highly unpredictable
- The traffic has low organization
- Represent high rate of change of relative distances

Low Lyapunov exponents represent low sensitivity to initial conditions:

- The traffic is highly predictable future
- The traffic is organized
- Relative distances has low rate of change

The air traffic complexity was computed also for the ATCC in Reims. The computed complexity was compared with number of air traffic controllers who work at ATCC in Reims. The red line describes normalized complexity of traffic. Correlation between two lines can be seen. When the complexity of the ATCC is increasing then the number of Air Traffic Controllers who work is increasing also. IF the complexity is not so high then the number of controllers who work is also decreasing.

Number of gridpoints	Computation time on CPU [s]	Computation time on GPU [s]
50000	11,0	0,15
100000	23,0	0,31
300000	68,0	0,73

TABLE I
LYAPUNOV EXPONENT COMPUTATION COMPARISON

Conclusion

This paper presented a method for dynamical systems based complexity computation. The number of aircraft that are in the airspace will increase in the future. We have evaluate how many aircraft can be controlled by one controller. If the complexity is estimated only as a number of aircraft in the sector then it may not describe the situation precisely. Air Traffic Controllers may sometimes accept more aircraft because the structure of traffic is different. So, if we know geometric features of traffic then Air Traffic Controllers may accept more aircraft. This metric can be used for analysing the pattern of air traffic. This complexity is computed at each grid point. The computation is implemented on GPU to reduce the computation time The complexity computation is implemented in CUDA language on the GPU. It improved the computation time because the algorithm consists of similar computation pattern for million of times. This way this complexity metric can be used for real air traffic problems.

REFERENCES

[1] G Aigoin. Air traffic complexity modeling. Master's thesis, Ecole Nationale de l'Aviation Civile, 2001.
[2] D Delahaye and S Puechmorel. Air traffic complexity : Towards intrinsic metrics. In *Proceedings of the Second USA/EUROPE ATM R&D Seminar*. Eurocontrol/FAA, 2000.
[3] Daniel Delahaye and Stephane Puechmorel. Air traffic complexity based on dynamical systems,atlanta, united states. *49th IEEE Conference on Decision and Control*, 2010.
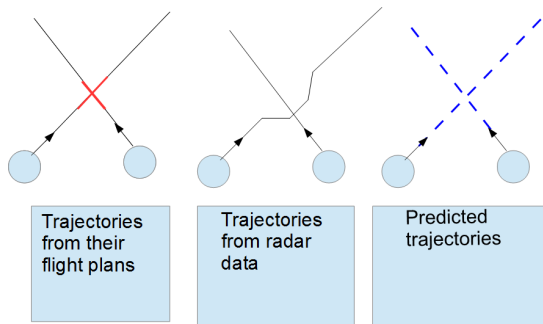
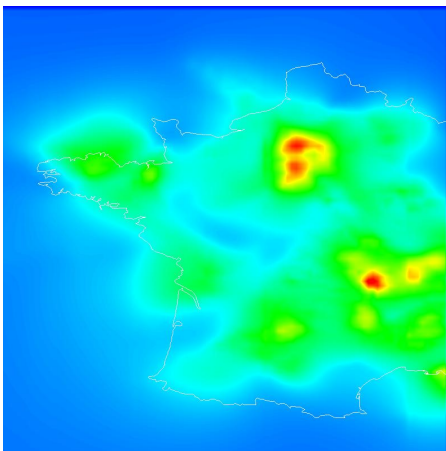Fig. 8.   Aircraft trajectory projection

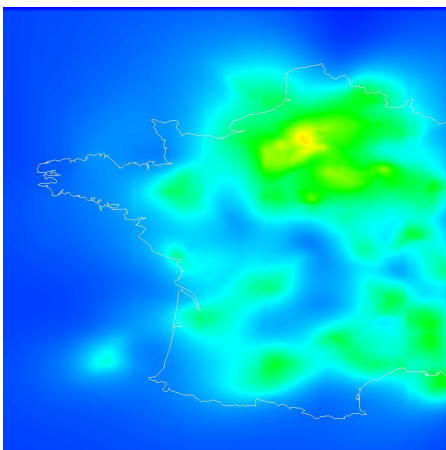Fig. 9.   Higher complexity in the French airspace

Fig. 10.   Lower complexity in the French airspace

[4] J.M Histon, G Aigoin, D Delahaye, R.J Hansman, and Puechmorel. S.   Introducing structural consideration into complexity metrics.   In *Proceedings of the Fourth USA/EUROPE ATM R&D Seminar*. Eurocontrol/FAA, 2001.

[5] J.M Histon, R.J Hansman, G Gottlieb, H Kleinwaks, S Yenson, D Delahaye, and Puechmorel. S.   Structural consideration and cognitive complexity in air traffic control. In *Proceedings of the 21th Air Traffic Management for Commercial and Military Systems*. IEEE,AIAA, 2002.

[6] Windemere inc. An evaluation of air traffic control complexity. Technical report, NASA 2-14284, 1996.

[7] B Kirwan, R Scaife, and R Kennedy. Investigating complexity factors in u.k. air traffic management. *Human Factors and Aerospace Safety*, 1(2), 2001.

[8] I.V Laudeman, S.G Shelden, R Branstrom, and C.L Brasil. Dynamic density : an air traffic management metric. Technical report, NASA TM-1998-112226, 1998.

[9] Michael McCool, Arch D.Robinson, and James Reinders. *Structure Parallel Programming*. Elsevier, 2012.

[10] B Sridhar, K.S Seth, and S Grabbe. Airspace complexity and its application in air traffic management. In *Proceedings of the Second USA/EUROPE ATM R&D Seminar*. Eurocontrol/FAA, 2001.