Entropy minimizing curves
Application to automated flight path design

S. Puechmorel

ENAC

29th October 2015
Problem Statement

Flight path planning

- Traffic is expected to double by 2050;
- In future systems, trajectories will be negotiated and optimized well before the flights start;
- But humans will be in the loop: generated flight plans must comply with operational constraints;

Muti-agent systems

- A promising approach to address the planning problem;
- Does not end up with a human friendly traffic!
- Idea: start with the proposed solution and rebuild a route network from it.
A curve optimization problem

An entropy criterion

• Route networks and currently made of straight segments connecting beacons;

• May be viewed as a maximally concentrated spatial density distribution;

• Minimizing the entropy with such a density will intuitively yield a flight path system close to what is expected.
Problem modeling

Density associated with a curve system

- A classical measure: counting the number of aircraft in each bin of a spatial grid and averaging over time;
- Suffers from a severe flaw: aircraft with low velocity will over-contribute;
- May be corrected by enforcing invariance under re-parametrization of curves;
- Combined with a non-parametric kernel estimate to yield:

\[
\tilde{d}(x) \mapsto \frac{\sum_{i=1}^{N} \int_{0}^{1} K(\|x - \gamma_i(t)\|) \|\gamma_i'(t)\| dt}{\sum_{i=1}^{N} \int_{\Omega} \int_{0}^{1} K(\|x - \gamma_i(t)\|) \|\gamma_i'(t)\| dtdx}
\]

(1)
The entropy criterion

- Kernel $K$ is normalized over the domain $\Omega$ so as to have a unit integral;
- Density is directly related to lengths $l_i$, $i = 1 \ldots n$ of curves $\gamma_i$, $i = 1 \ldots N$:

$$
\tilde{d}: x \mapsto \frac{\sum_{i=1}^{N} \int_{0}^{1} K (\|x - \gamma_i(t)\|) \|\gamma_i'(t)\| \, dt}{\sum_{i=1}^{N} l_i} \quad (2)
$$

- Associated entropy is:

$$
E(\gamma_1, \ldots, \gamma_N) = - \int_{\Omega} \tilde{d}(x) \log \left( \tilde{d}(x) \right) \, dx \quad (3)
$$
Optimal curve displacement field

Entropy variation

- \( \tilde{d} \) has integral 1 over the domain \( \Omega \);
- It implies that:

\[
- \frac{\partial}{\partial \gamma_j} E(\gamma_1, \ldots, \gamma_N)(\epsilon) = \int_{\Omega} \frac{\partial \tilde{d}(x)}{\partial \gamma_j}(\epsilon) \log(\tilde{d}(x)) \, dx
\]

where \( \epsilon \) is an admissible variation of curve \( \gamma_i \).
- The denominator in the expression of \( \tilde{d} \) has derivative:

\[
\int_{[0,1]} \left\langle \frac{\gamma_j'(t)}{\|\gamma_j'(t)\|}, \epsilon'(t) \right\rangle \, dt = - \int_{[0,1]} \left\langle \left( \frac{\gamma_j''(t)}{\|\gamma_j'(t)\|} \right)_N, \epsilon \right\rangle \, dt
\]
The numerator of $\tilde{d}$ has derivative:

$$\int_{[0,1]} \left\langle \left( \frac{\gamma_j(t) - x}{\|\gamma_j(t) - x\|} \right), \epsilon \right\rangle K' (\|\gamma_j(t) - x\|) \|\gamma_j'(t)\| dt \quad (6)$$

$$- \int_{[0,1]} \left\langle \left( \frac{\gamma_j''(t)}{\|\gamma_j'(t)\|} \right), \epsilon \right\rangle K (\|\gamma_j(t) - x\|) dt \quad (7)$$
Normal move

- Final expression yield a displacement field normal to the curve:

\[
\begin{align*}
&\left(\int_{\Omega} \left(\frac{\gamma_j(t) - x}{\|\gamma_j(t) - x\|}\right) K'(\|\gamma_j(t) - x\|) \log \tilde{d}(x) dx\right) \left(\frac{\gamma_j''(t)}{\|\gamma_j'(t)\|}\right)_{\mathcal{N}} \\
&= -\left(\int_{\Omega} K(\|\gamma_j(t) - x\|) \log \tilde{d}(x))dx\right) \left(\frac{\gamma_j''(t)}{\|\gamma_j'(t)\|}\right)_{\mathcal{N}} \\
&+ \left(\int_{\Omega} \tilde{d}(x) \log(\tilde{d}(x))dx\right) \left(\frac{\gamma_j''(t)}{\|\gamma_j'(t)\|}\right)_{\mathcal{N}} \sum_{i=1}^{n} l_i \end{align*}
\]
Implementation

A gradient algorithm

- The move is based on a tangent vector in the tangent space to $\text{Imm}([0, 1], \mathbb{R}^3)/\text{Diff}^+(0, 1)$;
- It is not directly implementable on a computer;
- A simple, landmark based approach with evenly spaced points was used;
- A compactly supported kernel (epanechnikov) was selected: it allows the computation of density $\tilde{d}$ on GPUs as a texture operation that is very fast.
A output from the multi-agent system

Integration in the complete system

- Route building from initially conflicting trajectories:

Figure – Initial flight plans and final ones
Conclusion and future work

An integrated algorithm

- Entropy minimizer is now a part of the overall route design system;
- Only a simple post-processing is necessary to output a usable airways network;
- The complete algorithm is being ported to GPU.

Future work: take the headings into account

- The behavior is not completely satisfactory when routes are converging in opposite directions;
- An improved version will make use of entropy of a distribution in a Lie group (publication in progress).