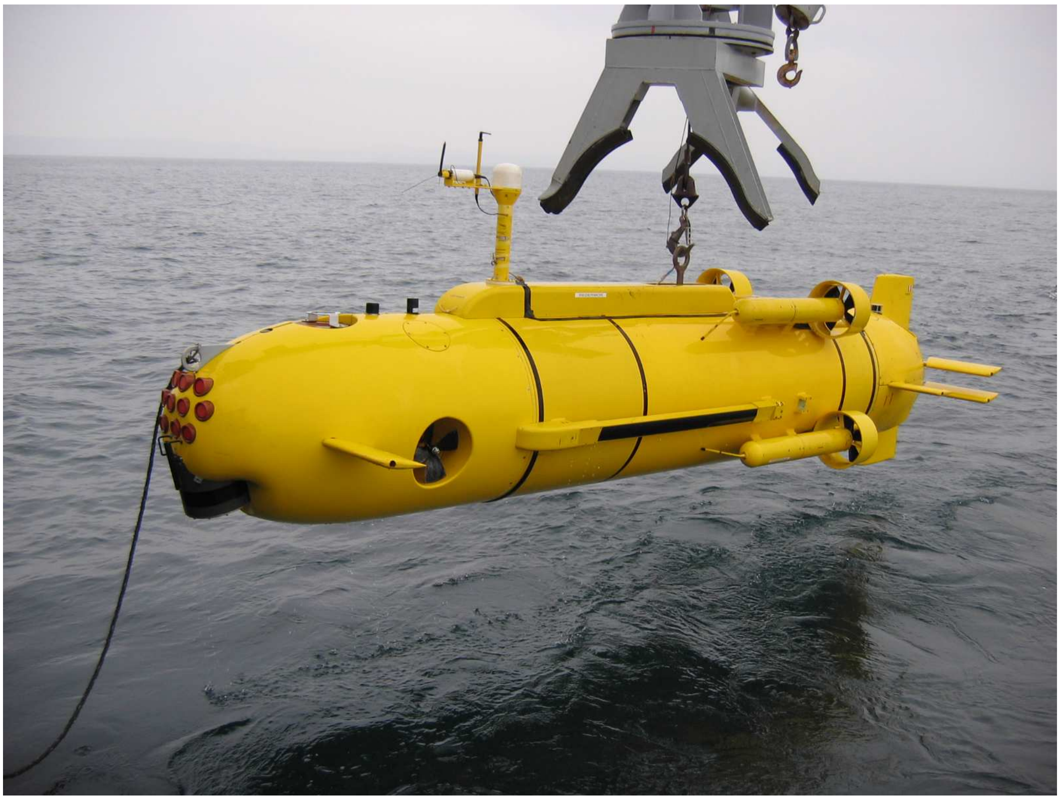
Interval constraint propagation for the localization of an underwater robot

Luc Jaulin
ENSIETA, Brest
GESMA (Groupe d'Etude Sous-Marine de l'Atlantique)

GICOLAG
Global optimization - Integrating Convexity,
Optimization, Logic and Algebraic Geometry
Tuesday, December 5^{th}, 2006.

1 The Redermor



The *Redermor*, made by GESMA
(Groupe d'Etude Sous-Marine de l'Atlantique)

The *Redermor* at the surface

Show simulation

2 SLAM

Localization

Given a map, determine the robot's location. The mark locations are known.

The localization problem is a state estimation problem. The model is

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}) \end{cases}$$

where $\mathbf{x} = (x, y, z, \phi, \theta, \psi, v)$.

SLAM (simultaneous localization and mapping)

The mark locations are unknown.

Determine the location of the robot as well as the location of the marks.

Why choosing an interval constraint approach ?

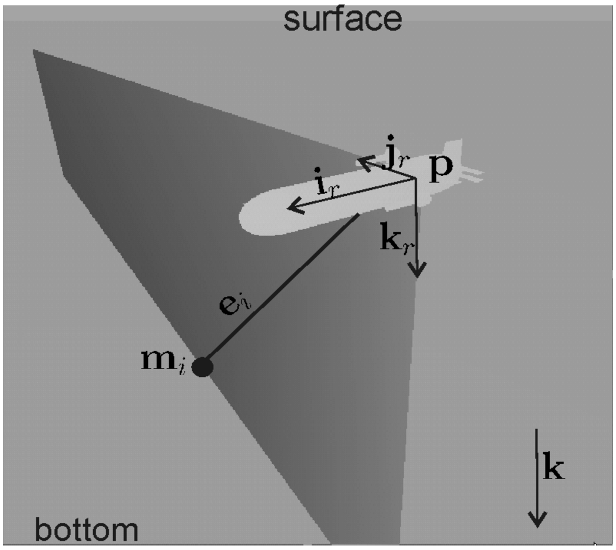
1) A reliable method is needed.
2) The model is nonlinear.
3) The noises are non Gaussian and their pdf are unknown.
4) Guaranteed error bounds are provided by the constructors of available sensors.
5) A huge number of redundant data are available.

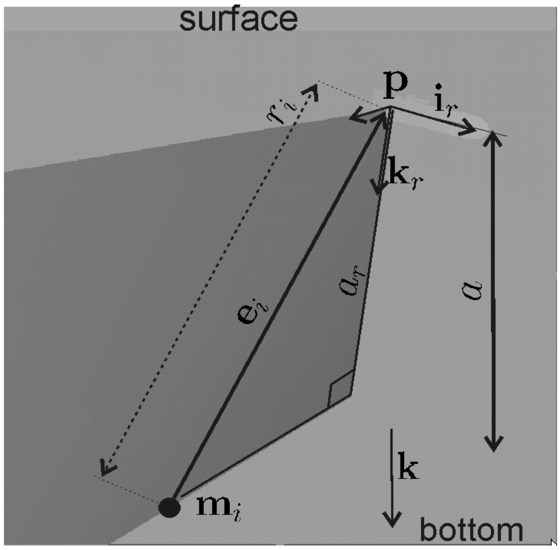
3 Sensors

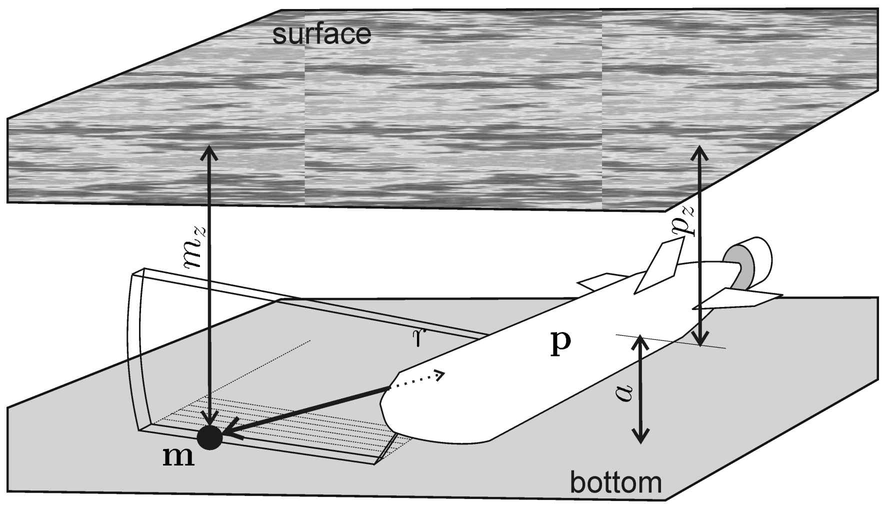
A GPS (Global positioning system), at the surface only.

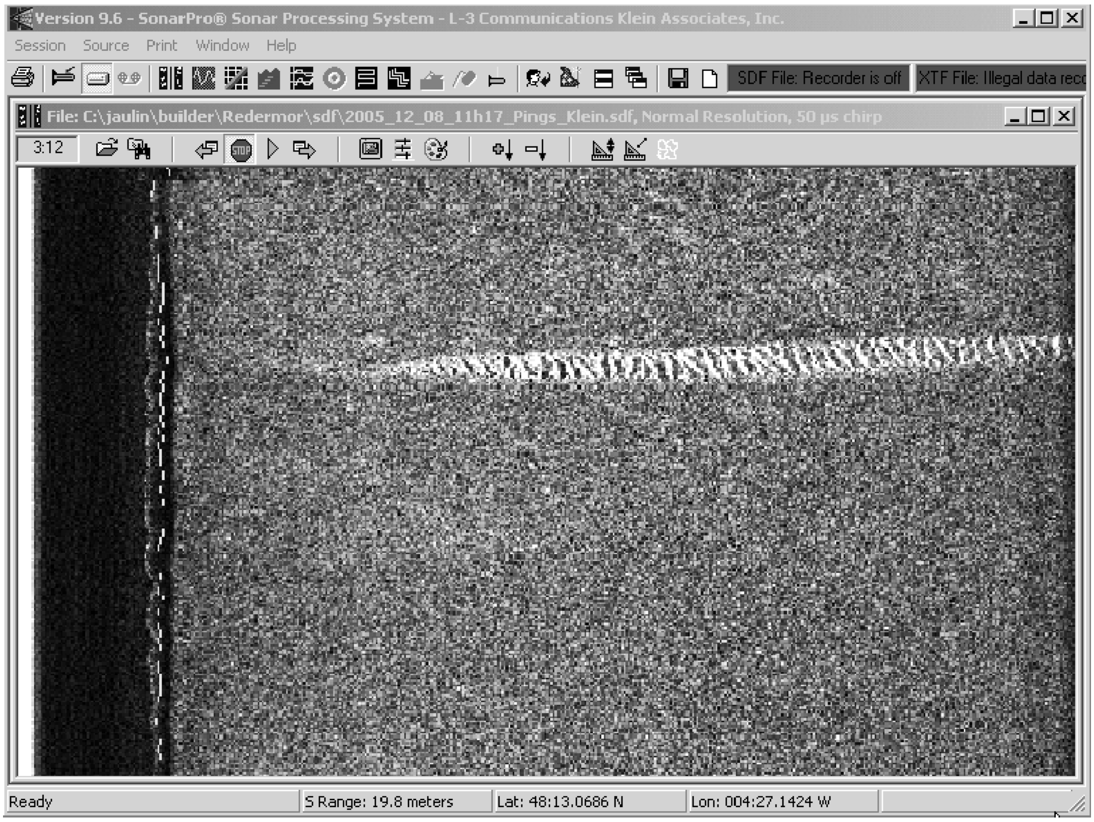
$$t_0 = 6000 \text{ s}, \quad \ell^0 = (-4.4582279^o, 48.2129206^o) \pm 2.5m$$
$$t_f = 12000 \text{ s}, \quad \ell^f = (-4.4546607^o, 48.2191297^o) \pm 2.5m$$

A sonar (KLEIN 5400 side scan sonar). Gives the distance r between the robot to the detected object.

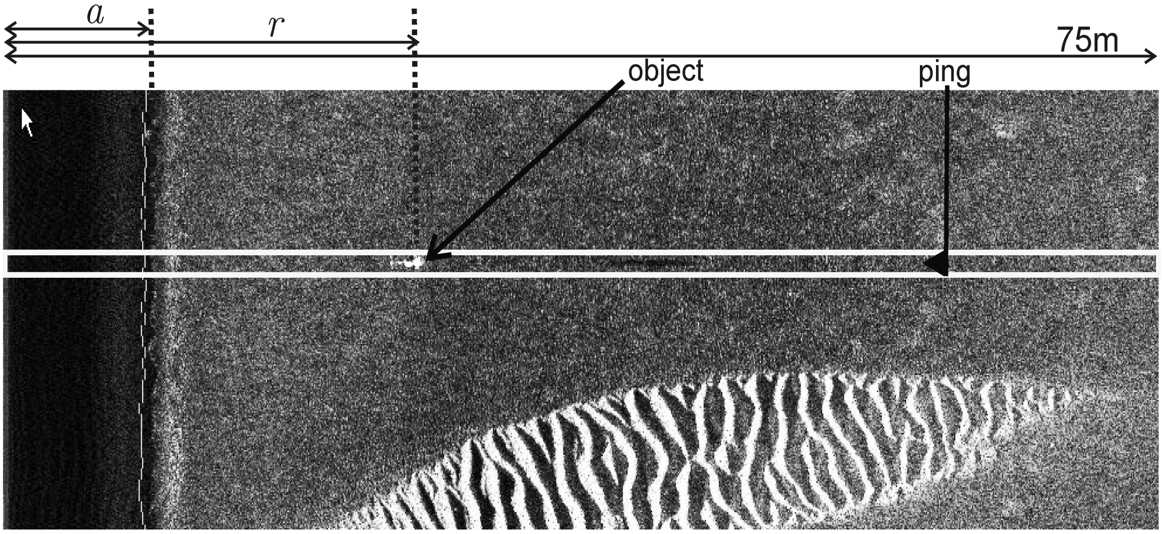





surface
m_z
p_z
r
$\mathbf{p}$
a
$\mathbf{m}$
bottom



Screenshot of the SonarPro software

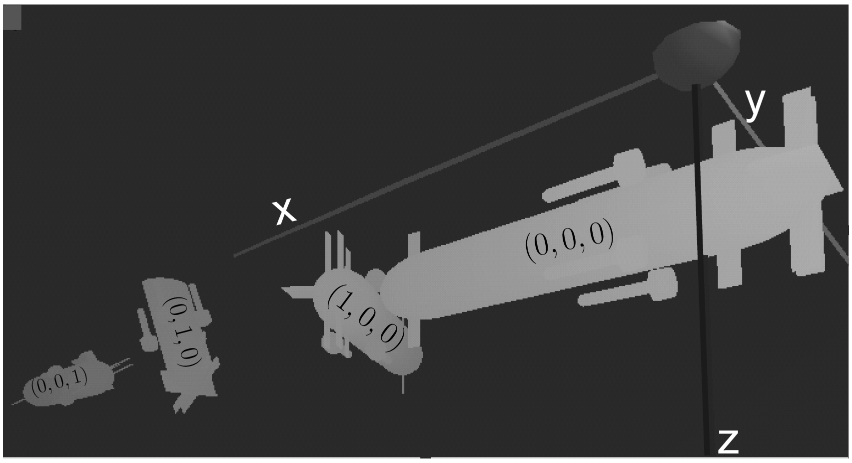


Detection of a mine using SonarPro

A Loch-Doppler. Returns the speed of the robot $\mathbf{v}_r$ and the altitude a of the robot $\pm$ 10cm.

A Gyrocompass (Octans III from IXSEA). Returns the roll ϕ, the pitch θ, and the head ψ the robots.

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} \in \begin{pmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{\psi} \end{pmatrix} + \begin{pmatrix} 1.75 \times 10^{-4}.\,[-1,1] \\ 1.75 \times 10^{-4}.\,[-1,1] \\ 5.27 \times 10^{-3}.\,[-1,1] \end{pmatrix}.$$



4 Data

For each time $t \in \{6000.0, 6000.1, 6000.2, \ldots, 11999.4\}$, we get intervals for

$$\phi(t), \theta(t), \psi(t), v_r^x(t), v_r^y(t), v_r^z(t), a(t).$$

Six mines have been detected by the sonar:

i	0	1	2	3	4	5	6	7	8	9	10	11
$\tau(i)$	7054	7092	7374	7748	9038	9688	10024	10817	11172	11232	11279	11688
$\sigma(i)$	1	2	1	0	1	5	4	3	3	4	5	1
$\tilde{r}(i)$	52.42	12.47	54.40	52.68	27.73	26.98	37.90	36.71	37.37	31.03	33.51	15.05

5 Constraints satisfaction problem

$$t \in \{6000.0, 6000.1, 6000.2, \ldots, 11999.4\},$$

$$i \in \{0, 1, \ldots, 11\},$$

$$\begin{pmatrix} p_x(t) \\ p_y(t) \end{pmatrix} = 111120 \begin{pmatrix} 0 & 1 \\ \cos\left(\ell_y(t) * \frac{\pi}{180}\right) & 0 \end{pmatrix} \begin{pmatrix} \ell_x(t) - \ell_x^0 \\ \ell_y(t) - \ell_y^0 \end{pmatrix},$$

$$\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t)),$$

$$\mathbf{R}_\psi(t) = \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) & 0 \\ \sin\psi(t) & \cos\psi(t) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{R}_\theta(t) = \begin{pmatrix} \cos\theta(t) & 0 & \sin\theta(t) \\ 0 & 1 & 0 \\ -\sin\theta(t) & 0 & \cos\theta(t) \end{pmatrix},$$

$$\mathbf{R}_\varphi(t) = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & \cos\varphi(t) & -\sin\varphi(t) \\ 0 & \sin\varphi(t) & \cos\varphi(t) \end{array} \right),$$

$$\mathbf{R}(t) = \mathbf{R}_\psi(t)\mathbf{R}_\theta(t)\mathbf{R}_\varphi(t),$$

$$\dot{\mathbf{p}}(t) = \mathbf{R}(t).\mathbf{v}_r(t),$$

$$||\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))|| \ = r(i),$$

$$\mathbf{R}^\mathsf{T}(\tau(i))\,(\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))) \in [0] \times [0,\infty]^{\times 2},$$

$$m_z(\sigma(i)) - p_z(\tau(i)) - a(\tau(i)) \in [-0.5, 0.5]$$

6 Contraction

Example 1. CSP:

$$\begin{aligned} z &= x + y \\ x &\in [1, 5], y \in [2, 4], z \in [6, \infty] \end{aligned}$$

Contractions

$$\begin{aligned} z = x + y \Rightarrow \quad z \in \; & [6, \infty] \cap ([1, 5] + [2, 4]) \\ & = [6, \infty] \cap [3, 9] = [6, 9]. \\ x = z - y \Rightarrow \quad x \in \; & [1, 5] \cap ([6, \infty] - [2, 4]) \\ & = [1, 5] \cap [2, \infty] = [2, 5]. \\ y = z - x \Rightarrow \quad y \in \; & [2, 4] \cap ([6, \infty] - [1, 5]) \\ & = [2, 4] \cap [1, \infty] = [2, 4]. \end{aligned}$$

Example 2

CSP:

$$\left\{\begin{array}{c}\mathbf{y}=\mathbf{A}\mathbf{x}\\ \mathbf{x}\in[\mathbf{x}]\subset\mathbb{R}^2,\mathbf{y}\in[\mathbf{y}]\subset\mathbb{R}^2\\ \mathbf{A}\in[\mathbf{A}]\end{array}\right.$$

Decomposition

$$\left\{\begin{array}{rcl}y_1&=&a_{11}x_1+a_{12}x_2\\ y_2&=&a_{21}x_1+a_{22}x_2\end{array}\right. \tag{1}$$

i.e.

$$\left\{\begin{array}{c}z_1=a_{11}x_1,\ z_2=a_{12}x_2,\ y_1=z_1+z_2\\ z_3=a_{21}x_1,\ z_4=a_{22}x_2,\ y_2=z_3+z_4\\ z_1\in[-\infty,\infty],\ldots,z_4\in[-\infty,\infty]\end{array}\right. .$$

Example 3

CSP:

$$\left\{\begin{array}{l}\mathbf{y}=\mathbf{R}\mathbf{x}\\ \mathsf{Rot}(\mathbf{R})\end{array}\right. , \qquad \begin{array}{c}\mathbf{x}\in[\mathbf{x}]\,,\mathbf{y}\in[\mathbf{y}]\\ \mathbf{R}\in[\mathbf{R}]\end{array}$$

Contractions

$$\begin{array}{rcl}[\mathbf{y}] & : & =[\mathbf{y}]\cap[\mathbf{R}]*[\mathbf{x}],\\ [\mathbf{x}] & : & =[\mathbf{x}]\cap[\mathbf{R}]^{\mathsf{T}}*[\mathbf{y}],\end{array}$$

Example 4. Consider the constraint

$$\begin{array}{c}\dot{\mathbf{p}}(t)=\mathbf{R}(t).\mathbf{v}_r(t)\\ \forall t\in[t_0,t_1],\mathbf{R}(t)\in[\mathbf{R}],\mathbf{v}_r(t)\in[\mathbf{v}_r]\end{array}$$

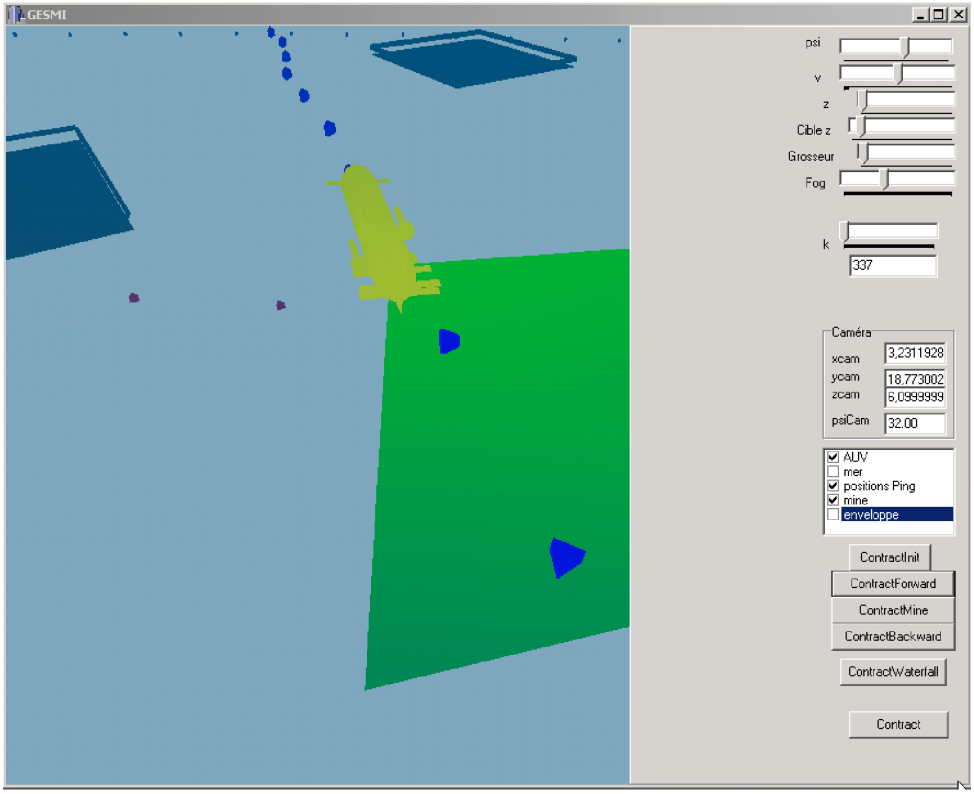
Since

$$\mathbf{p}(t_1)=\mathbf{p}(t_0)+\int_{t_0}^{t_1}\mathbf{R}(t).\mathbf{v}_r(t)\in\mathbf{p}(t_0)+(t_1-t_0).\left[\mathbf{R}\right].\left[\mathbf{v}_r\right],$$

the domains for $\mathbf{p}(t_0)$ and $\mathbf{p}(t_1)$ can be contracted as follows

$$\begin{array}{rcl}\left[\mathbf{p}\right](t_1) & = & \left[\mathbf{p}\right](t_1)\cap\left(\left[\mathbf{p}\right](t_0)+(t_1-t_0).\left[\mathbf{R}\right].\left[\mathbf{v}_r\right]\right),\\ \left[\mathbf{p}\right](t_0) & = & \left[\mathbf{p}\right](t_0)\cap\left(\left[\mathbf{p}\right](t_1)+(t_0-t_1).\left[\mathbf{R}\right].\left[\mathbf{v}_r\right]\right).\end{array}$$

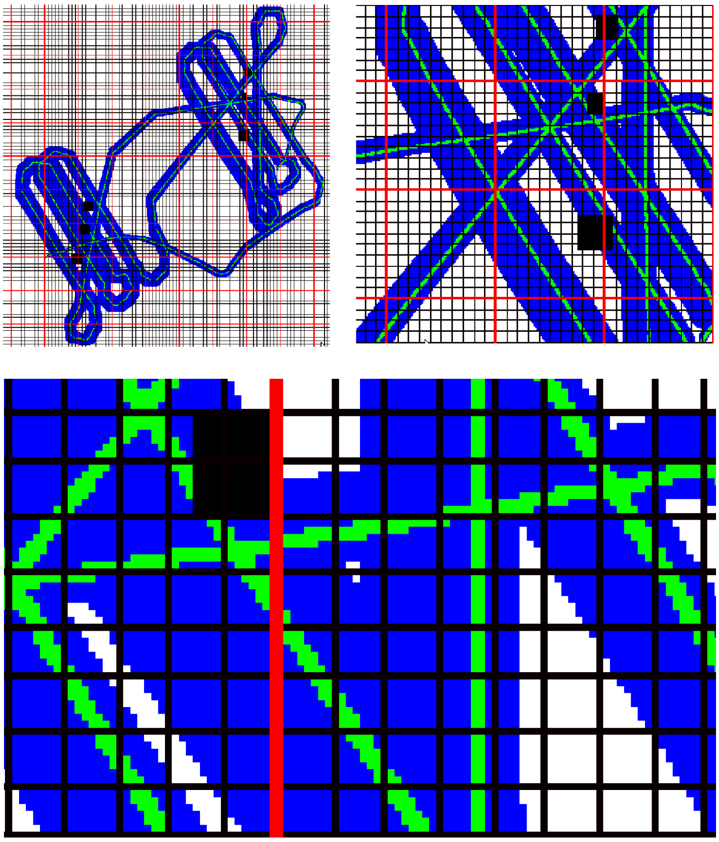
7 GESMI



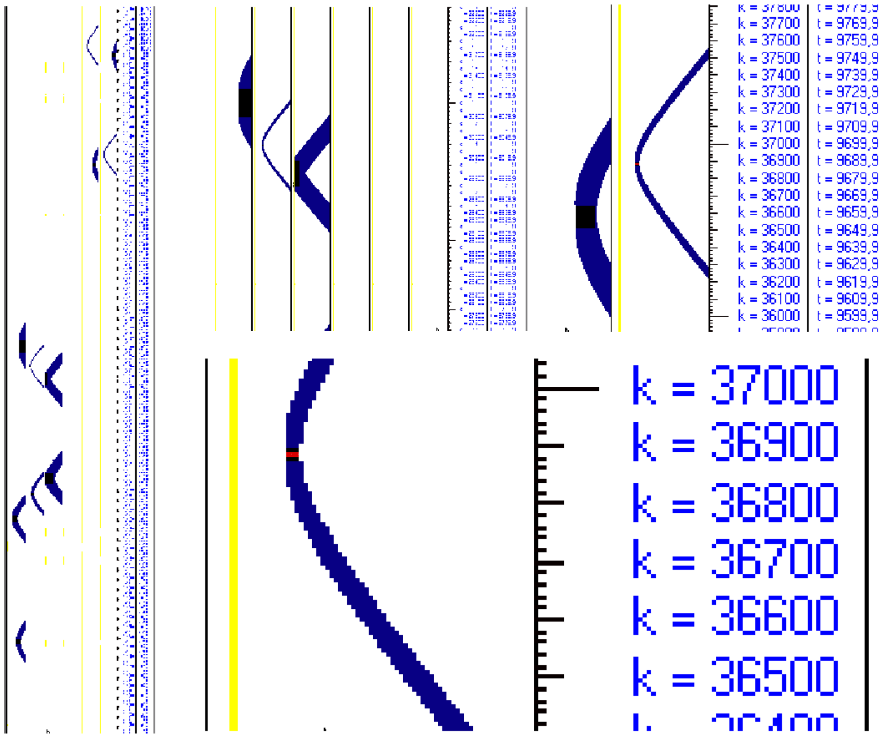
GESMI (Guaranteed Estimation of Sea Mines with Intervals)

```
//------------------------------------------------------------------
void Cmult(imatrix& C, imatrix& A, imatrix& B)
{     for (int i=1; i<=C.dim1() ; i++)
      for (int j=1; j<=C.dim2() ; j++)
          {  box a=Row(A,i);
             box b=Column(B,j);
             interval c=C.GetVal(i,j);
             CProdScalaire(c, a, b); C.SetVal(i,j,c);
             for (int k=1; k<=A.dim2(); k++)
               {A.SetVal(i,k,a[k]);B.SetVal(k,j,b[k]);};
}          }
//------------------------------------------------------------------
void Cmult(box& c, imatrix& A, box& b)
{     for (int i=1; i<=c.dim; i++)
          {  box a=Row(A,i);
             CProdScalaire(c[i],a,b);
             for (int k=1; k<=b.dim; k++)   A.SetVal(i,k,a[k]);
}         }
//------------------------------------------------------------------
void Crot(imatrix& R)
{     imatrix Rt=Transpose(R);
      imatrix I=iEye(R.dim1());
      Cmult(I,R,Rt);
}
//------------------------------------------------------------------
void Cantisym(imatrix& A)
{     for (int i=1; i<=A.dim1(); i++)
      { A.SetVal(i,i,interval(-0,0));
        for (int j=i+1; j<=A.dim1(); j++)
         {   A.SetVal(j,i,Inter(-A(i,j),A(j,i)));
             A.SetVal(i,j,Inter(A(i,j),-A(j,i)));
}     }  }
//
```

```
//----------------------------------------------------------------
int TForm1::Contract_Forward(void)
{     for (int k=0;k<kmax;k++)
            P[k+1].Intersect(P[k]+dT*Rot[k]*vr[k]);
}
//----------------------------------------------------------------
int TForm1::Contract_Backward(void)
{     for (int k=kmax-2;k>=0;k--)
            P[k].Intersect(P[k+1]-dT*Rot[k]*vr[k]);
}
//----------------------------------------------------------------
int TForm1::Contract_Mine(void)
{    for (int k=0;k<kmax;k++)
         for (int km=0;km<kmmax;km++)
             if (W[k].vu[km])
             {    Cplus(mines[km].P[3],P[k][3],a[k],1);
                  Cdistance(W[k].r[km],P[k],mines[km].P);
                  W[k].e[km].Intersect(P[k]-mines[km].P);
                  W[k].e[km].Intersect(Rot[k]*W[k].er[km]);
                  Cmoins(W[k].e[km],P[k],mines[km].P,-1);
}            }
//----------------------------------------------------------------
```



Trajectory reconstructed by GESMI



Waterfall reconstructed by GESMI