Télécom SudParis
École de l'institut Mines-Télécom
9 Rue Charles Fourier, 91000 Évry

KERDONCUFF Tanguy
TADROS Antoine

Final Year Project : Stereo Camera Tracking

Table des matières

Table des figures

Introduction

For our final year project we worked on an algorithm that can track targets seen by a set of two cameras. We followed the work of the "HOUSSINEAU et al. : a unified approach for multi-object triangulation" article.

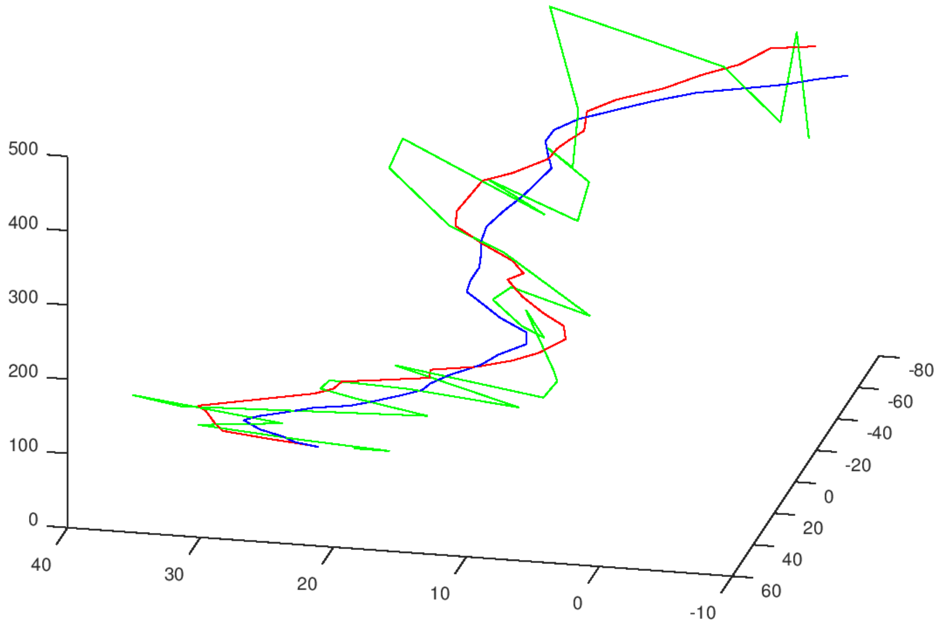

FIGURE 1 – Tracking example

1 Our Problematic

We want to track targets which are filmed by two calibrated cameras of focal length f. Those cameras will be our optical sensors. We want to track those targets from the data given by the cameras.

2 Modelisation

2.1 Projection on a single camera

Both cameras give use a projection of the real-world on a 2D space. The corresponding operation for a camera at $(0, 0, 0)$ is :

$$p_{camera}(x, y, z) = (\frac{fx}{z}, \frac{fy}{z})$$

For our work, we need this operation to be linear. Thus we will introduce homogeneous coordinates.

Homogeneous coordinates are a mean to linearize operations on vectors in order to write them as a matrix operation. We need to add an extra-coordinate to the vector state representation.

For example :
(x, y, z) becomes (x, y, z, α)
We can now write :

$$P_{camera} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

And our projection becomes, for $\alpha = 1$:

$$\begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = P_{camera} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

We can then get the projection by dividing the 1^{st} and 2^{nd} coordinate by the 3^{rd}

2.2 Projection on a single camera

We now assume that both cameras are calibrated, that the left camera C_l is located at $(0, 0, 0)$ and the right camera C_r is at $(b, 0, 0)$ where b is called the baseline. We denote the projection on the left camera as P_l and on the right camera as P_r.

Since C_l is at $(0, 0, 0)$, $P_l = f \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

And $P_r = f \begin{pmatrix} 1 & 0 & 0 & b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

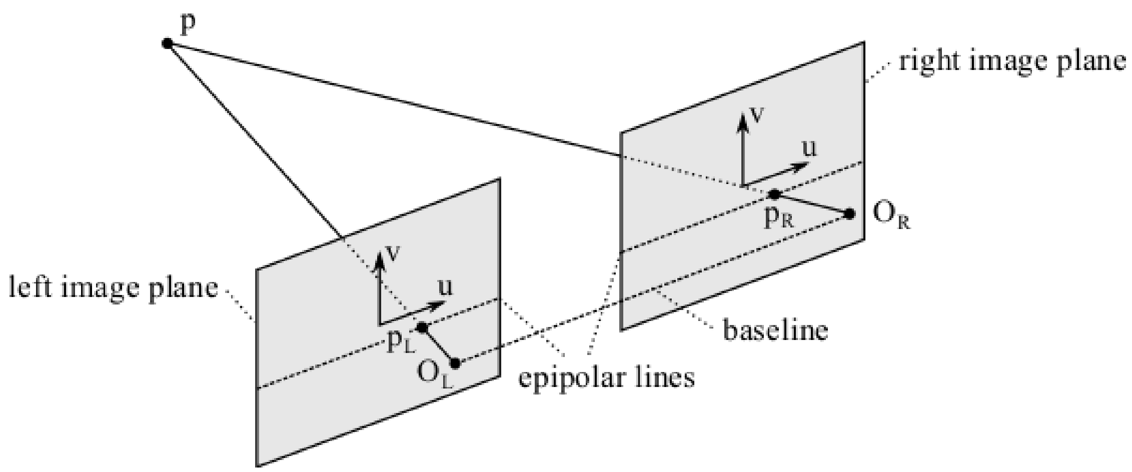

Figure 2 – Stereo geometry

2.3 Disparity Space

The uncertainty of the object coordinates on the cameras are gaussians. Since the projection on the cameras are non-linear, the uncertainty in the real world is not gaussian anymore. We have to use another space to represent the object and it's gaussian uncertainty since we want to use the Kalman update to track the target. This is where the disparity space is useful.

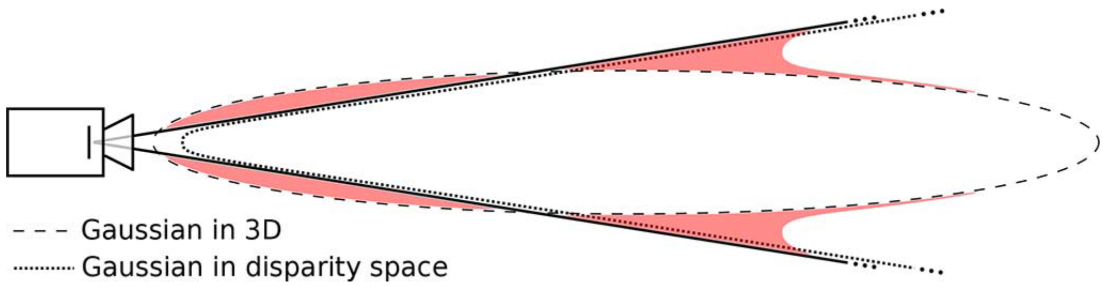

FIGURE 3 – Uncertainty

The projection from the measure space created by the cameras to the disparity space is linear.

Projection on the disparity space :

$$P = f\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & b \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

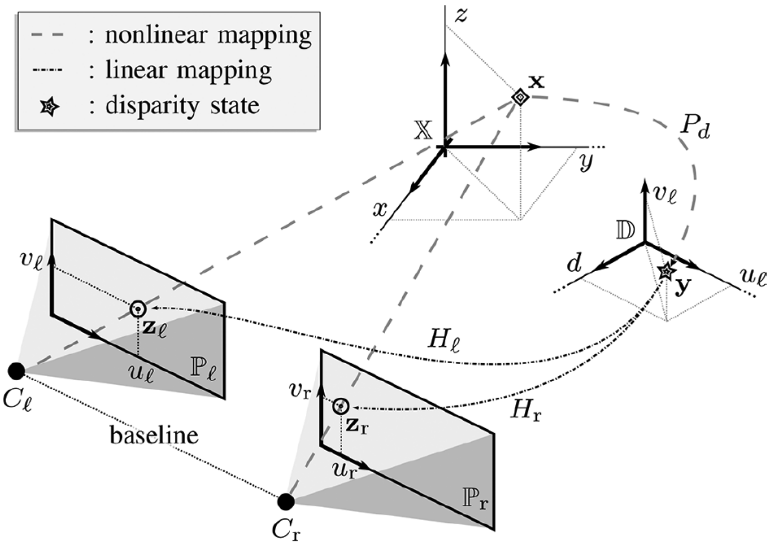

FIGURE 4 – Projections

3 Targets movements

We assume that the cameras get an observation at a frequency equal to $\frac{1}{T_e}$ and that the target have a linear movement with a uncertainty represented by σ_Q. The uncertainty on the observation is pictured by σ_{pl} and σ_{pv}

State equation :

$$X_t = X_t * F + U$$

$$F = \begin{pmatrix} 1 & T_e & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_e & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_e \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U \backsim N(0, Q)$$

$$Q = \sigma_Q^2 \begin{pmatrix} (T_e^3)/3 & (T_e^2)/2 & 0 & 0 & 0 & 0 \\ (T_e^2)/2 & T_e & 0 & 0 & 0 & 0 \\ 0 & 0 & (T_e^3)/3 & (T_e^2)/2 & 0 & 0 \\ 0 & 0 & (T_e^2)/2 & T_e & 0 & 0 \\ 0 & 0 & 0 & 0 & (T_e^3)/3 & (T_e^2)/2 \\ 0 & 0 & 0 & 0 & (T_e^2)/2 & T_e \end{pmatrix}$$

Observation equation :

$$Y_t = X_t * H + V$$

where H is the projection on one of the cameras and $V \backsim N(0, R)$ with
$R = \begin{pmatrix} \sigma_{pl}^2 & 0 \\ 0 & \sigma_{pv}^2 \end{pmatrix}$

4 Tracking

To track the target, we will :

1. produce particle around the observation mapped onto the disparity space
2. Map it back into the real space
3. Apply the Markov transition
4. Bring the particle back on the disparity space
5. Compute the new gaussian from the particle in the disparity space
6. Apply Kalman update in the disparity space
7. Sample the gaussian after the Kalman update to start again at the item 1

We keep the mean of the gaussian of the 6^{th} item at each iterations as the estimated position of the object.

5 Results

In this section we will comment the result. All the figures are representations of real trajectory (blue), the observations (green), and the estimated trajectory(red) in the real world. Purple points which correspond to particles. The green line is created by the direct mapping from the camera to the real space and the red one is created by the mapping from the disparity space to the real space.

The two cameras are in position (0,0,0) and (20,0,0).

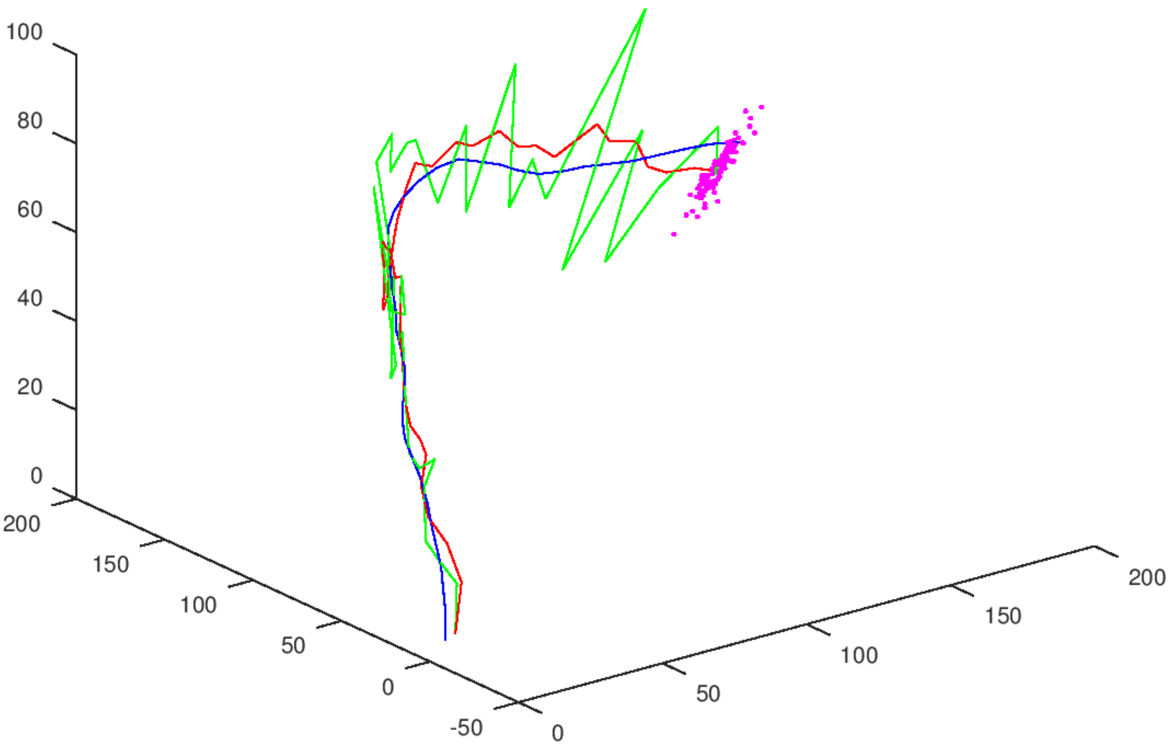

Figure 5 – Classical example

5.1 non-gaussian noise

It is not easy to see that the noise is not gaussian in a 2D plot but it is very clear in 3D. Anyway in the plot below we can feel that the green line doesn't seem to be gaussian around the blue line. Moreover we can see that that the noise change at each iteration. At the beginning their isn't a lot of noise, it is because the object is close to the cameras. When the object goes away from the cameras, it become really hard to track it because a small error in the camera will result on a big difference in the real space. That is why the noise is so important.

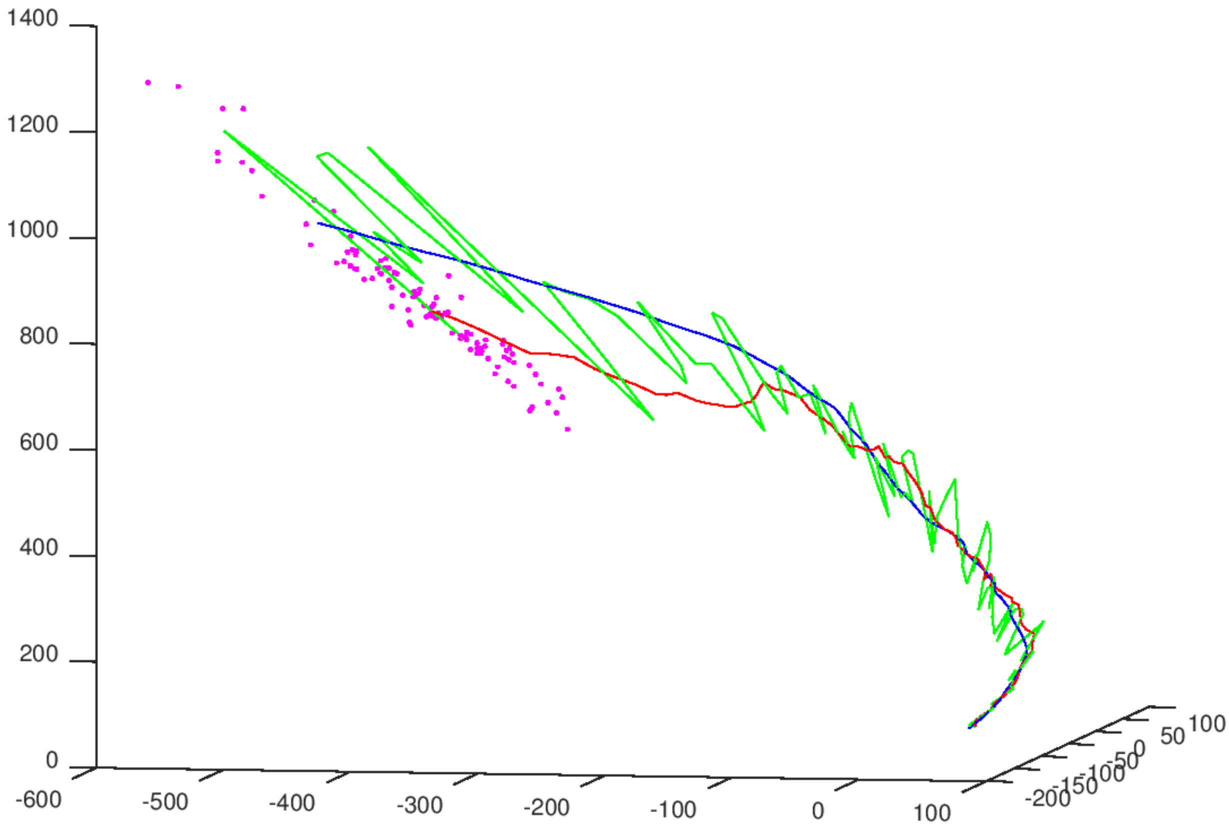

Figure 6 – Non-gaussian noise

5.2 Divergence

In the plot below we can see that their is a kind of divergence between the object and the estimation of the object which is clearly bad. This append when the object get really too far and the noise is too high. The algorithm don't trust the observation anymore and with the particles we can see that we are not confident on the position of the object.

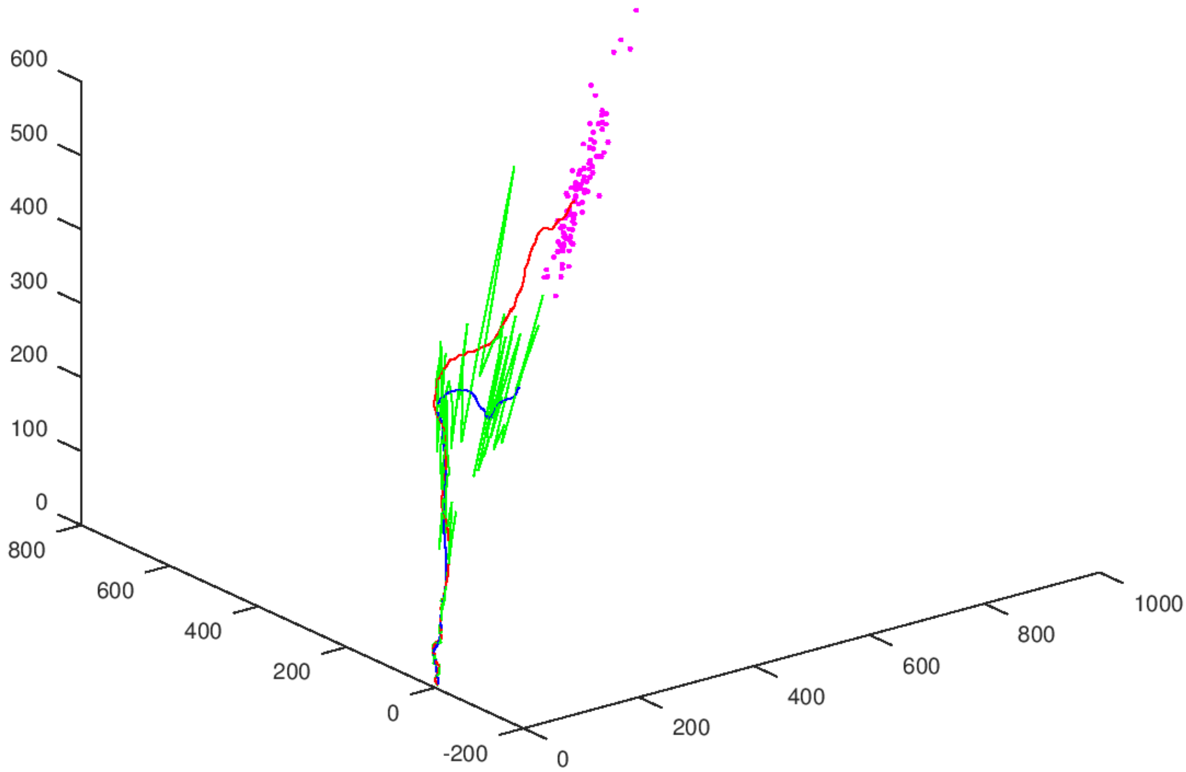

Figure 7 – Divergence

Sometime it is the exact opposite reaction when the object get too far, the algorithm trust only the observation and not anymore the prediction. This happen when the noise in the cameras is small. The particles are very close to the point which mean that we are very confident about the position. This happen at the end of the last plot (We can see the purple particles and the end).

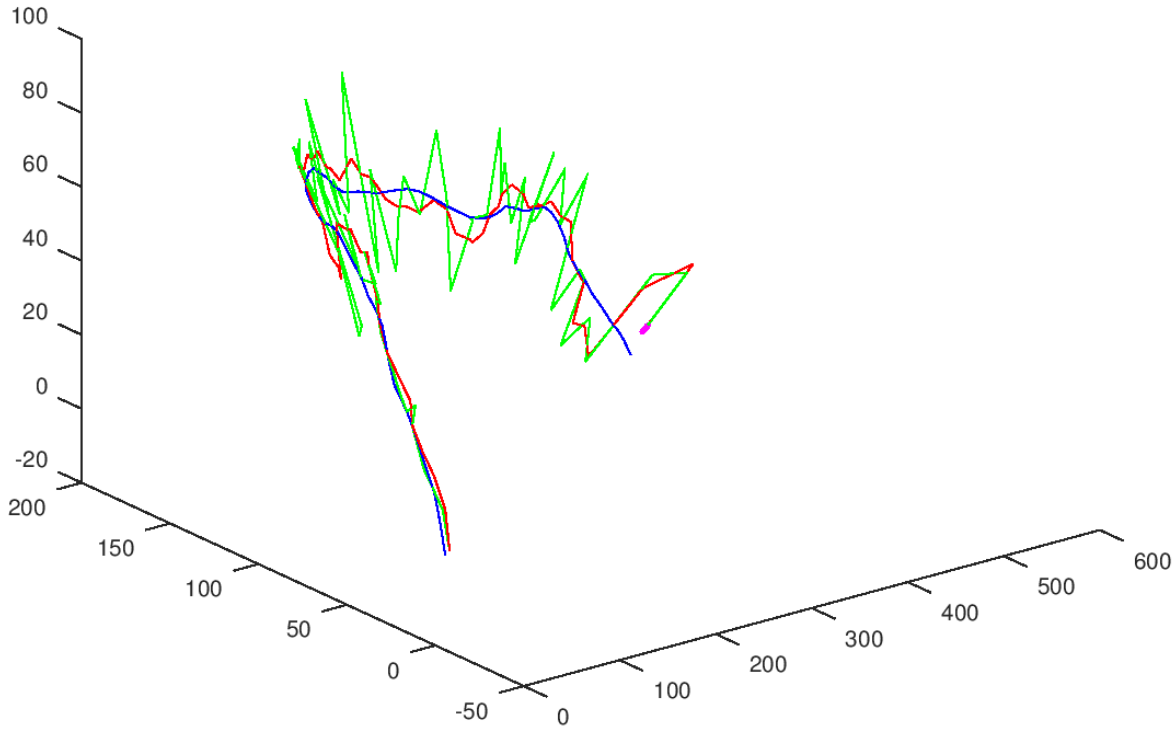

Figure 8 – Trust the observation

5.3 Changes through time

This is a classic example of what happen during the tracking when an object start close from the cameras and get away.

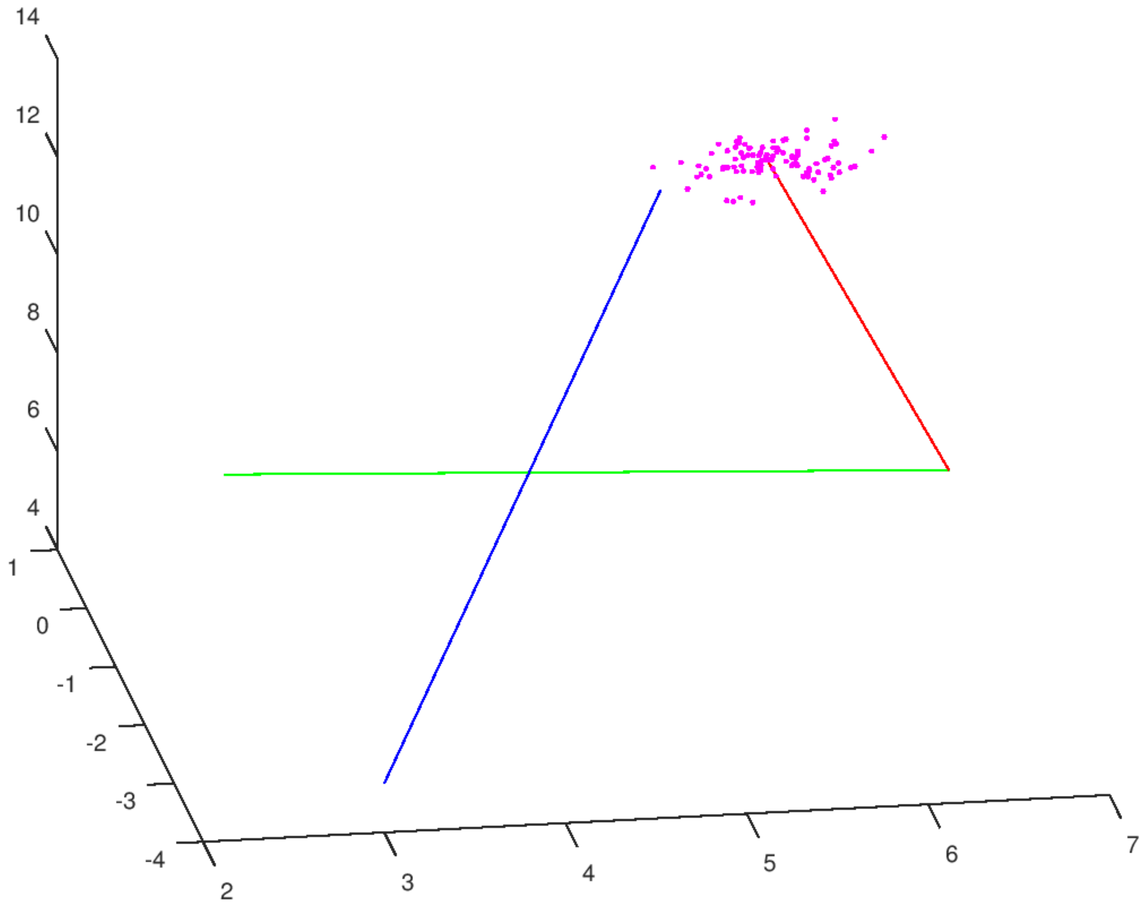

Figure 9 – Start 1

The plot below correspond to the position of the object at the iteration 0 and 1. Obviously the first estimated position (red) is the same as the observation (green). What is interesting is the next step : the next observation is not in a good position but with the prior knowledge of the movement of the object, the red line is very close to the blue at the second steps and we are quite confident about the position.

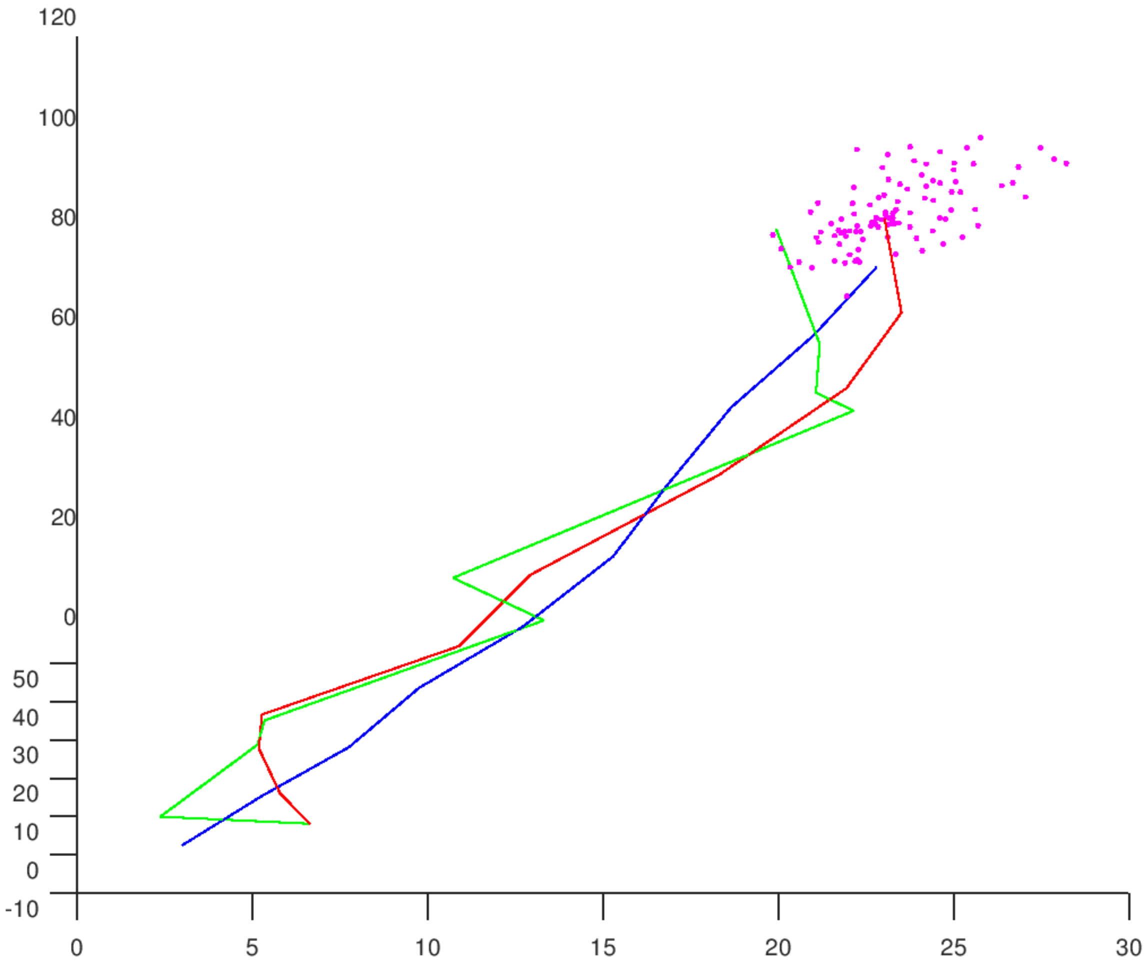

Figure 10 – Start 2

Then we track the object and we can notice that we are not confident anymore about the position of the object at this step.

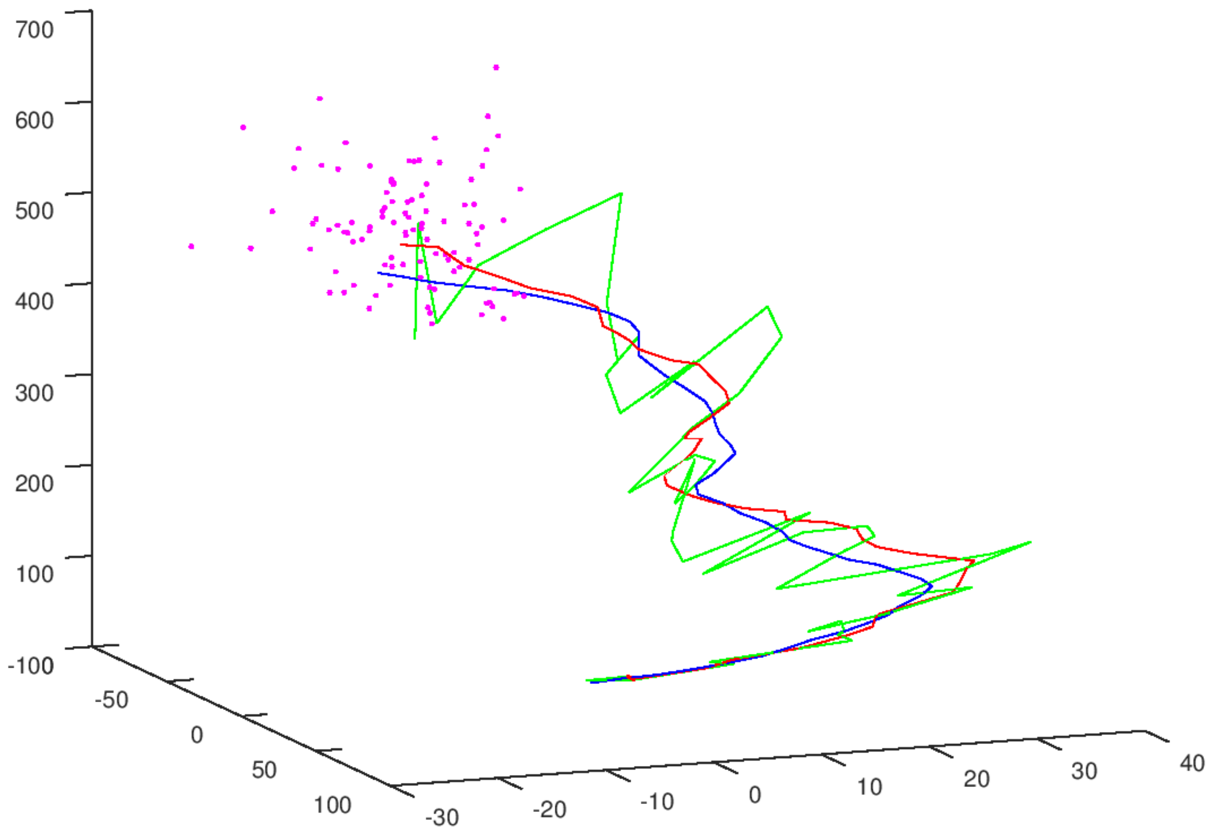

Figure 11 – Middle

The noise is getting higher by the time so we are less confident but the red line follow well the blue one.

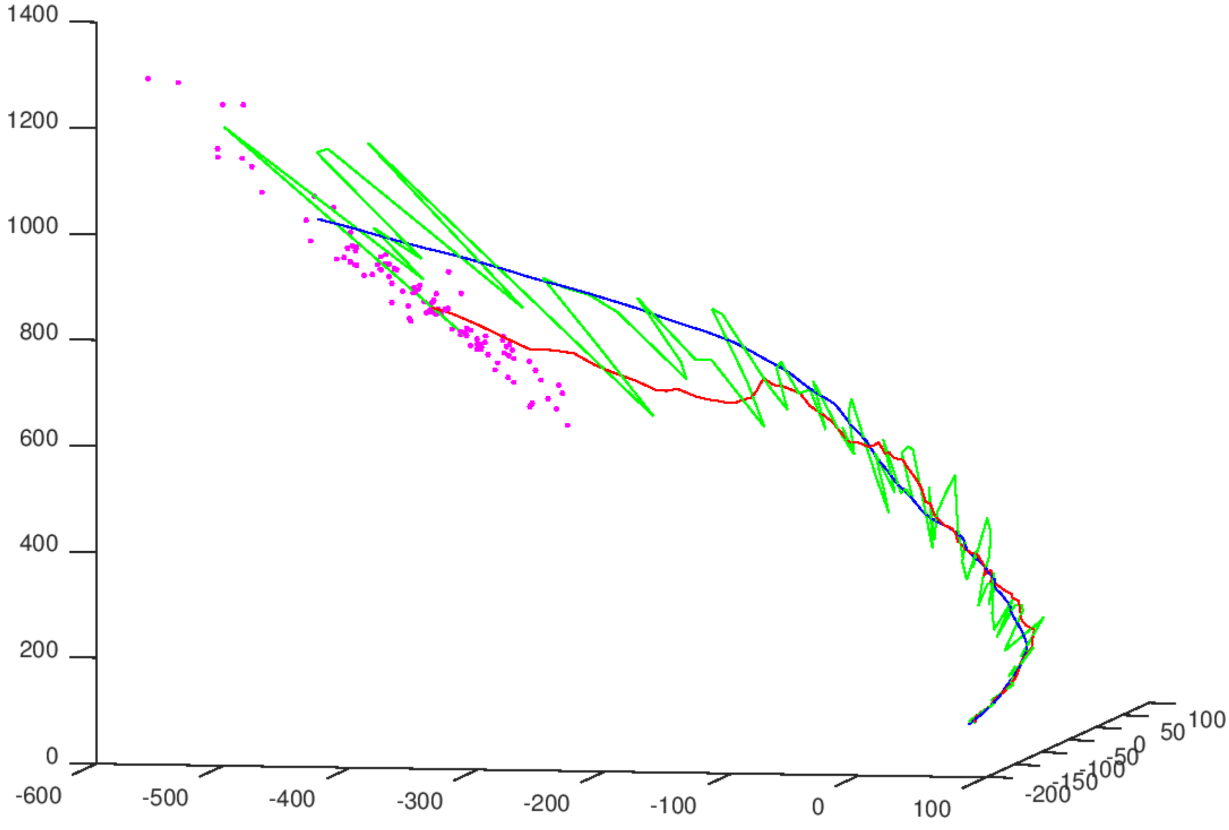

Figure 12 – End

At the end we have a divergence that we already talk about and which is common in this case. The object goes too far and the noise too high.

5.4 High noise in cameras

This situation appears when the observation of the object is really noisy. Obviously it is harder to track the object. The particles shows that we are not confident in the position but we still get a reasonable position of the object.

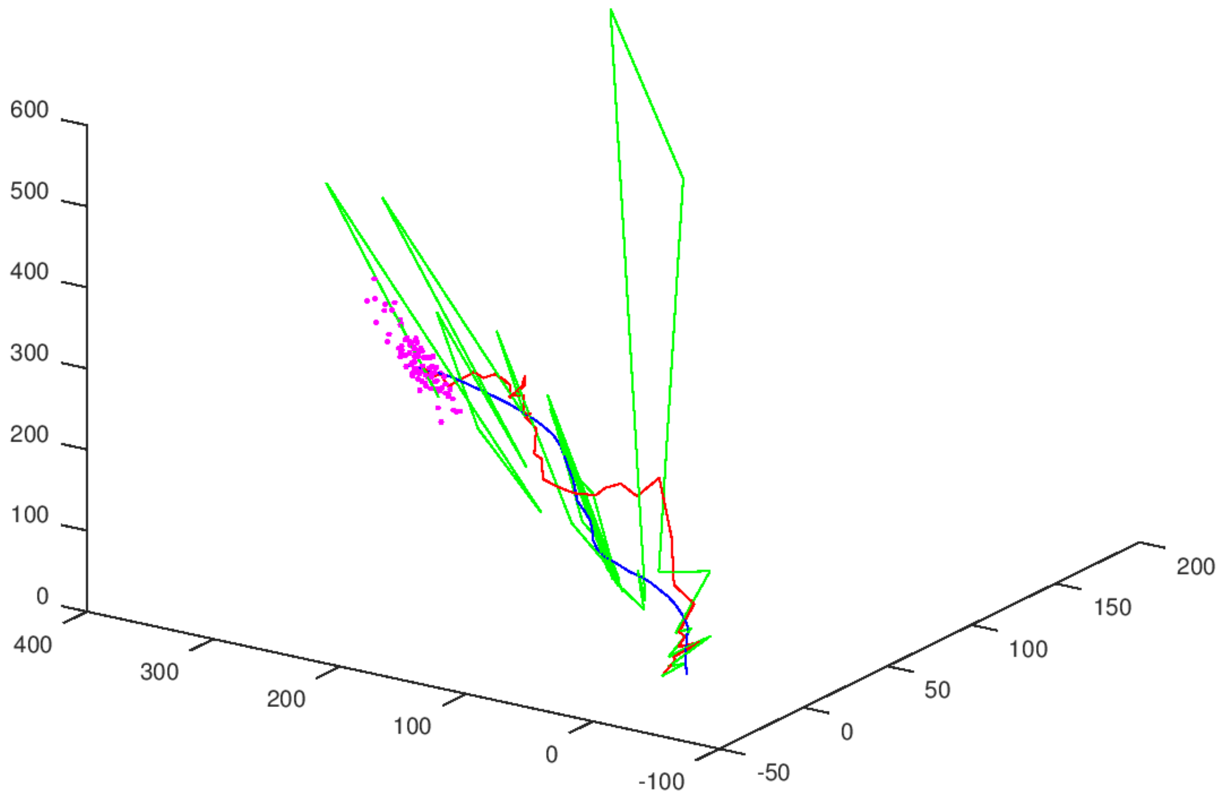

Figure 13 – High noise in cameras

5.5 Prediction fail totally

Sometimes, we get the plot below, we can see that the estimated position and the particles are not in the right position. Sometimes it happen also for the observation. I think this can appear because of the division by the last coordinate, and when this value is close to 0 we can easily get sometime totally wrong. A small mistake on this value will make a huge difference.

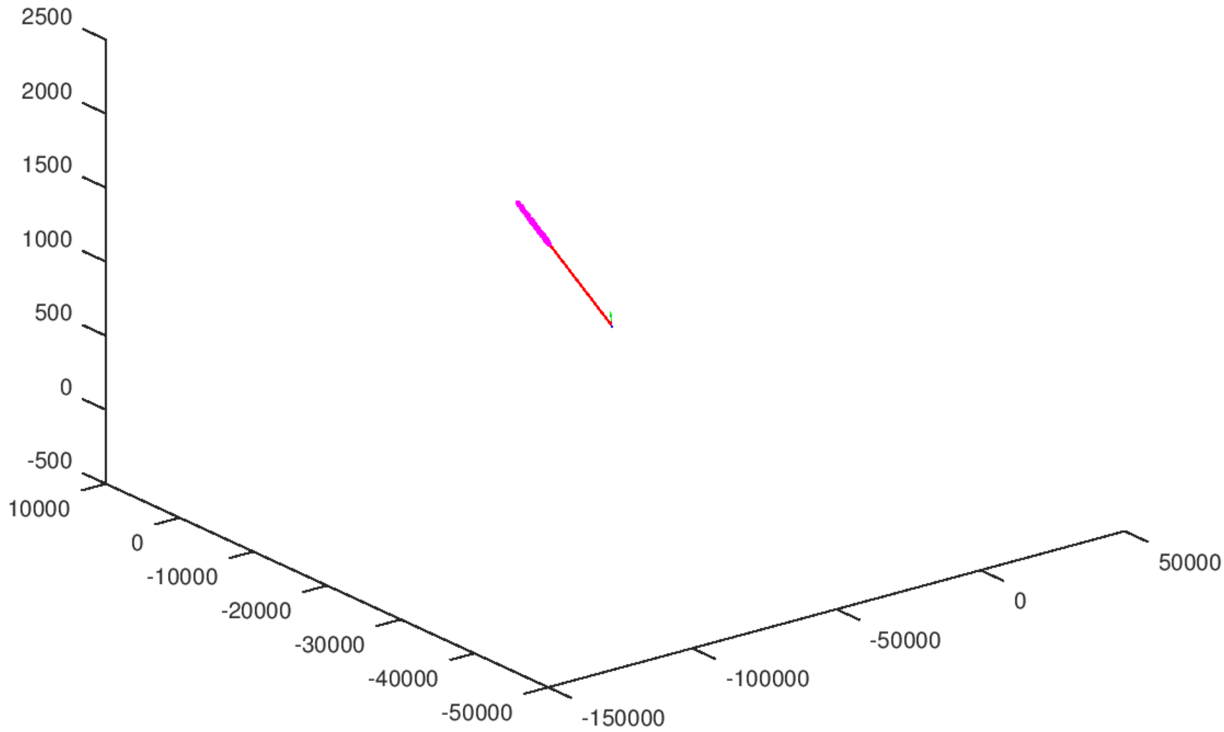

Figure 14 – Totally lost the position

Conclusion

In this report we have explain how we can track on object with two cameras by using a special algorithm. We have then discuss few situation of the problem to understand how to algorithm works. If we want to continue this work we would probably think about a multi-gaussian method. Instead of a simple Kalman update. We will approximate the particles that come back to the disparity space as a sum of gaussian instead of only one gaussian. This should work better because we can approwimate any distibution by a sum of gaussians (with enough gaussians).