Tracking 3D Ground Deformations From A Pair Of Point Clouds

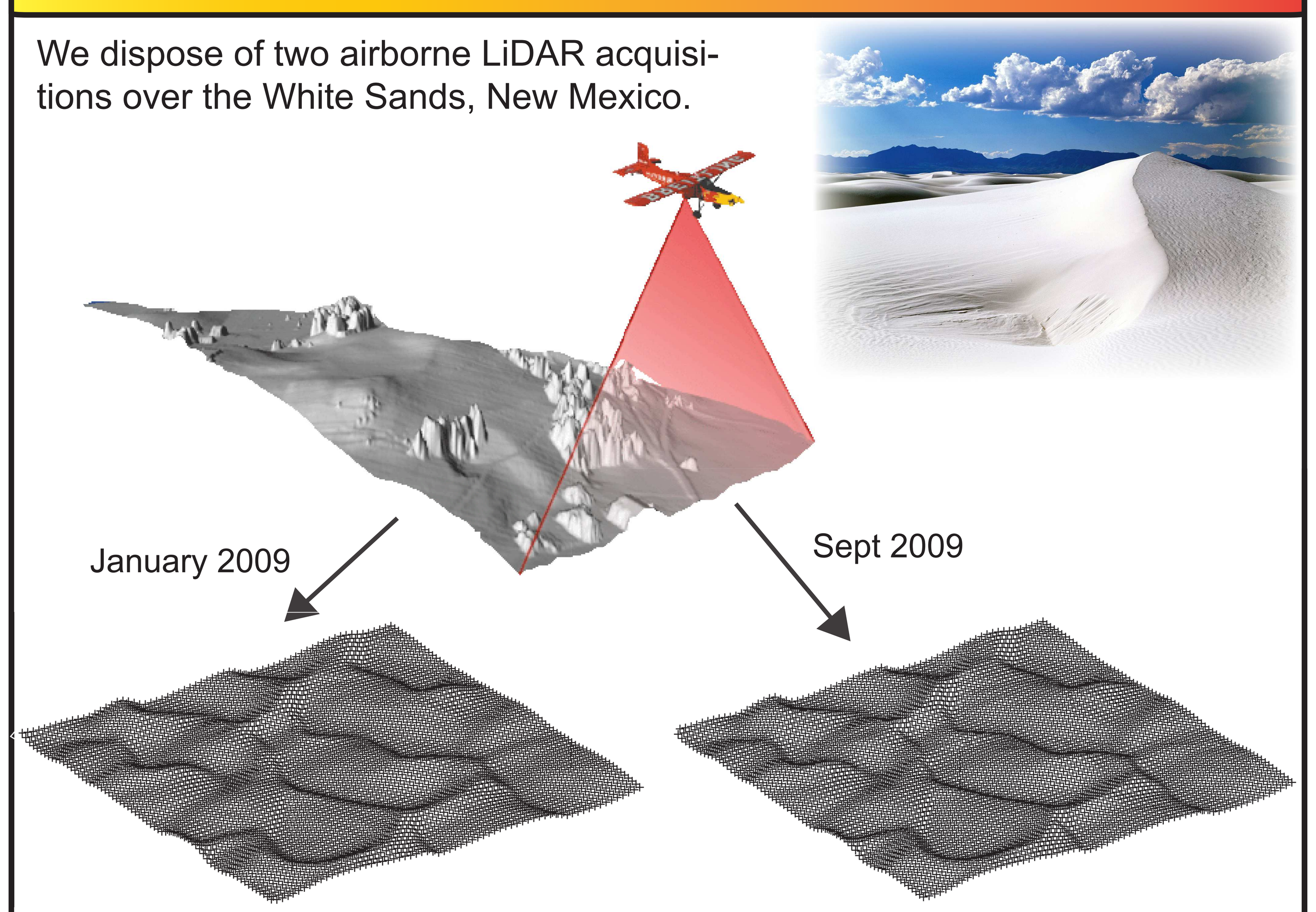
B. Conejo (Caltech), S. Leprince (Caltech), F. Ayoub (Caltech), J.P. Avouac (Caltech), and R.C. Ewing (Univ. Alabama)

1> Abstract

3D point clouds of the Earth topography have become largely available thanks to the explosion of LiDAR acquisitions and the progress of stereo-imaging. Recent studies have desmontrated the ability to use 3D point clouds to determine changes of Earth topography due to brutal events such as earthquakes or landslides but also due to continuous events such as dunes migration or glaciers flows.
We propose an approach to automaticly recover the deformation ground deformation using two point clouds. We apply our appraoch to track dunes of the White Sands desert, New Mexico.

2> Data

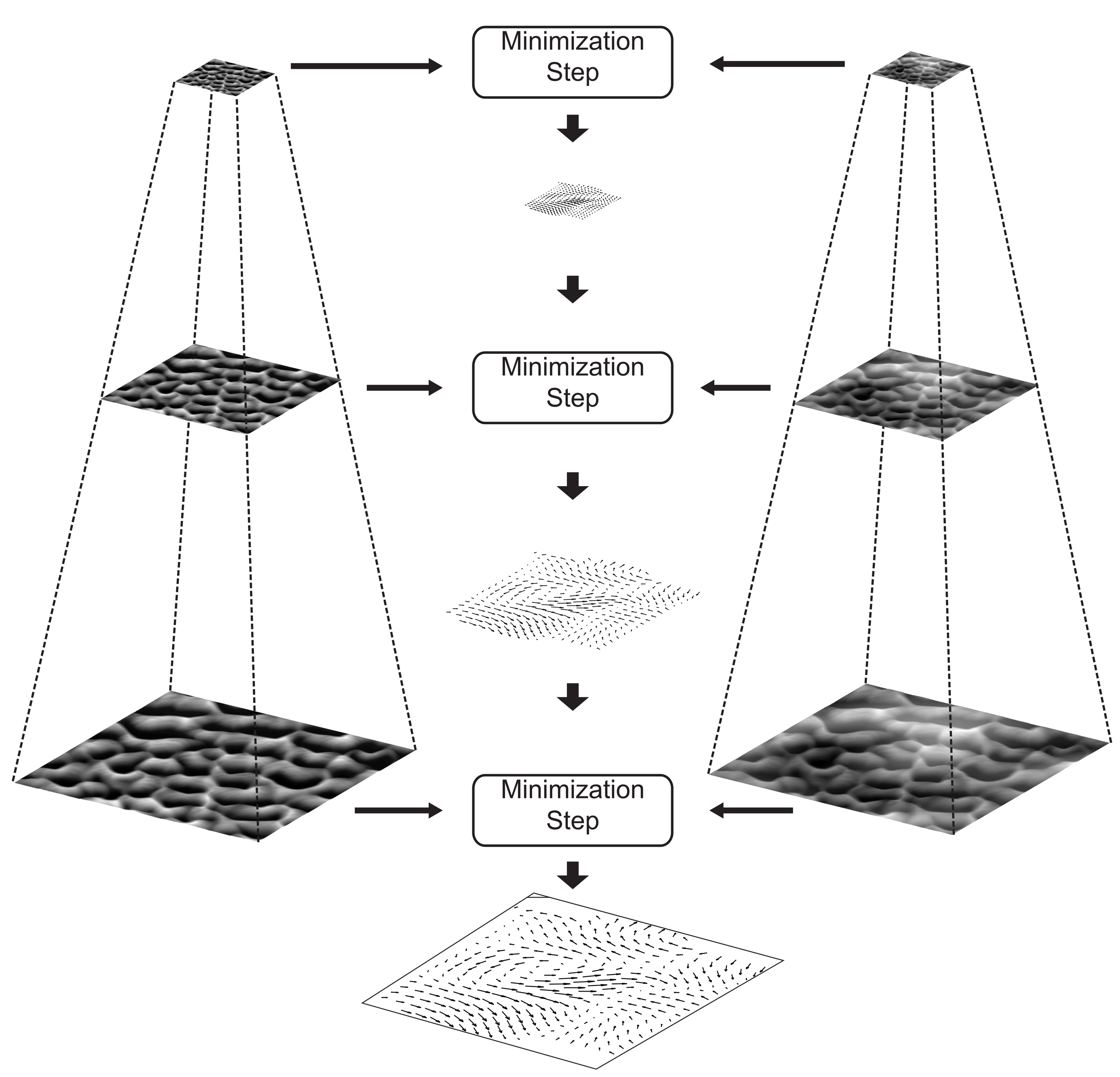
We dispose of two airborne LiDAR acquisitions over the White Sands, New Mexico.



What is the 3D ground deformation between two acquisitions?
How can we retrieve it automatically using computer vision algorithms?

5> Minimization strategy

As for optical flow, we proceed with a coarse to fine approach by building a pyramid for each point cloud. At each scale, we achieve a minimization step with a starting point provided by the previous scale.



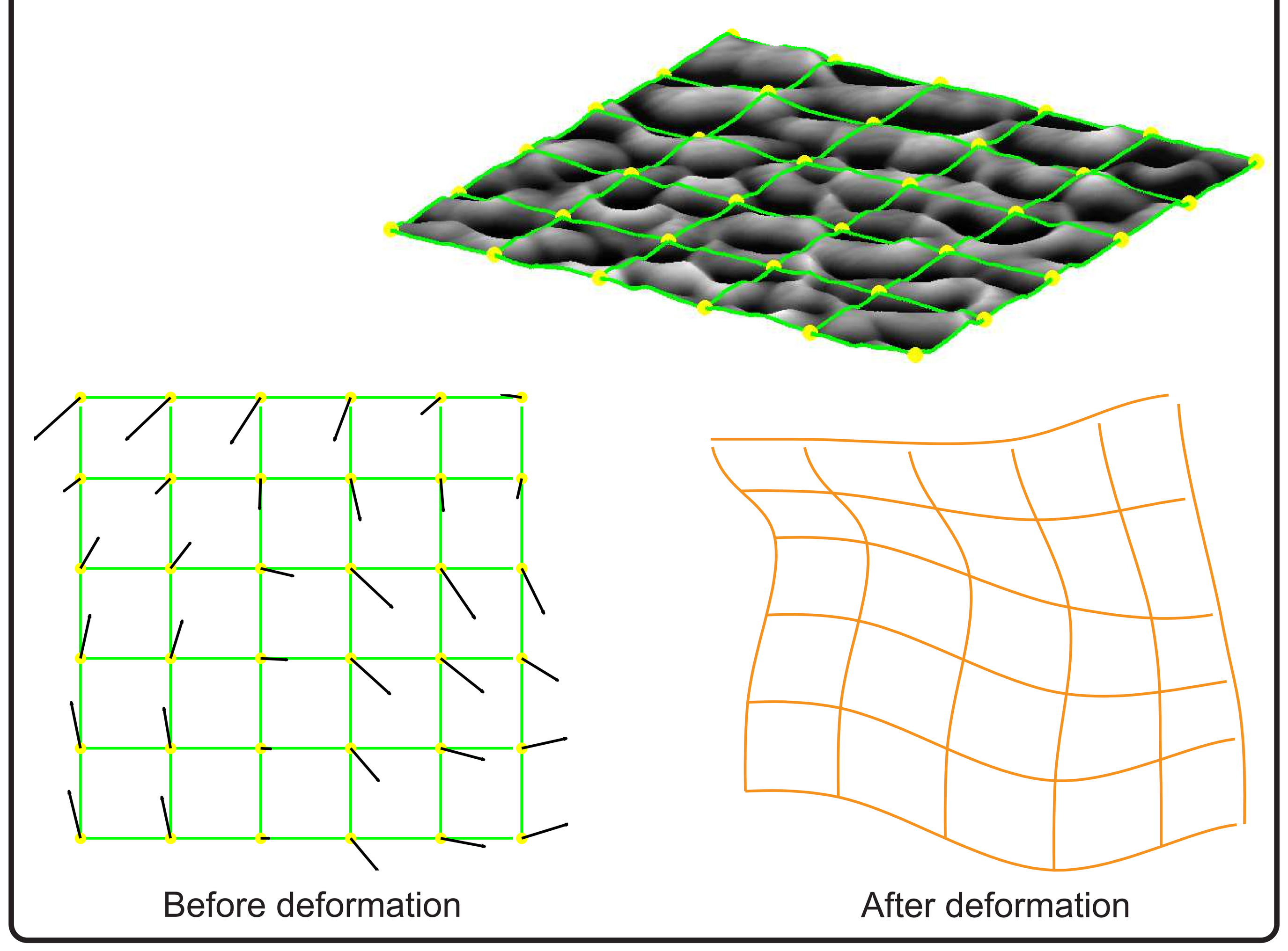
Two different methods can be used for the minimization step:

1) Gradient descent is fast but prone to local minimum (non-convex energy).

2) The MRF (Markov Random Field) solved by BLP (Binary Linear Programming) is slower (10 times more complex) but achieve better results.

3> Free Form Deformation (FFD)

The FFD framework (control points) is used to parametrize the 3D deformation. The yellow dots represent the control points.

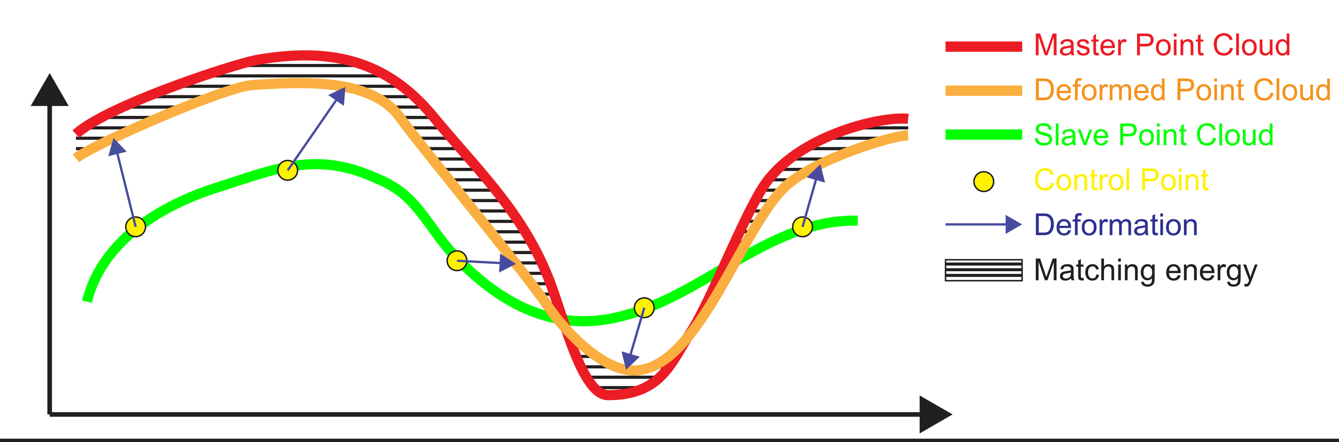


4> Cost function

We define a cost function to find the deformation that best transform one point cloud into another.

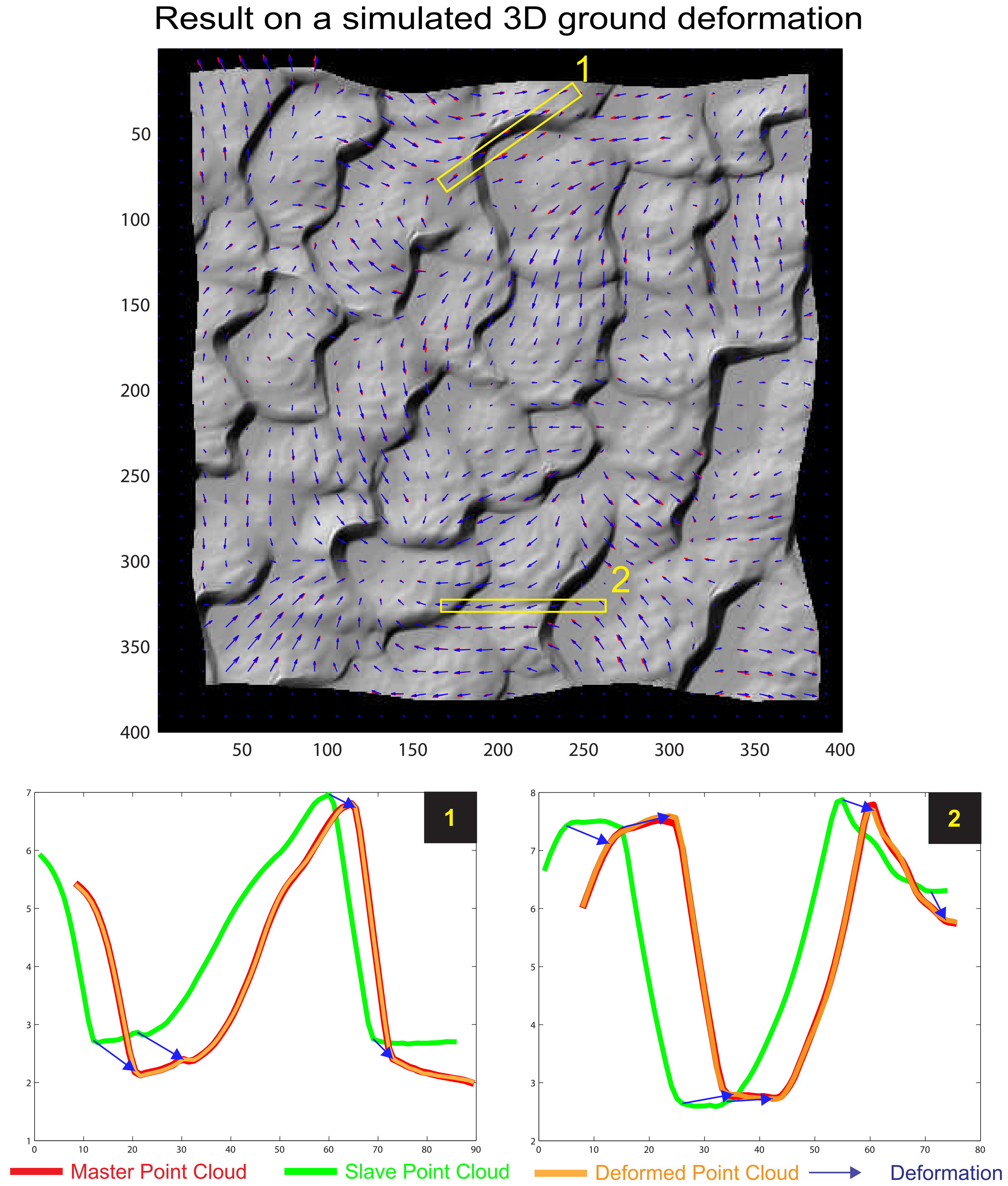
$$\mathbf{E} = \int \underbrace{\| \overbrace{P_M(x)}^{\text{Master Pt cloud}} , \overbrace{P_S \circ D(x)}^{\text{Deformed Pt cloud}} \|_2^2}_{\text{Enforces matching}} + \lambda \underbrace{\| \nabla \overbrace{D(x)}^{\text{Deformation}} \|_2^2}_{\text{Smothness of deformation}} dx$$

The figure below illustrates how the matching is enforced



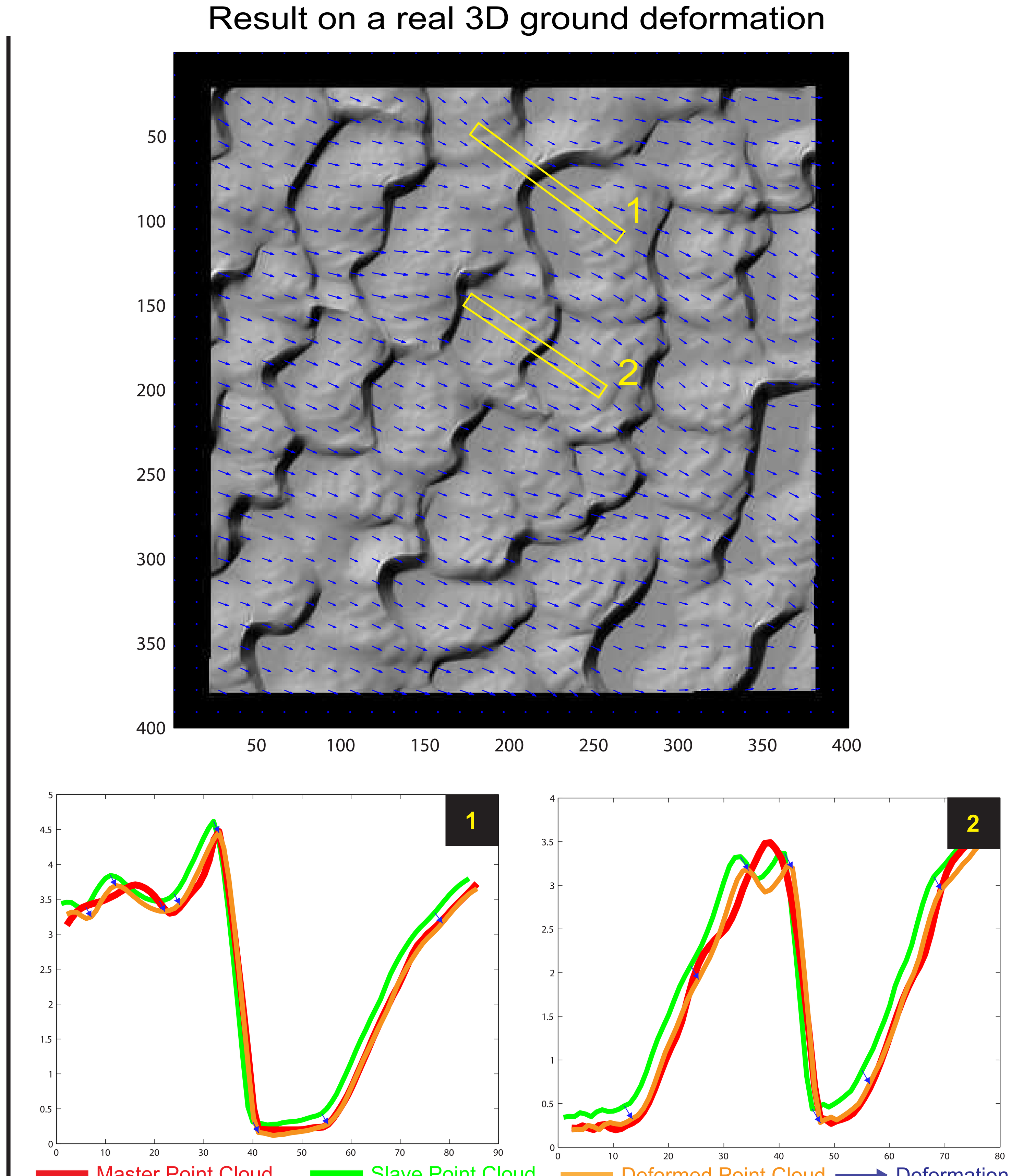
6> Results

Result on a simulated 3D ground deformation



We simulate a smooth and complex deformation of an amplitude up to 10 meters on a real LiDAR acquistion (1 meters resolution). Our algorithm produces good results as the mean error of the retrieved deformation is below 60 centimeters.

Result on a real 3D ground deformation



The retrieved deformation is conformed to a visual inspection. Nevertheless, the deformation retrived is a first order approximation as the algortihm does not track the ripples of the dunes.

7> Conclusions and Future Work

We have demonstatred the good behavior of our method on both synthetic and real smooth 3D ground deformations.

To achieve a better precision, we will have to modify the cost function. For instance, the conservation of local curvature or physical porperties such as volume conservation could be enforced.

The approach has to be enhanced to be able to track objects moving on top of each others such as ripples on dunes.

Finally, an efficient implementation in Python or C/C++ has to be done to process large datasets.