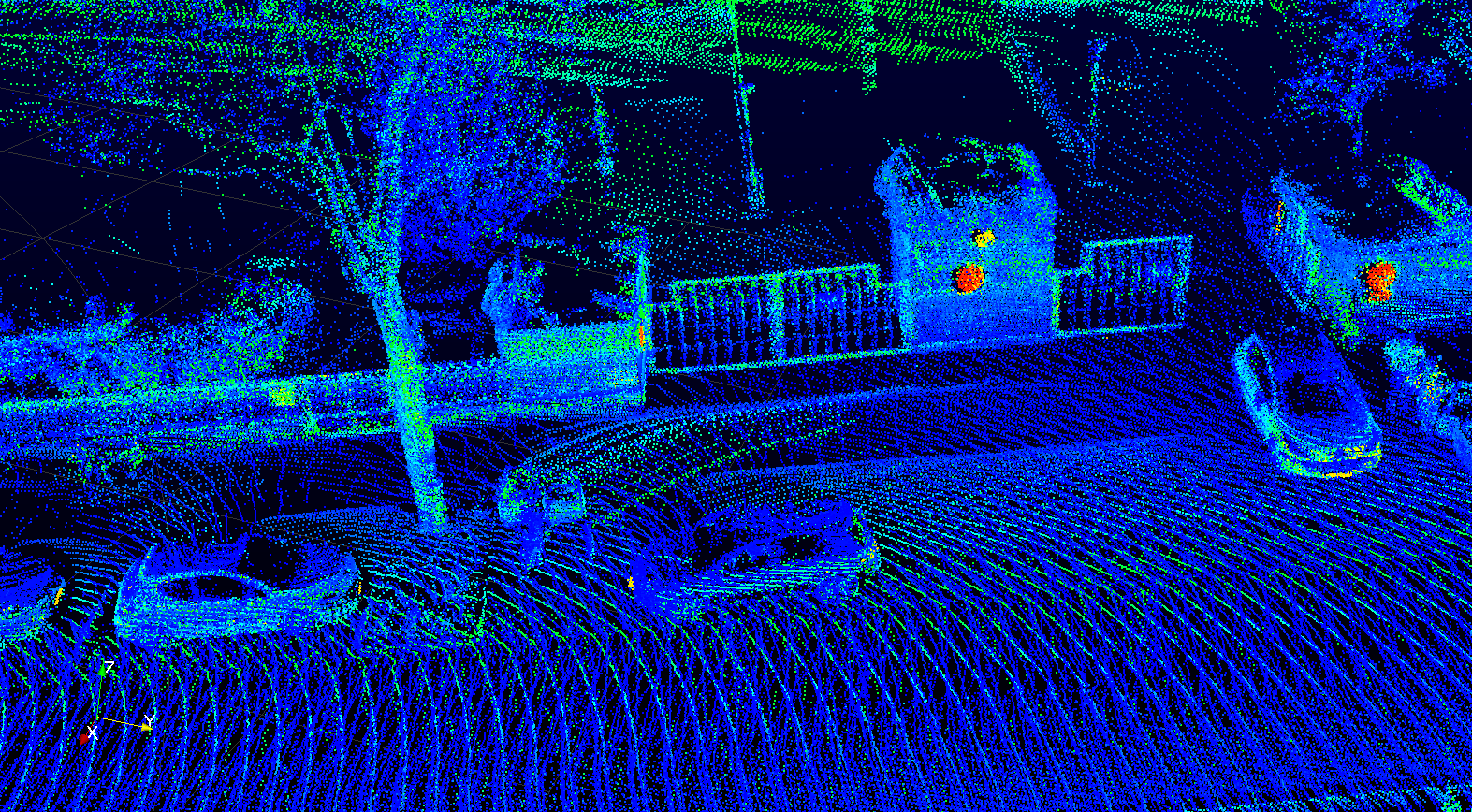


LiDAR based SLAM algorithm



Kitware French front office

# 0-Introduction

In this document, we will present you the Kitware solution for LiDAR based slam algorithm; working in real time and performing state of the art accuracy. In a first part, we will show you through some examples the interest and usage of our SLAM algorithm. Then, in a second part we will detail the mathematical problem and developed solution.

# II- Presentation and modelization of the problem

## II-0- Notations

Affine space:  
Let be a vector space associated with the field . An affine space of direction is a none empty set provided with an application : which associates each couple an element of (a vector) noted that satisfies:

Note:  
Let be an affine space. If we fixe an origin point O of , then it exists (by definition) from to which associates to a point a vector . This application is bijective (using (i) and (ii)). Hence, using an origin O of the affine space we can provide to the space a vector space structure isomorphic to .

Conversely, every vector space is canonically provided with an affine space structure using:

That is why it is not necessary to distinguish between and since they are isomorphic. We will sometimes make the distinction to be clearer.

Extrinsic / Intrinsic representation of a point:   
Let a point of a finite dimension affine space. We denote by the intrinsic representation of the point since it does not depend on any reference frame. Given an origin O of the affine space and a base of the direction (associated vector space) we can provide an extrinsic representation of X:

In other words, we represent using the coordinates of the vector expressed in the base .

Reference frame changes:  
Let a point of a finite dimension affine space and two references frame of . We denote by the change of basis matrix from to . We have two extrinsic representation of using the two references frames linked by the expression:

Euclidean affine space:  
 is called an Euclidean affine space if its direction vector space is a Euclidean vector space:  
i.e, is a -vector space of finite dimension provided with a scalar product:

With being bilinear, symmetric, definite and positive application from to .

With the introduction of a scalar product come notion as norm, distance, angles, orthogonality, …

Direct orthonormal references frames:  
Let be an Euclidean affine space with its scalar product . A reference frame of is said to be orthonormal if .  
An endomorphism of is said to be an orthogonal automorphism if:

If is an orthonormal basis for , and then:

and

This being true for all we have:

is said to be an orthonormal matrix; it represents an orthonormal automorphism in an orthonormal basis.  
In , the set of orthonormal matrix is a non-connected manifold consitute of :  
- Rotation matrix with det(.) = 1  
- Symmetry matrix with det(.) = -1

When a reference frame change is proceed from a orthonormal basis to another orthonormal basis then the change basis matrix is an orthonormal matrix.

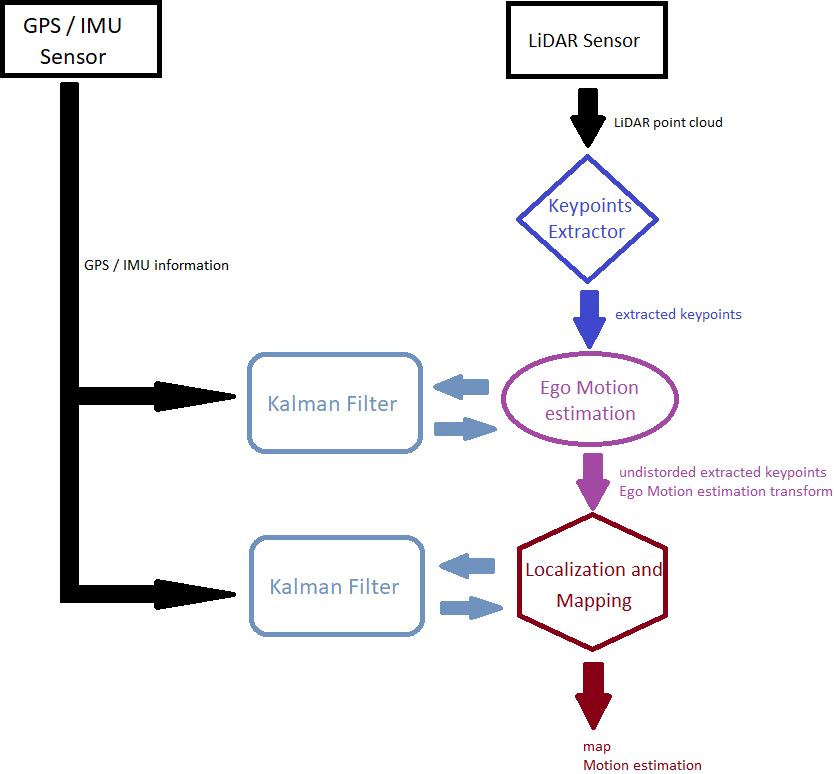
## II-1- Algorithm architecture presentation

The lidar based slam algorithm that we developed is composed of three parts (see Fig.1 Slam architecture)

The first part is the keypoints extraction one. The aim here is to extract from the full point cloud some interesting points that have good geometric properties (planes, edges, …).

The second part consists on estimating the ego motion of the sensor between two sweep / frames of the LiDAR. To do so we will use the two sets of keypoints extracted from the current and previous frame and estimate the motion of the sensor.

Finally, the third part will use the computed internal map of the environment and the ego motion output to estimate the localization of the sensor into the map and then completing the map with the new 3D data point cloud.

  
**Fig1**: LiDAR based SLAM algorithm architecture

## II-2- Keypoints extraction

The keypoint extraction consists of extracting points in the LiDAR output pointcloud that have specific geometric characteristics.

Basically, the LiDAR sensor is sampling the observable surfaces of the environment (see Fig.3). Let’s take a look at the figure 2 (Fig.2).

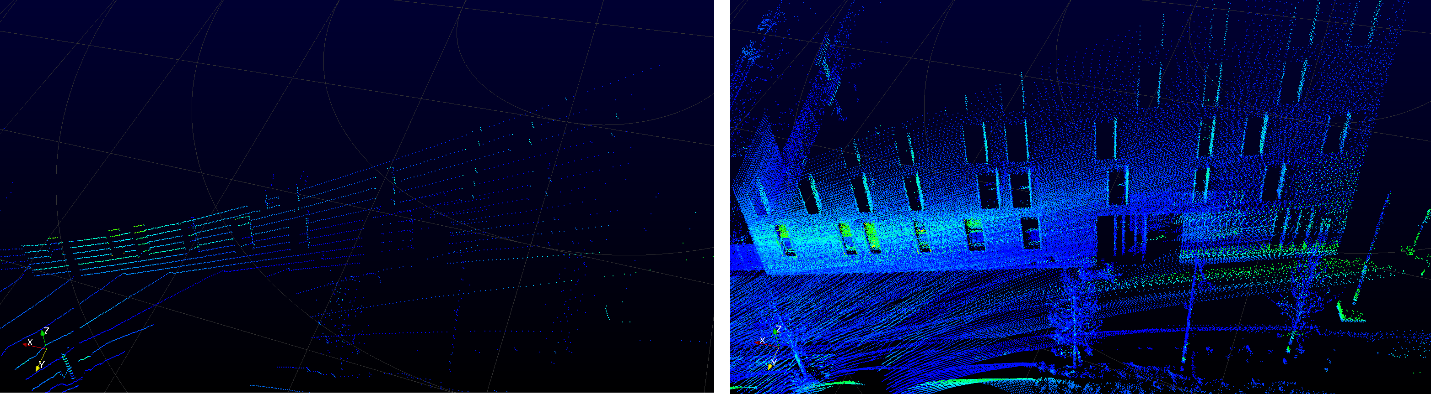
  
**Fig.2** Confluence museum in Lyon, France

What we call the observable surfaces are the surface of the water, the building (museum), the boats, surfaces of small objects, … These surfaces can have:

- discontinuity: borders (example: border of a traffic sign)  
- discontinuity: edges or corner points (example: salient points or edges on the building surface)  
- continuity: the surface can be locally approximated by a plane (example: smooth part of the building)

The keypoints we want to extract are the ones belonging to a or discontinuities that we call edges / corners points and the ones belonging to flat part of the surfaces continuities and curvature low enough that we call planar points. The idea is to extract all lines and planes geometric primitive of the scene.

We want to extract the geometric primitive instead of points due to the LiDAR low resolution sampling effect. Let’s take a look at figure 3 (see Fig.3):

  
**Fig.3:** Left: A velodyne VLP-16 frame of the front of a building. Right: Slam output of the same building

Due to the sparsity of point clouds generated by the manufactured LiDARs it is unlikely that a physical point on a observable surface in the scene will be reached twice by the LiDAR. However, we can observe and estimate the shape and properties of the samples surface. We can approximate some part of the surface by simple geometric primitives such as planes and lines. Even if we won’t observe the exact same points on these primitives we will observe the same primitive during several frames of the lidar. This is why a description of the scene using geometric primitive is better than using outputted points.

## II-3 Keypoints matching

If the geometric primitive is **plane**  whose normalized orthogonal direction is and is a point of the plane then:  
And then,   
With:

If the geometric primitive is a **line**  whose normalized direction is and is a point of the line then:

The orthogonal projection of a point on the line is  
Then,

Since is symmetric matrix:

And since is the matrix of a projection endomorphism:

With:

## II-4 Numerical optimization

In this step we have:

The set of keypoints extracted from the current frame of the lidar sensor:

Provided with its matches geometrics primitives: from either the previous frame or the map (depending if we are processing the Ego-Motion or the Mapping-Refinement step).

The nature of the geometric primitive is determined by the kernel dimension of the matrix . If the geometric primitive is:  
- A line whose normalized direction is then and its kernel dimension is 1  
- A plane whose normalized orthogonal direction is then A = and its kernel dimension is 2  
- An ellipsoid then where is the variance-covariance matrix of the blob

What we want is to find the rotation and translation that minimizes the sum of the distances (point-line, point-plane and point-ellipsoid):

While ones might think that it is a linear least square parameters estimation, it is important to remember that the matrix lie on a manifold embedded in and must satisfy:

We propose to use the Euler-Angle parameterization of the group using the following convention:

Where and:

We introduce:

And we want to minimize the cost function:

This is a non-linear least square parameters estimation.

Non contractual: 1cm of drift every meters of slam (average) => 1.08%