LASER SCANNER BASED SLAM
IN REAL ROAD AND TRAFFIC ENVIRONMENT

Olivier Garcia-Favrot and Michel Parent
INRIA Paris - Rocquencourt
domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex France.

Abstract - In this paper we will present a SLAM algorithm we have recently developed for our needs in autonomous automotive applications. Our approach has the particularity of making use exclusively of laser scanners to achieve our goals without using any other type of sensors or source of information. We concentrated on developing a self-contained system that could be placed on any kind of mobile platform and work in any kind of dynamic environment; this is why too at this point our approach does not make use of any model of the vehicle. Our SLAM system has been tested with success both on a car at full speed on a road and a human evolving indoors. We will present here the challenges we face that pushed us to develop the algorithm, the solutions we are exploring, discuss experimental results and suggest areas of future work.

I - Introduction

When trying to make a vehicle autonomously travel to its destination over many kilometers you soon realize you will have to overcome many challenges [1],[2],[3],[4]. The first need is to be localized globally in order to know what path to take for reaching your destination. The second need is to be localized locally in the surrounding environment to make sure you are keeping the road, the right speed, the right distance with the vehicle in front, that you do not collide with other vehicles and obstacles etc. A third need is to have the ability to detect certain features in the environment in order to know how to properly behave on the road. For example, you need to be able to detect lanes in order to stay centered, you need to detect intersections, crosswalks, continuous or dashed lines etc… Then you have to take cost into consideration if you expect your work to have one day any use in real life. While global localization is a problem that can be considered to be solved thanks to a wide variety of affordable GPS solutions, all the other issues still present a big challenge when it comes to real world applications.

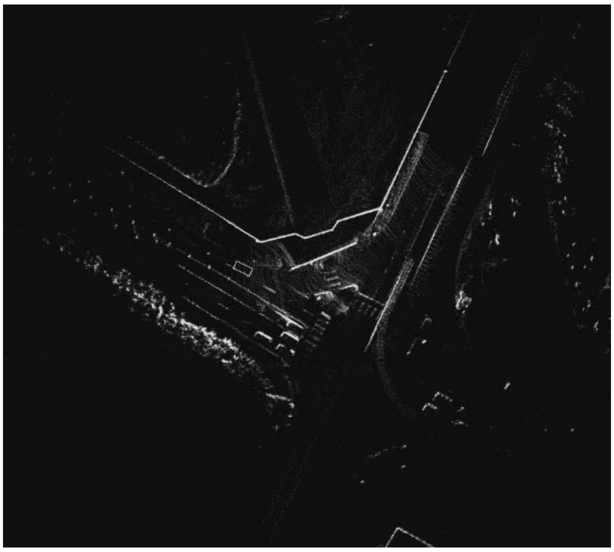

Fig 1. Map result showing road features.

In our struggle for achieving autonomous driving we have been exploring separately the potential of a single sensorial solution, the laser scanner, we believe eventually may give the necessary information to tackle all the issues mentioned above. We plan to use other types of sensors too but at this point the goal of our research is to make the most out of this type of sensor taken separately, before combining it with other sensorial input [5],[6].

Using a laser scanner only, we are able to keep track of our trajectory and our speed, in other words to localize, this, combined with the range data, allows us to create a model of our environment in the form of a map. This duality between creating consistent maps and localizing has been extensively studied as the Simultaneous Localization And Mapping (SLAM) problem [7],[8]. As we see in the figures presented (for example Fig 1), taken from real driving situations, the maps obtained contain all kind of useful information we plan to use for path planning and navigation, such as lane markings, intersections, crosswalks, empty parking space etc. But before entering into more details of our experimental results, we will first have an overview of the algorithm.

II - The algorithm overview

Our SLAM algorithm consists of two steps. A first step in which the relative movement of the vehicle is estimated and a second step in which this first estimation is refined by relocating the pose respect to the map we are progressively building. Figure 2 shows the different steps of the algorithm.

```
Pseudo code

void main ()
{
   //Init the pose and the robot's motion
   Pose = init_pose () ;
   Relative_motion = init_relative_motion () ;

   //Get the first scan and plot it on the map
   Scan = get_scan () ;
   Spikes = get_high_derivative_points (Scan) ;
   Previous_segments = calculate_segments (Spikes) ;
   Map = update_map (Pose,Scan, Map) ;

  //Localization loop
   while(true)
  {

    Scan = get_scan () ;
    Spikes = get_high_derivative_points (Scan) ;
    Current_segments  = calculate_segments (Spikes) ;
    Previous_motion = Relative_motion ;
    Relative_motion = calculate_motion (Previous_segments, Current_segments);
    if (Relative_motion = = NULL) Relative_motion = Previous_motion ;

    Pose_estimation = calculate_pose_estimation (Pose, Relative_motion) ;
    Previous_pose = Pose ;
    Pose = refine_pose_estimation (Pose_estimation, Scan, Map) ;
    Map = update_map (Pose,Scan, Map) ;
    Relative_motion  = recalculate_motion (Pose, Previous_pose) ;
    Previous_segments = Current_segments ;

   }//end of while
}//end of main
```

Fig 2. Steps of the algorithm.

II.1- Motion estimation: The first step works by tracking the motion between the current scan and the previous one. This is done by tracking points of high derivative, in other words by tracking the "spikes" that are apparent on the scan. This makes the algorithm absolutely universal as it will work in any structured or non structured environment. To find points of high derivative within the scan is easy but we need to identify them over the current scan and the previous one in order to discover the motion. What we do to perform this identification is to trace segments between the spikes and make use of invariants throughout the movement such as for example the length of those segments. For example, if we manage to find a segment of length "L" connecting two spikes in the current scan and we find also a segment of the same length in the previous scan, then we have pretty good chances that those two spikes at the ends of the segment are the same points of the environment. Now, by looking to our relative position to the segments in both the current and previous scans we are able to infer the relative translation and rotation we have done between the two scans. The problem is that many of the spikes are noise and are not consistent with the vehicle's movement. The challenge here was to develop the proper filter capable of sorting out only the spikes that are common to both the scans.

The typical error of the motion calculated this way is of ±20 cm in both x and y and ±2° in the orientation. Thankfully this error can be greatly reduced by refining this first estimation through a second step.

II.2- Relocation: The second step makes use of our near past experience by relocating on a map created by the previous scans. In our case a map is nothing else but a model, a discretization of the environment in the form of a bitmap containing a height field (Fig.3). The height field is formed as laser measurements are accumulated over time on certain pixels making the features of the environment become apparent. The amount of the laser data and its accuracy determine the level of discretization of the environment that can be achieved, that is, the map resolution (see section **II.3**). We believe that by studying the morphology of this height field we will be able to extract from it all the information necessary for navigation but this is a totally separate matter from localization.

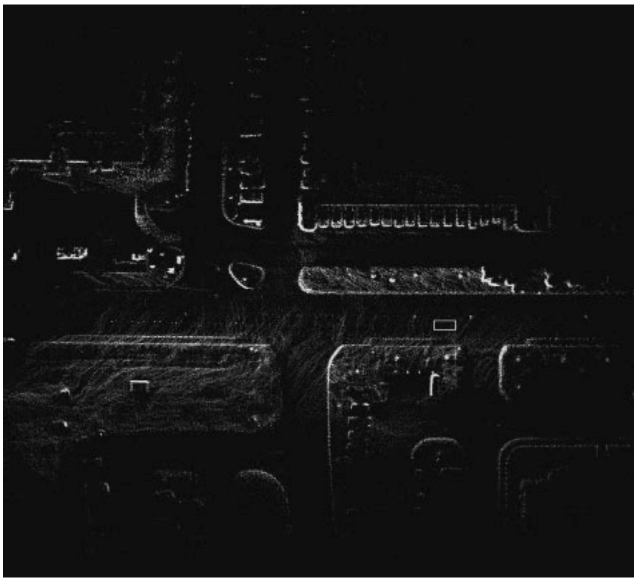

Fig 3. Bitmap containing the height field.

For relocation within the map, we use the fact that we know the actual pose of the vehicle is within the rather small error bounds of the pose estimation obtained from the first step. Since for relocation we are going to try to match the current scan with the map, the number of possible poses within the error bounds is limited by the resolution of the map. For example, if we have a 10cm per pixel map, the search area will be a square of 5x5 pixels and the discretization for the heading can be let say 0.2°. This gives us 25x21 = 525 candidate poses we then check exhaustively for the best match. We consider as the best match the candidate pose that maximizes the sum of the range value of the current scan points that hit non-empty pixels in the map. Of course, if our actual position is outside the typical bounds of error, then we are lost. For the rare cases when this happens we plan in the future to use more sophisticated techniques to find our position within the map but in practice this rarely happens except in the case the environment is very poor in information such as very large open spaces.

About this second step it is worth noting that the more information the map has about the zone of the current scan, the better it works. Although this seems evidence, it brings to an important conclusion which is that relocation works best when having a sensor looking in the opposite direction to the movement. This is so because unless we have done a loop, in general we are exploring what lies in front while we have already explored what lies behind us and therefore we have much more information. This is especially true at high speeds, looking at Figure 4 we can see the height field is much clearer in the opposite direction of the movement as we have gathered much more information in the zone of the map we have left behind than the amount we have managed to gather of what is coming in front of us. This means that simply having a sensor in the front of the vehicle is conceptually a bad policy if we plan to do localization, especially at high speeds. Also, in an automotive application it is of utmost importance to include the rear of the vehicle in the field of view as otherwise it would be like driving without any mirrors.

Finally, once we have refined our location within the map, we update the height field with the current scan data and we recalculate at posteriori the relative movement we have done.

It may happen to the first part of the algorithm not to give any output, in such cases we simply fill the gap taking as our current relative movement the last one we were able to calculate.

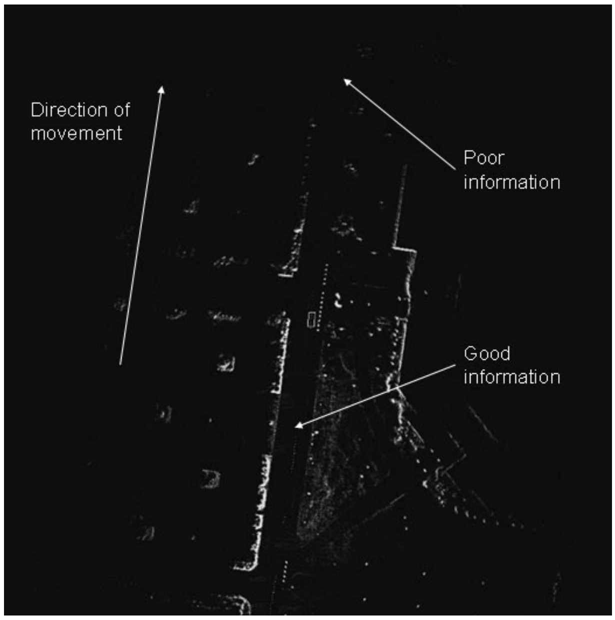

Fig 4. Best info is found at the back of the vehicle.

II.3- Algorithm's behavior: Although there is certainly a cumulative error out of this two step algorithm, it is so small that for practical purposes there is none. As we explained in the introduction it is not our goal to make use of this localization over hundreds of kilometers, this task is left to the GPS. Our use of the algorithm is to be localized locally, that is, to be able to generate an accurate map of the surrounding over 100 m of our vehicle in order to extract from it information about lanes, obstacles and other vehicles and take it as a base for path planning and trajectory control.

Even though, we have to say that our experimental results over few kilometers show very little accumulated error (see section III for more details). Errors mainly seem to occur as a result of singularities, such as having a surrounding environment very poor in information, that is, when we happen to go through large open spaces. This is to be expected as the accuracy and quality of the output of the algorithm varies depending on the quality and amount of the information you feed in it.

A study on the subject has not been realized yet at this point but globally we have the following:

The map resolution that can be achieved in general is very dependent on the angular resolution and the scan frequency. Without surprise the scan frequency is particularly important at high speeds. Other parameters like the laser range and field of view have no remarkable effect on the achievable map resolution. Actual tested implementations with different settings gave a 10 cm per pixel map resolution at 10 Hz and 0.5° angular resolution and a 5 cm per pixel map resolution at 20 Hz and 0.25° angular resolution.

When it comes to loss of localization, as we have explained, open spaces are our major enemy. It is therefore no surprise that the robustness of the algorithm is directly proportional to the range of the laser and how wide is our field of view, as this helps to reduce greatly the cases in which the algorithm has nothing to “hook” on.

Another potential source of error is moving objects. The algorithm normally filters out the segments including moving points because they do not maintain the invariants found in segments from static points. This works only if the scans contain a sufficient number of static points. The field of view is therefore quiet important for dealing with this. A 270° field of view is generally enough for completely removing this issue as a source of error even in very heavy traffic conditions.

III - Real road testing

In order to test our algorithm in a real driving situation we went to see our partners at the Southwest Research Institute in San Antonio, Texas. We have to thank the whole SSTI team for their help and for making this possible.

The SSTI vehicle (Fig 5) is a fully automated Ford Explorer equipped between other things with two Alasca Ibeo laser scanners and a differential omnistar GPS we will only use for reference. Each of the Alasca scanners have a range of up to 200 m, have 4 layers and a 270° field of view. They are located at the left and right front corners of the vehicle, so together they cover almost 360° except for the very rear part of the vehicle.

Fig 5. The SSTI vehicle at the SwRI in San Antonio.

Alasca laser scanners have both positive and negative specifications when it comes to our algorithm. On one side they have a long range, a 270° field of view and 4 simultaneous scanning planes at different angles (Fig 6). In the end we make use of only one of the planes in the localization process but the information of the four combined makes possible to obtain very rich and detailed maps. On the other side, each separate scanning plane has a very poor angular resolution which only gets worse when increasing the scanning frequency.

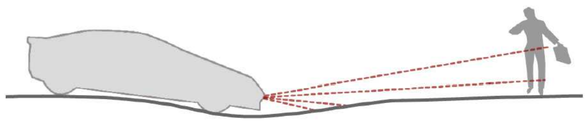

Fig 6. The laser scanners have 4 simultaneous layers.

Although the different scanning planes are interlaced and we have tried to combine them, this does not work because of the fact the points of high derivative we use keep information about the movement only when you derivate a continuous signal coming from the same scan plane. To make things worse, by default the angular resolution is not constant but higher in the front and lower on the sides and rear, which as we explained earlier is the best zone for relocation. As a result of this, the implementation on the SSTI vehicle was limited to achieve a 20 cm per pixel map resolution but turned out to be very robust thanks to the almost 360° field of view and possibly to its low map resolution. The scanning frequency was set to 12.5 Hz which may not be sufficient when driving at high speed but to increase the frequency would have lowered too much the angular resolution.

On the whole, results were very satisfactory as we were able to localize over kilometer drives at high speeds (60km/h to 70km/h) on a real road and heavy traffic environment.

Because the four scanning planes were plotted, rich and detailed maps were obtained with road features becoming apparent such as continuous and dashed lane markings (Fig 7). We believe this happens because as the lower scans sweep the road at a certain distance and angle, only reflective surfaces send back an echo. At this point we don't take advantage of this for localization because the lower scans are not used in the process.

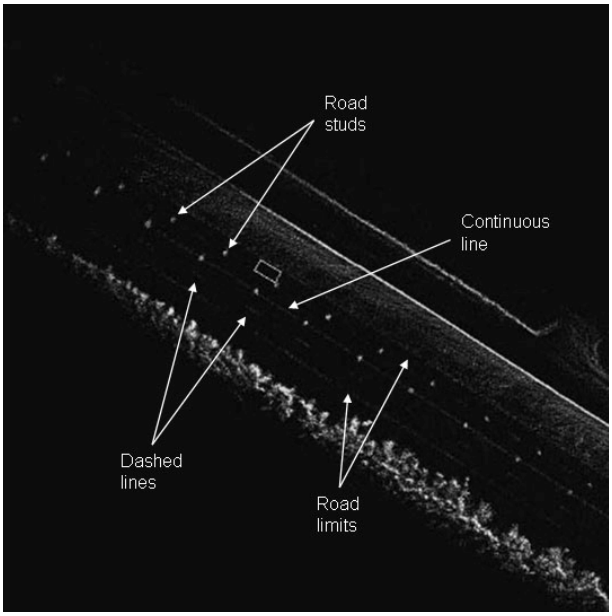

Fig 7. Road features become apparent.

We have to point out that when driving over large distances the use of a static map is not practical. This is why for the real road tests we have used a dynamic map centered on the vehicle and which we build from a circular buffer containing the N previous scans, that is, when new data comes in, the oldest data in the buffer disappears. The result covers a square area of 200m x 200m around the vehicle (1000 pixels x 1000 pixels) and we say the map is dynamic because it is being completely redrawn at each new scan. A dynamic map of the sort has been preferred during the tests to a static one primarily because the large distances involved in the testing would have required a huge bitmap. Then this is ok because in this application the map information is normally used for local navigation, which generally only involves a few meters around the vehicle. Other benefits of a dynamic map is its ability to regenerate rapidly in the case of a loss of localization and if a small buffer is used, the capacity of reflecting the changes in the environment. For example an opening gate that was initially closed will disappear from the map as it opens. Another example would be to update empty spaces in a parking lot while searching a place to park.

In the case we are dealing with an application in which we want to have a map over the complete trajectory of the vehicle, we need as we go to keep record of all measurements along with the calculated associated position. For drawing a portion of a map we then need to specify a location and we search in the recorded data all the positions found to be within a certain radius of the selected location. Now, we obtain a map by simply applying the appropriate coordinate change to the associated scans of those found positions. Because the map is kept not as a bitmap but reconstructed from the raw scan data every time, it is possible to easily zoom in or zoom out or rotate the map at will. This can be useful for example if we want the orientation of the map to follow in real time the current orientation of the vehicle.

Even though it is not the purpose of our algorithm, we are going to show here an example of its accuracy over a long distance drive (2.18Km). During the test, the differential GPS position was recorded to be used as a reference.

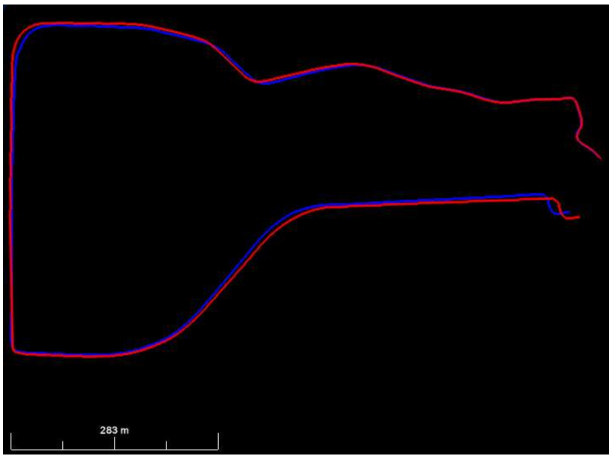

Fig 8. Laser(blue) and GPS(red) paths (2.18Km)

On this ride (Fig 8.), the differential GPS and the laser trajectories seem to match relatively well the first 400m but after an offset appears that remains for the rest of the way. Even though, comparing the relative shape of the trajectories and the distance involved, we can see the algorithm in general performs quite well even if it is almost inevitable to have at some moment some surrounding environment that the algorithm won't be capable of dealing with. We want to point that what we see here is the result of the raw output of the localization process, it could be certainly improved for example adding a Kalman filter which points towards interesting future developments.

IV - Suggested areas of future work

Immediate planned future work includes testing the algorithm with other laser configurations to see if we can improve its performance both in map definition and robustness. After that we plan to do:

- Tracking of moving objects, this should improve the localization process as we can eliminate the moving points from the next iteration of step one as well as eliminate those from the map, improving in this way the relocation in step two.

- Elaborate a good control law working with the map localization.

- Create classifiers able to extract from the map features such as lanes, intersections and so on.

- Combine the algorithm output with a low cost 1Hz GPS in order to transform the map coordinates into global latitude and longitude coordinates for global navigation.

REFERENCES

[1] S.Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aaron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "*Stanley: The robot that won the darpa grand challenge*," Journal of Field Robotics, vol. 23, no. 9, pp. 661-692, September 2006.

[2] C. Laugier, S. Petti, D. A. Vasquez Govea, M. Yguel, T. Fraichard, and O. Aycard, "*Steps towards safe navigation in open and dynamic environments*," in Autonomous Navigation in Dynamic Environments: Models and Algorithms, ser. Springer Tracts in Advanced Robotics Series (STAR), C. Laugier and R. Chatila, Eds. Springer, 2006.

[3] R. Benenson, S. Petti, T. Fraichard, and M. Parent, "*Toward urban driverless vehicles*", International Journal of Vehicle Autonomous Systems, Special Issue on Advances in Autonomous Vehicle Technologies for Urban Environment, vol. 1, no. 6, pp. 4 – 23, 2008.

[4] S. Kolski, K. Macek, L. Spinello,and R. Siegwart, "*Secure autonomous driving in dynamic environments: From object detection to safe driving.*" in Proceedings of the Workshop on Safe Navigation in Open and Dynamic Environments: Applications to Autonomous Vehicles at the IEEE International Conference on Robotics and Systems, San Diego, USA, 2007.

[5] Lars B. Cremean and Richard M.Murray, "*Uncertainty-based Sensor Fusion of Range Data for Real-time Digital Elevation Mapping (RTDEM).*" submitted, 2005 IEEE International Conference on Robotics and Automation.

[6] L. Conde Bento, Urbano Nunes, Fernando Moita and Antonio Surrecio, "*Sensor Fusion for Precise Autonomous Vehicle Navigation in Outdoor Semi-structured Environments.*" Proceedings of the 8^{th} International IEEE Conference on Intelligent Transportation Systems Vienna, Austria, September 13-16, 2005.

[7] S.Thrun. "*Robotic mapping: A survey.*" in G. Lakemeyer and B. Nebel, editors, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.

[8] Bailey, I. *Mobile robot localisation and maping in extensive outdoor environments*. PhD thesis, ACFR, University of Sydney, Australia, Aug 2002.