3D laser range finder simulation based on rotated LMS SICK 200.

Janusz Będkowski, Maciej Kretkiewicz, Andrzej Masłowski
Research Institute of Automation and Control
Intelligent Mobile Systems Division
Aleje Jerozolimskie 202
Warsaw, POLAND
januszbedkowski@gmail.com

Abstract

This paper describes the approach of the simulation of the robust 3D laser measurement system. The 3D range finder is constructed using LMS SICK 200 mounted on the rotational platform, therefore the robot can gather three-dimensional data in order to accomplish a 3D localization in soft real time during it's journey. The simulation uses multi core processing power to generate 3D data. The architecture of the simulator is based on the CORBA (Common Object Request Broker Architecture).

1. Introduction

The simulation of the robot perception based on LMS SICK 200 provides a lot of advantages. First of all it is cheaper than experimenting real device. The simulator gives opportunity to concentrate more on the artificial intelligence algorithms such as feature extraction, classification, approximation. Developing the simulator is easier than building new robot or perception system[5], due to the testability of many configurations. The newest technology offers multi core processors, which providing a base for real time simulator building of the 3D laser range finder. Therefore, the robust procedures of computing ray intersection is proposed as well as an inherent basic procedure to simulating the laser finder measurement. Presented work is the part of the simulation framework used in developing sophisticated algorithms such as classification, obstacle detection, map reconstruction, etc.

2. Simulator overview

The following scheme (figure 1) shows the concept of the components of the simulator. Simulator uses VRML format to describe the scene of the robot. The "init" block initializes all thread. Thread1 and thread2 (multi threading) compute the laser scans parallel in infinite loop. Parallel processing increase the performance of the algorithm, due to the computation can be obtain in real time mode. The problem of time consumption of the laser measurement simulation is solved by using multi threading. Therefore thread1 and thread2 compute an appropriate amount of laser beams, by dividing its' sets into two subsets from intervals: i = -90..0 for thread1 and i = 1.. 90 for thread2. ThreadOpenGL is responsible for data flow synchronization and renders the 3D scene.

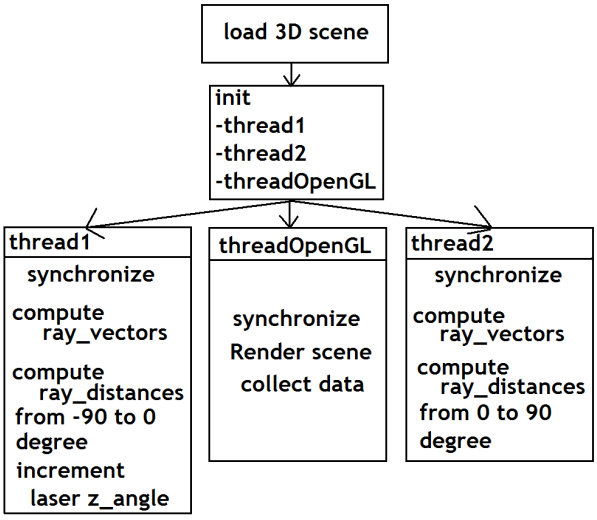

Figure 1: The components of the simulator.

2.1 The concept of the 3D range finder simulation

The idea of the 3D range finder is based on mounting LMS SICK 200 onto the rotating support. Therefore the scan measurement explore whole 3D scene round the robot chassis.

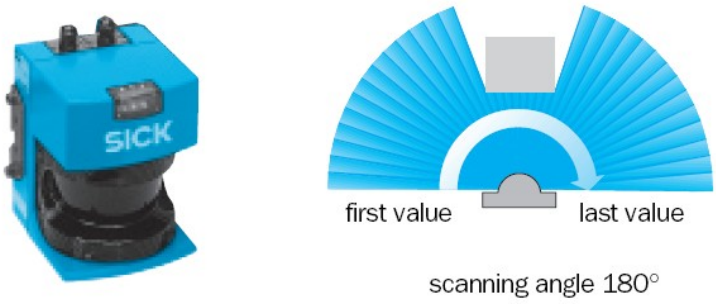

Figure 2. LSM SICK 200, and its scanning angle.

The LMS SICK 200 mounted on the rotating support is shown on figure 2. Laser reads data every 0.5^0 (-90^0; 90^0), the rotating support rotates 1^0 from 0^0 to 360^0. It gives 361 x 360 ~= 13 * 10^4 points of simulated measurement.

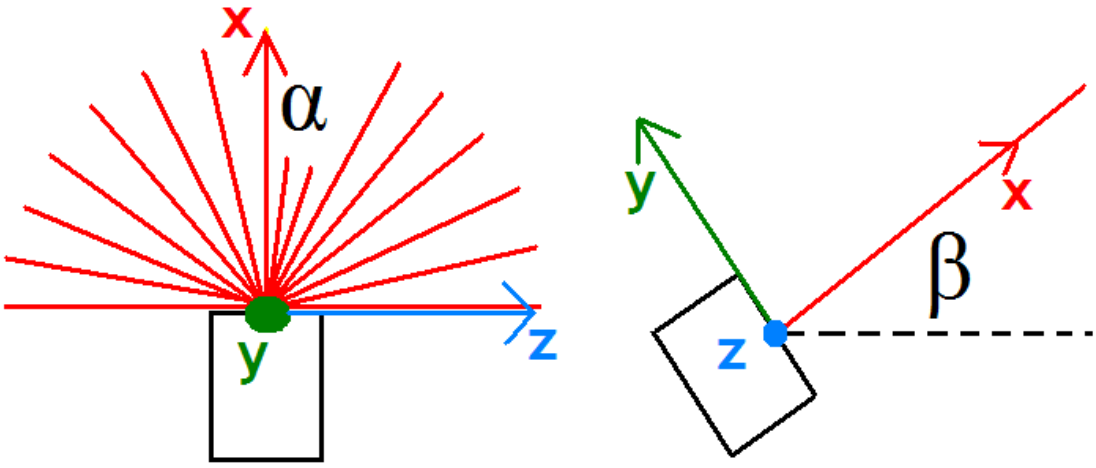

Figure 3 : The scheme of the LMS SICK 200 simulated rotation.
Simulated data is obtained in the polar coordinate system (r, α, β) where:

r – distance to an obstacle
α – horizontal angle of the laser beam
β – vertical angle of the laser beam
Data for point(r, α, β) from the simulated laser is obtained by transforming the point p [r, 0, 0, 1] by matrix $M = M_Z \cdot M_Y$, where:

$$M_z = \begin{matrix} \cos\beta & \sin\beta & 0 & 0 \\ -\sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad M_y = \begin{matrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

For each value of horizontal angle α and vertical angle β the data of laser beam is collected. The set of measured points is visualized as the set of triangles. This process is possible due to the reconstruction of the local map. The local map can be used in teleoperation with 3D visual feedback.

2.2 VRML Virtual Reality Modeling Language

The presented simulator uses VRML file format to define the environment which observed by the robot measurement system. VRML stands for ``Virtual Reality Modeling Language''. It allows to specify dynamic 3D scenes[4], through which users can navigate with the help of a VRML browser. VRML scenes can be distributed over the World-Wide Web and browsed with special VRML browsers, which most of them are plug-ins for Netscape or Internet Explorer. VRML is well integrated with the WWW, e.g. VRML scenes can be connected with other VRML scenes (as well as other WWW formats) via URLS. The VRML standard was chosen to define 3D scenes into the simulator. The idea is to create 3D scene usage, for example Wings3D program and then load into the simulator scene. Simulator allows the changing of the 3D laser position, due to it is possible build local map from any location. Therefore the sophisticated teleoperation of the mobile 3D sensor is available.

2.3 CORBA interface

CORBA (Common Object Request Broker Architecture) was chosen to develop interface to simulator. CORBA is supported on almost every combination of hardware and operating system. It supports a large number of programming object oriented languages (java, C++)[3]. The client server architecture is designed for the simulator. To invoke operations on a distributed object, a client must know the interface offered by the object. Required knowledge of the purpose and semantics of the operations are defined in IDL (Interface Definition Language) as follows:

```
module LMS3D{
     typedef sequence<float> Data;
     interface DataExchange{
          Data fetch();
     };
}
```

2.4 Robust algorithm of the ray intersection.

The equation for a plane is:

$$Ax + By + Cz - D = 0;$$

The triplet <A, B, C> represents what is called the normal of the plane - N. The D component in the equation represents the distance from the plane to the origin. The triplet <x, y, z> is any point that satisfies the equation. The set of all points <x, y, z> that solve the equation is exactly all the points that lie in the plane. The procedure that computes distance to plane is given:

```
procedure DistanceToPlane(point)
{
return    dotProduct(N, point) + D
}
```

The procedure of ray intersection is given:

```
procedure RayIntersection(ray_position, ray_direction)
{
     A = dotProduct(N, ray_direction);
     If (A == 0)
          return rey_position;

     return ray_position – ray_direction *
(DistanceToPlane(ray_position) / A)
}
```

Boosted method of the defining the location of a point with respect to a triangle polygon is based on the following idea. Figure 4 shows the rectangle polygon with a point defining ray intersection.

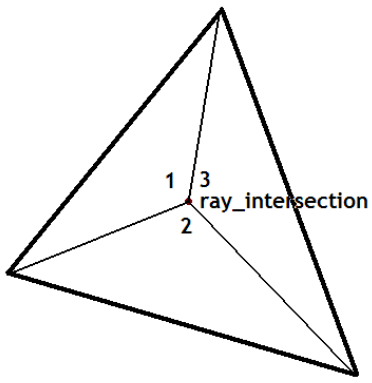

Figure 4: Triangle with ray intersection.

The is idea is based of the summation of the angles 1, 2, 3. If point – ray_intersection belongs to the polygon the sum of the points should be equal to 2Π.

3. Experiments

The scene was build in VRML format to obtain the laser 3D range finder simulation. Following figure shows the artificial environment of the robot (blue scene on the left), and local map build from computed data (green scene on the right). The blue scene is composed by 804 triangles, but local map from 361 * 360 ~= 13 * 10^4 triangles. This experiment shows the possibility of using 3D range finder into the teleoperation with 3D visualization feedback. Presented simulator is robust tool to test and validate 3D feature extraction algorithms.

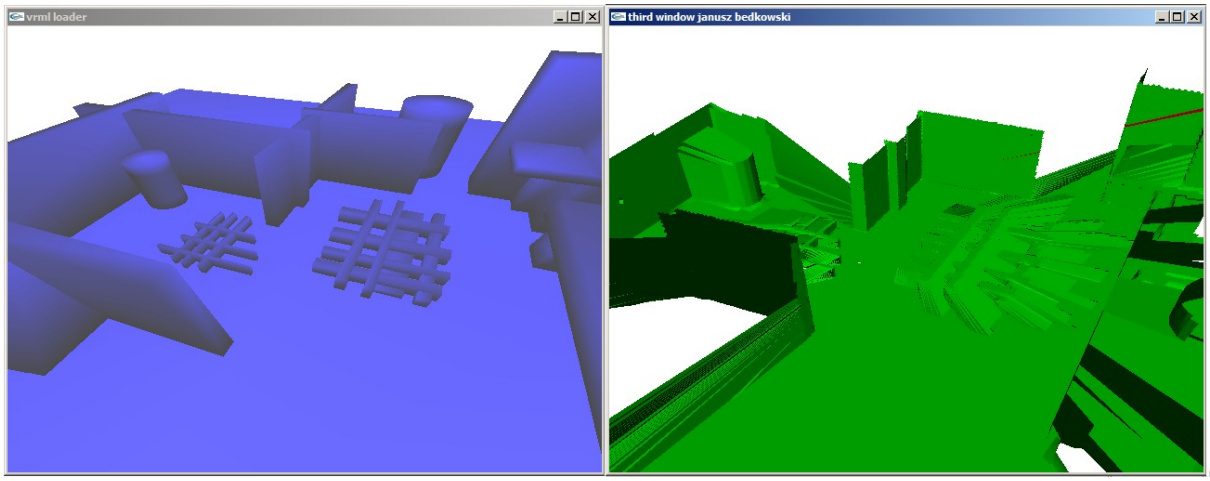

Figure 5. Artificial environment (blue on the left) and its local map (green on the right)
Second experiment shows the reconstruction of the teapot (well known object in 3D modeling).

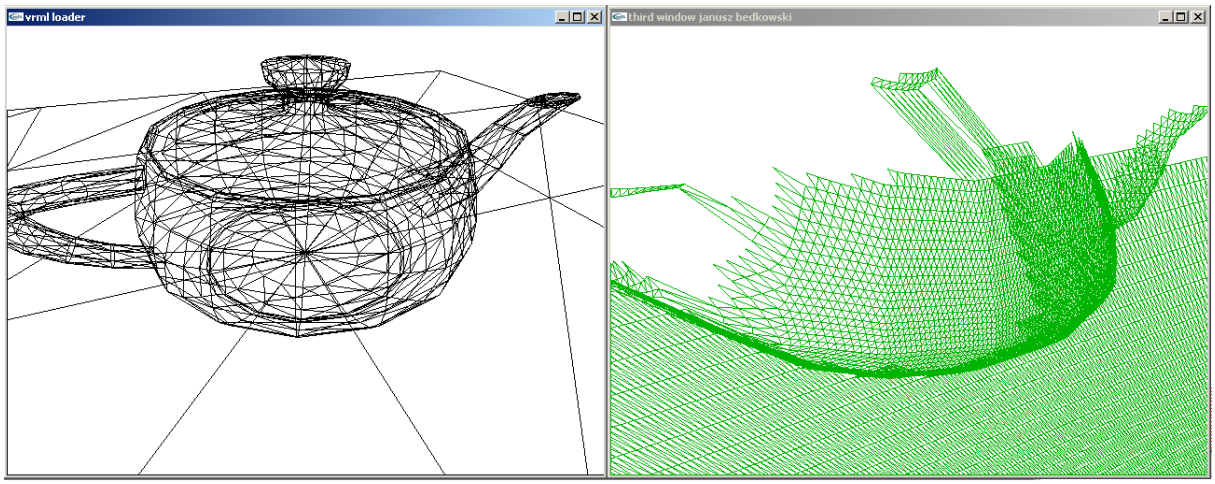

Figure 6: Teapot and its 3D laser reconstruction.

The third experiment shows the performance of the simulation. The computing ray intersections for the simulated LMS SICK laser range finder is a large time consuming task. This fact is determined by the need of computing large amount of rays intersection with respect to a triangle polygons. This problem can be solved by using multi processor parallel computing. It is possible to divide set of simulated beams into subsets. The number of subsets is limited by the number of available PCU. The following experiment was implemented to show the increased performance by usage of dual core processor.

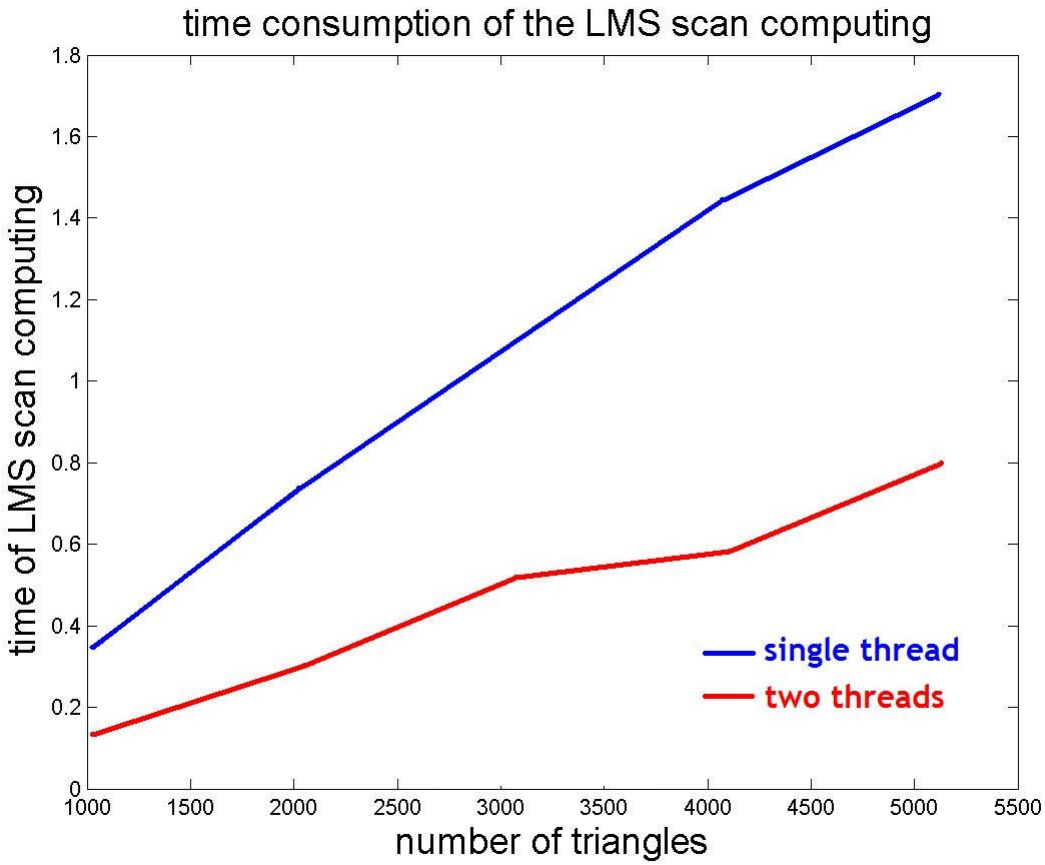

Figure 7: Time consumption of the LMS scan computing.

4. Conclusion

Described work shows the robust methods of computing ray intersection, therefore the concept of the new laser 3D range finder is proposed. It is proven that building on line simulator requires the usage of multi core processor, due to the application can be tested on the new Intel platforms. The approach shows advantages of building local map of the scene, due to the teleoperation with 3D visualization feedback can be obtained. Furthermore the 3D visualization is extremely a robust tool for operator while unknown terrain exploration is executed. Additional 3D image gives a lot of information that can be used for environment feature extraction. Presented simulation is a robust component of the simulation framework, therefore it gives opportunity to examine sophisticated performance algorithms such as classification methods or approximation. The presented simulator decreases the cost of testing mobile platform perception. Therefore, building the artificial environment is cheaper, and more effective than mounting real obstacles. Further more we can obtain the simulation of the multi robots control, to suppress the highly cost of the real robot testing in real environment.

References

[1] Filipe Ferreira, Luis Davim, Rui Rocha, Jorge Dias, Vitor Santos : *Presenting a technique for registering images and range data using a topological representation of a path within an environment,* Journal of Automation, Mobile Robotics & Intelligent Systems VOLUME 1, No 3 September 2007

[2] Barbara Siemiątkowska, Michał Gnatowski, Arkadiusz Zychewicz: *Fast method of 3D map building based on laser range data,* Journal of Automation, Mobile Robotics & Intelligent Systems VOLUME 1, No 2 June 2007

[3] Michi Henning, Steve Vinoski: *Advanced CORBA Programming with C++,* 1999

[4] Tangyun Dai, Zemin Wang, Shouheng Xu, *Research of Creating and Fetching 3D Models of Virtual Reality Based on OpenGL,* Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation June 25 – 26, 2006, Luoyang, China

[5] Eric Colon, Hichem Sahli, Yvan Baudoin, *MoRoS3D, a multi mobile robot 3D simulator,* Proceedings of the 9th International Conference on Climbing and Walking Robots Brussels, Belgium – September 2006

[6] ZHAO Gui-fen, DENG Jia-hao, SUN Wei, WANG Wei, *Study on Simulation Technique of Laser Detector,* 3rd International Conference on Computational Electromagnetics and Its Applications Proceedings 2004