Modeling and Simulation of Laser Rangefinder Architecture

Fábio B. Jesus
Embedded Systems Laboratory (LSE/UEFS)
Department of Technology
State University of Feira de Santana, BA, Brazil
fabiobispo.fsa@gmail.com

Joel P. C. Filho
Embedded Systems Laboratory (LSE/UEFS)
Department of Technology
State University of Feira de Santana, BA, Brazil
joelpcfilho@gmail.com

João C. N. Bittencourt
Embedded Systems Laboratory (LSE/UEFS)
Department of Technology
State University of Feira de Santana, BA, Brazil
joaocarlos@ecomp.uefs.br

Thiago C. Jesus
Embedded Systems Laboratory (LSE/UEFS)
Department of Technology
State University of Feira de Santana, BA, Brazil
jesus@ecomp.uefs.br

ABSTRACT

On robotics, autonomous navigation vehicles and intelligent transportation systems, the identification and calculation of distance to obstacles is critical. This information needs to be available as soon as possible in order to avoid accidents. A laser rangefinder can be used to solve this problem. Thus, this paper proposes modeling a line laser rangefinder based sensor. This model is described and validated in MATLAB software. Furthermore, the hardware architecture of the sensor is presented.

Categories and Subject Descriptors

I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis; C. [**Computer Systems Organization**]: [Embedded hardware, Sensors and actuators]

General Terms

Model Validation

Keywords

Laser Rangefinder, Sensor, Embedded System, Hardware Simulation and Emulation

1. INTRODUCTION

On robotics, autonomous navigation vehicles and intelligent transportation systems, the identification and calculation of distance to obstacles is critical. These informations need to be available as soon as possible in order to avoid accidents. Thus, a lot of works use data fusion between different types of distance sensors such as sonar, infrared. However, these sensors are directional, that is, only obstacles on the straight line whose the sensor is pointed can be noticed. An alternative is a line laser rangefinder. The use of this approach is efficient because it requires no directional pointing, i.e., all obstacles on the line laser of sight are perceived at the same time [1].

In [2] a laser rangefinder is used to detect obstacles and create a map for mobile robot. [3] and [4] use the laser rangefinder just to detect obstacles, but [4] developed the system on a general-purpose computer using $C\#$ language, reaching an error in the distance measurement to $\pm$3cm, and [3] developed the system for Android platform executing on a smartphone, tablet or netbook. In [5] is used a data fusion of stereo camera and laser rangefinder, which is used for gathering legs data in a process of detection and Tracking of human being. In all these applications, the processing is performed by a computer, generating a high throughput, high computational cost and obstructing the design of the embedded system. In order to obtain a completely on board embedded system, it is possible to implement the system in a FPGA (Field Programmable Gate Array).

A real-time embedded $3D$ vision measurement system based on the line laser is implemented in FPGA and DSP (Digital Signal Processor) in [6]. These devices are used to perform image processing and control of a high-speed camera. This kind of camera is too expensive, but provides more efficiency. The camera calibration is described in MATLAB software in [6]. The line laser and camera manipulation by FPGA is also performed by [7], however for the purpose to obtain $3D$ vision data and identify weld seam.

In this context, this paper proposes the modeling of a line laser rangefinder based sensor using a common camera, different than [6]. This model is validated in MATLAB software and partially simulated based on Verilog implementation, so it will be possible to implement it in FPGA later, aiming processing speed by pipeline and reconfiguration of architecture. No extra hardware is used to pre-process the image, like a DSP in [6] and [7], and the entire sensor is on board. Furthermore, the hardware architecture of the sensor is presented.

This paper is organized as follows. Section 2 presents some preliminary concepts about the sensor model, specifying the implemented algorithms and its simulations. The hardware architecture of the sensor is presented in Section 3. Finally, conclusions are drawn in Section 4.

2. MODELING AND VALIDATION

Laser rangefinder is a simple technique used to obtain three-dimensional information by calculating the deformation of a laser point (or a laser line) projected on the surface of a measured object [8]. The point method is relatively simple in sense the image processing algorithms, but once only a point is got from one image, so the measurement efficiency is rather low when compared to the line method.

Figure 1 shows a system to measure distance based on a laser rangefinder. A laser module (laser pointer) is used to project a brilliant red dot in the object in front of the robot. So the camera captures a frame and uses the projected dot position on its image plane to compute the distance to the obstacle based on simple trigonometry. This method is described by [3] and [9].

The laser is projected in an object at distance D from the robot. This red dot is reflected and projected in the camera's image plane. The distance pfc (pixels from center) between the center of the image plane (in the optical axis) and the red dot in the image plane is proportional to the distance D. It is possible to calculate D by the equation 1. The distance between the camera and the laser (h) is known previously, the number of pixels from the image center to the red laser dot (pfc) is obtained from the image. The radians per pixels (rpc) and the radian offset (ro) are obtained calibrating the system [3, 9].

$$D = \frac{h}{tan(pfc * rpc + ro)} \tag{1}$$

The calibration process is performed by taking a set of pairs of images taken at distances D known, as the height h, according the Algorithm 1. These pairs consist of an image of the environment to be sensed and another image of the same environment, illuminated by the laser. The binarization of the difference between these images will generate a black and white image only with the laser, and the Algorithm 2 shows how to calculate the laser coordinates. These coordinates are used to evaluate pfc. The next steps is solve the system of nonlinear equations to find a set of values of rpc and ro, and realize a polynomial interpolation to find the best values of rpc and ro that satisfy the expression $\theta = pfc * rpc + ro$. This is implemented at Lines 7 and 9 of the Algorithm 1, respectively. Once calculated the values rpc and ro, it is possible to execute the Algorithm 3 to distance calculation between camera and target according

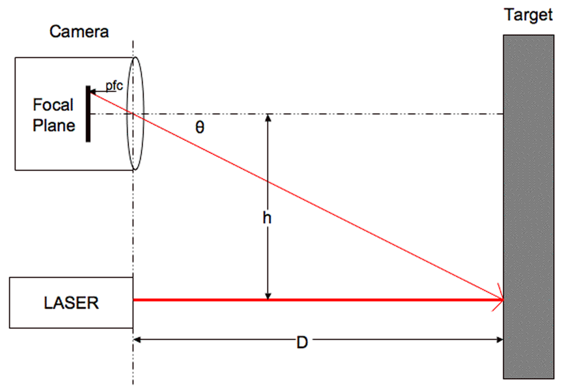

Figure 1: Geometric view of the laser rangefinder.

ALGORITHM 1: Camera Calibrate.

Input: Pairs of images (*imgBase*, *imgLaser*) with known distances D, and the parameter h.
Output: The parameters rpc and ro to the camera.

```
1 for each pair of images (imgBase(i), imgLaser(i)) in grayscale, such as $i = 1..size(D)$ do
2     $diff$ = abs($imgLaser$ - $imgBase$);
3     $binaryDiff$ = GRAYtoBIN($diff$, 85);
4     ($row$,$column$) = LaserDetection($binaryDiff$);
5     $centerRow$ = $binaryDiff$.rows/2;
6     $pfc$(i) = abs($row$-$centerRow$);
7     ($rpc$(i),$ro$(i)): Solve the system of nonlinear equations "$D(i) = h/\tan(pfc(i)*rpc + ro)$" for $rpc$ and $ro$;
8 end
9 ($rpc$,$ro$) = polyfit($pfc$, $rpc$, $ro$, '$pfc * rpc + ro$');
```

Equation 1. The calibration was performed in MATLAB as in [6].

It is important to remark that algorithm models were implemented in MATLAB, because this platform provides tools that support the generation of HDL (Hardware Description Language) code. However, it was used a minimum of native functions of MATLAB, giving preference to elementary operations contained in any HDL, in particular in Verilog, whose syntax is close to mid-level programming languages, facilitating the realization of testbenches.

More complex calculations, such as solving a nonlinear equations system (see Line 7 from Algorithm 1) or a polynomial interpolation (see Line 9 from Algorithm 1), is performed only in the calibration process, which is done offline. So there is no problem in using the functions *fsolve* and *polyfit*, respectively. In future work, the calibration can be implemented in hardware using numerical methods as an alternative to such functions, implementing, for example, the Quasi-Newton method and the Lagrange Interpolation, respectively [10]. In the Algorithm 3 at Line 3, the tangent function is performed offline and its results are saved in memory, as described in section 3, without problems for implementation in hardware.

ALGORITHM 2: Laser Detection.

```
Input: Binary image $binaryDiff$ where the white pixels
       represents the laser.
Output: Coordinates ($row$,$column$) of the center of the
        laser in the $imgLaser$.
1 for each column of $binaryDiff$ that contains a white
  pixel do
2     $height$ = max. number of white pixels per column;
3     $width$ = number of columns that contain white
      pixels;
4     row = max. row coordinate of a white pixel;
5     column = max. column coordinate of a white pixel;
6 end
7 column = column - floor($width$-1)/2);
8 row = row - floor(($height$-1)/2);
```

Also, although the project has been designed for the use of a line laser, this paper describes the implementation of a dot laser rangefinder, without loss of generality, since to extend this implementation to a laser line, is only necessary to perform the algorithms described here for each column of the image, considering $pfc = sqrt((laserRow - centerRow)^2 + (laserColumn - centerColumn)^2)$ and keeping in mind that there will be one laser coordinate by column.

ALGORITHM 3: Distance Calculation.

```
Input: Coordinates ($row$,$column$) of the center of the
       laser in $img$, and the parameters $h$, $rpc$ and $ro$.
Output: Distance $D$ between camera and target.
1 $centerRow = img$.rows/2;
2 $pfc$ = abs($row$-$centerRow$);
3 $D = h/\tan(pfc*rpc + ro)$;
```

Figures 2(a) and 2(b) show images processed by the Algorithms 1 and 3. Figure 2(c) shows the binary image highlighting the laser. With fifteen pairs of images ranging D from 70cm to 10.3cm, with h = 5.1cm, it was possible to estimate rpc = 0.0016128 and ro = 0.072809, and calculate distances on the images. The relation between the values of pfc achieved in the calibration and the respective values of D can be seen in Figure 3. The simulation results are shown in Table 1, which has a max measurement error of -3.89%.

3. ARCHITECTURE

Figure 4 shows the hardware architecture of the sensor. Each architecture block implements a functionality of the Algorithms 2 and 3. FIFO (First In Fisrt Out) Structures are used to synchronize the data production and consumption between blocks, enabling the parallel execution via pipeline. This is possible because all the processing is done pixel by pixel, without any correlation with past values or neighboring pixels values.

According to the architecture, the sensor operates as follows. The robot requests the sensor a distance measurement. So is captured an image ($imgBase$), sent pixel by pixel in grayscale and stored in memory. Then the laser is activated, a new image ($imgLaser$) is captured and sent pixel by pixel in grayscale to the module *Image Difference.* This module

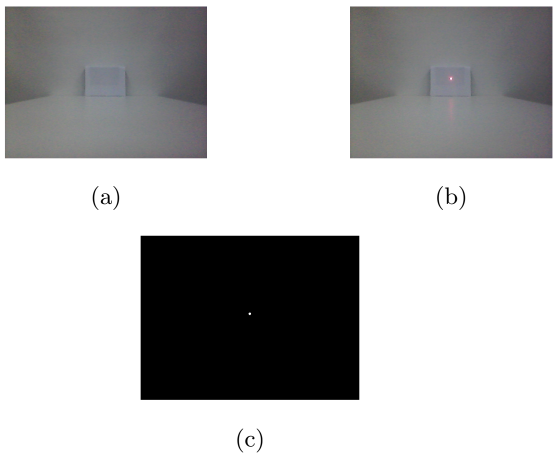

Figure 2: Images to distance calculation. (a) base image; (b) bright image; and (c) binary image.

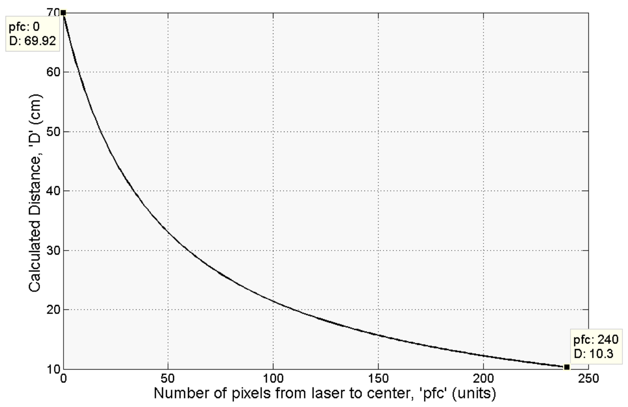

Figure 3: Relation between pfc and D.

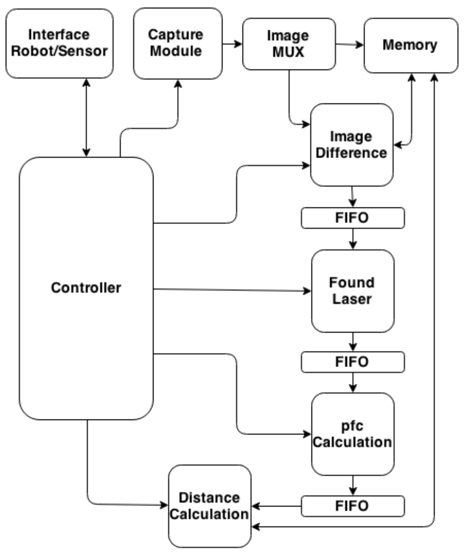

Figure 4: Hardware Architecture

Table 1: Results of simulation.

Pixels from Center (cm)	Real Dist. (cm)	Estimated Dist. (cm)	Error (Abs.)	Error (%)
1	70	68.40	-1.60	-2.28
5	65	62.92	-2.08	-3.19
9	60	58.25	-1.75	-2.91
13	55	54.23	-0.77	-1.41
18	50	49.91	-0.09	-0.19
25	45	44.89	-0.11	-0.25
34	40	39.74	-0.26	-0.66
45	35	34.83	-0.17	-0.48
57	30	30.68	0.68	2.26
78	25	25.34	0.34	1.36
107	20	20.37	0.37	1.83
161	15	14.77	-0.23	-1.53
213	12	11.53	-0.47	-3.89
231	11	10.68	-0.32	-2.87
234	10.9	10.55	-0.35	-3.19
238	10.8	10.38	-0.42	-3.88
240	10.5	10.30	-0.20	-1.94
240	10	10.30	0.30	2.97

calculates the difference between the pixel received and its corresponding in memory, then binarizes this pixel and sends it to the FIFO. The *Found Laser* block analyzes each pixel read from the FIFO. If it is a white pixel, its coordinates are sent to a new FIFO, which feeds the *pfc Calculation* block. This block is responsible for calculating the distance in pixels from the laser to the center of the image (pfc). Finally, the value of pfc is used to obtain the actual distance value D, which is returned to the robot.

Regarding the image capture, this task is performed by the *Capture Module* that uses a camera $OV7670$ at $30fps$ to take pictures in 640×480 resolution and $RGB555$ pattern, converts each pixel to grayscale and to sends one at a time. Another important note is about the *Distance Calculation*. In MATLAB model the Equation 1 is evaluated using the function $tan()$ to calculate the tangent of the angle $\theta = pfc * rpc + ro$. But, once all possibles values of pfc are known, and the parameters rpc and ro are constants for each camera [3], it is possible to calculate offline, during the calibration, each value D associated to each value of pfc, and save it in memory (these data can be seen at Figure 3). This way, the hardware complexity to calculate D is the complexity of one memory access.

An alternative to storing the D would be the implementation of the CORDIC algorithm (Coordinate Rotation Digital Computer). But its implementation in FPGA generates a very large area. It becomes an attractive solution if the project were synthesized in an ASIC.

4. CONCLUSION

We propose in this paper the modeling of a sensor based on a laser rangefinder line to detect obstacles and measure the distance. This model was implemented and validated in the Matlab software in order to provide an easily transcribed code for a HDL, especially for Verilog. Some tests were performed and distance calculation showed maximum error of -3.89%. The system was tested with a dot laser rangefinder, without loss of generality, since it is simple to extend this implementation to a laser line. It is also shown the sensor hardware architecture, designed for FPGA, which enables parallel processing with pipeline and a reconfigurable architecture. Future works can be performed to implement camera calibration in FPGA, making the sensor completely independent from external systems.

5. ACKNOWLEDGMENTS

We would like to thank the FAPESB, which partially funded this work.

6. REFERENCES

[1] Van-Dung Hoang, D.C. Hernandez, Han-Sung Park, and Kang-Hyun Jo. Closed-form solution to 3d points for estimating extrinsic parameters of camera and laser sensor. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pages 1932–1937, June 2014.

[2] O. Haffner and F. Duchon. Making a map for mobile robot using laser rangefinder. In *Robotics in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on*, pages 1–7, Sept 2014.

[3] Rafael V. Aroca, Aquiles F. Burlamaqui, and Luiz M. G. Gonçalves. Method for reading sensors and controlling actuators using audio interfaces of mobile devices. *Sensors*, 12(2):1572–1593, 2012.

[4] Ahmed A Hamad, Dhafir A Dhahir, Bushra R Mhdi, and Wadah H Salim. Laser distance measurement by using web camera. *IOSR Journal of Engineering*, 4(4):25–28, 2014.

[5] B. Ali, A.H. Qureshi, K.F. Iqbal, Y. Ayaz, S.O. Gilani, M. Jamil, N. Muhammad, F. Ahmed, M.S. Muhammad, Whoi-Yul Kim, and Moonsoo Ra. Human tracking by a mobile robot using 3d features. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 2464–2469, Dec 2013.

[6] Zhao Wang, Xuewei Cao, Haitao Song, Han Xiao, Wenhao He, and Kui Yuan. Embedded 3d vision measurement system based on the line structured light. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pages 6012–6017, June 2014.

[7] Yuanyuan Zou, Mingyang Zhao, Lei Zhang, and Chunying Jiang. Development of laser stripe sensor for automatic seam tracking in robotic tailored blank welding. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 3062–3066, June 2008.

[8] K. Ohtani, Li Li, and M. Baba. Laser rangefinder calibration based on genetic algorithm. In *SICE Annual Conference (SICE), 2012 Proceedings of*, pages 1234–1237, Aug 2012.

[9] Christian E Portugal-Zambrano and Jesús P Mena-Chalco. Robust range finder through a laser pointer and a webcam. *Electronic Notes in Theoretical Computer Science*, 281:143–157, 2011.

[10] K. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 1989.