Boarding and Acceleration on FPGA of an Obstacle Detection Module over Stereo Image

Mohamed Salah Salhi*, Manel Salhi, and Hamid Amiri

University of Tunis El Manar, National Engineering School of Tunis-Tunisia
Research Laboratory of Signal Image and Information Technology LR-SITI.
**Corresponding author: medsalah.salhi@enit.utm.tn*

Accepted: 1st May 2023

OPEN ACCESS

Abstract: This paper presents an accelerated algorithm realized on field programming gate array FPGA to mimic the human vision system. The FPGA is chosen as support owing to their real time propriety RTP. The proposed approach aims to detect any natural image and obtain a "cyclopean image" that provides information on the objects depth with a better details resolution. This strategy uses the common advantages of hybrid stereo imaging models in processing two images of the same scene captured from different viewing angles. Then, it merges the two views to produce a final image containing the depth information of different objects. The adopted algorithm is applied to obstacles detection in which execution time is experimented and shown in results below.

Keywords: Accelerated algorithm; Stereo imaging models; Cyclopean image; Real time propriety RTP.

Introduction

The stereo image history is linked to the human vision way, which exhibits many interesting properties. One of them is the ability to feel the depth of the scenes seen. The process behind this ability is called “stereopsis”. In fact, the brain processes the images from the left and right eyes and executes an image called a "cyclopean image". The cyclopean image then provides information on the objects depth with a better details resolution.

In order to mimic the human vision system, stereo imaging models have been created. The stereo vision system processes two images of the same scene captured from different viewing angles. Then, it merges the two views to produce a final image containing the depth information of different objects.

Stereo imaging has found its application in many fields such as engineering, architecture, science, education and the military. We can find systems based on stereo imagery in 3D computer graphics, simulators, training systems and robotics.

The primitives and dimensions established by a stereo image have been widely applied in home automation, robotic and other security measures.

A growing field of application is the autonomous driving of robots through guidance systems. The stereo imaging sensors used in this application type are the most often mounted in a "fish-eyes» topology. They, therefore, have good characteristics such as the wide detection angle: 3/4 of a surrounding circle, with vertical and horizontal resolution, the ability to detect obstacles in a short time and the simultaneous detection of different obstacles with information about their positions and shapes.

In this paper, we try to take advantage of stereo imagery to develop a ground obstacle detection module. This module will allow a robotic system to sense its environment to make better decisions while determining its orientation.

Indeed, ground is considered a reference when detecting obstacles and determining their real position.

Many methods have been used to tackle the ground plane estimation, such as Least Squares LS, Eigen Value EV, and Expectation Maximization Methods EMM, etc [1].

Ground plane estimation is about distinguishing the ground pixels in an image. For an application, like pedestrian detection, ground estimation is important.

This paper is presented under different sections whose study and analyzes respectively a literal review on stereo image techniques, the adopted strategy and the experimented results.

This section presents different ground detection methods. We will explain their principles and give an overview of their results.

The process of detecting traversable regions is of great importance especially for navigation systems [2], [3], so it needs a system capable of providing rich information to accomplish the task efficiently. For this reason we have chosen to use stereo imagery because it provides more information compared to other systems based on lasers, radar or monocular systems, etc.

Stereo vision calculates disparity by measuring distances between objects. Two ways to deal with the resulting image are built according to the disparity:

The first way is the 3D reconstruction according to the point cloud derived from a disparity map. Then, obstacles and the ground areas can be detected using edge detection, local safety map, level adjustment, etc [4]. However, this method is recognized by its high computational value, which makes it difficult to meet real-time requirements in robot navigation.

The second method is based on the construction of a v-disparity image. Labayrade introduced this technique to detect obstacles either on a flat or non-flat road. The advantage of this method, as mentioned earlier, is that it does not consider the flat road assumption or the need to extract specific structures like road edges. It consists of accumulating pixels having the same disparity along each row of a disparity map image. The resulting image includes significant patterns that reflect different objects constructing the actual scene [5].

Obstacles, arranged perpendicular to the robot, are represented by vertical lines of width indicated by the pixel intensity. In addition, the road which is modeled as a succession of flat surfaces will be mapped as an oblique line segment. This line is called the ground correlation line. It represents the traversable region that contains information on the coverage angle of the cameras [6], [7].

Considering the lack of hardware resources on an FPGA platform, the v-disparity-based processing system is more appropriate even if the computation of the disparity map looks like an inhibiting factor.

The work of Benenson showed that the construction of the v-disparity image can be done directly so that this method became the best candidate for our case. Additionally, inspecting the image required an algorithm to adjust the lines. Hough's transform was an option, but the 'Iteratively Reweighted Least Squares' IRLS algorithm presented a simpler and more robust solution to implement [8].

The computation of least squares weighted by the weights is performed iteratively to solve linear regression problems. The algorithm produces weights for the dataset that group observations in a tight linear relationship. The IRLS is used to calculate Maximum Likelihood Estimates' MLE. It is also popular for calculating robust estimates. It can offer a quick solution for difficult problems. Thus, the use of the IRLS algorithm is more practical in performing a linear regression that will allow the determination of the ground representative line in the constructed image [9], [10].

Determination of the Ground under the hypothesis of a flat road

The work conducted by Manolis and al. presents a method allowing a mobile vehicle to locate obstacles using a pair of images taken from its environment. Their method assumes that the vehicle is traveling on level ground [11]. Their applied technique calculates a homography of the ground, which makes it possible to determine the movement of the ground and then allows the obstacles detection. Thus, the camera calibration is not necessary and the technique applies to both single and stereo images, so it does not rely on the disparity map calculation nor go through 3D reconstruction [12]. Their proposed technique starts by selecting and matching a set of corners from a pair of images, then, using the Least Squares Algorithm LSA based on the Least Squares Median calculation, it identifies which of these corners belong to the ground. The following figure illustrates the processing flow of this technique.

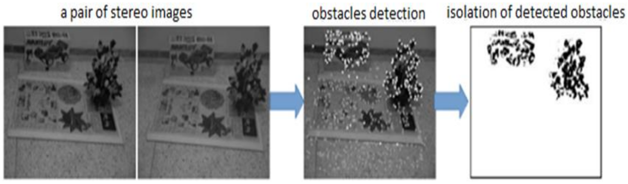


Figure 1. Ground sensing design flow based on plane geometry assumption

The picture in the middle represents the detection result of aberrations by the median of least squares when estimating the ground homography.

Role of field knowledge in obstacle detection

The work of Zhongfei Zhang and al. [13] investigated the role of a priori knowledge of the ground region in obstacle detection. Three types of algorithms were examined and a conclusion was made that the algorithm, which continuously estimates the ground plane is the most robust and real. In fact, their research compared three types of algorithms: the first, which assumes a priori knowledge of the ground region, the second, which determines the ground region without prior knowledge and the third continuously, estimates the ground surface. The first two processes were sensitive to noise caused by ground surface irregularities, such as bumps and pits, but they were quick in determining the ground area. Their weakness is that they measure the deviation of data acquired from a known or unknown planar configuration, but it is not usually the case for the ground plane to be planar [14].

The ground surface estimation algorithm has been continuously proven to be robust to noise. The results given in the work of Zhongfei Zhang and al. show that the continuous estimation of the ground plane is better compared to other algorithms. Yet, its use offers an advantage which is summed up in its ability to determine the height of obstacles although it requires partial calibration. However, it becomes ineffective in the absence of calibrated cameras [15].

Extraction of image primitives in obstacle detection

Dieter Koller and al. [16], in their research, proposed to use the features of track marks to detect the ground surface. Their project used stereo imaging techniques to extract features from the road scene to help guide a vehicle while assuming the road level. Their developed model, using stereo analysis, continually predicts the position of lane markings that dramatically improves overall robustness and accuracy. Nevertheless, this technique depends closely on the quality of the acquired images, even if the track marks are not visible, which is a frequent case that limits the functionality of this technique [17].

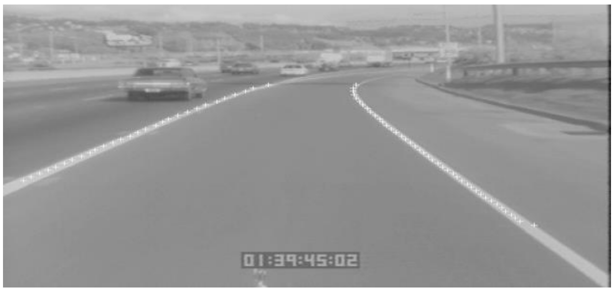


Figure 2. Estimation of markings on a lane

Ground determination by monocular sequences

Ground can be determined by processing monocular images. However, special algorithms need to be developed. Jin Zhou [18] presented in this work, an approach to detect the ground plane based on homography. The images acquired are monocular with the advantage of being cost effective and the technique does not need calibration as in stereoscopic approaches. A homograph of the ground plane is determined from the derived constraints related to the positioning of the camera. Then, specially designed algorithms determine the ground surface. The algorithms presented are efficient, robust and precise [19].

In this figure, a set of random corners is selected and the ground is determined by green crosses using appropriate algorithms.

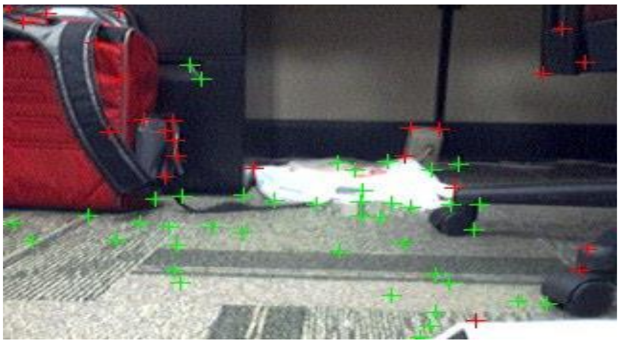


Figure 3. Illustration of a ground detection based on the monocular image

The ground detection by the "v-disparity" image

Usually, the ground surface is not planar. It has its ups and downs due to its topography.

Therefore, the often assumption based on planar ground is invalid. In fact, this idea causes various problems, such as imprecision and false detection of obstacle positions.

Labayrade and al. [20] have proposed a new method, which can increase the reliability of the obstacle detection process. His method could detect obstacles without taking plane geometry; it was also original, fast and robust.

The advantage of this approach is its ability to adapt to the up and down slopes. It is based on the construction and processing of the v-disparity image, which includes the projection of different features forming the scene being watched. The 'v-disparity' image provides a semi-global match. Obstacle detection is robust to partial occlusion or errors made during its construction. In addition, it is not necessary to extract external structures like lane markings or road edges to detect the ground region from a pair of stereo images.

The construction of the v-disparity image is relatively simple. Based on it, it is possible to determine an obstacle detection method in both cases of plane and non-plane geometry.

After estimating the road profile, objects located above the ground surface will be considered potential obstacles. Therefore, an accurate calculation of the points of contact between the ground and obstacles can be determined. See figure 4 below.

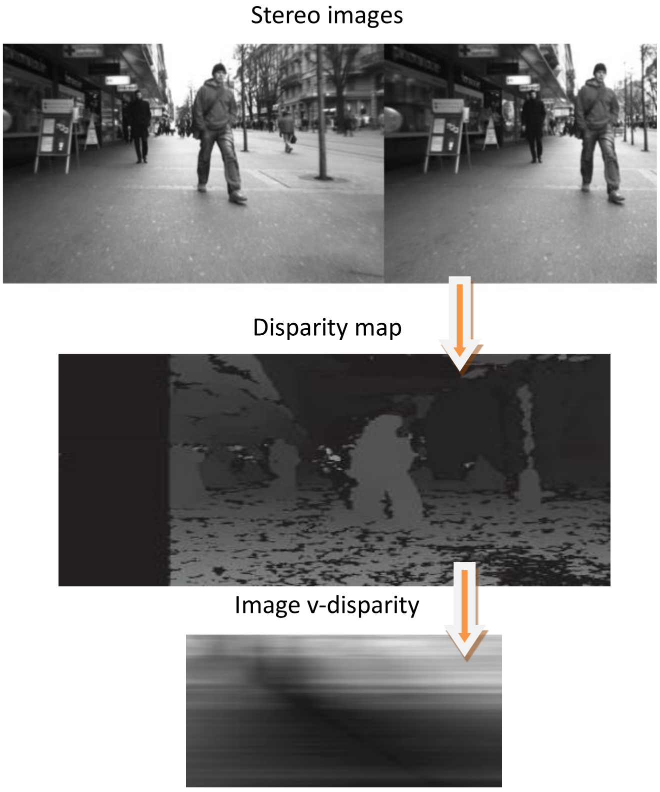


Figure 4. Representation of the V-disparity approach

Comparison between existing ground detection techniques

The following table compares the different algorithms presented in ground detection. Obviously, the image processing v-disparity technique is more advantageous than the other methods. However, the passage through the construction of the disparity map complicates the implementation of this algorithm on an FPGA.Currently, a new approach is introduced in the sense of ignoring the calculation of the disparity map and directly constructing the v-disparity from a pair of stereo images while maintaining the same quality as that proposed by Labayrade [20].

Materials and Methods

Although the presented techniques give robust and fast results, they require General Purpose Processors GPPs and Graphic Processor Units GPUs implementation. Therefore, these solutions are not suitable for applications that lack power and space resources like small robots. Therefore, implementation on an FPGA platform is necessary to have low power consumption and to preserve the same processing quality.

Due to the considerable speed gain and the simplicity of the calculation, we will adopt the concepts developed by R.Benenson and make them suitable for implementation on an FPGA.

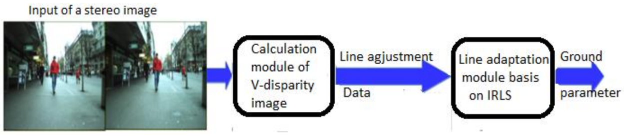

Input of a stereo image
Calculation module of V-disparity image
Line agjustment
Data
Line adaptation module basis on IRLS
Ground
parameter

Figure 5. The stages of soil detection

In this work we propose to implement on FPGA a ground detection module. The developed module could meet real-time execution requirements. It should allow an autonomous driving system, such as a robot, to integrate a low power, high performance and effective cost processing system.

It could therefore enable the vehicle to process the stereoscopic images in flow and to determine the ground region in a scene, making it possible to detect obstacles and to avoid them.

The best suited method to implement on the FPGA was chosen from other techniques. This method is based on building a v-disparity image and determining its ground region. The verification of the chosen algorithms was conducted by implementing them on Matlab and studying its feasibility by dividing it into smaller modules.

This approach leads to the design and implementation of specialized state machines, which calculate from a given pair of stereo images an image of v disparity. The latter will allow the system to determine the ground and any obstacles in the scene being watched.

Table 1. Comparison between different approaches in ground detection

Techniques	Inconveniently	Advantages
Flat road assumption	• Complex algorithm. • Need a textured floor. • The algorithm can be faulty if the data are wrongly chosen	• No camera calibration. • Stereo or monocular application. • Efficient and robust.
Prior knowledge of the ground	• Sensitive to noise. • There are cases where the system is	• Robust in continuous ground estimation. • Fast when you have prior knowledge of the

	unresolvable. • Complex algorithm.	ground surface.
Lane marking method	• Depends on the clarity of the ground characteristics • Real-time issue	• Good performance in clear scenes • Simple and robust
Monocular Sequence Methods	• Complex algorithm • Poor results if the ground is not dominant in the scene	• Efficient and robust • Profitable • No calibration required
V-disparity method	• Using the disparity map slows down the process • The disparity map needs more processing such as smoothing and filtering	• No plane geometry assumption • Fast, reliable and robust against noise • Precise • Simple algorithms • Rich information included in the v-disparity image

V-disparity image computation

Given a stereo-image (left and right images), we take only the lower half part to compute the V-disparity image. The algorithm can be written as follows:

```
For each row in the left image

{
get left_row_iterator_beginning;
get left_row_iterator_ending;
get right_row_iterator_beginning;
for (disparity = 0; disparity <disparity-max; disparity++);
{
left_row_iterator = left_row_iterator_beginning + disparity + offset;
right_row_iterator = right_row_iterator_beginning;
V-disparity_cost = (Σmin(cost, cost_sum_saturation ))/3;
v-disparity-image_row[d] = V-disparity_cost;
}
}
```

The cost function is computed as follows:

```
cost = |left_r − right_r| + |left_b − right_b| + |left_g − right_g|;
_r = red pixel;
_b = blue pixel;
_g = green pixel;
```

Line parameter extraction:

The line parameter extraction is based on the Iteratively Reweighted Least Squares (IRLS).

This algorithm is made to minimize the system: w(i) ∗ (b−A∗x)

“w” stands for the weights chosen for computation. The algorithm for this part can be written as follows:

Select points for line fitting:

```
for ( disparity = 0; disparity <disparity-max; disparity++); {
if (V-disparity_row <= threshold); {
store the point of coordinates “ x= disparity “ and “y= row_number”;
}
}
```

Initialization of A,b,x and w:

```
A = x11. .. .. .. .xn1;
B = y1....yn;
X = ab;
W = w1....wn;
```

“a” stands for the “slope of the line to be detected ” .
“b” stands for “the coordinate at the origin”.

Main algorithm

```
for(I = 0; I < number_of_iterations, i++); {
recompute weights
w(i) = w(i)∗w(i−1);
solve w(i)∗A∗x = w(i)∗b;
}
```

Recomputed weights

The error vector can be written as:

$$e = b - A*x \qquad (1)$$

We can compute weights according to “Tukey” as follows:

$$w(i)=(1-(e\ (i)/c\ )2)2 \qquad (2)$$

Hardware implementation

In this section we will present our solution for the ground detection module implementation on an ARM-FPGA platform. The processing frequency of our module is 100 MHz.

Proposed hardware solution

We have chosen to implement the v-disparity image construction module on the programmable logic part of the FPGA. We decide to implement the line adaptation module on the processing system.

Communication between the modules will be maintained in the AXI Bus interface.

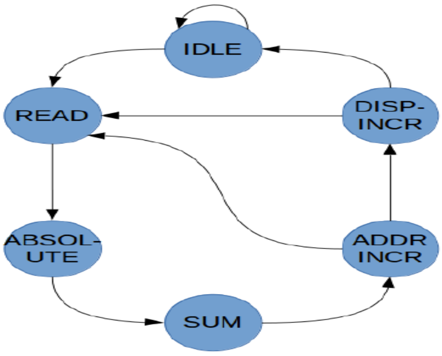


Figure 6. State machine for v-disparity image construction (first draft)

Implementation of the V-disparity module and results

A state machine for building v-disparity images has been designed and implemented as shown in the following figure. However, the execution time was enormous and its image processing capacity was nowhere near 25 fps (frames per second). One row of the stereo image pair took a time of 3 ms, so we got a disparity image for 3ms * 240 rows = 720 ms, which is less than 2 fps.

In the state "IDLE_State", the FSM for "First State Machine" is inactive.

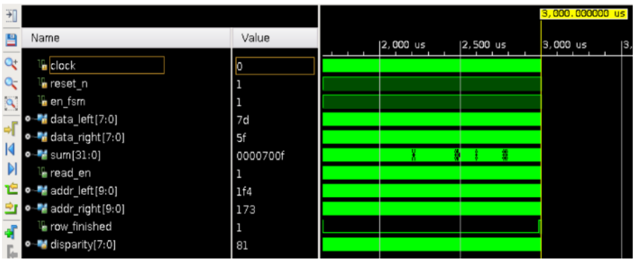
In the 'READ_State' state, the FSM starts reading buffers.

In the 'ABSOLUTE_State' state, the FSM calculates the absolute difference between the right and left pixels.

In the 'SUM_State' state, the FSM adds the new difference to the cost.

In the "ADDRINCR_State" state, the FSM increments the address of the buffers from which it reads the different pixels values. Then, it starts to read if it has not reached the end of the left buffer, otherwise it will increase the disparity.

In the 'DISPINCR_State' state, the FSM increments the disparity. Then, it starts reading if it does not reach the maximum disparity value; otherwise it goes into the inactive state.



Name	Value
clock	0
reset_n	1
en_fsm	1
data_left[7:0]	7d
data_right[7:0]	5f
sum[31:0]	0000700f
read_en	1
addr_left[9:0]	1f4
addr_right[9:0]	173
row_finished	1
disparity[7:0]	81

Figure 7. Runtime for the first draft implementation (3000us)

Fast Track Solution

To achieve higher performance, a new design of finite state machines has been proposed. Algorithm profiling has shown that the cost calculating for each disparity is the part which consumes the most time of the algorithm. Optimization involves the parallelism of this process and the processing of 8 pixels at a time.

Because the luminescence value of the pixels is a positive integer, the cost can be written as follows;

$$cost = \sum_{n=1}^{n\_data} |a_n - b_n| = \sum_{n=1}^{n\_data} max(\,a_n, b_n) - \sum_{n=1}^{n\_data} min(\,a_n, b_n) \tag{3}$$

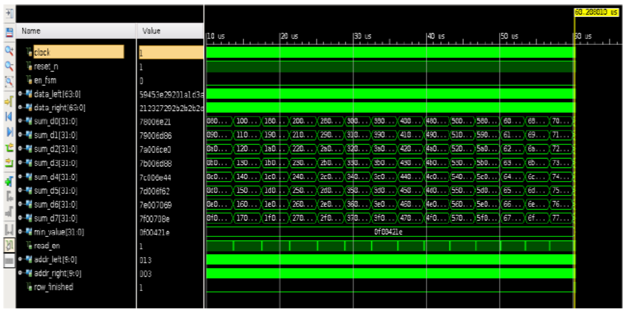
With (an) is a left pixel value and (bn) is a right pixel value. In this way we can focus only on the sum of the max and min values in parallel, for the difference.

So, by processing more data (8 pixels), the process could achieve more processing speed. It can process an image in 15.2ms and it can reach higher speeds. The acceleration rate was 740/15, 2 = 48 times.

The following figure shows the computational time taken to complete a row in the v-disparity construction. The state machine can calculate 8 disparities at a time that corresponds to the variables sum_dn.

Implementation of the IRLS algorithm

The algorithm was developed and implemented using the C language. A runtime test was performed on an ARM (Advanced RISC Machine) Processor integrated on the FPGA. The results showed that it can run in enough 18ms for our system to get it up to 25fps of speed.



Name	Value
clock	1
reset_n	1
en_fsm	0
data_left[63:0]	59453e29201a1d3a
data_right[63:0]	212327292b2b2b2e
sum_d0[31:0]	78006e21
sum_d1[31:0]	79006d86
sum_d2[31:0]	7a006ce0
sum_d3[31:0]	7b006d88
sum_d4[31:0]	7c006e44
sum_d5[31:0]	7d006f62
sum_d6[31:0]	7e007069
sum_d7[31:0]	7f00708e
min_value[31:0]	0f00421e
read_en	1
addr_left[9:0]	013
addr_right[9:0]	003
row_finished	1

Figure 8. Execution time after acceleration (60us)

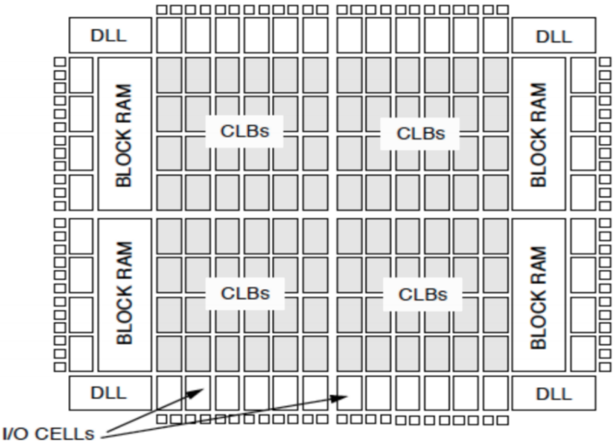


Figure 9. The FPGA architecture

Field-Programmable Gate Arrays FPGAs are semiconductor chips incorporating Configurable Logic Blocks in a 2D array order. CLB's are connected via programmable interconnects. In fact, the interconnects make a network of vertical and horizontal wires that constitute links between the CLBs. Each intersection between the wires lodge a switcher that permits reconfiguration of CLBs. Modern FPGA's embed hundreds of thousands of CLBs with the inclusion of hardened functional units.

The FPGA construction permits fast and efficient development of common functions. It is possible to program electronically the FPGA device by loading a bit stream configuration file into the device. The configuration is memorized in an SRAM memory, consequently an FPGA can be reprogrammed more than once.

Validation of the v-disparity image construction module

The simulation of the v-disparity image construction module showed that the module could process one frame every 15.3ms. The quality of the resulting image was acceptable, but the destruction of the soil parameters had to be accurate due to the good quality of the resulting image.

Validation of the IRLS module

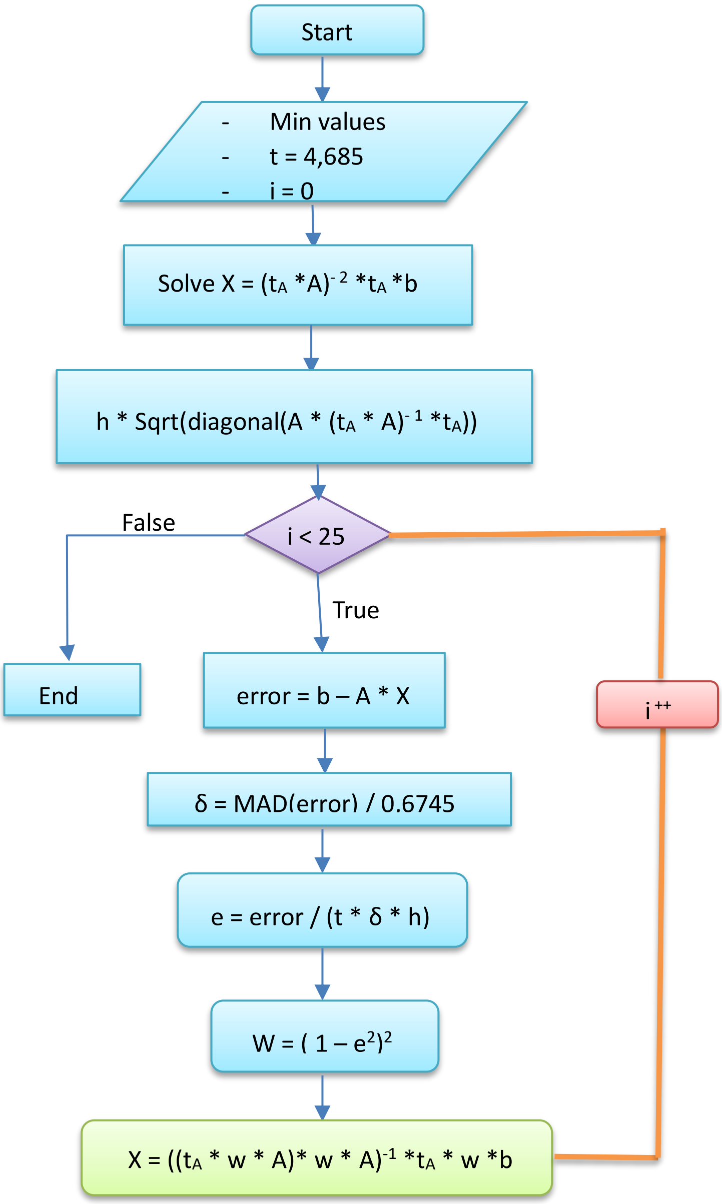

Start
- Min values
- t = 4,685
- i = 0
Solve $X = (t_A * A)^{-2} * t_A * b$
$h * \mathrm{Sqrt}(\mathrm{diagonal}(A * (t_A * A)^{-1} * t_A))$
False
$i < 25$
True
End
$error = b – A * X$
i^{++}
$\delta = \mathrm{MAD}(error) / 0.6745$
$e = error / (t * \delta * h)$
$W = (1 – e^2)^2$
$X = ((t_A * w * A)^* w * A)^{-1} * t_A * w * b$

Figure 10. Flow chart for IRLS algorithm implementation

The IRLS calculation module was developed using C code and executed the ARM Processor embedded on an FPGA. The results showed that the module can process a frame in 18ms. The total time taken to process a frame is 15.3 ms + 18 ms = 33.3 ms. Therefore, the system could process: 1/33, 3 ms = 30 fps.

Conclusion

This topic presented the design, implementation and optimization of ground detection modules on FPGAs. We have chosen a method based on the construction of a v-disparity image, because; it is simpler to implement and robust. A direct method for constructing the v-disparity image made the implementation more eligible to try.

We studied the algorithms for constructing v-disparity images and the line adaptation used to detect the ground plane. For the implementation, we used Xilinx's Vivado development tool and zedboard development board.

The design included two main modules; the first allows the construction of v-disparity image. It was implemented on the programmable logic of the FPGA device. The second module aims at ground detection by the line alignment algorithm. It was implemented on the processing system.

Communication between the two modules was provided by the AXI bus interface.

The implementation obtained reached a speed of 30 fps at a frequency of 100 MHZ but the communication between the modules is still to be optimized.

References

[1] P. Ke, C. Meng, J. Li and Y. Liu, "Homography-based Ground Area Detection for Indoor Mobile Robot Using Binocular Cameras". IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM), 2011. https://doi.org/10.1109/RAMECH.2011.6070451

[2] M. I. A. Lourakis and S.C. Orphanoudakis, "Visual detection of obstacles assuming a locally planar ground". Conference, 3rd ACCV, volume II, pages 527-534, 1998.

[3] Z. Zhang, R. Weiss, and R.A. Hanson, "Qualitative obstacle detection". International Conference on Computer Vision and Pattern Recognition, pages 554-559, 1994. https://doi.org/10.1109/CVPR.1994.323881

[4] D. Koller, T. Luong, and J. Malik, "Binocular stereopsis and lane marker flow for vehicle navigation: lateral and longitudinal control". Technical Report UCB/CSD 94-804, University of California at Berkeley, Computer Science Division, 1994.

[5] J. Zhou, B. Li, "Homography-based ground detection for a mobile robot platform using a single camera", Robotics and Automation, IEEE International Conference. ICRA 2006. https://doi.org/10.1109/ROBOT.2006.1642332

[6] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection on non flat road geometry through v-disparity representation". In IVS, 2002.

[7] R. D. King, "A brief history of stereoscopy". WIREs Computational State, 5: 334340. 2013. https://doi.org/10.1002/wics.1264

[8] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares", Communications in Statistics - Theory and Methods, 6:9,813-827,1977. https://doi.org/10.1080/03610927708827533

[9] M. Galassi, J. Davies , J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich,"GNU Scientific Library". Reference Manual, Edition 2.2.1, for GSL Version 2.2.1,31 August 2016.

[10] M. S. Salhi, M. H. Rahmouni, and H. Amiri, “Evolutionary Deep-indicators Algorithm in Facial Recognition improvement over SoC”, International Journal of Advanced Science and Technology, Vol. 29, No. 5, pp. 5376 – 5387. 2020.

[11] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2), 328-341. 2009. https://doi.org/10.1109/TPAMI.2007.1166

[12] D. Scharstein, and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”. International Journal of Computer Vision, 47(1-3), 7-42. 2002.

[13] Q. Yang, L. Wang, and N. Ahuja, “A constant-space belief propagation algorithm for stereo matching”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(3), 467-480. 2010. https://doi.org/10.1109/CVPR.2010.5539797

[14] C. Vázquez, A. J. López, J. Amat, and I. Parra, “Real-time stereo vision system for obstacle detection and tracking on UAVs”. Sensors, 11(5), 5124-5142. 2011.

[15] Zbontar, J., & LeCun, Y., “Stereo matching by training a convolutional neural network to compare image patches”. Journal of Machine Learning Research, 17(1), 2287-2318. 2017.

[16] Z. Luo, T. Shen, L. Zhou, L. Zhang, H. Zhang, and Y. Li, “Obstacle detection for unmanned surface vehicle using stereo vision”. Journal of Marine Science and Technology, 22(1), 196-205. 2017.

[17] Y. Song, Q. Liu, and Y. Zhang, “An obstacle detection method for autonomous underwater vehicle

based on stereo vision". Ocean Engineering, 153, 67-77. 2018.

[18] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 340-349. 2018.

[19] X. Pang, Y. Sun, J. Ren, L. Chen, and X. Mei, "Deep stereo matching with cross-modal consistency". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1285-1294. 2019.

[20] J. Yin, J. Shi, M. Zhang, and Q. Zhang, "Fast stereo matching using adaptive joint bilateral filtering". IEEE Transactions on Circuits and Systems for Video Technology, 31(4), 1167-1180. 2020.

Publisher: Chinese Institute of Automation Engineers (CIAE)
ISSN: 2223-9766 (Online)

Copyright: The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.