Noise-reduction method based on color channel discrimination applied to laserline 3D reconstruction

Carlos Daniel Díaz Cano*, Jesús Carlos Pedraza Ortega, Cyntia Mendoza Martínez, Saúl Tovar Arriaga, and Juan Manuel Ramos Arreguín.

Facultad de Informática, Universidad Autónoma de Querétaro
carlosd.dc.it @gmail.com*, caryoko@yahoo.com, isc_cmendoza@hotmail.com, saulotov@yahoo.com.mx, jramos@mecamex.net

Abstract. In this paper, we present a noise-reduction method which relies on a RGB model, which most imaging device by default has, and the laser-line colors commonly used for image processing such as digitizing. It was observed that there is a white line on the projected surface and the noise could be interpreted as the light's laser color. We proposed an approach using the standard format RGB where the two color channels different to the laser color are used and tested along three different threshold values as a pre-processing methods for noise-reduction on surface reconstruction. We digitized six different objects, adding a linear interpolation, and comparing the results against the border detection method. As a result, the proposed approach was faster using almost the same number of points and its reconstruction was smoother compared to the border detection method. This could lead to a light algorithm for noise-reduction not requiring heavy methods unavailable in some languages.

Keywords: noise-reduction, laser-line, digitizing, RGB model, surface reconstruction.

1 Introduction

In this paper it is proposed a method for noise reduction focused on the color channels, resulting in a very light algorithm consuming less resources and having faster performance in the preprocessing step for noise-removal and binary image generation, which could be implemented along with more complex steps for digitizing methods.

There are noise reduction methods that use statistical, probabilistic, calibration [1], or primitive matching [2] approaches, some even go far by adding a training module for pixels which uses Gaussian Mixture Model, k-means clustering and covariance [3], making it a heavier algorithm. Such methods are implemented in modern desktop and specialized computers, however other less developed and limited systems such as microcontroller development boards or mobile devices could have slow processing.

It has been the standard that most image capturing devices employs the RGB color space, which is also the same color space used in the laser-line surface reconstruction.

* Corresponding author

When the laser-line is projected over a surface, it is noted there is a white line that by additive color is the result from the max value of all color channels. When a channel representing the same as the laser-line color was set to zero, the resulting image has a more defined line with lesser noise which is later converted to a grayscale image.

To binarize the image, three different methods were tested, Otsu's method [4], Kapur (or maximum entropy) [5], and a basic differentiation for an array. Applying such methods to a set of images it was found its global threshold. With the threshold found, the binary step was applied and the binary image reduced to one single pixel thick. This thickness was found by average of the pixel's position, and for the gap in the images, it was used a linear interpolation that always fills each gap found in the line, thus forming a false surface.

With each range image as a bi-dimensional matrix it was possible to map the position for each voxel in a 3D space once we get its rectangular form.

With the rectangular form of each point in the matrix, the mesh was formed letting to have a visual representation of the point cloud where it was possible to make an esthetical comparison between each result from a similar method [6].

2 Color and Image Devices

Image devices and sensors, both works as a radiation convertor into electrical signal [7]. This is because the "color" perceived by the human eye is in reality radiative energy absorbed or reflected by the different materials surrounding us. This creates a wavelength where at a certain level a color is perceived, this wavelength is known as the visible spectrum, measured in nanometers. This spectrum goes from 380nm to 740nm, any value over or below these values goes unperceived by the human eye, however the image sensors are different. Image sensors (commonly CCDs) work by accumulating electrical charges proportional to the intensity of received light [8], this is done with a photo-cells set that follow a pattern of Green, Red and Blue to produce an image, these produced images can be altered by wavelength not perceived by the human eye, being a reason it could give some undesired data.

Most image sensors use the RGB format, which it is named after R (Red), G (Green) and B (Blue) since it simplifies the design and architecture of the system which can take advantage of a large number of software routines due this space color has been around for years [9] and can also be transformed to other color spaces, however this space is limited to the number of bits used to store an image, 8-bits is the most common.

However, there have been a numerous color spaces and variations of RGB, where some color space model is used to reduce the effects of white ambient light [3].

3 Proposed Approach

The whole algorithm for surface reconstruction is shown in Fig.1. Said method was first tested on a ceramic bell painted in black shown in Fig. ***2***. The distance between the object and the camera is 30 cm, while the camera and laser had 25 cm distance and a 90° between said distances. Once the system is setup, then the lights are turned off leaving a dark room where a finite number of images are captured. The image set is

loaded and cropped into an interest region to get the object's center, leaving a shorter image reduces the time used in the processing required in each step. To get a better data of the line, the channel corresponding to the laser line color is set to zero, from there the three color channels are reduced to one single channel (grayscale image) to later get the binary image with an integer number as a threshold. Such value was found by applying a threshold selection method, among three different methods: Kapur, Otsu and the basic differentiation.

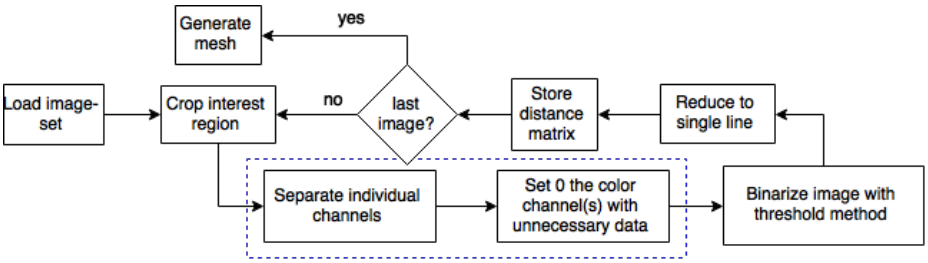


Fig.1. Flown diagram of the whole algorithm, the marked part is the proposed approach.

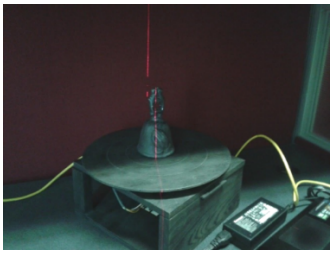


Fig. 2. One of the used objects was this ceramic bell. The rotatory platform was manually operated.

First tested method was a basic differentiation where one vector v with size n returns a vector v' with size n-1 where each element i is the result of subtracting the i element from the next i element, Equation (1):

$$v'(i) = v(i+1) - v(i) \tag{1}$$

First method had unsatisfactory results since sometimes the selected threshold yields a high value giving a totally black image, or with a low value gives a distorted image.

Second method, Equation (2), was Otsu's method [4] which is one of the most employed threshold selection methods:

$$\omega_0 = \sum_{i=1}^{t} p_i \qquad \omega_1 = 1 - \omega_0$$
$$\mu_0 = \frac{\sum_{i=1}^{t} i * p_i}{\omega_0} \qquad \mu_1 = \frac{(\sum_{i=t+1}^{N} i * p_i) - \mu_0}{\omega_1} \tag{2}$$
$$\max\left(\sigma_w^2(t)\right) = \omega_0 \omega_1 (\mu_1 - \mu_0)^2$$

Where the t variable is the threshold value currently evaluating while N is the maximum value in the range the pixel can take, or as in RGB model this is 255 (meaning t

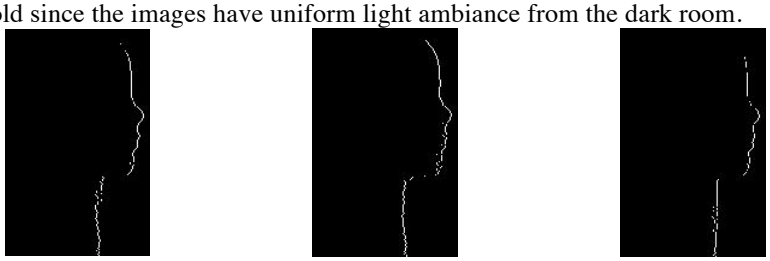
goes from 0 to 255), and p values are the probability obtained from the image's histogram, being ω_0 the cumulative probabilities from 1 to t while ω_1 from $t+1$ to N, similarly μ_0 and μ_1 are the cumulative moments of the histogram. The optimal threshold is the maximum variance σ_w^2.

Third method, Equation (3), was the maximum entropy [5], using the same variables as Equation (2) it is added the log of the probabilities, resulting in:

$$H_f(t) = -\sum_{i=0}^{t} \frac{p_i}{\omega_0} log \frac{p_i}{\omega_0}$$
$$H_b(t) = -\sum_{i=t+1}^{N} \frac{p_i}{\omega_1} log \frac{p_i}{\omega_1} \quad (3)$$
$$t_{opt} = \max (H_f(t) + H_b(t))$$

The maximum value of the sum of both entropies H_f & H_b, is the selected threshold.

By comparing the results, Figure 3, from each method, it was selected the Otsu's method with a complexity of O (N), this because even though some images had a few distortion, the Kapur method had a considerable loss in information, while the differential method sometimes had a completely black result. It was employed a global threshold since the images have uniform light ambiance from the dark room.



(a) (b) (c)

Fig. 3. Results from the reduction to a single line after getting the binary image from each threshold. (a) differential; (b) Otsu's method; and (c) maximum entropy.

As it is shown in **Fig. *4***, the binarized image gives better results when two channels are used instead of three when the threshold method, Equation (4), is applied.

$$f(x,y) \begin{cases} 0 \; or \; false, & p(x,y) \leq u \\ 1 \; or \; true, & p(x,y) > u \end{cases} \quad (4)$$

Where $p(x,y)$ is the current pixel in x,y and u is the global threshold value.

To reduce to a single pixel the center's line was calculated by using Equation (5). As a measure to get the correct center's object, a c corrective constant is added to address possible distortion in depth due the laser-line was a little off the center in the object since the system is not in a fixed position and the camera is not completely centered on the object due lack installations in the workplace requiring to assemble the system from zero each time it was used. This c corrective constant was found by heuristic ways.

$$md(x,y) = \frac{\sum p(x,y) * x}{\sum p(x,y)} + c \quad (5)$$

md is the distance matrix; p is the current pixel and c is the constant factor to rectify the depth.

For the missing data in each line, a linear interpolation, Equation (6), is used, it searches for the rows where there are only zeroes after the first pixel in the image is

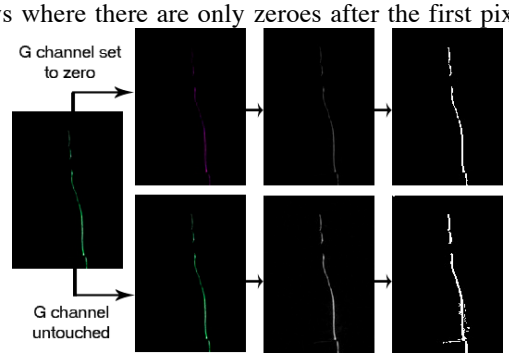


Fig. 4. Results from making color channel representing laser color equal to zero. As it is shown the noise is reduced when two channels are used instead of three.

found until the last one, once identified it takes the rows after and before each column with missing data, and makes use of those two points by using its x and y values and takes the slope from the first known value pixel in the interval as the missing value.

$$md(x,y) = (y - y_e + 1) * \frac{md(x_l, y_l) - md(x_e, y_e)}{y_l - y_e} + x_e \quad (6)$$

Where:

- x_l & y_l *: are the later known x and y position.*
- x_e & y_e*: are the earlier known x and y position.*
- x & y*: are the current x, y position of the missing pixel.*
- md*: is the distance matrix.*

The result for each described step until now is shown in **Fig. *5***. As it appears in (b), the binary image is almost identical to the original image (a), and the result for the line reduction (c) gives a slight deformation. However, since the filling gap is made to totally seal the model, any hole in the object will be given a false surface (d), requiring a better method for more complex objects where holes are part of said object.

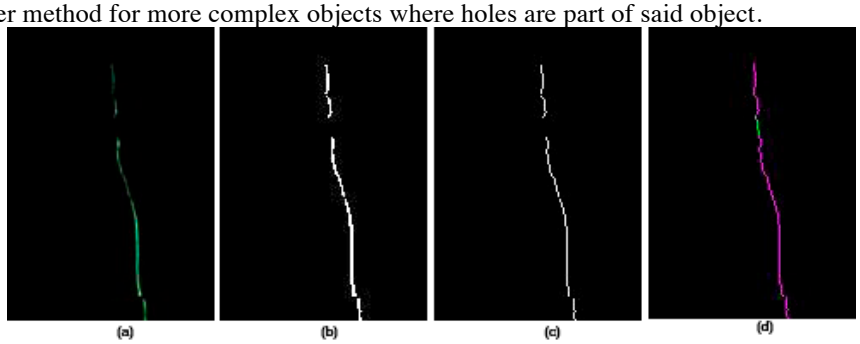


Fig. 5. Results from each step in the proposed approach. (a) original image. (b) binary image, (c) single line reduction. (d) interpolation.

Finally, with the distance matrix completed, each value in the matrix is converted to rectangular form (7).

$$x = D * cos\theta \quad (7)$$

$$y = D * sin\theta$$

Where D is the Matrix value and θ the displacement angle.

Applying the triangulation principle for surface reconstruction, the mesh is then generated by using the distance matrix in its rectangular form.

The used computer is described in Table 1 listing some technical aspects like CPU, GPU and RAM. The camera was a digital camera Lumix brand. And the whole algorithm is shown in Algorithm 1 as pseudo code for surface reconstruction which can be implemented on a variety of systems, being this algorithm one contribution of this paper.

Table 1. Hardware specs.

Hardware	
Processor	Intel® Core™ i7-4700MQ CPU 2.4 GHz.
Memory (RAM)	16.0 GB.
GPU	NVIDIA GeForce GT 755 M
Camera	Digital Camera Lumix

Algorithm 1: Algorithm for surface reconstruction

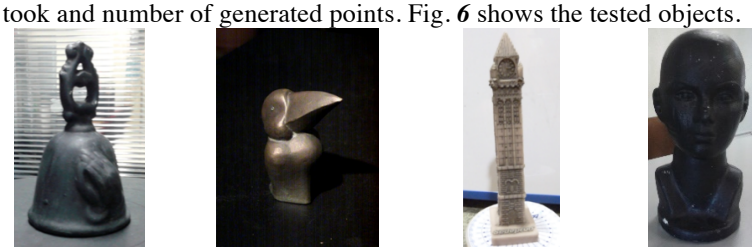
Load n images (p); interest region [x1:x2] & [y1:y2]; dm[n, y2-y1]=NaN; c=distance between laser-line and object's center; θ[n]; U=threshold

1: **for** i=0 until n
2: ***Cut interest region*** x2:x1 & y2:y1
3: ***Matrix channel equal Laser-Color*** (LC), i[:,:,LC]=0;
4: Convert to grayscale
5: Binarize according threshold U: p(x,y) $\begin{cases} 0, & p(x,y) \leq U \\ 1, & p(x,y) > U \end{cases}$
6: **for** k=0 until y2-y1 **do**
7: pd=$\frac{\sum p(xp,k)*xp}{\sum p(xp,k)} + c$;
8: **if** pd>0 **then**
9: dm[i,k]=pd;
10: **end if**
11: **end for**
12: **for** k=first known position(x_l, y_l) of dm[i,k]!=NaN until last known **do** (x_e, y_e)
13: **if** dm[i,k]==NaN **then**
14: dm[i,y]= $(y - y_e + 1) * \frac{md(x_l,y_l) - md(x_e,y_e)}{y_l - y_e} + x_e$;
15: **else**
16: find next missing interval
17: **end if**
18: **end for**
19: **end for**
20: deg=360/n;
21: **for** k=0 until n **do**

```
22:       θ[k]= deg*π / 180;
23:  end for
24:  for i=0 until n do
25:       for j=0 until y2-y1 do
26:            X[i,j]=dm[i,j]*cos(θ[i]);
27:            Y[i,j]=dm[i,j]*sin(θ[i]);
28:            Z[i,j]=j
29:       end for
30:  end for
31:  Build mesh with base matrices X, Y & Z.
```

4 Experimental Results

A similar method [6] was taken as comparison which uses also RGB as basis, although the method couldn't be completely implemented since there was no underwater nor more complex system with four color lasers at disposal, the core for noise-reduction relies on an edge detection method as a common method [10], a subtraction for noise reduction for each color channel, and the interpolation method (6) for missing data. Results are shown in **Fig. *7*** (a), (c), (e), (g), (i) and (k). With the proposed method, the noise reduction essentially uses only two color channels. And its result are shown in **Fig. *7*** (b), (d), (f), (h), (j) and (l), while the **Table *2*** contains the number of used images, time it took and number of generated points. Fig. ***6*** shows the tested objects.



(a) (b) (c) (d)

(e) (f)

Fig. 6. Tested objects: (a) bell, (b) raven, (c) tower, (d) polystyrene head, (e) pig & (f) skull.

Finally it was made another digitizing with the bell and head, Fig. **6** *(a) & (d) respectively. This time with 256 images and subsets for both objects, using both methods where it was measured the time and points between both and the results are shown in*

Table *3* and **Table *4***.

In this new reconstruction both objects were better represented. In the bell the hands appearing in Fig. ***6*** (a) are more defined, while the head had a better definition, notably

in the ears when comparing **Fig. *7***(j) against **Fig. *8***(d). The 3D reconstructions by using 128 and 64 images also show defined objects, while using 32 images and less have an object without a defined curvature from the base and instead were too squared.

Table 2. Results for each reconstruction with proposed method

Object	Images	Threshold	Time	Points
Bell	36	5	5.947295 seconds	5364
Raven	56	15	10.844818 seconds	13982
Tower	72	8	3.890804 seconds	12159
Pig	64	25	7.143672 seconds	9534
Head	64	15	8.713225 seconds	10918
Skull	256	10	15.574079 seconds	36714

Table 3. Results from bell object

Bell				
	Border Detection		Proposed Approach	
Images	Time	Points	Time	Points
256	25.311303 sec.	36218	14.12059 sec.	36288
128	12.910331 sec.	18100	6.284609 sec.	18137
64	6.721035 sec.	9051	3.475062 sec.	9064
32	3.600887 sec.	4527	2.034301 sec.	4529
16	2.323144 sec.	2256	1.284535 sec.	2255
8	1.493166 sec.	1116	0.947402 sec.	1116

Table 4. Results from head object

Head				
	Border Detection		Proposed Approach	
Images	Time	Points	Time	Points
256	26.103478 sec.	42541	12.053969 sec.	43050
128	12.536345 sec.	21246	6.207357 sec.	21523
64	6.957200 sec.	10618	3.535140 sec.	10764
32	3.857809 sec.	5300	2.007538 sec.	5360
16	2.237772 sec.	2642	1.356919 sec.	2680
8	1.494562 sec.	1299	0.937214 sec.	1320

As it is shown in the results, while both methods had almost the same points, the time with the proposed method was almost half the time the border detection method in all cases as it is shown in **Fig. *8*** where the 256 images were used.

5 Conclusion and Future Work

This paper presents a novel approach for noise reduction using the data from just two color channels from the RGB space color model for laser based images, and a whole algorithm for surface reconstruction. This method was tested along a linear interpolation on six different objects, resulting in a sealed surface reconstruction not letting any holes even if the original object had them.

By applying the proposed approach it is possible to reduce the time in the pre-processing phase, therefore, improving the computing time performance in comparison with previous work. It was found with 64 and more images the object's features could be seen or noticed in the reconstruction and the object is almost rounded meaning it works with some curved surfaces, while with less than 64 the result is an almost squared object.

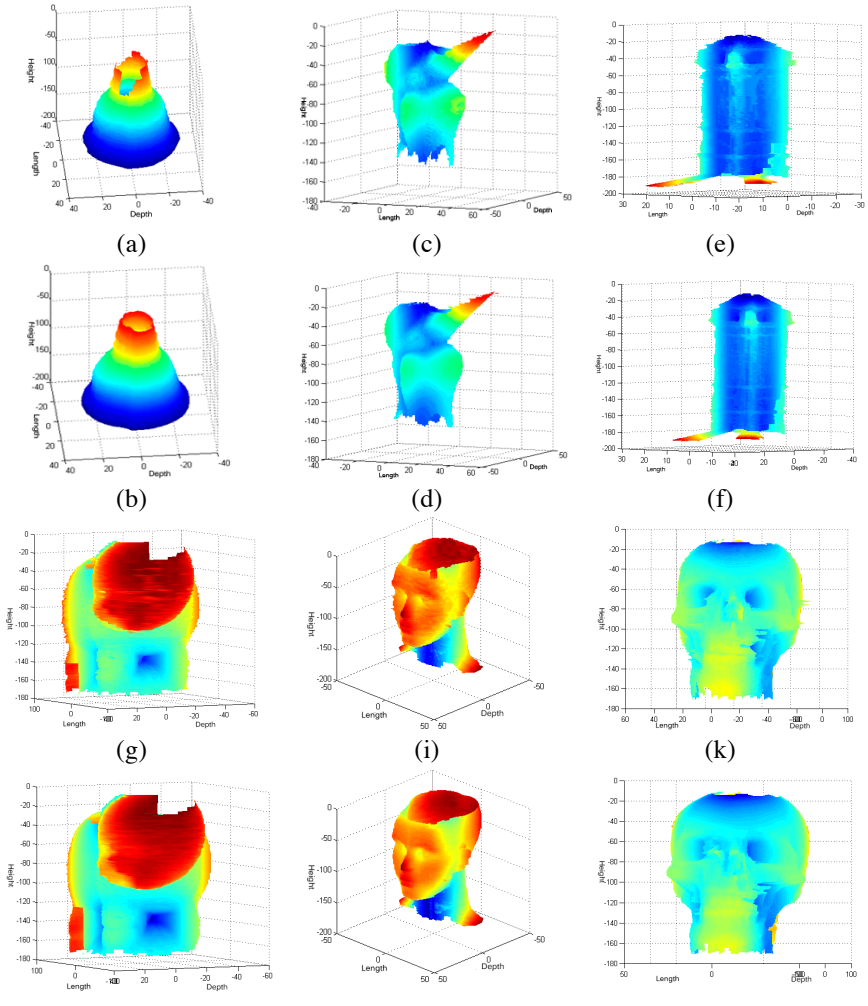
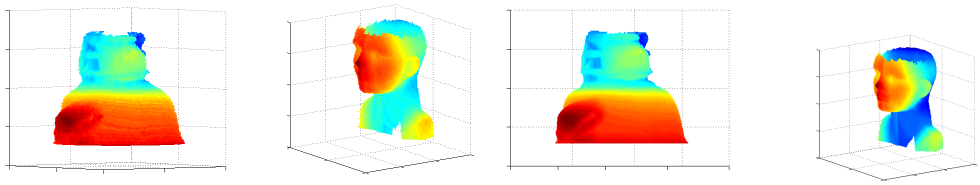


Fig. 7. Reconstruction results with both methods using in (a) and (b) bell, (c) and (d) raven, (e) and (f) tower, (g) and (h) pig, (i) and (j) head, (k) and (l) skull. (a), (c), (e), (g), (i) and (k) were made with the edge detection method, (b), (d), (f), (h), (j) and (l) with the proposed method.

Since the RGB standard model has a limited range and not the most accurate model to represent the color spectrum of the human vision, the method could be tested with more color spaces such as XYZ, HSV, HSI, YCbCr, CIELuv, CIELab, etc., expecting to

obtain results with less noise. By using different space colors it is expected to get a narrow range for the threshold.



(a) **(b)** **(c)** **(d)**

Fig. 8. The surface reconstruction with 256 images. (a) and (b) were achieved with the detection border method. (c) and (d) with the proposed method.

And finally implementing it in a less powerful system, being a reason this method is proposed since it is going to allow a reconstruction system by employing different devices not necessarily a personal computer, but could be employed a mobile devices and microprocessor based embedded development boards.

Acknowledgments. This work is supported by the CONACyT through the project number 308449. Also, we would like to thank the Universidad Autonoma de Queretaro for the FOPER's fund for projects and research, and for the facilities and support.

References

1. Winkelbach, S., Molkenstruck, S., & Wahl., F. M. (2006). Low-Cost Laser Range Scanner and Fast Surface Registration Approach. *28th DAGM Symposium, Berlin, Germany, i*, 718-728.
2. Ding, M., Xiao, Y., Peng, J., Schomburg, D., Krebs, B., & Wahl, F. M. (2003). 3D reconstruction of free-formed line-like objects using NURBS representation. *Pattern recognition, 36*(6), 1255-1268.
3. Chmelar, P., Ladislav, B., & Nataliia, K. (2015). The laser color detection for 3D range scanning using Gaussian mixture model. *Radioelektronika (RADIOELEKTRONIKA), 25th International Conference. IEEE*, (pp. 248-253).
4. Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 23-27.
5. Kapur, J. N., Sahoo, P. K., & Wong, A. K. (1985). A New Method for Gray-Level Picture Thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 273-285.
6. Yang, Y., Zheng, B., Kan, L.-Y., Yu, J., & Wang, J.-C. (2014). 3D color reconstruction based on underwater RGB laser line scanning system. *Optik*, 6074–6077.
7. Jahne, B. (2004). *Practical Handbook on Image Processing for Scientific and Technical Applications* (Second ed.). Boca Raton, FL, USA: CRC Press, Inc.
8. Chityala, R., & Pudipeddi, S. (2014). *Image Processing and Acquisition using Python* (First ed.). Chapman and Hall/CRC .
9. Jack, K. (2005). *Video Demystified A Handbook for the Digital Engineer* (4 ed.). Oxford: Elsevier.

10. Fu, G., Menciassi, A., & Paolo, D. (2012). Development of a low-cost active 3D triangulation laser scanner for indoor navigation of miniature mobile robots. *Robotics and Autonomous Systems*, 1317-1326.