Proseding Int-S

by Sukemi Sukemi

Submission date: 11-Oct-2021 08:56AM (UTC+0700)
Submission ID: 1670517533
File name: paper.pdf (1,013.38K)
Word count: 2805
Character count: 14612

Optimization of Underwater Image Objects with Noise Analysis Using a Gaussian Filter Selected Algorithm

Sukemi[1], Yogi TIARA PRATAMA[2]

[1] Lecture of Computer Engineering Department, Universitas Sriwijaya, Palembang, Indonesia

[2] Teacher of SMK Muhammadiyah 2 school, Palembang, Indonesia

e-mail: sukemi@ilkom.unsri.ac.id[1]

Abstract. Different types of images when testing will give different results. Between still images and moving images requires speed and accuracy in the computation process. This is due to the availability of time in processing which tends to narrow at any time along with environmental changes. Retesting of processors that have been built through previous research requires the selection of new image data and processes. The selection of the new image refers to a different image and has never been used before. Next, for the new process by applying Gaussian filtering selection (filters block). The results of the first stage in testing of some images obtained that the 'bit space adder/sub' accuracy value and using filters block for underwater objects image data test was 90.91% in the first cycle. However, when compared to the architecture of least significant bit only obtained an accuracy of 0.01% so that there is a very significant difference in accuracy, which is equal to 90.90%. This result will improve if added using a filter block, the accuracy value rises to 95.75%.

Keywords: *filters block, variable bitspace adder, under water image*

Introduction

Giving the correct 'answer' at the right time (timeliness) is one of the key factors in the system that is continually being researched to improve the reliability of rapid changes (eg changes in underwater objects that are swift flowing). The timeliness or time delay to obtain an 'answer' from a process affects the results of the calculation process in the timeframe available / provided. The timeliness of 'answers' comes out for all systems, depending on the change in the environment in which the system works, which varies from a very short span of time to a long time. The failure of many systems (or parts of the system) to produce the right 'answer' within each available time frame has the potential to pose risks that threaten the success of other systems, such as the sustainability of the navigation system, transportation, electricity and security.

As an approach that has been taken to meet the 'deadline' in a timely calculation system, it is classified into software groups and hardware groups, for example, Rate-Monotonic Scheduling algorithm [1], on Non-preemptive [2], EDF [3], IRIS [4], Imprecise computation [5], MSB-First Interval-Bounded Variable-Precision Arithmetic Unit in Real-Time [6] and Priority-based computation: Results of studies on shifting paradigm changes in real time calculations [7]. There have been many approaches to show that software and hardware solutions have not been maximally addressed in a timely system. Although several new scheduling techniques have been found. In addition, publications that have not yet been implemented have not yet been collected and there are still several fields of research that do not support each other to realize this.

Purpose Increase the 'clock' on the processor aims to do calculations quickly by minimizing the average value of the time span for 'job/task' in a process [8]. This deviates from the main goal of the real-time system, which is trying to fulfill the timeframe of each task. However, the most important thing of all real-time systems is predictability, not just speed which has a very relative 'meaning'. Predictability is defined as functional certainty and temporal behavior for each task, which must be as detailed as possible so that it is sufficient to meet the system specifications. Every fast calculation will be very helpful in fulfilling a short span of time, but even a quick calculation prediction cannot guarantee a real-time system. The real-time system can only work correctly if no time failure occurs. However, time failures can occur in various time frames, when the system is overloaded (due to changes in environmental conditions) so the system must rearrange the order of tasks that must be performed to respond to these changes.

The level of accuracy is determined by the limited time available to reach the system in real-time or on time [9]. Therefore, the range of the calculation results must be able to provide 'answers' in the decision- making process. However, the fact is that the determination of the accuracy of arithmetic calculations generally cannot be done before the calculation is completed. so the calculation process can be vulnerable to failure if there is not enough time available.

In addition, the selection of performance factors such as time, accuracy and precision is needed to set the system to meet the goals of actual timeliness. Until now, there is no mechanism that can play the role of the voter represented by the hardware or software assigned to choose it. Each selection will consider several parameters, such as time, accuracy and precision so that the real-time system reaches its target [10].

The main assertion, the selection of performance factors such as time, accuracy and precision is very necessary in the system settings to meet the goals of actual timeliness. The initial idea was that a mechanism that could play the role of the voter was represented by the hardware assigned to select it via a 'bit space' selector. However, in this paper, we suggest a selection of filters block [11][12][13][14], following on from the 'bit space' selector for processing images of objects/victims under moving water. Each selection has taken several parameters into account, such as time, accuracy and precision so that the real-time system reaches its target [10].

Many factors determine the pattern of objects under the sea. rivers, lakes, ponds, such as streams of water, mud and sand. All of these factors have not been considered in this study so that the pattern of images of objects under the lake/pond taken at this time only refers to the condition of the depth and speed of the 'prototype boat'. The object/victim image will be filtered on the filter design that has been proposed in previous studies so that the optimization of the bit space processor variable continues to be tested [15]. The initial hypothesis can logically be estimated that optimization increases because before filtering the optimization rate ranges from 90.91%. This figure is obtained from the bit space processor whose image input is entered without going through the filtering process. The filtering process requires a minimum system to reduce the error rate of underwater object images described in section 2.

Underwater Objects Image Filtering Process

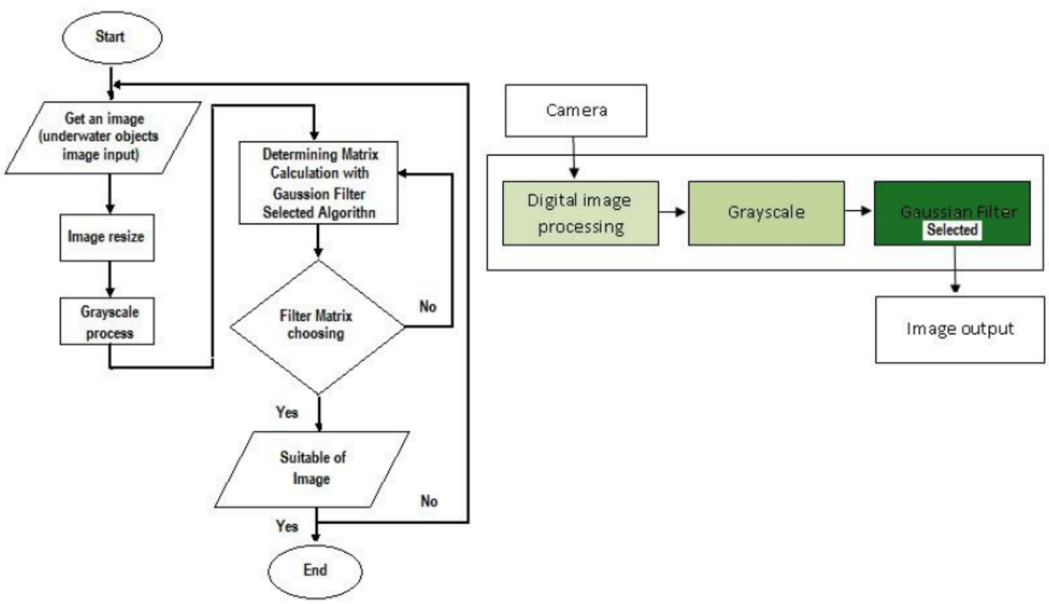


Figure 1. Gaussian filter selected algorithm and it Filters block system

The initial stage, image modification is done through a scaling process. Scaling is done so that the image can be represented in a more complete form so that it requires less memory and computational time.
in addition, the scaled image must be of good quality. The scaling process can be done through the use of an interpolation method that uses the average value of the drawing area to represent that area. The images obtained in digital form are arrays containing real and complex values represented by certain rows of bits. Furthermore, the image has a matrix consisting of rows (N) and columns (M). At coordinates (x, y) the values contained in coordinates are $f(x, y)$, which is the amount of intensity or color found in pixels in the area. If the x, y, and amplitude f values are limited and have a discrete value, then it can be said that the image is a digital image. Digital images are written in matrix form, which is shown in equation 1.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & f(N-1,M-1) \end{bmatrix} \quad \dots(1)$$

Furthermore, a gray image is obtained, with each pixel having a layer whose intensity value is in the 0-255 interval. Through this intensity, the pixel values in the gray image can be represented in the matrix to facilitate the calculation process in subsequent operations. The gray image intensity value is calculated from the intensity of the Red Green Blue (RGB) image using equation 2.

$$Grayscale = ((Red * 0{,}299) + (Green * 0{,}587) + (Blue * 0{,}114)) \quad \dots(2)$$

The next process, the convolution stage, namely the sum of the multiplication operations with the operation notation (*), which multiplies the image with a mask or kernel. The convolution for 2 functions $f(x)$ and g (x) are defined as follows:

$$h(x) = f(x)*g(x) = \sum f(a)g(x-a) \quad \dots(3)$$

$$\sum_{y=-N}^{N} f(u,v)g(x+u, y+u) \quad \dots(4)$$

where:
$f(x,y)$: original image; $h(x,y)$: linier-position invariant operator; $g(x,y)$: result image convolution; x, y, u and v : position of points in the image.

The final process of the contents of the filters block is the selection of Gaussian filters that are used to blur the image or eliminate noise [11]. This filtering process is to perfect the image which looks a bit more blurry for use in the next process. It also aims to produce the right edge of the image. If this process is not used, fine line detection will also be detected as edges. The Gaussian filter used is a 2-dimensional filter which is described through the following equation:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \dots(5)$$

where $G(x,y)$ Gauss matrix elements in position (x,y), π = 22/7, e = 2.71828182846, σ is the standard deviation value =1 [13].

Underwater Image Processing by Gaussian Filter Selector

The bit space variable embedded in the processor design is compiled based on the data bit width synchronized with the data input width in figure 2 [12]. The initial step before being processed into a bit space adder, the underwater object image will be processed first by selecting the optimal filter option [16] to get the lowest noise level image. This paper proposes a model for selecting underwater object images and is a step further from previous research [10]. The filter block is built with several 3x3, 5x5 to 9x9 matrix filter options arranged in parallel so that you can process images at the same time. For the initial stage of testing these filters block, each filter is connected to different input and output lines to produce images of underwater objects with lower noise levels than before. The resulting image will be processed further into variable bit space components that are designed to achieve a high degree of accuracy despite the fact that the computational process is not always finished due to environmental factors.

The Accuracy obtained from the new variable bit space adder device uses equation 6;

$$A_j(t)_{ars-range} = \left[1 - \frac{\sum_{i=1}^{n} f(x)_{upper} - \sum_{i=1}^{n} f(x)_{lower}}{\sum_{i=1}^{n} f(x)_{upper}} \right] * 100\% \qquad \ldots(6)$$

where; $A_j(t)_{ars-range}$: accuracy j at a certain time (ns); $f(x)_{upper}$: upper limit computing value; $f(x)_{lower}$: lower limit computing value [15].

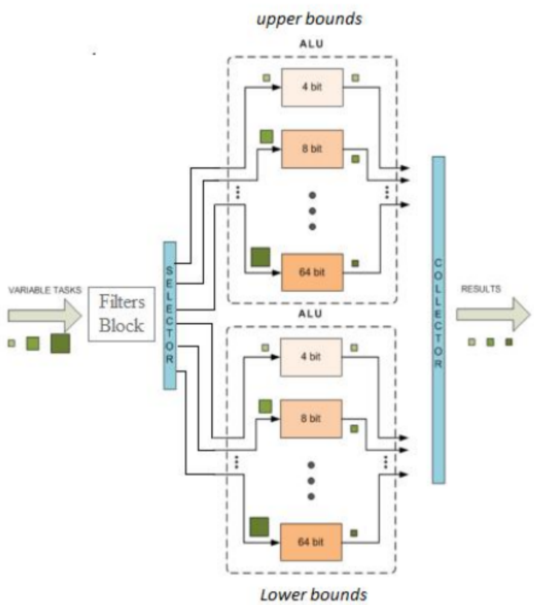
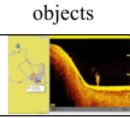
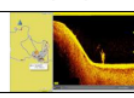
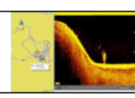
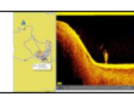
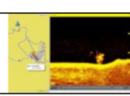
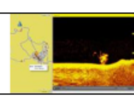
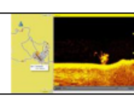
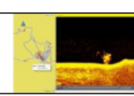
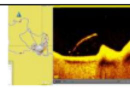
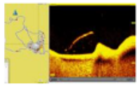
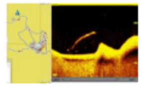
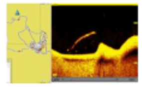
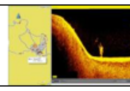
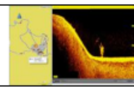
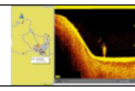
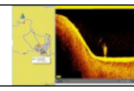
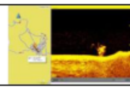
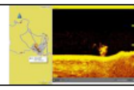
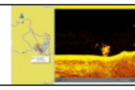
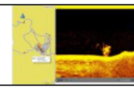
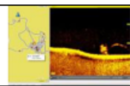
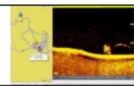
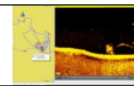
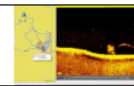
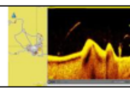
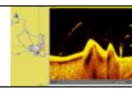
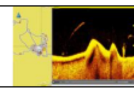
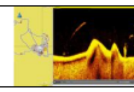
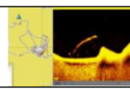
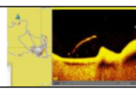
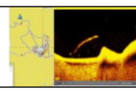
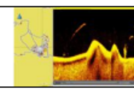


Figure 2. New variable bitspace adder with filters selection

RESULTS AND DISCUSSION

Tabel 1.The results of processing underwater objects data using a 5x5, 7x7 and 9x9 filters block

Time (WIB)	Underwater objects	Underwater objects image on matrix		
		5 x 5	7 x 7	9 x 9
00.00				
Processing time		00:00:09.3277430	00:00:10.5571198	00:00:12.1041920
03.00				
Processing time		00:00:08.2378418	00:00:10.1561178	00:00:13.2051700

06.00				
Processing time		00:00:09.8888625	00:00:12.9453205	00:00:14.2538669
09.00				
Processing time		00:00:08.2254926	00:00:10.2573505	00:00:12.6058814
12.00				
Processing time		00:00:08.2528098	00:00:10.1889900	00:00:12.4187543
15.00				
Processing time		00:00:08.7291704	00:00:11.9187543	00:00:16.0058695
18.00				
Processing time		00:00:08.9927239	00:00:09.7708184	00:00:12.1113061
21.00				
Processing time		00:00:10.6160280	00:00:12.0431847	00:00:12.1113061

Tabel 2.The results of time consumption data using a 5x5, 7x7 and 9x9 filters block

Processing time	matrix 5x5 (millisecond)	matrix 7x7 (millisecond)	matrix 9x9 (millisecond)
0:00	9.3277430	10.5571198	12.1041920
3:00	8.2378418	10.1561178	13.2051700
6:00	9.8888625	12.9453205	14.2538669
09:00	8.225454926	10.2573505	12.6058814
12:00	8.2524926	10.1889900	12.4187543
15:00	8.7291704	11.9187543	16.0058695
18:00	8.9927239	9.7708184	12.1113061
21:00	10.6160280	12.0431847	12.1113061

Referring to the filters block test results table that the 5x5 matrix filtering process speed shows the processing time is faster than the 7x7 and 9x9 matrices, which is an average of 2.034126823 millisecond, this is because the convolution process is smaller, but for accuracy with low noise, the 7x7 matrix and 9x9 higher.

Underwater object image results from the filters block will be processed last in the 'bit space adder', to determine the accuracy of time-limited real-time calculations. Accuracy is generated through a regression process for the average accuracy obtained from the computational time function.

average accuracy	98.75247871	99.97253	99.99944	99.99993	100	100	100	100

t (ns)	accuracy
0.476190476	98.75247871
0.952380952	99.97253113
1.428571429	99.99944406
1.904761905	99.99993466
2.380952381	100
2.857142857	100
3.333333333	100
3.80952381	100

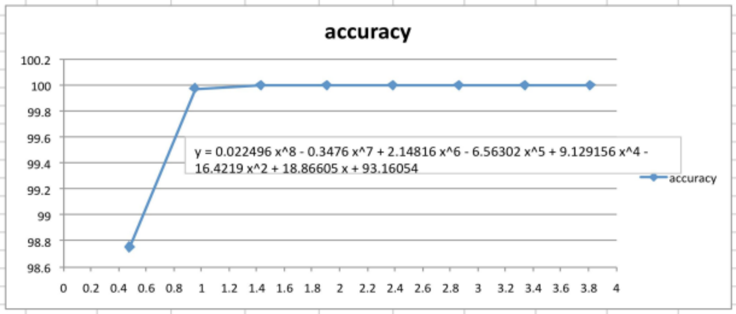


Figure 3. Accuracy over a period of time

Referring to Table 3, Using a 5x5 filter has resulted in high accuracy in the first step of the bit space adder with a consumption of processing time of only 9.034 ns. this shows that the limited processing time for images or objects that are always moving and under the surface of the water will be recognized with a minimum time as in the first step of the bit space adder above. the accuracy will be maximum in step 5 with the required computing time of 2,381 ns.

SUMMARY

Placement of the filters block the adder variable bit space adder architecture creates optimal time and guarantees the accuracy of predictability about 85%.

FUTURE WORK

The time for taking underwater object images should be taken from dynamic underwater flow.

ACKNOWLEDGMENT

This research was funded by a Research Grant organized by UNSRI in 2019 called HIBAH KOMPETITIF.

REFERENCES

[1] Lehoczky, J.; Sha, L.; Ding, Y. "The rate monotonic scheduling algorithm: exact characterization and average case behavior", *IEEE Real-Time Systems Symposium,* pp. 166–171, 1989.

[2] Jeffay, K., Donald FS., Charles UM., On Non-Preemptive Scheduling of Periodic andSporadic Tasks Proceedings of the Twelfth IEEE Real-Time Systems Symposium, SanAntonio, Texas, *IEEE Computer Society Press*, pp. 129-139, 1991.

[3] Deadline scheduling for real-time systems, EDFf and related algorithms., *The Springer International Series in Engineering and Computer Science.,* Vol. 460, ed. 1, 1998.

[4] Dey JK, Kurose J, Towsley D, On-Line processor scheduling for a class of IRIS, *Technical Report*, Univ. of Massachusetts Amherst, MA, USA, 1993.

[5] Liu, JW, Imprecise computation, *Proceedings of the IEEE* 82(1):83 – 94, 1994.

[6] Kerlooza, YK., Yudi, SG., Agus M, MSB-First Interval-Bounded Variable-Precision RealTime Arithmetic Unit, *ITB Journal Publisher*, Vol. 4 No. 1, 2010.

[7] Sukemi, Riyanto, Priority based computation: An initial study result of paradighm shift on real time computation, *ComEnggApp Journal*, Vol. 6 No.1. 2017.

[8] Stankovic, J.A., Spuri, M. and Ramamritham, K. "Deadline Scheduling for Real-Time Systems : EDF and Related Algorithms," *Kluwer Academic Publishers (Kluwer International Series in Engineering and Computer Science),* 1998.

[9] Mora-Mora, H., Mora-Pascual, J. and Garcia-Chamizo, J.M. "Real-Time Arithmetic Unit," *Real-Time Systems Springer Science + Business Media, LLC*, DOI 10.1007/s11241-006-8753-z, 2006, pp. 53-79.

[10] Sukemi, Ratna AAP., Sudibyo H. "Priority based computation: A Study of paradigm shift on real time computation," *Compuatational Intelegence and Cybernetics (Cyberneticscom), 2012 IEEE International Conference on Digital Object Identifier*, *Bali, Indonesia*. Pp. 129-132, Juli 2012.

[11] Sukemi, Yogi Tiara Pratama. "Optimalisasi image analisis noise citra menggunakan algogritma Gaussian Filter" *Annual research Seminar,* Vol.4, No.1, pp. 278-282, 2018.

[12] Sukemi, Ratna A.A.P, and Sudibyo H, "Variable Bitspace of Variable Precision Processor," *Journal of Advanced in Information Technology*, *Academy Publiser*,Vol. 5 No. 2 2014, pp. 65-70.

[13] H. S. Javadi and H. R. Mahdiani, "Efficient utilization of imprecise blocks for hardware implementation of a Gaussian filter," *Proc. IEEE Comput. Soc. Annu. Symp. VLSI, ISVLSI*, vol. 07–10-July-2015, pp. 33–37, 2015.

[14] T. Ohtani, Y. Kanai, and N. V. Kantartzis, "A 4-D subgrid scheme for the NS-FDTD technique using the CNS-FDTD algorithm with the shepard method and a gaussian smoothing filter," *IEEE Trans. Magn.*, vol. 51, no. 3, pp. 3–6, 2015.

[15] Sukemi, "Development of the arithmetic unit bitspace adder/sub real time architecture for optimal performance", *Desertation of Doctoral Program*, Universitas Indonesia, 2016.

[16] Sukemi, Rianto, BS. Bambang, Yogi Saputra, "Optimizing Accuracy of the Variety Bitspaces for Weather Forecasting for Early Flight Safety", *IOP Publisher: Conference series*, 2019.

Proseding Int-S

ORIGINALITY REPORT

6%	%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Data Analytics for Monitoring the Satisfactory Parameters of Airline Passengers using Machine Learning Algorithms in Python", International Journal of Innovative Technology and Exploring Engineering, 2020 Publication	1%
2	Suharjito Suharjito, Alpha Epsilon, Abba Suganda Girsang. "Human Ear Recognition Methods Based on Image Rotation", International Review of Automatic Control (IREACO), 2017 Publication	1%
3	I. Hrynaszkiewicz. "Preparing raw clinical data for publication: guidance for journal editors, authors, and peer reviewers", BMJ, 01/28/2010 Publication	1%
4	Kenny Hong, Stephan K. Chalup, Robert A.R. King. "A component based approach for classifying the seven universal facial expressions of emotion", 2013 IEEE	<1%

Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), 2013
Publication

---

5 | Zeliang Zhang, Kang Xiaohan, Mohd Nor Akmal Khalid, Hiroyuki Iida. "Bridging Ride and Play Comfort", Information, 2021
Publication | <1 %

---

6 | "Ubiquitous Computing and Ambient Intelligence", Springer Science and Business Media LLC, 2017
Publication | <1 %

---

7 | Huda Ubaya, Sukemi, Ayu Purnama Sari. "New Model of Vegetation Monitoring Using Flying NIR Cameras with NDVI Parameters and C-means", Journal of Physics: Conference Series, 2019
Publication | <1 %

---

8 | Sukemi, Sukemi, Anak Agung Putri Ratna, and Harry Sudibyo. "Variable Bitspace of Variable Precision Processor", Journal of Advances in Information Technology, 2014.
Publication | <1 %

---

9 | Lun, D.P.K.. "Orthogonal discrete periodic Radon transform. Part II: applications", Signal Processing, 200305
Publication | <1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		