Petroleum and Coal

Article **Open Access**

Features extraction from carbonate modern platforms satellite imagery using Computer Vision

Grisel Jiménez Soto[1], Mirza Arshad Berg[1], Michael C. Poppelreiter[2], Khaidir Rahmatsyah[3]*

[1] *South East Asia Carbonate Research Laboratory, Department of Geosciences, Universiti Teknologi PETRONAS*
[2] *Shell Kuwait Exploration and Production BV, Kuwait*
[3] *Baker Hughes Netherlands*

Received March 30, 2020; Accepted June 26, 2020

Abstract

Firstly, a Python code load massive satellite imagery from a specific folder RGB format, then each raw coral reef image is resized, converted from RGB band to gray, smoothed, and binarized using Open Computer Vision Library available in Python 3.0. The coral reef edge information contains very prominent geometric attributes that characterize their behavior, thus morphological changes were applied to define the contour of the carbonate platform. Furthermore, a structural analysis and shape descriptors were made in a set of images in order to numerically calculate the characteristics of the carbonate platform. A total of 27 satellite images were processed by the algorithm successfully at the same time, only two images were not segmented correctly because of the illumination and intensity of the predominant colors, especially blue color. Finally, this dataset was exported from Microsoft Excel spreadsheets and CSV format respectively.

Keywords: *Opencv; Automatic coral features extraction; Modern carbonate build ups.*

1. Introduction

Feature extraction techniques are extensively being used in satellite imagery and getting impressive attention for remote sensing applications. The state-of-the-art feature extraction methods are appropriate according to the categories and structures of the objects to be detected [1].

Non-hyperspectral image data has been used in many applications. The ease of obtaining free images from any region of the world has directed many investigations towards the interpretation using RGB format. In this study, an algorithm is developed to massively process dataset of satellite images and automatically extract several geometric properties. Development of GIS geographic information systems and large amount of high resolution images that can be downloaded from Google Earth make possible implement computer vision techniques for their interpretation.

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it is open source and released under the BSD 3-Clause License. It is free for commercial use and have a powerfull toolkits to solve many problems. The major challenges in automatic extraction in massive satellite imagery include presence of carbonate reefs in different sizes, different orientations and carbonate reefs without not continue shape. However, this problem was solved by applying an improvement and smoothing treatment of the raw images.

In this work, we introduce an automatic solution for carbonate reef shape analysis for applications that require monitoring changes, including structural analysis and shape descriptor analysis. A new Python module called **"geometryIsland**" was created for entire segmentation and estimated more than ten geometric parameters. Input data are the RGB image and area and length scale factor and output are segmented images linked to each set parameters as shown in conceptual level block diagram (Figure 1).

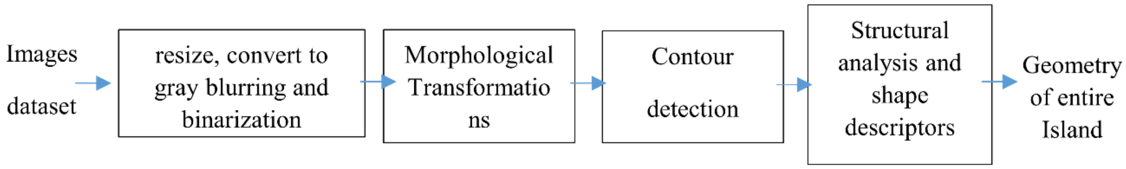

Figure 1. Conceptual level block diagram of "geometryIsland" module

The method implemented of this system is explained in detail in Section 3, the principal results in Section 4, followed by conclusions in Section 5.

2. Study areas

The Maldives are among the largest coral reef and carbonate platform areas in the world. The Maldives archipelago is located in the northwestern Indian Ocean and extends for about 1,000 km from north to south and is some 150 km wide. Coral reefs cover a total area of 21,370 km^2. The archipelago is composed of 21 atolls, 14 larger and 7 smaller ones [2]. For this study, four smaller atolls were selected: Makunudu, Goidhoo, Thoddoo and Gaafaru (Figure 2).

Kepulauan Seribu or Thousand Islands is situated on the outer shelf margin of Sumatra in the Java Sea and consists of numerous coral reef islands ranging in dimension from a few meters across to length greater than 7 km. The archipelago commencing about 25 km north-west of the Java coastline, extends for about 40 km into the Java Sea [3]. For this study, six modern platforms were selected: Pari, Tidung Barat, Kotok, Semak Daun, images 11 and 14.

Tun Sakaran Marine Park (TSMP), also known as Semporna Islands Park, lies off the east coast of Sabah, Malaysia, at the entrance to Darvel Bay (latitude 4°33′N to 4°42′N and tude 118°37′E to 118°51′E) [4] (Figure 1). It is the largest marine Park in Malaysia and one of. It is the largest marine Park in Malaysia and one of the prime coral reef sites of the country. It includes eight coral reef islands and associated reefs (Bodgaya, Boheydulang, Sebangkat, and Selakan, the sand cays of Maiga, Sibuan, and Mantabuan, the patch reefs of Church and Kapikan) and covers an area of approximately 340 km^2 of sea and coral reefs and 10 km^2 of land [5]. For this study, seven platforms were. For this study, seven platforms were selected: Gaya, Church reef, Mantabuan, Selakan, Sibuan, Kapikan and Maiga.

The South China Sea is situated between roughly the equator and 22°N, and includes the Spratly Chain (in the south, also called the Nansha Chain or Trường Sa) and the Paracel Chain (to the north, also termed the Xisha Chain or Hoang Sa) Isolated carbonate platforms of the Spratly chains examined here range in size from 5 to 195 km^2, and include circular, ovoid and elongate shapes; many include long axes north-east to south-west (Spratlys) [6]. For this study, ten carbonate platforms were selected: Layan Layang, East reef, Cuarteron reef, Pearson, Alison reef, Cornwallis, Commodore reef and images 15 and 17.

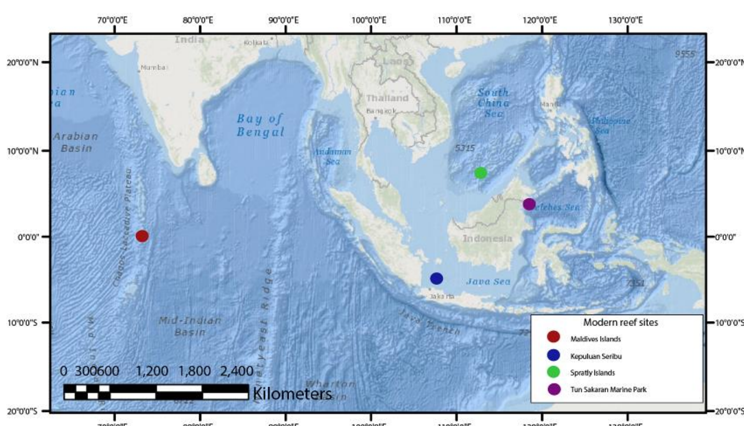

Figure 2. Location map of modern reef sites used for this study

3. Methods

A Python script for automatic feature extraction using Open Computer Vision Library was developed. A total of 29 islands were analysed from Kepulauan Seribu, Semporna, Spratly Islands and Maldives areas. For each georeferenced image the scale factors are known (*fa, fd*) using the following equations:

$$f_a = \frac{area\ real\ image\ (km^2)}{area\ digital\ image\ (pixel^2)}$$

$$f_d = \frac{lenght\ real\ image\ (km)}{lenght\ digial\ image\ (pixel)}$$

In the Figure 3 is a scheme that explains how the algorithm works.

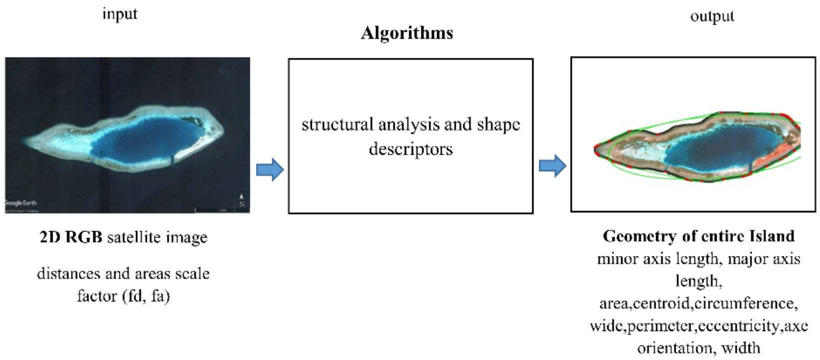

Figure 3. Algorithm workflow with the input and output dataset

3.1. Massive satellite imagery loading from a specific folder using Python code

In this phase of the work, a routine was programmed in Python that performs a selective search in the working directory. This code identifies the extension of the files name that are inside the folder (*.jpg) after that it calculate metadata (width, height, bands/channels) of images then send results from this folder to preprocessing module.

3.2. Preprocessing RGB images

Preprocessing routines are as follows: resize, convert to gray, blur Gaussian and binary (threshold). Resizing island image means changing the dimensions of it, be it width alone, height alone or both. Also, the aspect ratio of the original image could be preserved in the resized image. To resize an image, OpenCV provides cv2.resize () function. To simplify the images processing, it was implemented a method to convert RGB (9 values by pixel) image to gray (3 values by pixel) There are more than 150 color-space conversion methods available in OpenCV, but we applied into only most widely used, BGR ↔ Gray. The, it proceeded to smoothen image using custom-made filters to images (2D convolution) to reduce blurring. Image blurring is achieved by convolving the image with a low-pass filter kernel which was useful for removing noise. This step removed high frequency content (e.g: noise, edges) from the image resulting in edges being blurred when this filter is applied. OpenCV provides mainly four types of blurring techniques but for this case Gaussian blur was used.

Finally, for better accuracy, it is recommended use binary images before finding contours using threshold or canny edge detection, in that sense, island image was binarized (white and black pixel only), for every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to 0 (black), otherwise it is set to a maximum value.

3.3. Morphological transformations

The previous step to draw the polygon that defines the outline of the island image, is to apply different morphological operations such as erosion and dilation. In this case, the technique of dilation worked as follows: A pixel element is '1' if at least one pixel under the kernel is '1'. So it was increased the white region in the image or size of foreground. Normally, in cases like noise removal, erosion is followed by dilation. Erosion was removed white noises, but it also shrank the object. So, we dilated it, since noise is gone, it may not return, but the object area increased. This step was useful in joining broken parts of an object. Several kernel were tested to select 10x10 size.

3.4. Drawing the coral reef island contour

The OpenCV library has a powerful module available to find the contours of an image. This algorithm travels the lines whose pixels are similar and detects the difference between these pixels and neighboring pixels. Contours are defined as the line joining all the points along the boundary of an image that are having the same intensity. To draw the contours, cv.draw. *Contours function* is used. For this function, first argument is the binary image, second argument is the contours which should be passed as a Python list, third argument is index of contours. To identify the contour of the coral reef island, the area of all the contours detected by the algorithm is calculated, then a definitive polygon was draw and the rest of the external region was painted with white color.

3.5. Structural analysis and shape descriptors

Table 1. Features, functions and parameters

Feature	Function	Parameters
minor axis Length	(x,y),(MA,ma),angle = cv2.fitEllipse(cnt)	cnt:contour
major axis Length	(x,y),(MA,ma),angle = cv2.fitEllipse(cnt)	cnt:contour
area	area = cv2.contourArea(cnt)	cnt:contour
perimeter	perimeter=cv2.arcLength(cnt, closed=True)	cnt:contour
centroid (cX,cY)	M = cv2.moments(cnt) cX = int(M["m10"] / M["m00"]) cY = int(M["m01"] / M["m00"])	cnt:contour
eccentricity(eX.eY)	eX=a b s (x-c X) eY=a b s (y-c Y)	
circumference	c=2*math . pi*math . sqrt ((ma**2+MA**2)/2)	ma: minor axis MA:major axis
axe orientation	(x,y),(MA,ma),angle = cv2.fitEllipse(cnt)	cnt:contour
wide	hull=cv2.convexHull(cm, returnPoints = False) leftmost = tuple(cnt [cnt [:,:,0].argmin()][0]) rightmost = tuple(cnt [cnt [:,:,0].argmax()][0]) topmost = tuple(cnt [cnt[:,:,1].argmin()][0]) bottommost = tuple(cnt [cnt[:,:,1].argmax()][0]) wide= rightmost- leftmost	cnt:contour

At this stage, routines were programmed to quantify the geometric properties of each coral reef island. This code was used to extract the features of entire coral reef island like major and minor axis, area, perimeter, axe orientation, centroid, etc. Table 1 contains all features extracted.

Axis orientation is calculated from horizontal axis always counter-clock wise as it shown in the Figure 4. Additionally, *cv2.convexHull* function is used to obtain the coordinates of the extreme points of the contour as show in the Figure 4. Extreme Points means topmost, bottommost, rightmost and leftmost points of the object as shown. The Convex Hull of a shape or a group of points is a tight fitting convex boundary around the points or the shape, in this case the wide was calculate as distance between leftmost and rightmost (Figure 4).

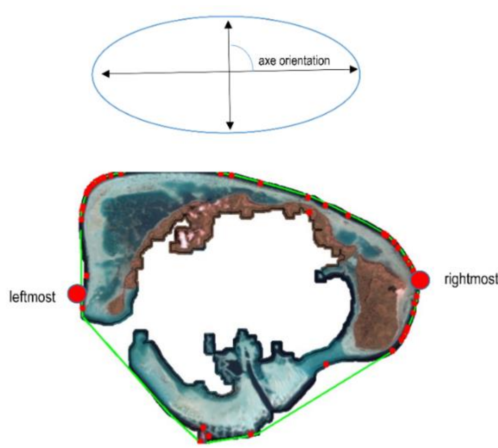

Figure 4. Convex Hull of a shape or a group of points is a tight fitting convex boundary around the points or the shape.

3.6. Create database

Once the numerical values of all the parameters were obtained, a database was developed. This database was designed exporting data from "pandas library (dataFrame format)" in Python to Microsoft Excel spreadsheet.

Excel database files make it easy to enter, store, and find specific information. The basic format for storing data in an Excel database is a table. Once a table has been created, we used Excel's data tools to search, sort, and filter records in the database to find specific information. In addition, the option to export the data using csv format is programmed and can be executed by the user at any time.

4. Results

The data was extracted from Google Earth, all images are made of 3-Bands - red, green and blue (RGB). The original task was an individual binary-prediction problem, for example detect elements like: sea and Not-sea, shallow water and Not-shallow water at the same time how to estimate binarization thresholds to segment the land regions and sea portions, knowing that each image has different reflectance levels. The following flowchart was used for this study (Figure 5).

To segment the coral reef islands, the color information was not useful to identify important edges or other features. Performance (compute time) of the algorithm was also evaluates, specifically when it analyzed the image pixel-by-pixel, for this reason just one channel of color was used as shown in the Figure 6.

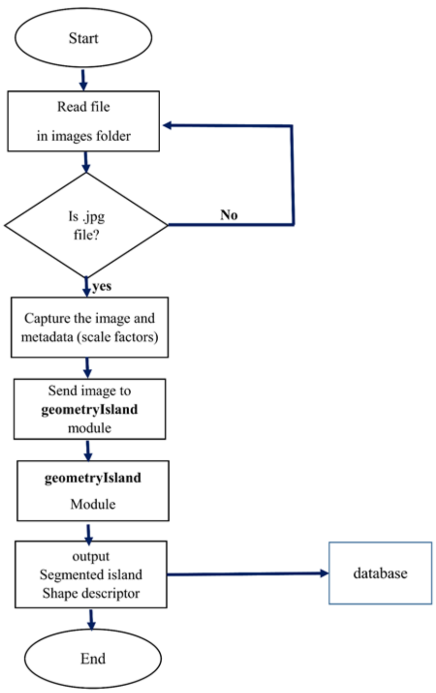

Figure 5. Main Flow chart for binarization thresholds to segment the land regions and sea portion

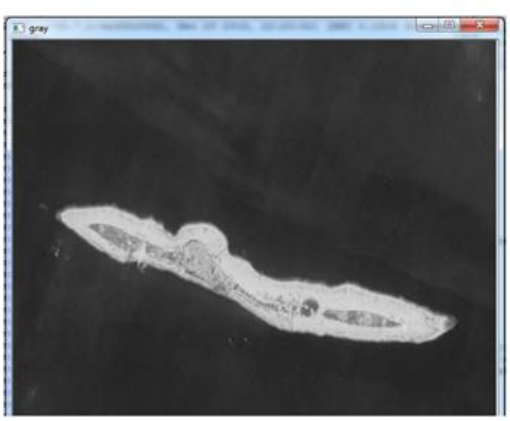

Figure 6. Gray scale image

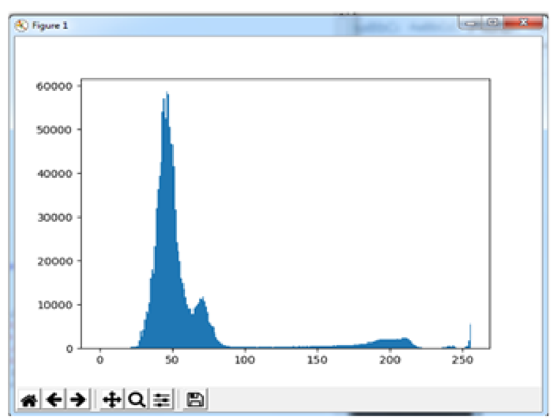

Figure 7. Gray scale histogram

It is observed in the histogram (Figure 7) that most of the pixels have an intensity between 25 to 90 corresponding to sea color. This histogram helped to decide what value of threshold to use when converting a grayscale image to a binary one by thresholding, in this case values between 100 to 120 were tested.

As in any other signals, images also can contain different types of noise, especially because of the source (camera sensor). Image Smoothing techniques as Gaussian filters helped in reducing the noise and minimizing too much blurring. (Figure 8 and 9).

Figure 8. Gaussian filter

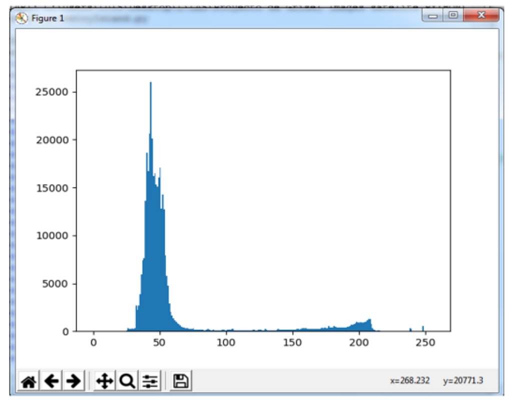

Figure 9. Gaussian filter histogram

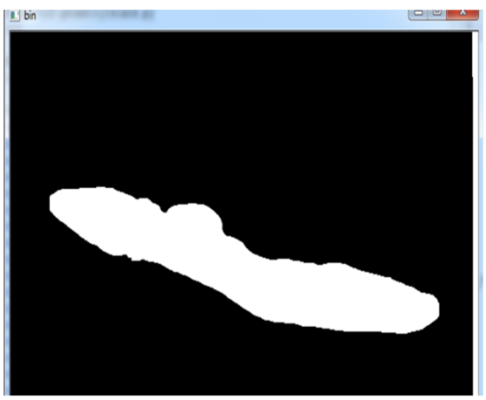

Figure 10. Satellite Image Segmentation using binarization

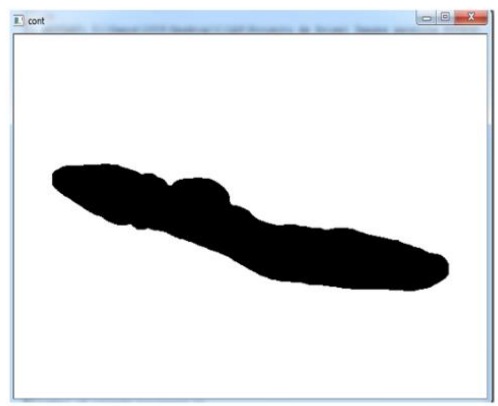

Figure 11. Drawing contour (inverted mask)

Contours are simply the boundaries of the objects. Using 105 value as threshold (Figure 9) these pixels can be classified into groups setting an upper and lower bound to each group. The resulting image in this case is a binary image (Figure 10 and 11).

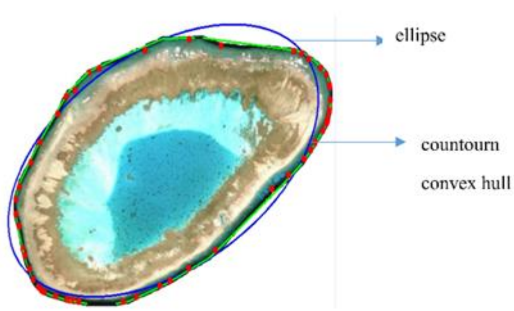

Figure 12. Drawing contour (inverted mask)

Then, a quantitative analysis of coral reef island shapes by drawing a circumscribed ellipse in the segmented polygon was performed. The final step is to visualize the convex hulls (Figure 12 and 13). The database has 12 fields as shown in the following Table 2.

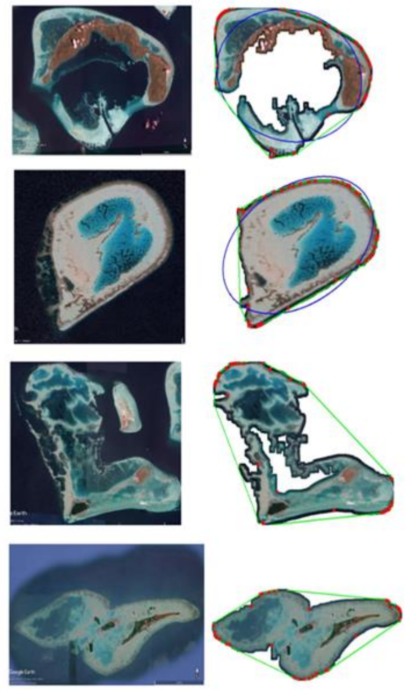

Fig. 13. Example results from 27 modern reef satellite images

Table 2. Features results from 27 modern reef satellite images

ID	Major axis Length (km)	minor axis Length (km)	Area (ha)	Perimeter (km)	Centroid_ X	Centroid_ Y	eX (km)	eY (km)	Circunference	Axe Orien-tation	Wide (km)
imagen1 jpg	7.18	4.76	2489.35	48.12	4.39	2.34	0.11	0.10	38.26	172.61	7.70
imagen2.jpg	7.07	3.86	2909.84	20.40	4.69	2.43	0.16	0.03	35.76	17.95	6.34
imagen3 .jpg	5.45	3.34	2054.48	16.28	4.13	2.85	0.03	0.02	28.40	139.94	4.81
imagen4 .jpg	7.72	4.15	2729.84	50.61	4.28	2.90	0.07	0.01	38.94	146.85	8.02
imagen5. jpg	7.03	3.63	2749.86	19.14	5.24	2.30	0.15	0.01	35.16	19.12	6.36
imagen6.jpg	4.50	2.96	1520.10	14.36	4.03	1.94	0.00	0.00	23.91	141.66	4.02
imagen7.jpg	6.55	3.72	2603.28	18.76	4.02	2.96	0.08	0.19	33.48	133.07	5.62
imagen8.jpg	7.40	3.05	2240.64	22.03	4.30	3.19	0.02	0.07	35.54	3.88	8.23
imagen9.jpg	7.84	1.00	792.83	16.68	4.38	2.95	0.15	0.03	35.13	167.56	7.29
imagen10.jpg	8.49	2.61	2331.09	19.28	3.73	3.19	0.17	0.02	39.49	179.46	7.84
imagen11.jpg	5.92	2.26	1541.28	14.73	4.00	3.09	0.01	0.01	28.16	176.54	5.79
imagen12.jpg	8.58	2.79	2572.14	28.45	4.13	2.37	0.03	0.04	40.08	12.82	8.39
imagen13.jpg	7.14	2.18	1741.54	17.06	4.63	2.85	0.02	0.02	33.16	5.20	7.22
imagen14.jpg	6.21	2.28	1595.71	15.66	3.51	2.13	0.05	0.02	29.40	25.24	5.47
imagen15.jpg	8.31	2.44	1668.96	21.30	4.36	2.47	0.07	0.01	38.47	148.89	7.94
imagen16.jpg	8.58	3.66	3397.61	21.48	5.10	2.52	0.15	0.03	41.43	7.73	8.28
imagen17.jpg	8.45	4.30	4110.28	22.30	4.41	2.76	0.01	0.05	42.13	22.41	8.11
imagen18.jpg	8.08	3.54	1959.83	38.75	4.32	2.26	0.40	0.22	39.21	177.84	8.39
imagen19.jpg	9.06	3.72	3501.29	23.85	4.36	2.99	0.12	0.05	43.53	159.99	8.39
imagen20.jpg	8.15	2.42	1971.81	20.86	4.37	2.82	0.07	0.04	37.77	22.19	7.75
imagen21.jpg	6.89	4.67	877.53	20.80	6.41	1.85	1.04	1.28	37.00	144.44	5.89
imagen22.jpg	7.57	5.00	3676.74	37.94	5.15	2.56	0.26	0.14	40.32	151.01	7.84
Imagen23.jpg	8.50	4.09	3036.20	27.90	4.16	2.90	0.15	0.11	41.93	157.72	8.40
Imagen24.jpg	4.97	2.00	649.74	27.92	2.85	2.83	0.20	0.00	23.80	45.10	4.29
Imagen25 .jpg	6.33	3.11	1284.36	33.63	4.15	3.06	0.17	0.12	31.33	11.98	6.57
Imagen26 .jpg	7.29	4.17	1454.04	39.82	3.09	2.70	1.26	0.33	37.32	8.09	5.69
Imagen27.jpg	8.18	3.90	964.59	35.16	4.60	3.34	0.38	0.33	40.27	3.91	8.37

5. Conclusions

The results show a promising way to extract meaningful features from a satellite images data. The main advantage of our method is that the results are not just a series of connected pixels or sets of line segments. The final results offer spatial information and connectivity for free.

The system responded very satisfactorily for most images processed. The complete algorithmic procedure was coded in Python 3, using OpenCV library mainly. The programming logic is of low computational cost because is possible to perform simple pixel-by-pixel processing of a megapixel island image in milliseconds using any computer practically.

These concepts can lead to developing more studies among the comparative studies between modern and ancient carbonate platforms morphology and the surrounding oceanographic conditions. The results can be used in geographical object recognition for reservoir modelling of ancient carbonate platforms using techniques like object-based process mimicking, multiple-point statistics, and remote sensing.

Acknowledgements

We would like to thank the fully support of South East Asia Carbonate Research Laboratory (SEACARL) to complete this study.

References

[1] Karim S, Zhang Y, Asif MR, Ali S. Comparative analysis of feature extraction methods in satellite imagery. Journal of Applied Remote Sensing, 2017; 11(4): 042618.
[2] Gischler E, Storz D, Schmitt D. (2014) Sizes, shapes, and patterns of coral reefs in the Maldives, Indian Ocean: the influence of wind, storms, and precipitation on a major tropical carbonate platform. Carbonates and Evaporites 29: 73–87. https://doi.org/10.1007/s13146-013-0176-z.
[3] Utami, Dwi Amanda; Reuning, Lars; Cahyarini, Sri Yudawati (2018). Satellite and Field Based Facies Mapping of Isolated Carbonate Platforms from the Kepulauan Seribu Complex, Indonesia. The Depositional Record 4(2), N 255-273. http://doi.org/10.18154/RWTH-2018-229836.
[4] Chalabi, A., Pierson, B. (2009) Remote sensing analysis of recent carbonate platforms east of Sabah: potential analogues to Miocene carbonate platforms of the South China sea; current update: advantage of multispectral bands. Conference: National Postgraduate Conference on Engineering, Science and Technology.
[5] Montagne A, Naim O, Tourrand C, Pierson B, Menier D. Status of Coral Reef Communi-ties on Two Carbonate Platforms (Tun Sakaran Marine Park, East Sabah, Malaysia) Jour-nal of Ecosystems, 2013; ID 358183.
[6] Rankey E. On facies belts and facies mosaics: Holocene isolated platforms, South China Sea. Sedimentology, 2016; 63(7): 2190-2216.

To whom correspondence should be addressed: Grisel Jiménez S., South East Asia Carbonate Research Laboratory, Department of Geosciences, Universiti Teknologi PETRONAS, Malaysia, E-mail: jimenezsotogrisel@gmail.com