The application of image processing algorithms to the analysis of SPI images

Ovidiu Ghita, Paul F. Whelan
Vision Systems Group
School of Electronic Engineering
Dublin City University
Dublin 9, Ireland

and

Robert Kennedy
Martin Ryan Science Institute
National University of Galway
Galway City, Ireland.

Abstract

Sediment Profile Imagery is a pollution monitoring technique that is widely used to evaluate whether the marine sediments provide the suitable habitat for benthic fauna. Currently, the SPI data is interpreted by human operators and as a consequence the results are inherently affected by intra and inter observer variability. SPI Analyser is a novel software environment that has been specially designed to perform quantitative measurements on SPI data. The devised software is a collection of automatic and semi-automatic image processing algorithms that are suitable to identify important features present in SPI images. Semi-automatic techniques have been developed to identify features such the sediment water interface or custom segmentation of various image details including burrows and voids. In addition an automatic technique for identification of the oxidised sediments has been devised. The software environment has a user-friendly mouse-driven visual interface that facilitates easy manipulation of the input data and outputs the results in a format that can be easily accessed or modified at any stage. The developed application has been evaluated on a collection of various SPI images and encouraging results have been achieved.

Keywords: SPI, pollution monitoring, image processing techniques.

1. Introduction

Sediment profile imagery (SPI) is a remote sensing technique that is currently used extensively to evaluate the activity of the resident marine fauna and the overall level of aquatic sediment contamination [1,2]. Although this sediment monitoring technique has been used by marine benthic scientists for several years, SPI analysis is based on a visual qualitative and quantitative evaluation performed by the human operator [8]. Independent tests reveal that this method is affected in terms of sensitivity and reproductibility by the subjectivism of the human operator.

The aim of this paper is to describe a software environment that is able to perform repeatable quantitative and qualitative measurements from SPI images. As the generic image processing environments that are widely available are not suited to achieve repeatable measurements as they involve considerable user intervention, it became clear that the best option is to develop a custom design image processing environment where measurements are performed either automatically or involving minimal user intervention.

In its current form, the present implementation is able to perform automatic identification of oxidised sediments and can also be successfully applied to the identification of other features present in SPI images such as burrows and voids. It is important to note that the extraction of oxidised sediments is particularly challenging as the color of the oxidised sediments can vary to a relatively large extent and they may have a heterogeneous distribution within the image. The identification of burrows and voids requires minimal user intervention and consecutive measurements reveal good repeatability. Other useful tools include the semiautomatic identification of the sediment water interface and various image processing procedures including adaptive smoothing [7] and histogram analysis [10].

The devised software environment (MRS 2.1) is well under development, with a freely available prototype system currently in use (code and user manual can be downloaded from: *http://www.eeng.dcu.ie/~whelanp/vsg/vsgcode.html*). We plan to add extra features such as automatic identification of the sediment water interface and the presence of voids and burrows in the image. In the next sections we present in detail the main computational components of the devised software environment.

2. Graphical interface

In order to facilitate an easy access to the computational and graphical components the main menu has been divided in a numbers of items that are grouped in agreement with the type of operations performed. The image processing operations are applied in succession and in case of a mistake the user can reset the operations and start over again. The most important operations include file access, application of standard image processing operations, semiautomatic segmentation and data clustering.

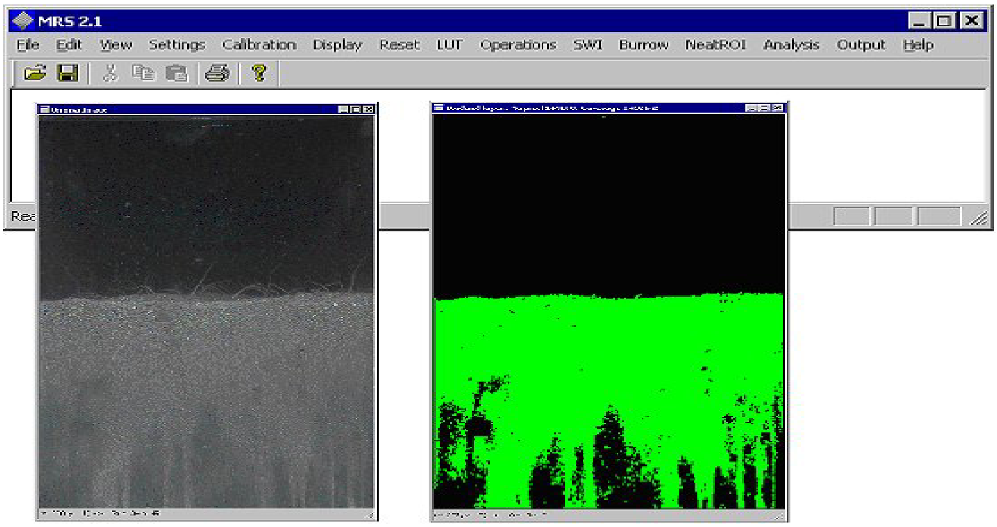


Figure 1. SPI Analyser – graphical interface.

3. File access, parameter settings and display options

The software has been developed to load and save the image data in bitmap format. The application has been designed to save the data associated with the last operation performed. This feature allows the user to save the data after any processing tool has been applied and it is useful to prevent the repetition of standard operations. The input/output operations are grouped in the "File" sub-menu.

The parameters required by computational components such as edge detection, Gaussian blurring [10] and clustering [3] can be modified through the "Settings" sub-menu. Also an option to revert the settings to default values is provided.

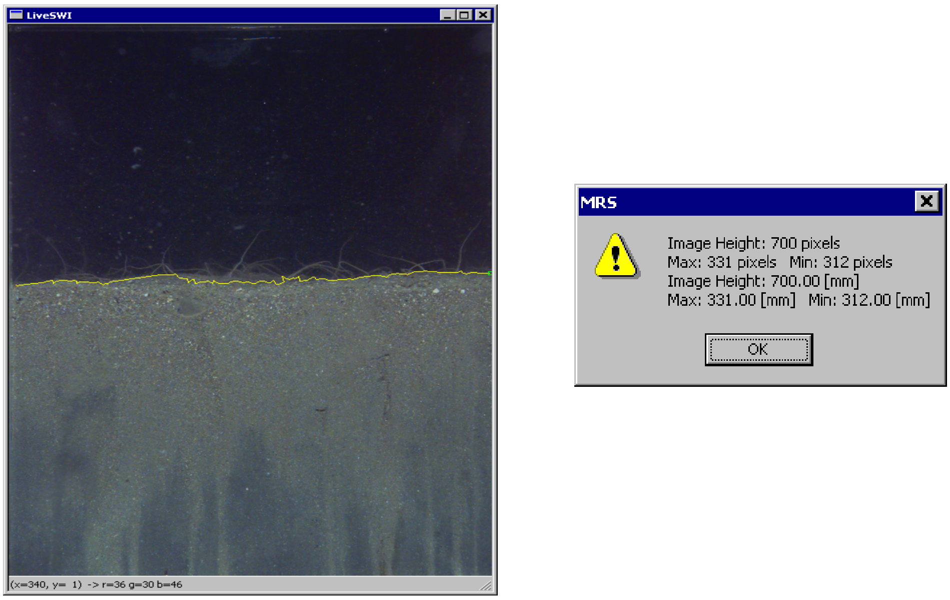
The data can be displayed in various graphical forms using the options that are contained in the "Display" sub-menu.

4. Histogram operations and image filtering

The histogram operations are grouped in the "LUT" sub-menu and allow the user to apply greyscale transformations in order to enhance the image contrast [10]. The low-level operations are grouped in the "Operations" sub-menu and allow the user to perform blurring or image sharpening depending on the objective of the subsequent operations (i.e. sharpening enhances the high frequency information while blurring improve the local homogeneity by removing image noise). An important operation is the edge extraction and this operation must precede the watershed-based segmentation described in Section 6.

5. Identification of the sediment water interface (SWI)

The "SWI" sub-menu contains the interface to computational components employed to identify the SWI in SPI images. For this purpose a semi-automatic method (intelligent scissors) that requires minimal user intervention has been developed [4,6].



(x=340, y= 1) -> r=36 g=30 b=46

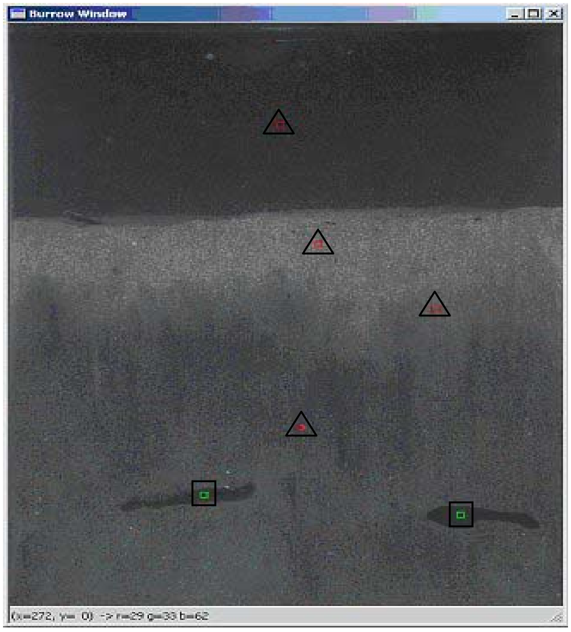
Image Height: 700 pixels
Max: 331 pixels   Min: 312 pixels
Image Height: 700.00 [mm]
Max: 331.00 [mm]   Min: 312.00 [mm]

Figure 2. Semi-automatic detection of SWI.

The user first selects an initial point by placing the mouse cursor as close as possible to the interface between the water and sediments. Then the user moves the mouse cursor over the image regions that represent SWI points and an optimal path from the initial point to the current point defined by the mouse cursor is shown. In image areas were the gradient information is not strong enough the user can add points to the optimal path by clicking the left button of the mouse. The last point on the SWI line is set by clicking the right button of the mouse. Once the right button is clicked the SWI line is finalised and the user can obtain information regarding how deep the SPI camera has penetrated the seabed. Figure 2 illustrates the SWI detection process.

6. Burrows/voids detection

The "Burrow" sub-menu provides the graphical environment to computational components required to extract features such as burrows and voids. Currently, two distinct techniques that can be successfully applied to the detection of the aforementioned benthic features are implemented. The first implementation is based on seeded watershed algorithm [11] and before its application the edge information should be calculated using the "Edges" selection available in the "Operation" sub-menu. The second method is built on a region growing strategy [9]. The graphical interface required to select the Positive (marked in green) and Negative seeds (marked) in red is very intuitive and involves only simple clicks with the left and right buttons of the mouse on the image areas that represent voids/burrows (for positive seeds) and sediments (negative seeds). Figures 3 to 5 illustrate the seed selection process and the results of the application of the burrow/void detection algorithms. Also information regarding position within the image or the size of the detected benthic features can be obtained by selecting the "Info" option. It is useful to mention that the region growing detection tool can be used to identify other benthic features that present similar color/texture attributes. A typical example is illustrated in Figure 6 where an image area with similar texture attributes is identified.



(x=272, y= 0) -> r=29 g=33 b=62

Figure 3. Selection of seed points for burrow detection algorithms (green (squares) – positive seeds, red (triangles) – negative seeds).

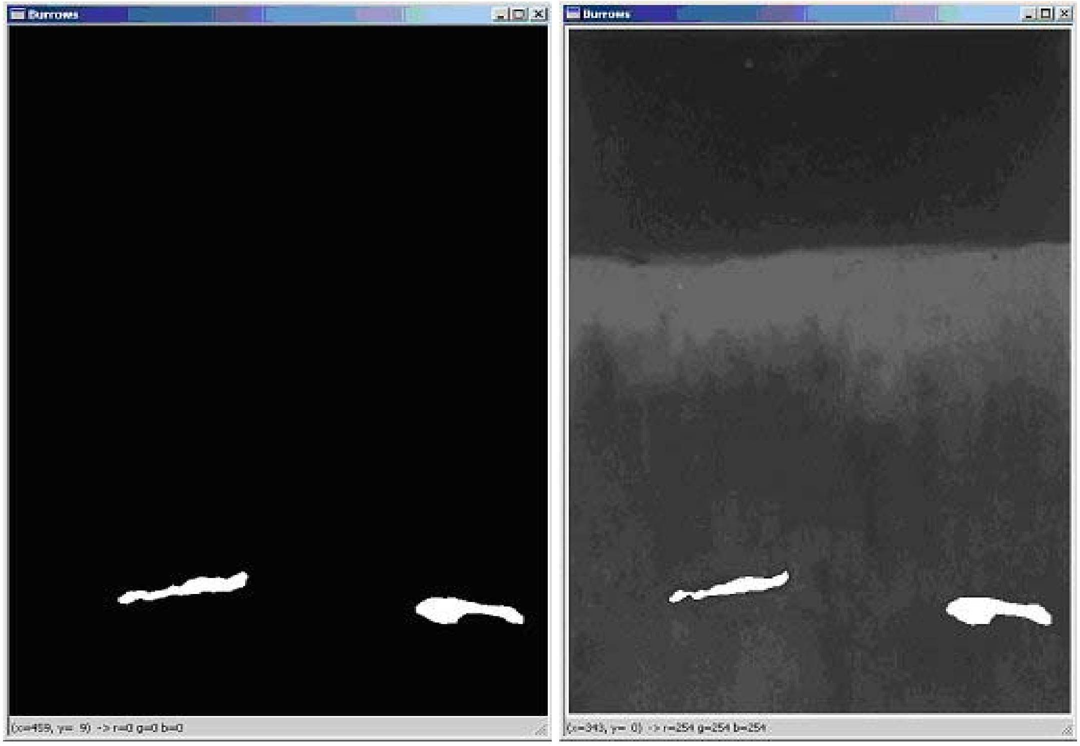

Burrows
Burrows
(x=459, y= 9) -> r=0 g=0 b=0
(x=343, y= 0) -> r=254 g=254 b=254

Figure 4. Burrow detection results. (Left) Watershed–based algorithm. (Right) Region growing technique.

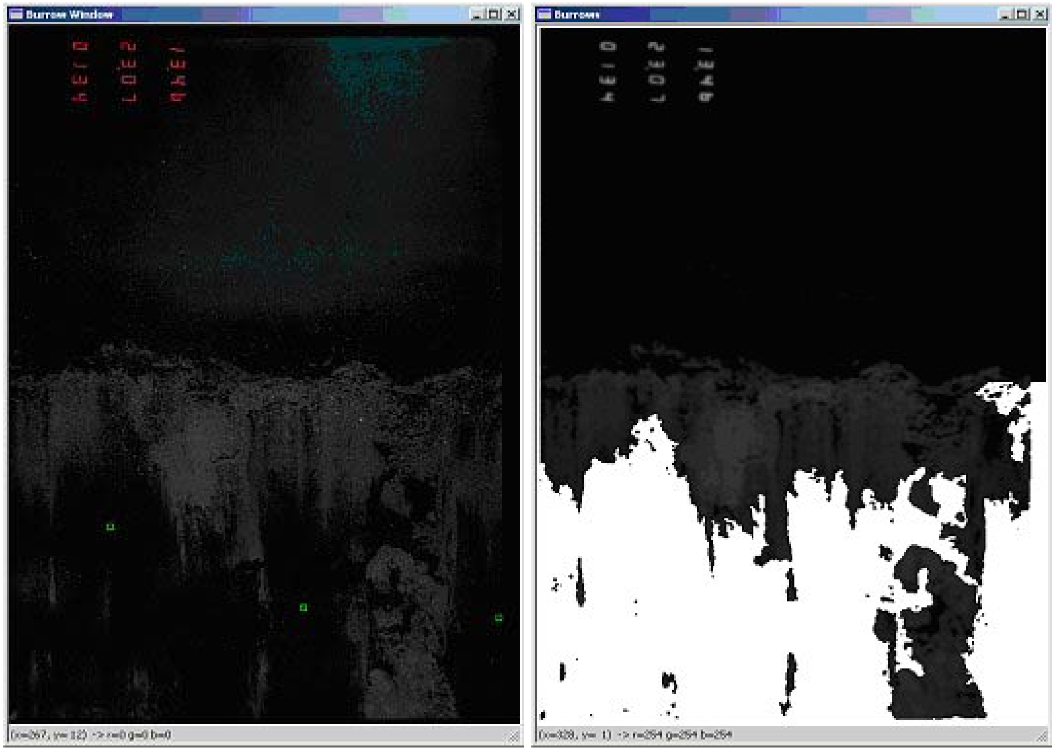

Burrow Window
Burrows
(x=267, y= 12) -> r=0 g=0 b=0
(x=328, y= 1) -> r=254 g=254 b=254

Figure 5. Detection of large burrow formation using the region growing technique. (Left) Input image with the positives marked in green. (Right) Output image.

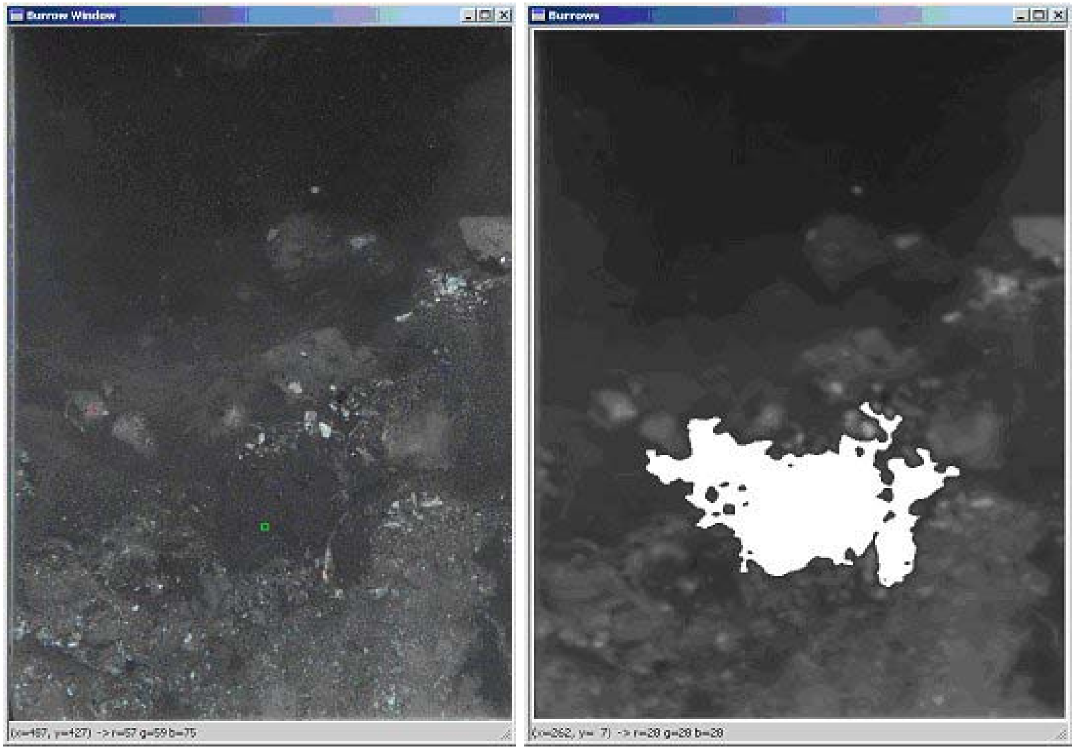

Burrow Window
(x=487, y=427) -> r=57 g=59 b=75
Burrows
(x=262, y= 7) -> r=28 g=28 b=28

Figure 6. Feature detection using region growing. (Left) Input image with the positive seed marked in green. (Right) Output result.

7. Semi-automatic tool for segmentation of closed contours

NeatROI is another semi-automatic segmentation tool that has been included in the development of the SPI Analyser software environment. The functionality of this tool is closely related with that required to identify the SWI line.

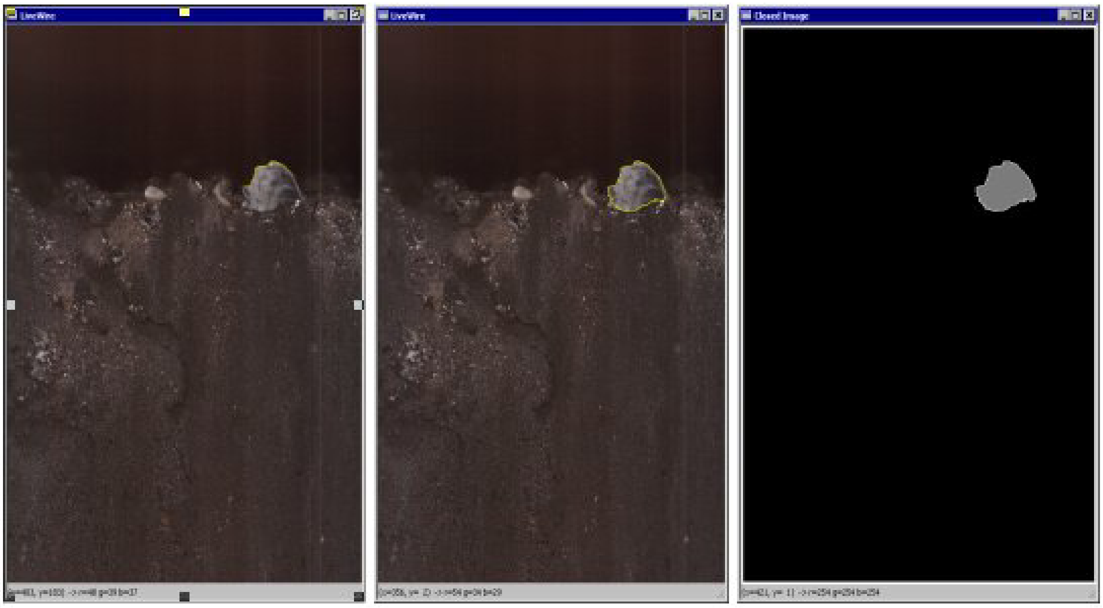
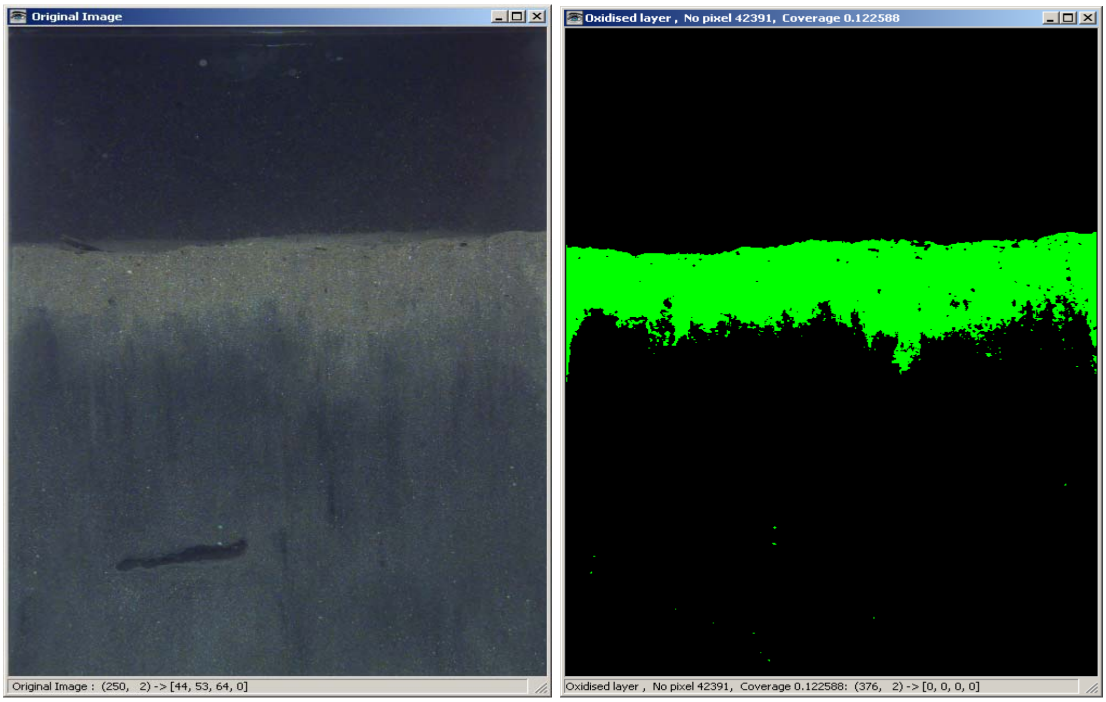


Figure 7. Feature segmentation using the NeatROI tool.

Thus the user has to initially choose the option “Initialise” to extract the gradient information. Then the user can pop-up a window by clicking the “TrackROI” option. The user has to select a point on the boundary of the object he/she wants to segment and then an optimal path is inferred from the starting point to the current point indicated by the mouse position. The segmentation is complete when the contour is closed. Then an additional window will pop-up automatically and will show the segmented object. This process is illustrated in Figure 7.

8. Identification of the oxidised sediments

The segmentation of the sediments in distinct layers is performed by the operations grouped in the “Analysis” sub-menu. The computational routines mainly implements supervised and unsupervised clustering techniques [3]. In this regard, for supervised color and greyscale clustering techniques the user has to set the number of resulting clusters in the window provided in the “Settings” sub-menu. The unsupervised technique will identify the number of final clusters analytically [5]. The supervised techniques are very efficient when the separation between the layers contained in the analysed sediment is well defined. Also the computational time required by the supervised techniques is generally lower than that required by the unsupervised clustering method. After clustering is performed the user can review each identified cluster in the image and evaluate its color histogram. Also a classification algorithm indicates if the cluster is oxidised or reduced. Alternatively, the user can select the option “Detect oxi layers” and the algorithm will identify and display in an output image all oxidised layers contained by the sediment. Information about the image coverage of the oxidised layer is also provided. A graphical example is illustrated in Figure 8.



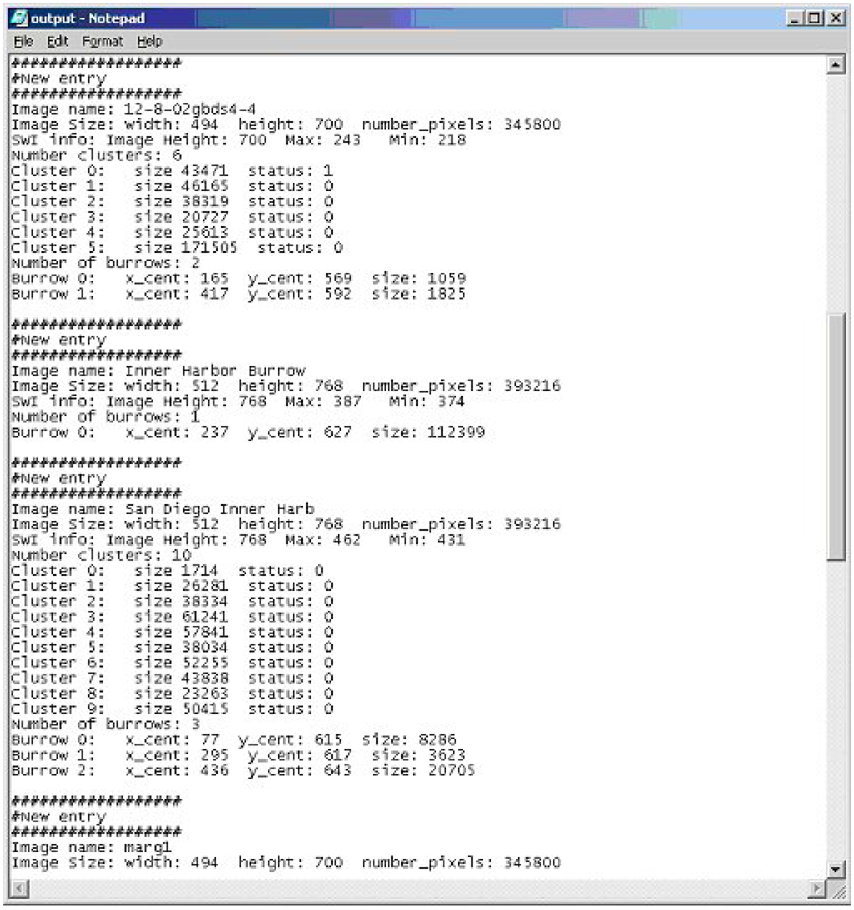
Original Image : (250, 2) -> [44, 53, 64, 0]

Oxidised layer , No pixel 42391, Coverage 0.122588: (376, 2) -> [0, 0, 0, 0]

Figure 8. Identification of the oxidised sediments. (Left) Input image. (Right) Output result.

9. Results presentation

The program outputs the results as a text file for ease of access and interface with other popular applications. The user can add a new entry to the master file when a sequence of successful operations is completed. The output entry is a collection of tags that are associated with the sequence of operations. The information revealed by the output entry includes: image name, image size, SWI data, clustering information and details about burrow detection. The user can access the output information at any time by selecting the "Open output file" option. A section of a typical output file is depicted in Figure 9.



```
##################
#New entry
##################
Image name: 12-8-02gbds4-4
Image Size: width: 494  height: 700  number_pixels: 345800
SWI info: Image Height: 700  Max: 243   Min: 218
Number clusters: 6
Cluster 0:   size 43471  status: 1
Cluster 1:   size 46165  status: 0
Cluster 2:   size 38319  status: 0
Cluster 3:   size 20727  status: 0
Cluster 4:   size 25613  status: 0
Cluster 5:   size 171505  status: 0
Number of burrows: 2
Burrow 0:   x_cent: 165  y_cent: 569  size: 1059
Burrow 1:   x_cent: 417  y_cent: 592  size: 1825

##################
#New entry
##################
Image name: Inner Harbor Burrow
Image Size: width: 512  height: 768  number_pixels: 393216
SWI info: Image Height: 768  Max: 387   Min: 374
Number of burrows: 1
Burrow 0:   x_cent: 237  y_cent: 627  size: 112399

##################
#New entry
##################
Image name: San Diego Inner Harb
Image Size: width: 512  height: 768  number_pixels: 393216
SWI info: Image Height: 768  Max: 462   Min: 431
Number clusters: 10
Cluster 0:   size 1714  status: 0
Cluster 1:   size 26281  status: 0
Cluster 2:   size 38334  status: 0
Cluster 3:   size 61241  status: 0
Cluster 4:   size 57841  status: 0
Cluster 5:   size 38034  status: 0
Cluster 6:   size 52255  status: 0
Cluster 7:   size 43838  status: 0
Cluster 8:   size 23263  status: 0
Cluster 9:   size 50415  status: 0
Number of burrows: 3
Burrow 0:   x_cent: 77  y_cent: 615  size: 8286
Burrow 1:   x_cent: 295  y_cent: 617  size: 3623
Burrow 2:   x_cent: 436  y_cent: 643  size: 20705

##################
#New entry
##################
Image name: marg1
Image Size: width: 494  height: 700  number_pixels: 345800
```

Figure 9. Output file format.

10. Conclusions

The aim of this paper is to present a brief overview of a new software environment that has been specially designed to analyse SPI images. Currently, the analysis of the data returned by an SPI camera is interpreted by human operators. The procedure required to analyse the SPI data is not only laborious but also is affected by a poor reproductibility and require a considerable expertise.

Thus, the development of the SPI Analyser software environment is the first step towards the automation of the process to interpret SPI data. We are aware that considerable work has to be done before a robust image processing application will meet the current standards in practice for analysing SPI images, but the initial results returned by the current implementation are encouraging. Also additional steps need to be taken towards the standardisation of the SPI cameras.

References

[1] J. Cutter and R.J. Diaz, “Novel optical remote sensing and ground truthing of benthic habitat using the Burrow-Cutter-Diaz ploughing sediment profile camera system (BCD sled)”, Journal of Shellfish Research, 17(5), 1443-1444, 1998.

[2] R.J. Diaz and L.C. Schaffner, “Comparisons of sediment landscapes in Chesapeake Bay as seen by surface and profile imaging”, Chesapeake Research Consortium.

[3] R. Duda and P.E. Hart, Pattern classification and scene analysis**,** John Wiley & Sons, New York, 1973.

[4] A. Falcao, J.K. Udupa, S. Samarasekera and S. Sharma, “User-steered image segmentation paradigms: Live Wire and Live Lane”, Graphical Models and Image Processing, vol. 60, pp. 233-260, 1998.

[5] O. Ghita, P.F. Whelan and R. Kennedy, "A practical approach for analysing SPI images", Systemics, Cybernetics and Informatics (SCI 2003), July, Florida, USA, 2003.

[6] E.N. Mortensen and W.A. Barrett, “Interactive segmentation with intelligent scissors”, Graphical Models and Image Processing, vol. 60, pp. 349-384,1998.

[7] Michael Lynch, Kevin Robinson, Ovidiu Ghita, Paul F. Whelan (2004), “A Performance Characterisation in Advanced Data Smoothing Techniques”, Proceedings of the Irish Machine Vision & Image Processing Conference 2004, Trinity College Dublin, September 1st-3rd, 2004.

[8] M. Solan, R. Kennedy and B. Keegan, “Sediment profile imagery: Are we looking at the same picture?”, Technical report, National University of Galway, Ireland, 2002.

[9] M. Sonka, V. Hlavac and R. Boyle, Image Processing, Understanding, and Machine Vision, 2nd edition, PWS Boston, 1998.

[10] P.F. Whelan and D. Molloy, Machine Vision Algorithms in Java: Techniques and Implementation, Springer (London), 2000.

[11] L.Vincent, P. Soille, ”Watersheds in digital spaces: an efficient algorithm based on immersion simulations”, IEEE Trans. Pattern Anal. Machine Intell. (PAMI), 13(6), pp. 583-598, 1991.