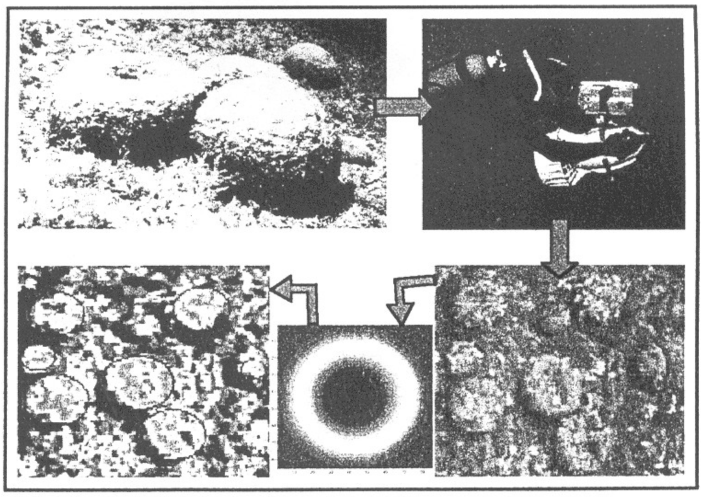
Université de Liège

Matlab designed Image Analysis Software for detection of Codium elisabethae and calculation of population statistics.

Summer training work realised within the MAROV Project (Mapping of Marine Habitats of the Azores using Robotic Ocean Vehicles), co-ordinated by the Department of Oceanography and Fisheries, University of the Azores, Faial Island.



Sirjacobs Damien
European Diploma (DEA) in Modelling of the Marine Environment
August 2002

I am sincerely grateful to Prof. Ricardo Santos for welcoming my participation to this summer 2003 Marov campaign, to Frederico Cardigos for his help in making this stage possible, and multiple advises. I would also like to thank Fernando, Madalena, Rogerio and Carla for their diverse and daily support within the ImagDOP department.

Index

1. Introduction

This work was realised in the general context of the MAROV project (Mapping of Marine Habitats of the Azores using Robotic Ocean Vehicles (http://www.horta.uac.pt/projectos/marov/). The specific objecive of this project is the generation of detailed maps of a set of Sites of Community Importance (SCIs- EU Habitats Directive) in the Azores. The particular aim targeted here is the development of a methodology for studying the distribution and dynamics of Codium elisabethae population with the help of submarine image analysis. In the present work, we focused on designing a image processing program allowing automatic Codium detection and the calculation of some population parameters.

The Codium elisabethae O.C. Schmidt is green algae characterised by a globular shape and sponge texture (Fig. 1). This alga may reach 30 cm in diameter and is exclusively found on rocky substrate. The depth of occurrence is ranging from 8 to 30 meters with highest density found between 15 to 25 meters. This algae was for a long time considered as endemic to the Azores, but was also recorded in 1985 around Porto Santo island, in the archipelago of Madeira (Audiffred & Prud'home van Reine, 1985). This Codium elisabethae can only be distinguished from the Codium bursa (present in the Mediterranean and the oriental coasts of the Atlantic) at the microscopic level.

Fig.1- Views of Codium elisabethae in the Azores
(from http://www.horta.uac.pt/species/Algae/Codium_elisabethae/)

2. Image description

Image acquisition technique:

The submarine video images were acquired with a Sony digital video camera protected by a *Amphibico* waterproof housing in the sites called Caldeirinhas and Ponta dos Radares (close to Monte da Guia, Faial island, Azores, Portugal). The video camera was mounted on submarine scooter with 45 degree angle from the vertical (Fig. 2).

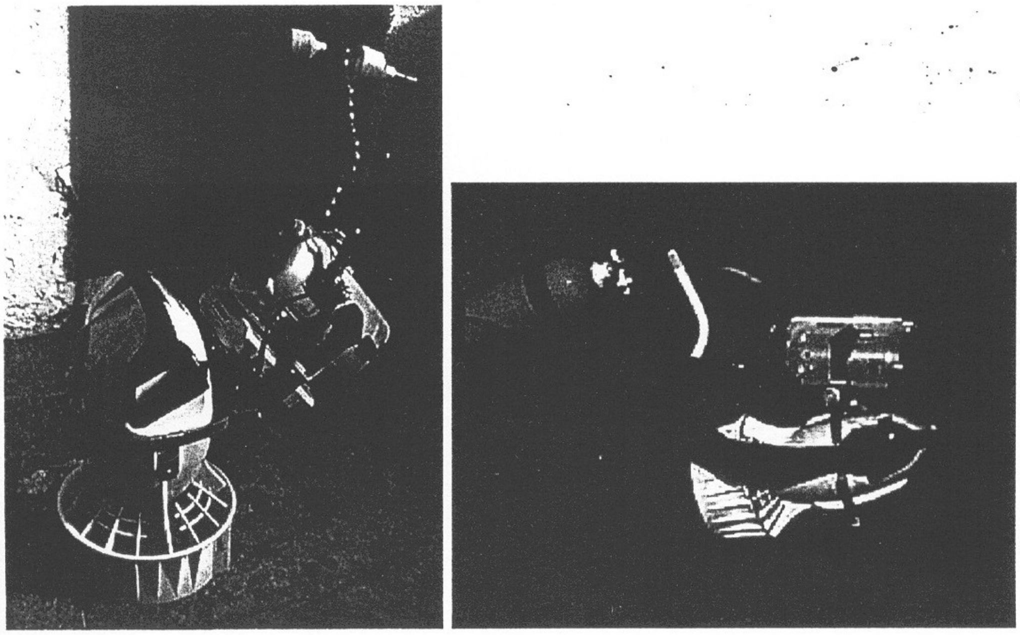


Fig.2 - Video and scooter equipment (Source: http://www.horta.uac.pt/projectos/marov)

The depth of sea bed filmed were ranging from 24 to 26 meters, whereas distance to target (from camera to sea bed) was around 4 meters.

Mosaic building:

The complex process of creation of a mosaic consists in the alignment of multiple individual images in a single image, creating thus a panoramic vue. This work was realised with a software called "Adelie" (Salgado et al., 2001). Because of variations in light conditions and automatic reaction of the camera, as well as various movements of the camera, some artefacts linked with mosaic building may be observed on the single panoramic vue, such as: intensity shifts, straight borders, or 'waves' probably linked with the correction of some camera rotation movements.

General image characteristics:

The true color RGB image presented in Fig 3-a is decomposed into its three color components as an illustration (Red in Fig 3-b, Green in Fig 3-c, Blue in Fig 3-d). Most information is recorded in green and blue band (with a certain shift, but high correlation), as red light is mostly absorbed by upper layers of water.

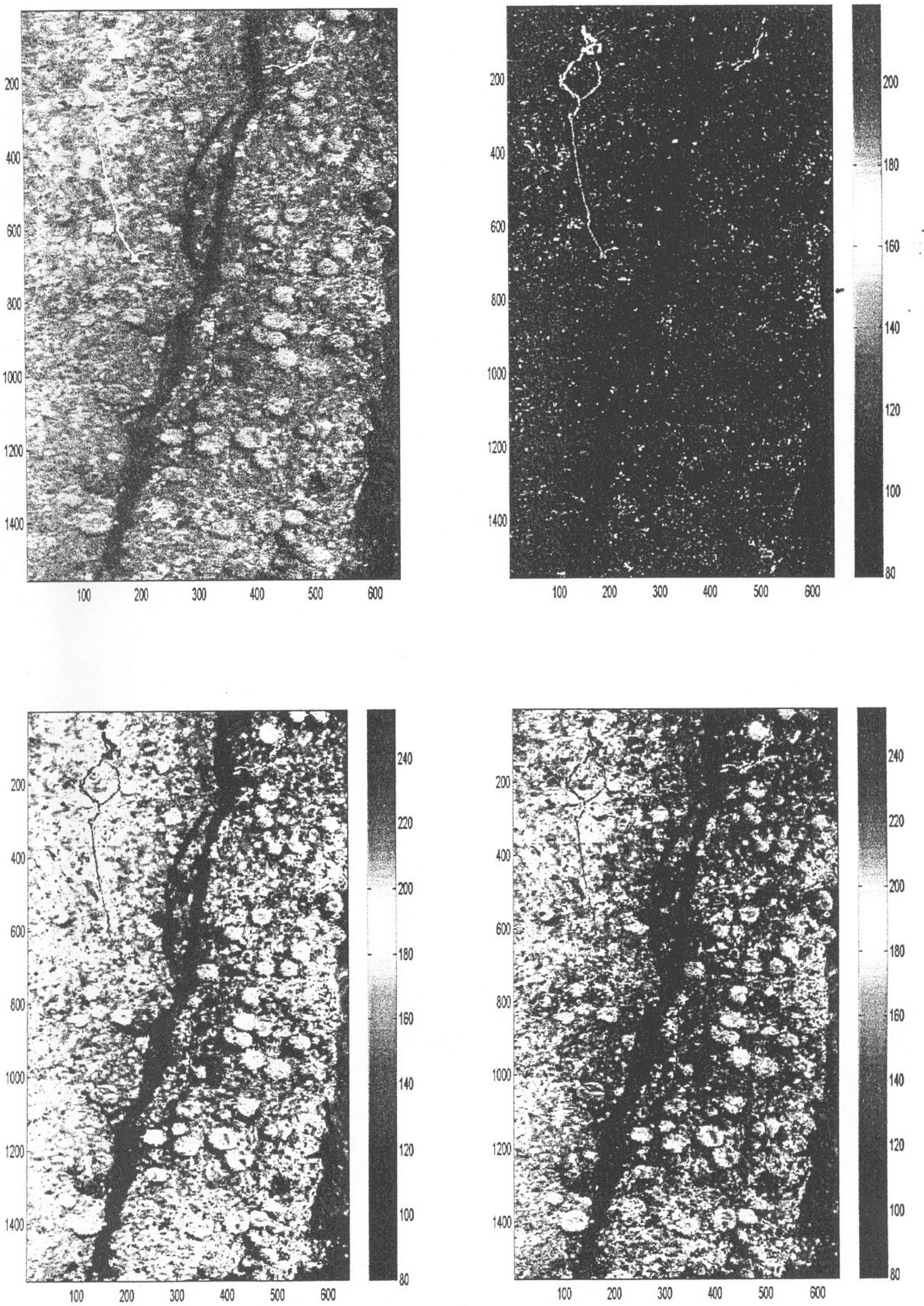


Fig. 3 – True color image of the rocky substrate with Codium elisabethae (a), and the corresponding red (b), green (c) and blue (d) isolated color components.

3. Matlab designed software

Any filename or variable used in the program are given here in *italics*.

The image processing program developped for this application with Matlab 5.3 software consists of two executable matlab files:

ConfigC.m This script is used to define most of the information (as the name of input image and all user definable processing parameters) required by the main program to treat the image. After giving desired value to all parameters and saving this file, the user may execute it by typing 'config' and 'ENTER' in the Matlab Command Window (see annexe 9-a for synthetic definition of all parameters, as green comments). This action will generate a file called *fconf.mat* which contain all information about the following image processing. After the processing, this file may be renamed and kept for helping later result analysis.

Codium3.m This script contain the main program (see annexe 9-a). Main steps of work are:

Pre-processing of the image:

Cuting or defining a region of interest (ROI) within the image;
Extracting the Green band and display for threshold limits user definition;
Treshold application ;
Median filtering:
Histogram equalization;

Calcul of the centroid gradient signature over the region of interest:

Calcul of the centroid class filter: *matclass*. This filter is used in the following step to class all image pixel's values belonging to analog concentric rings around a central pixel into separate vectors (see illustration of classes in Fig. 4).

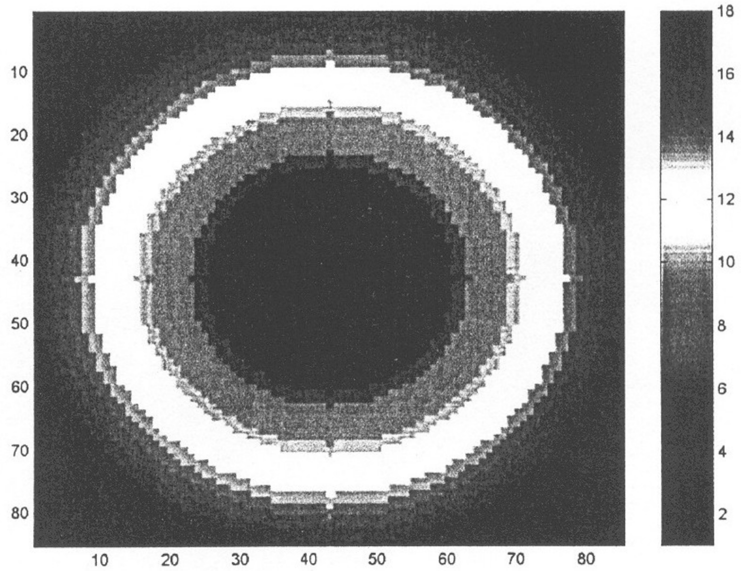


Fig. 4 – Exemple of centroid class filter: *matclass (Rinti=10; Rintf=40; dR=2).*

Calcul of *MGC:* For every pixel tested as a 'potential center of Codium' over the image, the previously described vectors will be used for deriving the signature fonction defined as follows. Allong a radius growing from minimum (*Rinti*) to maximum (*Rintf*), the difference between the medians of the internal cumulated concentric classes and the external ring class is computed and saved in the *MGC* matrix.

Detection of spheric structure:

Detection of pics in the signature function: Along the signature function calculated for each potential center, the algorithm search for presence of a pic along the radius (value superior to previous and following radius value) and react if this pic is superior to a user defined value (tpicpot). Cpic and Cray are matrix recording the intensity and radius of such detected pics.

Mean filtering of the pics: Cpic is spatially filtered (with a filter size window of (2*dzmn+1)) and recorded in Cpic1 matrix if more then minneighb neighbours pics were present in the space window.

Extraction of local maximum pics: Upon pics recorded in Cpic1, the spatial local maximums (searched within a window of (2*dzmx+1)) are recorded into Cpic2 matrix.

Drawing selected cycles: Any pic selected at this stage is drawn over the preprocessed image (centered around the same central position, with a radius corresponding to the analog value found in Cray).

Elimination of overlapping detections: Any conflict between overlapping sphere detected is solved by reducing by half of overlap the radius of simply intersecting circles, and by eliminating any circle which center would be included in another circle. The final detection result is again displayed (see Figure 5).

Extracting Codium population statistics: After this last selection, some synthetic characteristics of the Codium population detected are calculated and saved as text file filename:

- Total number of Codium detected
- Density of Codium (nb/sqr meter)
- Proportion of area covered by Codium (%)
- Histogram of Codium population according to radius

4. Preliminary results

Detailed configuration settings, time consumption and results of most important steps of the preliminary runs are given in the annexe 9-b in table format. Some new parameters appears along with the processing of new images, as new optional functions were developped to improve detection efficiency.

Amongst the Codium image database, the '2105median2.jpg' image, showing optimal combination of contrast, resolution, lightening conditions and regularity of background was used to begin detection experiments.
The first approach for pretreatment was to work on the following combination of the green and blue bands (most information, although highly correlated):

separate histogram equalization applied to both green and blue bands, multiplication composition of the equalized bands, linear intensity strectch of the composition and finally application of 3*3 median filtering. The resulting image (called g2b2s1.jpg) was used for first runs of the program (treatments MGC until MGC5). Nevertheless, better results were then achieved with the green band only (MGC6). As a matter of fact, as the two bands were highly correlated, combining them into intensity resulted in additional noise instead of increased information. This approach was kept for further tests.
Pre-treatment defined as median filtering, cutting intense value and equalization further improved the Codium detection. MGC8 treatement showed optimal results, with cuting values superior to 225 and reducing them to value of 125. This process actually avoid intense reflection from naked rocks or sand (I>225) to be selected as Codium and increase in the same time the intensity gradient of the Codium edges (by incrasing, before equalization, the frequency of pixels showing value analog to Codium edge: 125).

The following parameters (determined by carefull human observation of each image) are used to evaluate detection efficiency:
NbC Vi video/eye = the total number of Codium visible by human eye from the original video visualisation.
NbC Iv image/eye = number of Codium invisible (or easily seen) by human eye on the mosaic image.
NbC Iv edge/soft = number of Codium present in the image edge and thus invisible to the sofware
NbC exp. img/soft = number of Codium expected as optimal estimation from the software
NbC announced = number of Codium estimed by the software
CD = real Codium identified
NCD = non Codium identified (any other roundish structure selected as Codium)
Missing C = missing Codium (compared to expected estimation)

Detection efficiency was then synthetised by:

Confidence in the computer counting compared to human work on image: the ratio between algorithm answer and the expected Codium detection
%Pop expected = *NbC announced / NbC expected*

Confidence in the population histogram according to radius: the proportion of the announced detection corresponding to real Codium
%Accurate detection = real Codium identified / *NbC announced*

Confidence in the computer counting compared to human work on the video film : the ratio between announced population and total present Codium population
%Pop total = *NbC announced / (NbC Vi video/eye - NbC Iv edge/soft)*

Synthesis of detection efficiency:

For each image processing realised, some basic parameters tuning were applied to look for optimal efficiency. The optimal detection efficiency obtained at this stage are presented in the table 1.

Image (.jpg)	2105m2	2105lmed	2105amed-1	2105amed-2	2105amed-3	2105c2med
NbC exp. img/soft	55	43	25	28	8	70
NbC announced	53	45	22	25	10	74
% pop detect/eye	**96**	107	88	89	125	106
% accurate detection	**79**	50	68	60	0	47
% pop tot existing	**80**					

Table 1- Synthesis of detection efficiency for the preliminary image processing.

The best results were reached for the image 2105m2.jpg, and are very encouraging (as illustrated in figures 5). Each Codium was identified by a number, some were considered as hardly visible to human eye on the image are (but visible on the video sequence, before mosaic processing, whereas others were invisible to the software because of their lateral position in the image.

Image 2105m2.jpg:

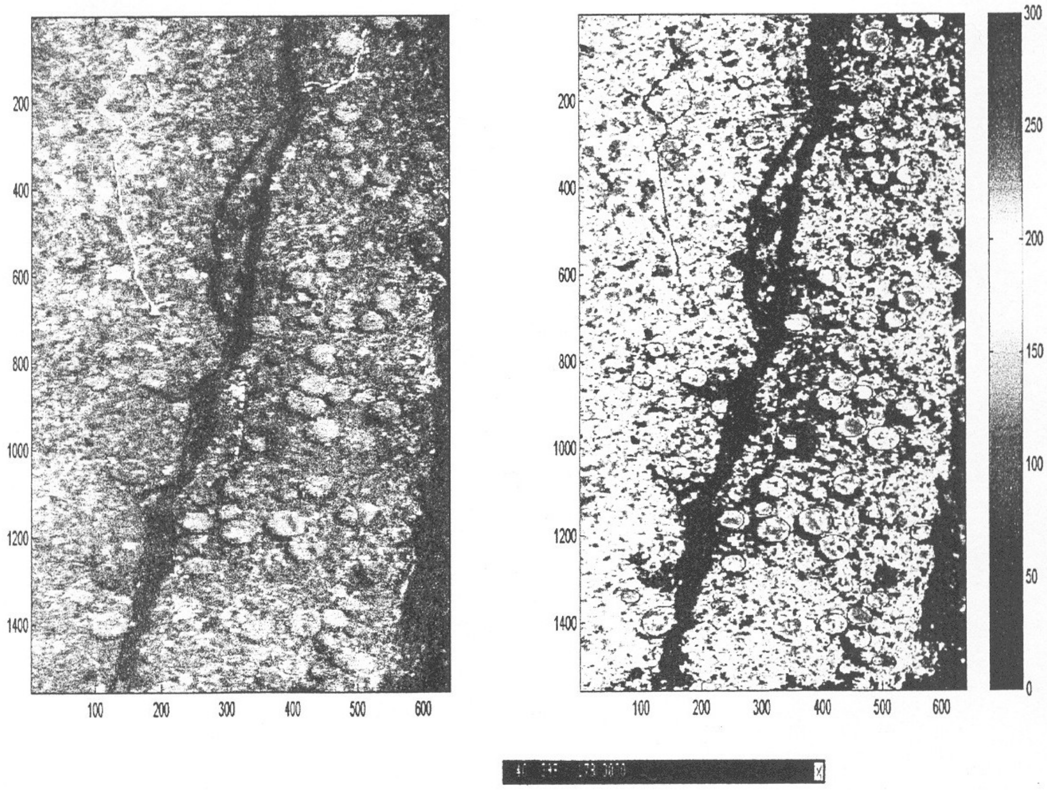


Fig. 5 - Original and processed image 2105m2.jpg

The software produces the following informations: visually illustration of the identifications (Fig X), plot of the radius histogram of identified Codium population (Fig 6), as well as the parameters (given by the matlab command window):

Estimation of number of Codium present:	nbC = 53;
Density of Codium (nb/sqr meter):	densC = 4,06;
Proportion of area covered by Codium (%):	PropcovC = 9,7;

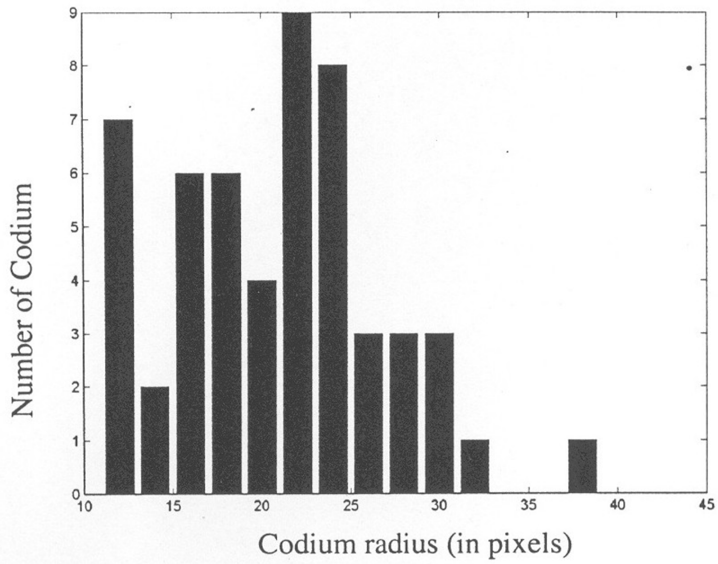


Fig. 6 – Radius histogram of estimated Codium population

The detection efficiency criteria reached for this image are very promising.

The results obtained for the other processed images are illustrated in the figures n° 7 to 11. Details about efficiency and tuning parameters may also be found in annexe 9-b. The short worktime invested in their detection tuning during this summer stage was devoted towards optimal estimation of population number, and secondly towards detection accuracy. This work could already show the general stability of tuning parameters, the important effect of image quality (resolution, lightning and background conditions) on the results, and give some ideas for sofware improvement.

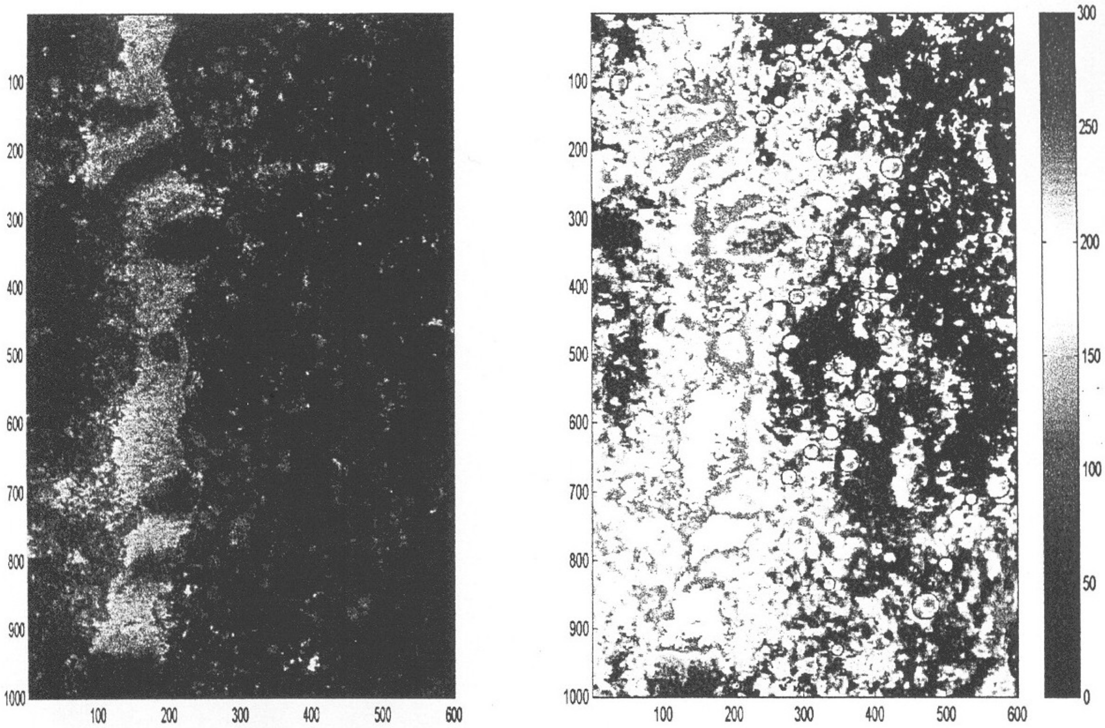


Fig. 7 - Original and processed image **2105lmed**.jpg

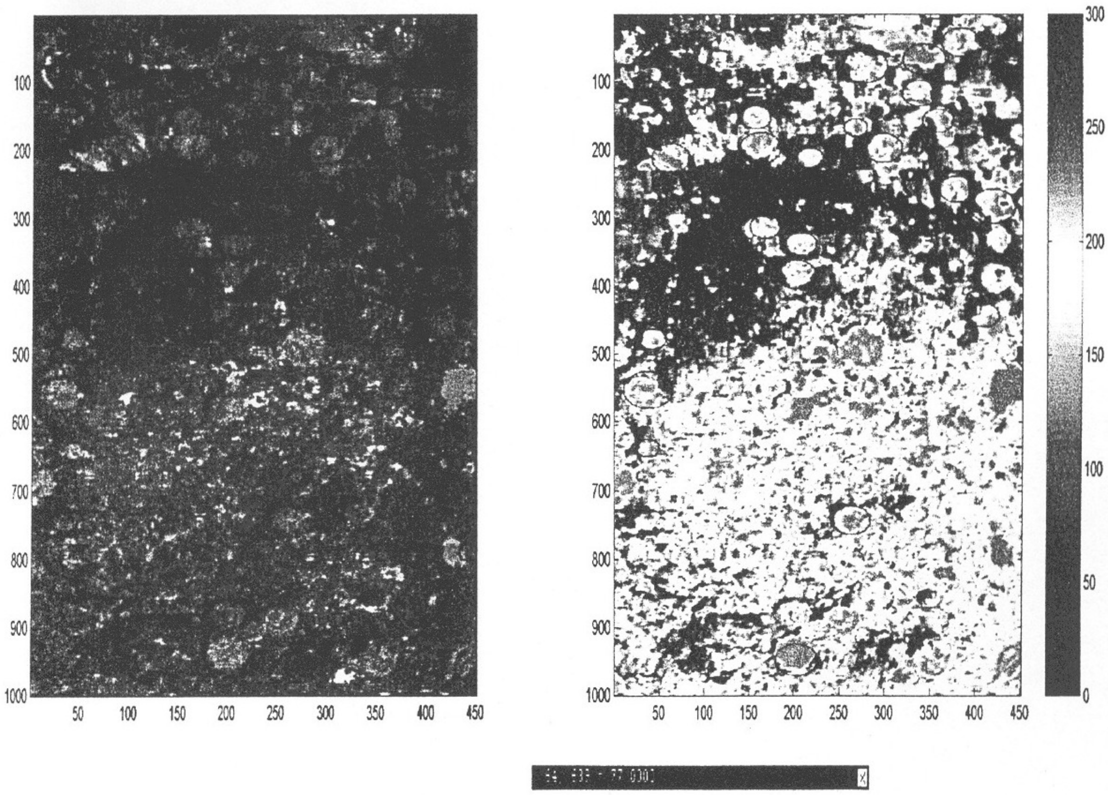


Fig. 8 - Original and processed image 2105amed.jpg (ROI 1).

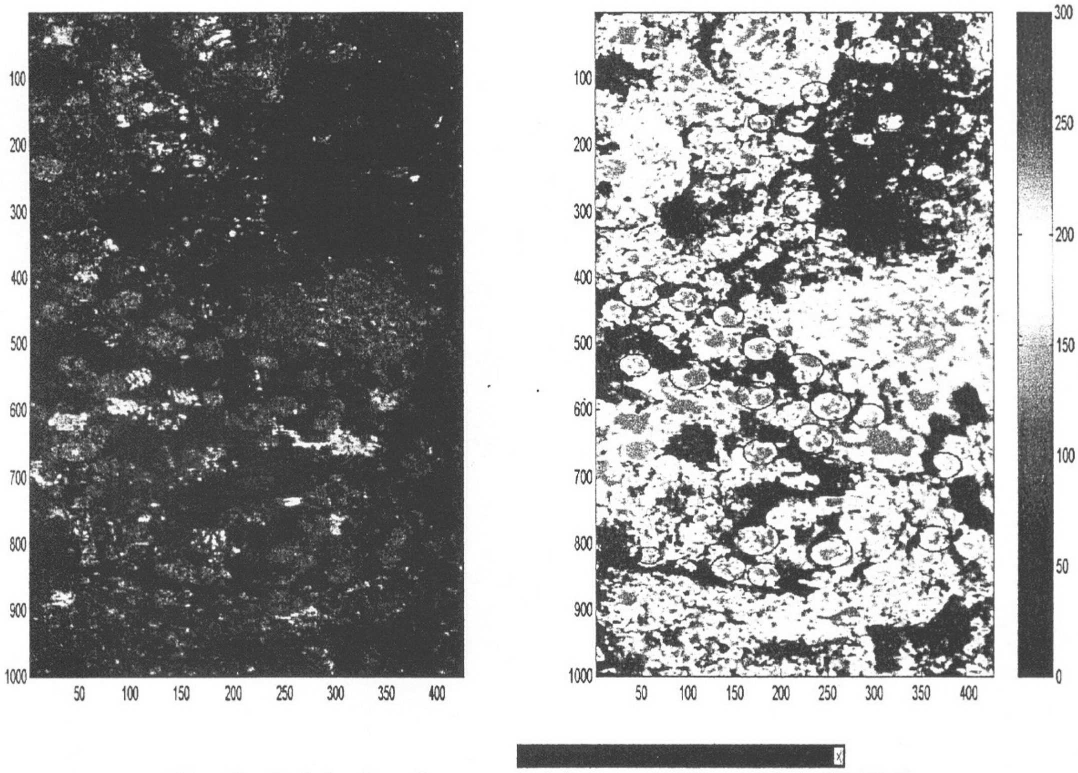


Fig. 9 - Original and processed image 2105amed.jpg (ROI 2)

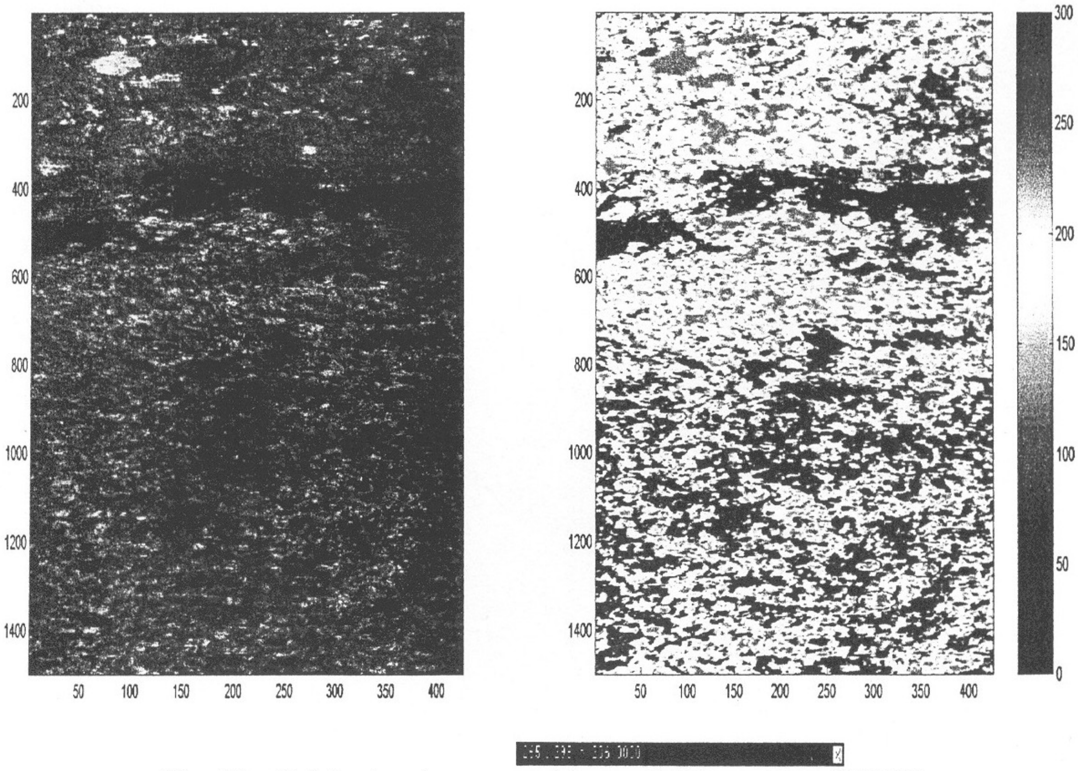


Fig. 10 - Original and processed image 2105amed.jpg (ROI 3)

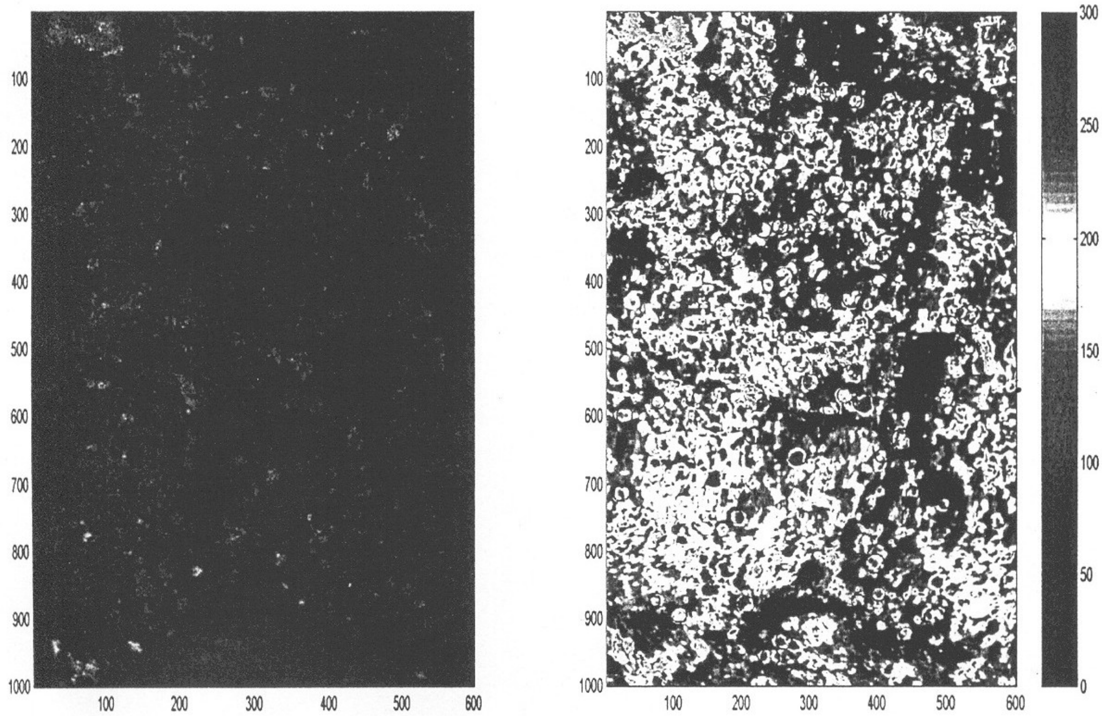


Fig. 11 - Original and processed image **2105c2med**.jpg

Efforts to reduce time consumption allowed to reach an order of 4 to 10 minutes / 100.000 pixels, depending on the user defined radius and spatial coverage.

5. Gaining experience with further image analysis processing

Additional images and regions of interest should be processed for gaining more experience about the algorithm efficiency and for identifying optimal image acquisition conditions and optimal sites for Codium population dynamics survey. Such work can be done by the following procedure:

Run 1: definition of working parameters:

Determine requested action:

Setting parameters: `imsodis=1; imcut=0;`

User defined parameters: `name of input file NAO`

Save and execute ConfigC.m, then execute Codium3.m

This first run will display the whole image to allow user to:

- Choose cutting limits of the region of interest (`cutxi;cutxf;cutyi;cutyf`)
- Define minimum and maximum research radius for Codium (`Rinti;Rintf`)
- Adapt covering step interval if necessary (`pascov`)
- Define intensity threshold and reduced value (`highcut;reduceh`)

Run2: Image processing and first Codium detection

Determine requested action:

Setting parameters: `imsodis=0; imcut=1; tune=0; meanfilt=0;`

User defined parameters: `those defined by the first run`

Save and execute ConfigC.m, then execute Codium3.m

This second run will calculate the MGC matrix and save it under the filename matfile.m. A first attempt of Codium detection will be realised by this run, with the predefined tuning parameters (set according to previous user experience with this image processing software).

Rename matfile.m as MatGCX.m, with X beeing the identification number of the MGC file loaded for following tuning runs.

Following runs: tuning selection parameters

Determine requested action:

Setting parameters: `imsodis=0; imcut=1; tune=1;`

Main tuning parameters: `tpicpot; tmaxpic; minneighb; meanfilt;`

If yet kept constant for the preliminary studies, the combined effect of other parameters could be explored: `tmoypic; pdcm; dzmx and dzmn.`

These tuning parameters should be iteratively modified according to the previous detection results obaitned. The user may benefit from observing the Cpic, Cpic1 and Cpic2 matrix displayed by the previous run to define logical progressive modification of the tuning parameters.

Save and execute ConfigC.m, then execute Codium3.m

Repeat tuning run until reaching optimal detection.

It is the combination of tuning parameters and image characteristics (acquisition conditions and background type) that can be improved by gaining more experience with this software and may constraint optimal Codium detection to a single detection run with constant parameters.

6. Listing potential future improvements

6.a- Image acquisition

- Constant distance to target, not higher then 3 meters (optimal conditions seems to be those of the image 2105m2.jpg) ;

- Put reference distance on the sea bed for pixel size determination (ex: 20 cm white bar, in the beginning and end of video transect).

- Testing results obtained from digital camera pictures (high resolution), or from a specifically designed submarine camera (with ideally high sensitivity to weak light intensity and several spectral channels in the green-blue bandwith).

6.b- Software

Improving Codium detection efficiency

- Implementing local histogram equalization in the pre-treatment

- Trying other pre-treatment to enhance Codium visibility for the centroid gradient algorithm;

- Padding image matrix with a Rext-thick band of 125 grey level pixels for allowing detection of Codium in lateral zones of images;

- Trying a detection based on the gradient calculated between non adjacent ring-classes, without aggregation of internal population;

- Taking into account criteria of internal homogeneity of concentric zones;

- Refining the algorithm of elimination of overlaping detections

- Exploring the first and second derivative specificity of Codium which may be found in the centroid gradient signature (MGC matrix);

- Designing otpion for user defined corrections with the mouse (erase invalid Codium and indicate missing Codium (by center position and diameter), and uptading population's statistics;

Reducing time and ressource consumption

- Modify the MGC matrix (as well as reading, testing and display options) in order to keep only the useful information (this might lead to a factor 25 reduction of RAM memory for this variable: from 100 Mb to 4 Mb for a 1.000.000 pixel image).

- Traducing the program from Matlab to IDL language, for reducing the processing time consumption.

7. Conclusions

An image processing program was specifically designed with Matlab software for Codium detection. It produces the following informations: visual illustration of the identifications, plot of the radius histogram of identified Codium population, number of Codium detected, density of Codium (nb/sqr meter), proportion of area covered by Codium (%).

Efforts to reduce time consumption allowed to reach an order of 4 to 10 minutes / 100.000 pixels, depending on the user defined radius and spatial coverage.

The detection efficiency criteria reached for the 2105m2.jpg image are very promising. The results obtained for the other images processed during this short summer stage could already show the general stability of tuning parameters, the important effect of image quality (resolution, lightning and background conditions) on the detection efficiency.

A detailed methodology description was given in order to allow the users to gain further experience with this software and other image processing.

In addition, this work produced valuable information and identified some proposal for further inprovement of the method, considering image acquisition, detection efficency and time consumption.

8. Bibliography

Salgado. M, Cardigos, F., Santos, R. S., (2001) Elaboration of Mosaics using Adelie Software; Imar/DOP internal report.

Imar/DOP, (2001), Comunidades Biologicas Marinhas dos Sitios de Interesse Comunitario do Canal Faial-Pico; September 2001 Report of the European Maré Project concerning Integrated Management of the Costal and Mazrine zones of the Azores.

Azevedo Neto, I., (1997), Studies on Algal Communities of São Mguel, Azores, PhD thesis in Marine Biology, University of the Azores, Ponta Delgada, 1997.

Vidondo, B. and C. M. Duarte, (1998), Population structure, dynamics and production of the mediterranean macroalga Codium bursa (chlorophyceae), Phycol. (34), p. 918-924.

Vidondo, B. and C. M. Duarte, (1995), Seasonal growth of Codium bursa, a slow growing Mediterranean macroalga: in situ experimental evidence of nutrient limitation, Mar Ecol Prog Ser. (123), p. 185-191.

9. Annexes

9-a- Matlab Codes

Program ConfigC.m, with synthetic definition of all parameters as green comments:

```
%CODIUM DETECTION AND POPULATION STATISTICS
%Configuration program
%Damien SIRJACOBS 08/2002
%d.sirjacobs@student.ulg.ac.be

close all;
clear all;

%definition of input images and files
%matfile=MGC12;
NA0 ='2105amed.jpg'; % name of input image
%A0 = imread('g2b2s1.jpg','jpeg');

%Pretreatment parameters
n=1; % number of repetitions of image pre-treatment
highcut=250; %threshold for reduction of highest pixel values
reduceh=200; %reduced value attributed to highest pixel values
lowcut=0; %threshold for reduction of lowest pixel values
reducel=0; %reduced value attributed to lowest pixel values

%logical variables for optional functions
impretreat=1; % =1 if pre-treatment should be applied to the image
imsodis=0; % =1 if only displaying image covering and pre-treatment is
requested
imcut=1; % =1 if image should be cut before any processing
wincov=0; % =1 if only a sub-window of the cutted image should be processed
if wincov==1 % definition of sub-window coverage
   xcovi=440;
   xcovf=445;
   ycovi=940;
   ycovf=945;
end
meanfilt=1; % =1 if Mean filtering should be applied
tune=1; % =1 for loading MGC matrix and tuning purposes
pix=0.2/50;% dimension of pixels (meters) ; as 50 pix. large medium coidium
is about 20 cm diameter..?
cutxi=300; % definition of region of interest coverage (initial cut along x
axis)
cutxf=725; % definition of region of interest coverage (final cut along x
axis)
cutyi=500;
cutyf=1500;

%Centroid gradient filter definition
pascov=5; % distance between successive pixels tested (in pixels)
Rinti=10; % minimum internal radius of the ring defining pixel class (in
pixels)
Rintf=30; % maximum internal radius of the ring defining pixel class (in
pixels)
dR=2; % thickness of the rings defining pixel class (in pixels)
pasR=2; % radius step increase between successive rings defining pixel
class (in pixels)
```

```
%thresholds definition
tpicpot=70; %threshold for detection of centroid gradient intensity pic
tmoypic=0; %threshold for spatial averaging of detected centroid gradient
pics
tmaxpic=10; %threshold for detection of local maximum centroid gradient pic

%filters definition
dzmn=2*pascov; % dimension of mean window filter
pdcm=1.5; %ponderation of central pixel in local mean calculation
minneighb=2; % minimum number of selected neighbours in the close
neighbourhood
dzmx=2*pascov; % dimension of maximum window filter
cf=3; % dimension of median window filter

%intermediate variables
Rextf=Rintf+dR;
ctt=2*Rextf+1;
xct=Rextf+1;
yct=xct;

save fconf
clear all
```

Program Codium3.m :

```
close all;
clear all;
defintern=0;
if defintern==1;
%logical variables for optional functions
impretreat=1;
imsodis=0;
imcut=1;
wincov=0;
meanfilt=1;
tune=0;
%image coverage definition
%surfIm=150; %in square meters !slope of bottom, altitude of diving flight
:)
%put ref dim for pixel size determination (3)
pix=0.2/50;% in meters ; cause 50pix large medium coidium is about 20 cm
diameter..?
cutxi=300;
cutxf=770;
cutyi=2500;
cutyf=3500;
%Centroid gradient filter definition
pascov=5;
Rinti=10;
Rintf=30;
dR=3;
pasR=3;
%thresholds definition
tpicpot=80;
tmoypic=0;
tmaxpic=10;
%pic filters size definition
dzmn=1*pascov;
pdcm=1.5;
minneighb=2;
dzmx=2*pascov;
draw=0;
Rextf=Rintf+dR;
ctt=2*Rextf+1;
xct=Rextf+1;
yct=xct;
else
   load fconf
end

%definition of input images
A0 = imread(NA0,'jpg');
%A0 = imread('g2b2s1.jpg','jpeg');
if imcut==1
   A1=A0(cutyi:cutyf,cutxi:cutxf,:);
else
   A1=A0;
end
clear A0;
figure
imagesc(A1)
%saveas(gcf,'imo6','bmp')

[ci,cj,n]=size(A1);
```

```
%B = rgb2gray(A1);
%clear A1;
if impretreat==1
   %Red=A1(:,:,1);
   Gre=(A1(:,:,2));
   %Blu=double(A1(:,:,3));
   %figure
   %imagesc(Red)
   figure
   imagesc(Gre)
   colorbar
   %figure
   %imagesc(Gre)
   %colorbar
   %imagesc(Blu)
   clear A1;
%figure
%imshow(B);
%colormap(gray)
%colorbar
%[counts,x] = imhist(B);
%nmed=1;
for i=1:npt
   for l=1:ci
      for c=1:cj
         if Gre(l,c)>highcut
            Gre(l,c)=reduceh;
         elseif Gre(l,c)<lowcut
            Gre(l,c)=reducel;
         %else
         %   Gre(l,c)=255;
         end
      end
   end
figure
imagesc(Gre);
Gre = medfilt2(Gre,[cf,cf]);
%Red = double(histeq(Red));
  Gre = (histeq(Gre));
   %colorbar
   %Blu = double(histeq(Blu));
   %figure
   %imagesc(Gre)
   %figure
   %imagesc(Blu)
   %for li=1:ci
   %   for co=1:cj
   %   end
   %end
   % B=round(B.*B);
%   mn=min(B);
%   mn2=min(mn);
%   mx=max(B);
%   mx2=max(mx);
%   p1=255/(mx2-mn2);
%   B=B-mn2;
%   B=B*p1;
   %B=round(B);
   %B = imadjust(A2,[mn2 mx2],[0 1]);
end
B=double(Gre);
```

```
   figure
   imagesc(B);
clear Gre;
%for i=1:nmed
%    B = histeq(B);
%    %B = imadjust(B,[0 mx],[0 1]);
%   B = medfilt2(B,[cf,cf]);
%    %figure
%    %imagesc(B);
%end
%end
end

%figure
%imshow(B);
%colormap(gray)
%B = rgb2gray(A0);
%[ci,cj]=size(B);
%A1=A(:,:,1);
%A2=A(:,:,2);
%A3=A(:,:,3);A1=double(A1); A2=double(A2); A3=double(A3);
%B=double(B);
B2=B;
Cray(1:ci,1:cj)=0;
Cpic(1:ci,1:cj)=0;
vectdim=[Rinti:pasR:Rintf];
lvd=length(vectdim);
%definition of area of interest
if wincov==0
   xcovi=xct;
   xcovf=cj-xct;
   ycovi=yct;
   ycovf=ci-yct;
end

if imsodis==0
%Calcul MGC or loading MGC from HDrive
%calcmatclass
%matclass(1:yct,1:xct)=0;
      for x=1:ctt
      for y=1:ctt
         d=sqrt((x-xct)*(x-xct)+(y-yct)*(y-yct));
         if d<=Rinti
            %nbcat(1)=nbcat(1)+1;
            matclass(y,x)=1;
         elseif d<=(Rextf)
            for cat=lvd:-1:1
               if d>vectdim(cat)
                  %nbcat(cat+1)=nbcat(cat+1)+1;
                  matclass(y,x)=cat+1;
                  break
               end
            end
         else
            %nbcat(lvd+2)=nbcat(lvd+2)+1;
            matclass(y,x)=lvd+2;
         end
      end
      end
if tune==0;
%image covering
```

```
for xcw=xcovi:pascov:xcovf
   for ycw=ycovi:pascov:ycovf
      nbcat(1:lvd+2)=0;
      clear Z1;
      clear Z2;
      xcw
      ycw
      %vect class extractions
      for x=-Rextf:1:Rextf
         for y=-Rextf:1:Rextf
            readcat=matclass(yct+y,xct+x);
            nbcat(readcat)=nbcat(readcat)+1;
            matcat(readcat,nbcat(readcat))=B(ycw+y,xcw+x);
         end
      end

%radius covering
for cat=1:lvd
   if cat==1
      Z1(1:nbcat(1))=matcat(1,1:nbcat(1));
      Z2(1:nbcat(2))=matcat(2,1:nbcat(2));
   else
      Z1=[Z1,Z2];
      Z2(1:nbcat(cat+1))=matcat(cat+1,1:nbcat(cat+1));
   end
   %Z1=double(Z1(:));
   %Z2=double(Z2(:));
   %GC(cat,1)=mZ1;
   %GC(cat,2)=mZ2;
   MGC(ycw,xcw,cat)=median(Z1)-median(Z2);
end

%clear nbcat;
%clear matcat;
end
end
else
   load (matfile)
end
%pack

%detection of Sphere
for xcw=xcovi:pascov:xcovf
for ycw=ycovi:pascov:ycovf
for Rdet=2:(lvd-1)
   if MGC(ycw,xcw,Rdet-1)<MGC(ycw,xcw,Rdet) &
MGC(ycw,xcw,Rdet)>MGC(ycw,xcw,Rdet+1)
      if MGC(ycw,xcw,Rdet)>tpicpot
         Cpic(ycw,xcw)=MGC(ycw,xcw,Rdet);
         Cray(ycw,xcw)=((Rdet-1)*pasR+Rinti);
         break
%         X=[Rinti:pasR:Rintf];
%         xi=xcw-X(Rdet);
%        yi=ycw-X(Rdet);
%        xf=xcw+X(Rdet);
%        yf=ycw+X(Rdet);
%         for x=xi:xf
%           for y=yi:yf
%           d=sqrt((x-xcw)*(x-xcw)+(y-ycw)*(y-ycw));
%              if d<=X(Rdet)
%                  B(ycw,xcw)=255;
```

```
%                 end
%              end
%           end
        end
    end
end

%save plots median curves
%X=[Rinti:pasR:Rintf];
%figure
%plot(X,GC(:,1:2))
%saveas(gcf,'GC4a.bmp')
%figure
%plot(X,GC(:,3))
%saveas(gcf,'GC4b.bmp')
%figure
%plot(X,vmZ2)
end
end

%Selection of Algae
Cpic1(1:ci,1:cj)=0;
%Mean Filtering of pics

if meanfilt==1
%    for hi=1:pascov:(2*dzmn+1)
%       for hj=1:pascov:(2*dzmn+1)
%          if hi==dzmn+1
%         h(hi,hj)=1;
%    end
%    end
%   Cpic1 = filter2(h,Cpic)
for xcw=xcovi:pascov:xcovf
   for ycw=ycovi:pascov:ycovf
      if Cpic(ycw,xcw)>tmoypic
         li=0;
         nz=0;
         clear loc1;
         for di=-dzmn:pascov:dzmn
            for dj=-dzmn:pascov:dzmn
            li=li+1;
            loc1(li)=Cpic(ycw+di,xcw+dj);
            if loc1(li)>0
               nz=nz+1;
            end
            if di==0 & dj==0
               loc1(li)=pdcm*loc1(li);
            end
         end
      end
      if nz>minneighb
         Cpic1(ycw,xcw)=mean(loc1);
      end
      end
   end
end
else
   Cpic1=Cpic;
end
%Extraction of local maximum pics
Cpic2(1:ci,1:cj)=0;
```

```
Rcer=Rintf+dR;
for xcw=xcovi:pascov:xcovf
   for ycw=ycovi:pascov:ycovf
      if Cpic1(ycw,xcw)>tmaxpic
         li=0;
         for di=-dzmx:pascov:dzmx
            for dj=-dzmx:pascov:dzmx
               li=li+1;
               loc(li)=Cpic1(ycw+di,xcw+dj);
            end
         end
         maxloc=max(loc);
         if Cpic1(ycw,xcw)==maxloc
            Cpic2(ycw,xcw)=2;
            B(ycw,xcw)=300;
            xi=xcw-Rcer;
            yi=ycw-Rcer;
            xf=xcw+Rcer;
            yf=ycw+Rcer;
            for x=xi:xf
            for y=yi:yf
            d=sqrt((x-xcw)*(x-xcw)+(y-ycw)*(y-ycw));
               if d>=(Cray(ycw,xcw)-2) & d<(Cray(ycw,xcw))
              % Cpic2(y,x)=1;
               B(y,x)=300;
            end
            end
            end
         end
      end
   end
end
%Elimination of overlapping detections
for xcw=xcovi:pascov:xcovf
   for ycw=ycovi:pascov:ycovf
      if Cpic2(ycw,xcw)>0
         Ray1=Cray(ycw,xcw);
         xi=xcw-2*Ray1;
         yi=ycw-2*Ray1;
         xf=xcw+2*Ray1;
         yf=ycw+2*Ray1;
         if xi<1
            xi=1;
         end
         if yi<1
            yi=1;
         end
         if xf>cj
            xf=cj;
         end
         if yf>ci
            yf=ci;
         end
         for x=xi:xf
            for y=yi:yf
               %if ((Cpic2(y,x)>0) & ((x~=xcw)!(y~=ycw)))
               if Cpic2(y,x)>0
                  if (x==xcw)&(y==ycw)
                  else
                  Ray2=Cray(y,x);
                  dcc=sqrt((x-xcw)*(x-xcw)+(y-ycw)*(y-ycw));
```

```
                    Rt=Ray1+Ray2;
                    if Rt>dcc
                       if dcc>Ray1 & dcc>Ray2
                          Cray(ycw,xcw)=round(Ray1-(Rt-dcc)/2);
                          Cray(y,x)=round(Ray2-(Rt-dcc)/2);
                       elseif Ray1>=Ray2
                          Cpic2(y,x)=0;
                       else
                          Cpic2(ycw,xcw)=0;
                       end
                    end
                    end
                 end
              end
           end
        end
     end
  end
%Drawing selected cycles and extracting Coidium population statistics
dc=0;
for xcw=xcovi:pascov:xcovf
   for ycw=ycovi:pascov:ycovf
      if Cpic2(ycw,xcw)>0
         B2(ycw,xcw)=300;
         Ray=Cray(ycw,xcw);
         dc=dc+1;
         VRC(dc)=Ray;
         xi=xcw-Ray;
         yi=ycw-Ray;
         xf=xcw+Ray;
         yf=ycw+Ray;
         for x=xi:xf
            for y=yi:yf
               d=sqrt((x-xcw)*(x-xcw)+(y-ycw)*(y-ycw));
               if d>=(Cray(ycw,xcw)-2) & d<(Cray(ycw,xcw))
               B2(y,x)=300;
               end
            end
         end
       end
    end
 end
 %pack
 fprintf(1,'number of Coidium detected:')
 nbC=length(VRC)
 pix2=pix*pix;
 surfim=(xcovf-xcovi)*(ycovf-ycovi)*pix2;
 fprintf(1,'density of Coidium (nb/sqr meter):')
 densC=nbC/surfim
 covC=0;
 for nc=1:nbC
    covC=covC+pi*VRC(nc)*VRC(nc)*pix2;
 end
 fprintf(1,'Proportion of area covered by Coidium (%):')
 PropcovC=covC/surfim
 VRCx=[Rinti+pasR:pasR:Rintf+pasR];
 [n,xout] = hist(VRC,VRCx);
 figure
 bar(xout,n)

%Displaying process and results
```

```
figure
imagesc(matclass)
colorbar
clear matclass;


figure
imagesc(Cray)
colorbar
pixval on
%saveas(gcf,'ray','bmp')
clear Cray;

figure
imagesc(Cpic)
colorbar
pixval on
%saveas(gcf,'pic','bmp')
clear Cpic;

figure
imagesc(Cpic1)
colorbar
pixval on
%saveas(gcf,'picfilt','bmp')
clear Cpic1;

figure
imagesc(B)
colorbar
pixval on
%saveas(gcf,'detectB','bmp')
clear B;

figure
imagesc(B2)
colorbar
pixval on
saveas(gcf,'detectB2-6','bmp')
clear B2;

if tune==0;
   save 'matfile' MGC
end

%save MatGC MGC -ascii -double -tabs;
end
fclose('all')
```

9-b- Detailed configuration and results of preliminary tests

Image/ROI/treatment definition

	2105m2.jpg					
image	g2b2s1.jpg	g2b2s1.jpg	2105m2.jpg	2105m2.jpg	2105m2.jpg	2105m2.jpg
band	Intensity	Green		Green	Green	Green
pre-treatment		median		median/cut/equal	median/cut/equal	(median/cut/equal) 5*
MatGC	**MGC**	**MatGC5**	**MatGC6**	**MatGC8**	**MatGC9**	**MatGC12**
impretreat	1	1	1	1	1	1
imsodis	0	0	0	0	0	0
imcut	0	0	1	1	1	1
draw	0	0	0	0	0	0
wincov	0	0	0	0	0	0
xcovi						
xcovf						
ycovi						
ycovf						
meanfilt	1	1	1	1	1	1
tune	1	1	1	1	1	1
pix	0,004	0,004	0,004	0,004	0,004	0,004
cutxi	220	220	220	220	220	220
cutxf	860	860	860	860	860	860
cutyi	1	1	1	1	1	1
cutyf	1556	1556	1556	1556	1556	1556
Pretreatment parameters						
cf	3	3	3	3	3	3
npt	1	1	1	1	1	5
highcut	none	none	none	225	250	250
reduceh	none	none	none	125	220	220
lowcut	none	none	none	0	0	0
reducel	none	none	none	0	0	0
Centroid gradient filter definition						
pascov	5	5	5	5	5	5
Rinti	10	10	10	10	10	10
Rintf	40	40	40	40	40	40
dR	3	3	2	2	2	2
pasR	3	3	2	2	2	2
pic filters size definition						
dzmn=…….*pascov;	2	2	2	2	2	2
dzmx=….*pascov;	2	2	2	2	2	2
pdcm	1,5	1,5	1,5	1,5	1,5	1,5
minneighb	none	none	none	0	2	2
thresholds definition						
tpicpot	115	190	25	80	80	70
tmoypic	0	0	0	0	0	0
tmaxpic	0	50	7	10	27	25
Results						
NbC Vi video/eye	76	76	76	76	76	
NbC Iv image/eye	11	11	11	11	11	
NbC Iv edge/soft	10	10	10	10	10	
NbC exp. img/soft	55	55	55	55	55	
NbC announced	37	35	57	53	53	53
CD	31	33	40	42	not better	36
NCD	6	2	17	11		
Missing C	24	22	15	12		
% pop detect/eye	67	64	104	**96**		
% accurate detection	84	94	70	**79**		68
% pop tot existing	56	53	86	**80**		
image size (nb pixels)	995200	995200	995200	995200	995200	995200
Processing time (minutes)	130	130	130	130	130	130
Processing speed **(min/100000 pix)**	13,1	13,1	13,1	13,1	13,1	13,1

Image/ROI/treatment definition

image	**2105lmedian.jpg**			**2105amed.jpg**				
band	Green	Green	Green	Green	Green	Green	Green	Green
pre-treatment	cut/median/equal			cut/median/equal				
MatGC	**MatGC10**	**MatGC10**	**MatGC10**	**MatGC14**	**MatGC14**	**MatGC14**	**MatGC14**	**MatGC14**
impretreat	1	1	1	1	1	1	1	1
imsodis	0	0	0	0	0	0	0	0
imcut	1	1	1	1	1	1	1	1
draw	0	0	0	0	0	0	0	0
wincov	0	0	0	0	0	0	0	0
xcovi								
xcovf								
ycovi								
ycovf								
meanfilt	1	1	1	1	1	1	1	1
tune	1	1	1	1	1	1	1	1
pix	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004
cutxi	100	100	100	300	300	300	300	300
cutxf	700	700	700	750	750	750	750	750
cutyi	1000	1000	1000	1500	1500	1500	1500	1500
cutyf	2000	2000	2000	2500	2500	2500	2500	2500
Pretreatment parameters								
cf	3	3	3	3	3	3	3	3
npt	1	1	1	1	1	1	1	1
highcut	155	155	165	250	250	250	250	250
reduceh	75	125	90	200	200	200	200	200
lowcut	0	0	0	0	0	0	0	0
reducel	0	0	0	0	0	0	0	0
Centroid gradient filter definition								
pascov	4	4	4	5	5	5	5	5
Rinti	4	4	4	10	10	10	10	10
Rintf	24	24	24	30	30	30	30	30
dR	2	2	2	2	2	2	2	2
pasR	2	2	2	2	2	2	2	2
pic filters size definition								
dzmn=…….*pascov;	2	2	2	2	2	2	2	2
dzmx=….*pascov;	2	2	2	2	2	2	2	2
pdcm	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5
minneighb	2	2	-1	2	1	1	1	1
thresholds definition								
tpicpot	35	35	65	80	80	70	70	70
tmoypic	0	0	0	0	0	0	0	0
tmaxpic	0	0	15	27	27	27	15	10
Results								
NbC Vi video/eye								
NbC Iv image/eye								
NbC Iv edge/soft								
NbC exp. img/soft	43	43	43	25	25	25	25	25
NbC announced	34	45	46	20	20	22	31	38
CD	14	21	23	14	15	15	18	21
NCD	20	24	23	6	5	7	13	17
Missing C	29	22	20	11	10	10	7	4
% pop detect/eye	79	105	**107**	80	80	**88**	124	152
% accurate detection	41	47	**50**	70	75	**68**	58	55
% pop tot existing								
image size (nb pixels)	600000	600000	600000	450000	450000	450000	450000	450000
Processing time (minutes)								
Processing speed	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

(min/100000 pix)

Image/ROI/treatment definition

image	**2105amed.jpg**				**2105amed.jpg**			
band	Green	Green	Green	Green	Green	Green	Green	Green
pre-treatment	cut/median/equal				cut/median/equal			
MatGC	**MatGC15**	**MatGC15**	**MatGC15**	**MatGC15**	**MatGC16**	**MatGC16**	**MatGC16**	**MatGC16**
impretreat	1	1	1	1	1	1	1	1
imsodis	0	0	0	0	0	0	0	0
imcut	1	1	1	1	1	1	1	1
draw	0	0	0	0	0	0	0	0
wincov	0	0	0	0	0	0	0	0
xcovi								
xcovf								
ycovi								
ycovf								
meanfilt	1	1	1	1	1	1	1	1
tune	1	1	1	1	1	1	1	1
pix	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004
cutxi	300	300	300	300	300	300	300	300
cutxf	725	725	725	725	725	725	725	725
cutyi	500	500	500	500	2500	2500	2500	2500
cutyf	1500	1500	1500	1500	4000	4000	4000	4000
Pretreatment parameters								
cf	3	3	3	3	3	3	3	3
npt	1	1	1	1	1	1	1	1
highcut	250	250	250	250	250	250	250	250
reduceh	200	200	200	200	200	200	200	200
lowcut	0	0	0	0	0	0	0	0
reducel	0	0	0	0	0	0	0	0
Centroid gradient filter definition								
pascov	5	5	5	5	5	5	5	5
Rinti	10	10	10	10	10	10	10	10
Rintf	30	30	30	30	30	30	30	30
dR	2	2	2	2	2	2	2	2
pasR	2	2	2	2	2	2	2	2
pic filters size definition								
dzmn=…….*pascov;	2	2	2	2	2	2	2	2
dzmx=….*pascov;	2	2	2	2	2	2	2	2
pdcm	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5
minneighb	1	1	1	2	1	1	2	2
thresholds definition								
tpicpot	70	80	70	70	70	80	80	80
tmoypic	0	0	0	0	0	0	0	0
tmaxpic	10	10	27	10	27	27	27	32
Results								
NbC Vi video/eye								
NbC Iv image/eye								
NbC Iv edge/soft								
NbC exp. img/soft	28	28	28	28	8	8	8	8
NbC announced	49	37	25	48	26	13	13	10
CD	19	16	15	19	1	0	0	0
NCD	30	21	10	29	25	13	13	10
Missing C	9	12	13	9	7	8	8	8
% pop detect/eye	175	132	**89**	171	325	163	163	**125**
% accurate detection	39	43	**60**	40	4	0	0	**0**
% pop tot existing								
image size (nb pixels)	425000	425000	425000	425000	637500	637500	637500	637500
Processing time (minutes)								
Processing speed	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

(min/100000 pix)

Image/ROI/treatment definition

image	**2105c2med.jpg**							
band	Green	Green	Green	Green				
pre-treatment	cut/median/equal							
MatGC	**MatGC17**	**MatGC17**	**MatGC17**	**MatGC17**				
impretreat	1	1	1	1				
imsodis	0	0	0	0				
imcut	1	1	1	1				
draw	0	0	0	0				
wincov	0	0	0	0				
xcovi								
xcovf								
ycovi								
ycovf								
meanfilt	1	1	1	1				
tune	1	1	1	1				
pix	0,004	0,004	0,004	0,004				
cutxi	200	200	200	200				
cutxf	800	800	800	800				
cutyi	500	500	500	500				
cutyf	1500	1500	1500	1500				

Pretreatment parameters

cf	3	3	3	3				
npt	1	1	1	1				
highcut	165	165	165	165				
reduceh	90	90	90	90				
lowcut	0	0	0	0				
reducel	0	0	0	0				

Centroid gradient filter definition

pascov	5	5	5	5				
Rinti	10	5	5	5				
Rintf	30	20	20	20				
dR	2	2	2	2				
pasR	2	2	2	2				

pic filters size definition

dzmn=…….*pascov;	2	2	2	2				
dzmx=….*pascov;	2	2	2	2				
pdcm	1,5	1,5	1,5	1,5				
minneighb	2	2	0	1				

thresholds definition

tpicpot	70	80	80	75				
tmoypic	0	0	0	0				
tmaxpic	10	10	15	15				

Results

NbC Vi video/eye								
NbC Iv image/eye								
NbC Iv edge/soft								
NbC exp. img/soft	70	70	70	70				
NbC announced	137	194	74	89				
CD	50	50	35	37				
NCD	87	144	39	52				
Missing C	20	20	35	33				
% pop detect/eye	196	277	**106**	127				
% accurate detection	36	26	**47**	42				
% pop tot existing								
image size (nb pixels)	600000	600000	600000	600000				
Processing time (minutes)	47	24	24	24				
Processing speed	7,8	4,0	4,0	4,0				

(min/100000 pix)