Automated detection of fossils in microscope images

Andreas Richter

26.07.2016

Agenda

Overview of Image Analysis and Task to Solve

Matlab: PST+Hough-Transforms

OpenCV: Thresholding, Edge-Images &
FindContours

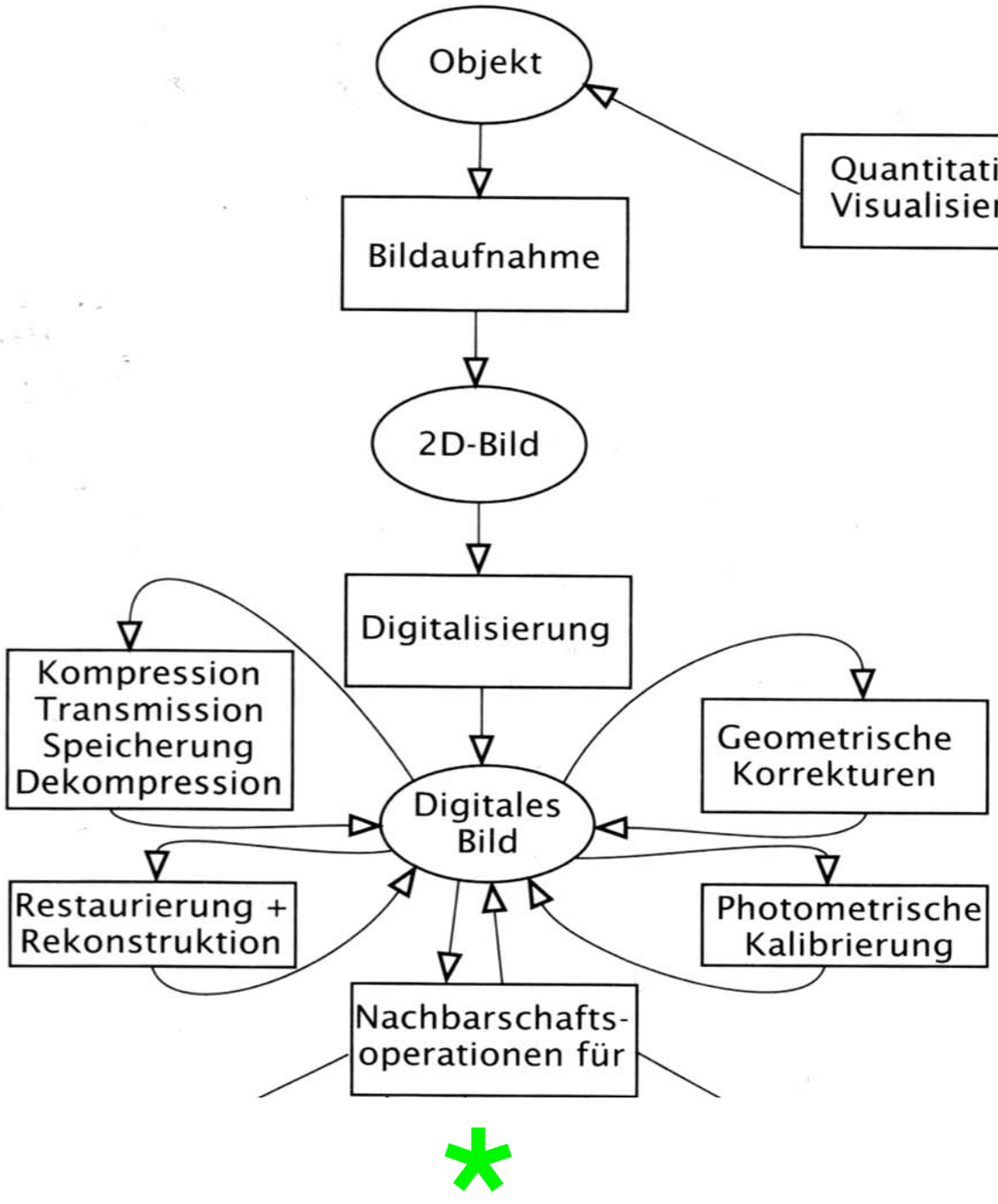
OpenCV: Haar-Classifiers

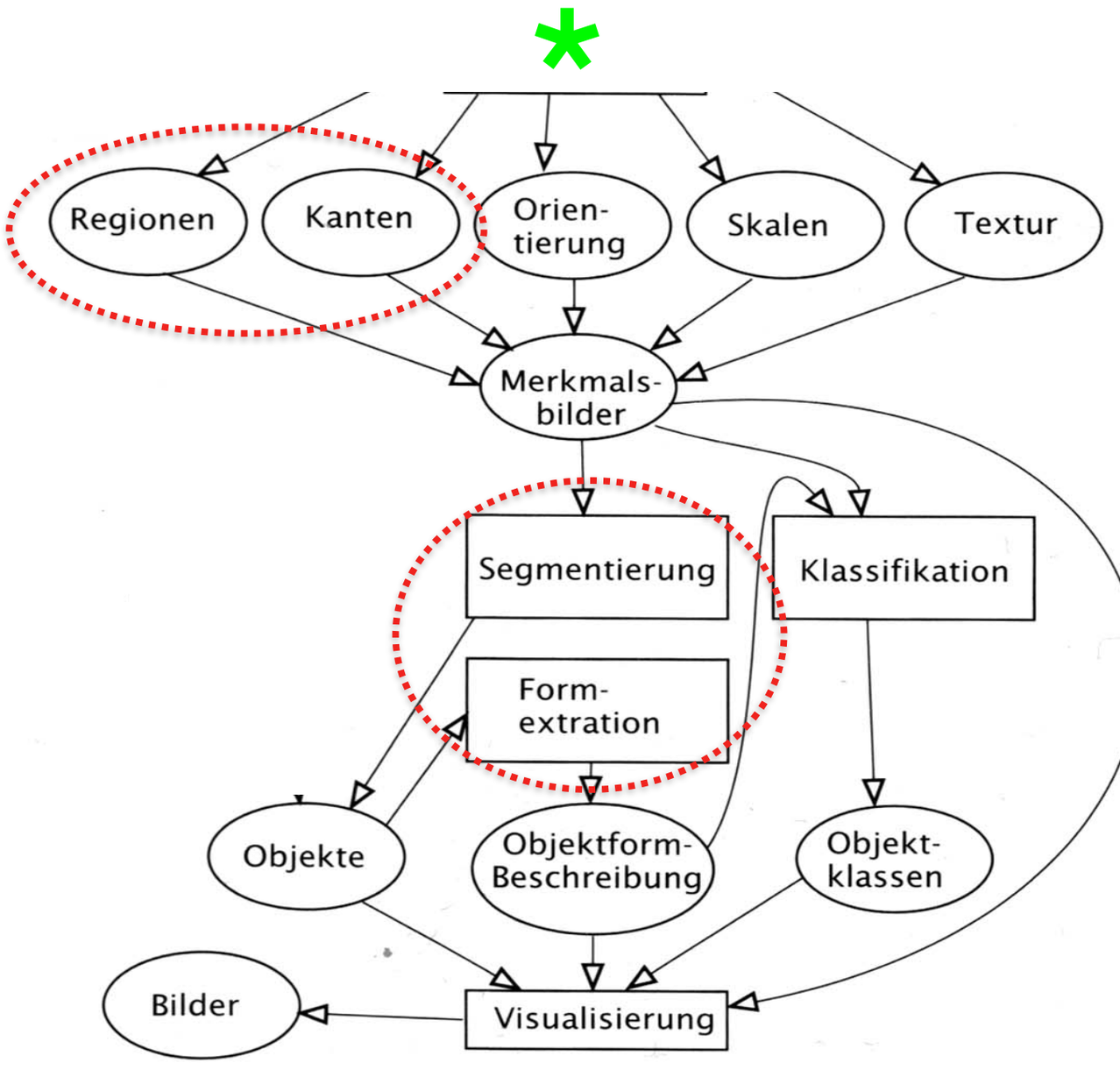
Outlook

How many objects are there?

How can their shapes be extracted to vectors?

Intro: Overview of Image Analysis

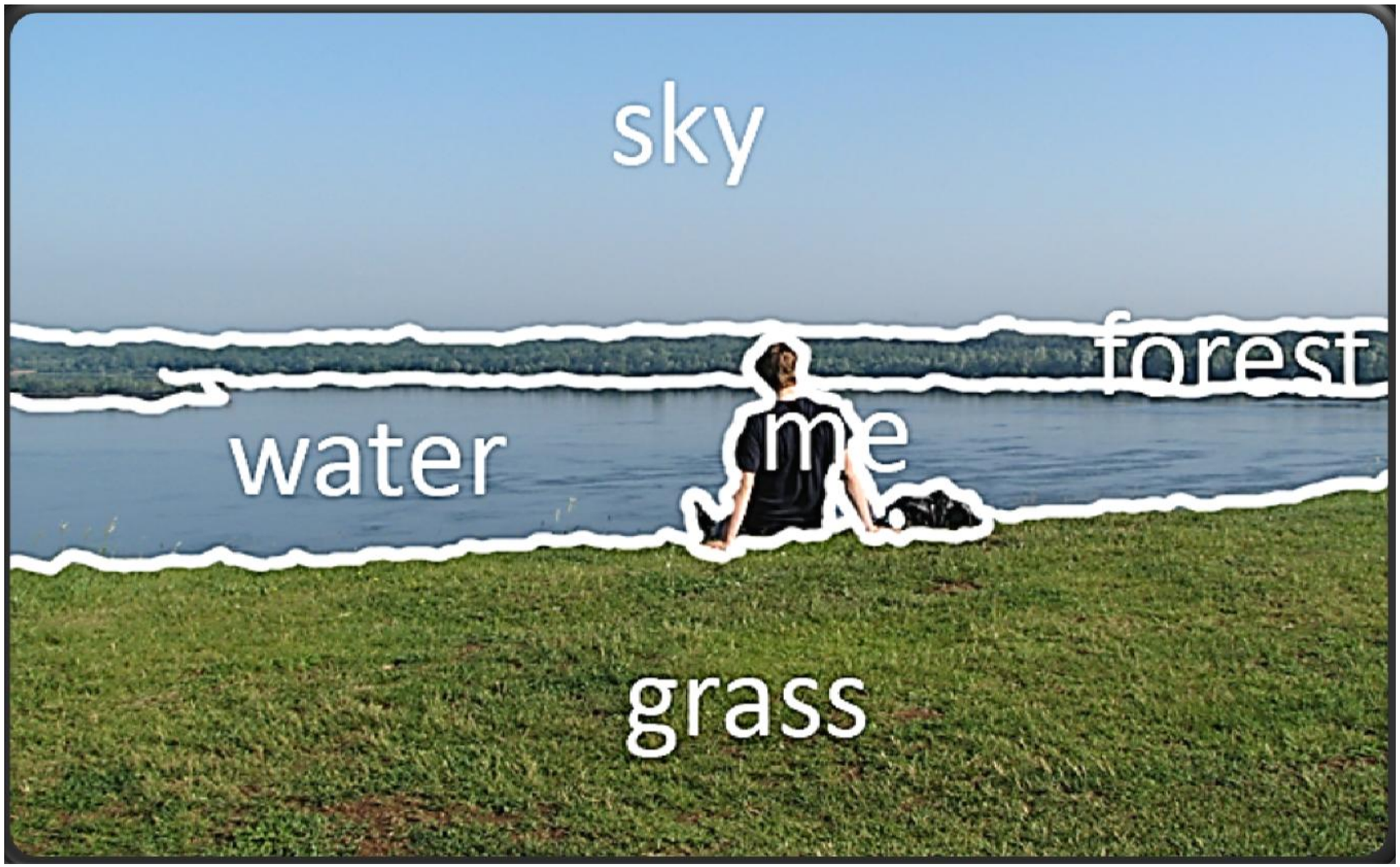




Quelle: http://www2.in.tu-clausthal.de/~reuter/ausarbeitung/Ralf_Bruder_-_Digitale_Bildverarbeitung.pdf

Machine vision

Segmentationalgorithm: „MARKOV-RANDOM-FIELDS with GIBBS-POTENTIAL“



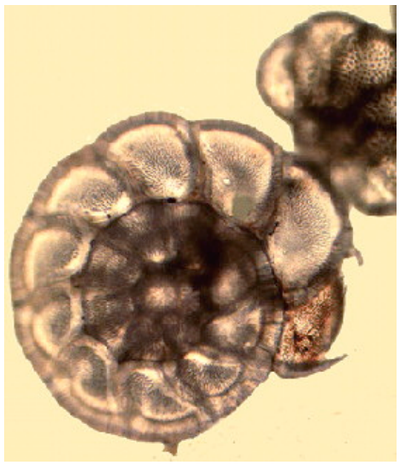
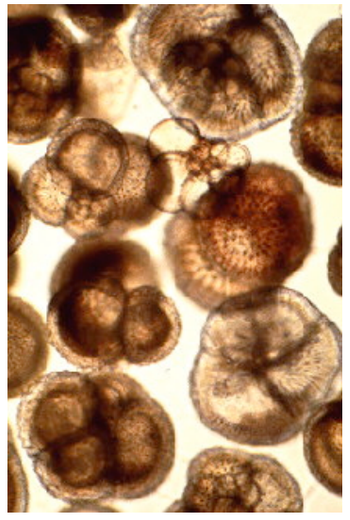
Source: https://www.htw-dresden.de

Task to Solve and trailed paths

Automated detection of fossils in microscope images

Foraminifera



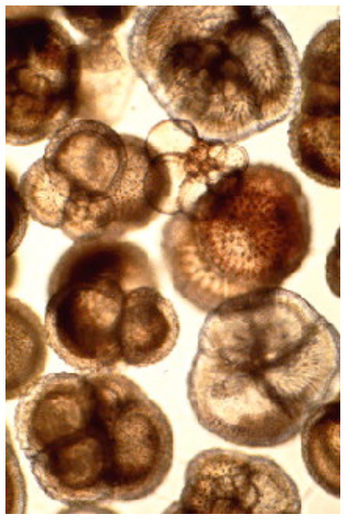




Source: http://www.microscopy-uk.org.uk/mag/indexmag.html?http://www.microscopy-uk.org.uk/mag/artnov98/bdforam3.html

Task to Solve and trailed paths

- **Foraminifera** are marine microorganisms, unicellular creatures
- **SIZE**: most species are between 0.2 und 0.5 mm in size (smallest 40 micrometers, biggest 20 cm)
- **AGE**: can get some month or up to some years old
- **TYPES**: bentic(deep sea), planctonic, freshwater, …
- **SPECIES:** today about 10.000 recent and 40.000 fossil species

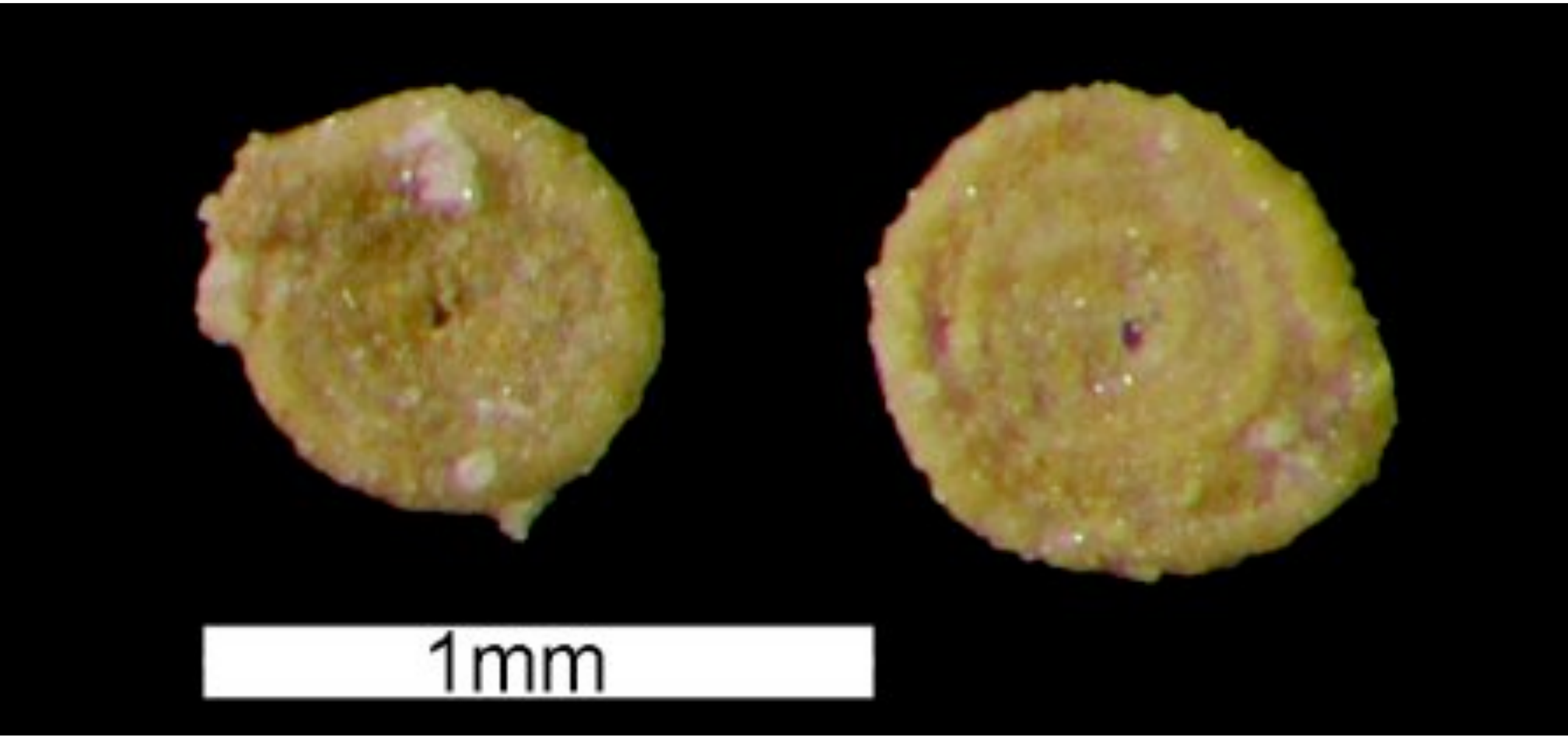


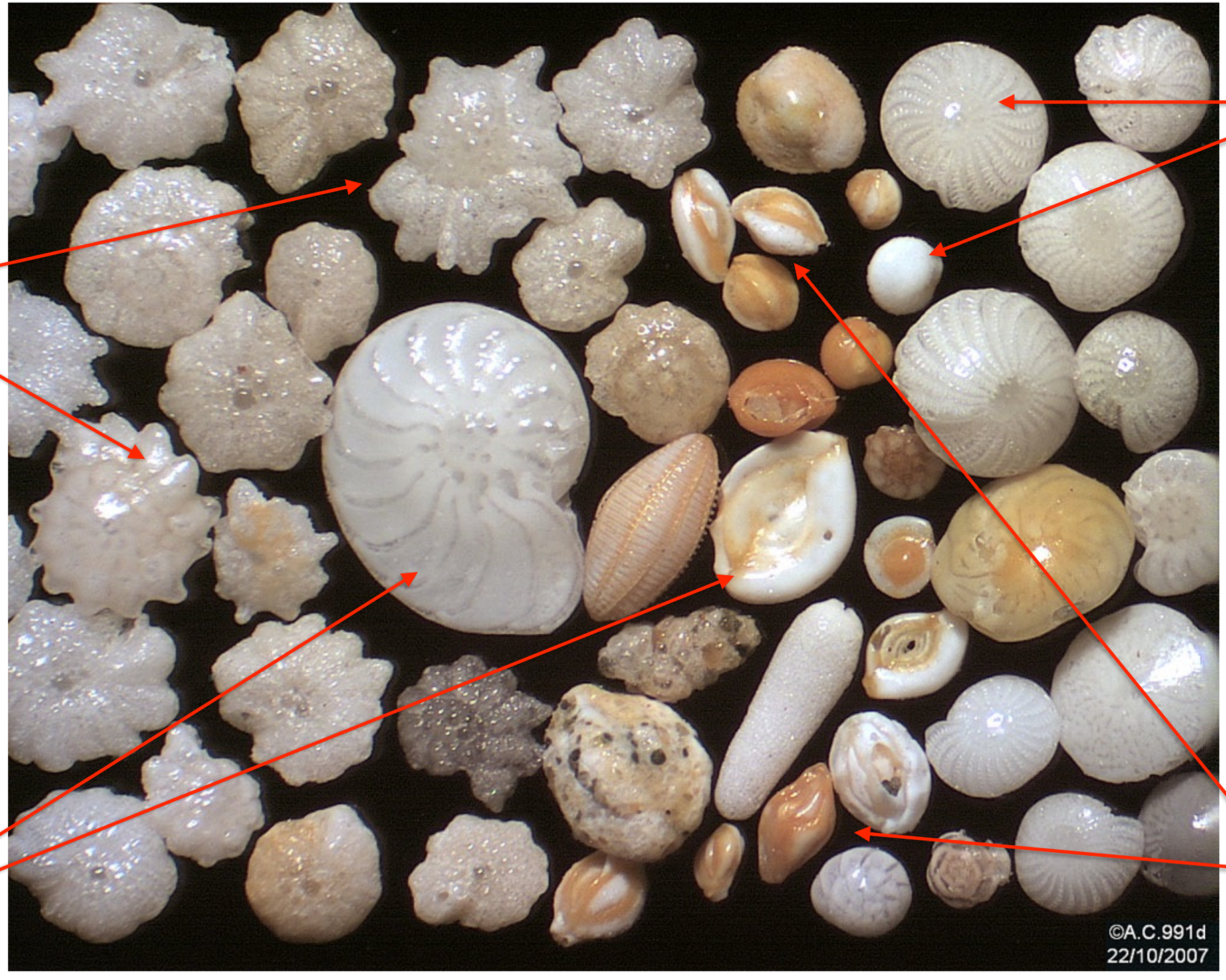

Source: http://www.microscopy-uk.org.uk/mag/indexmag.html?http://www.microscopy-uk.org.uk/mag/artnov98/bdforam3.html

Source: http://haraldmaage.de.tl/Foraminiferen

Task to Solve and trailed paths



Task to Solve and trailed paths



How many objects are there?

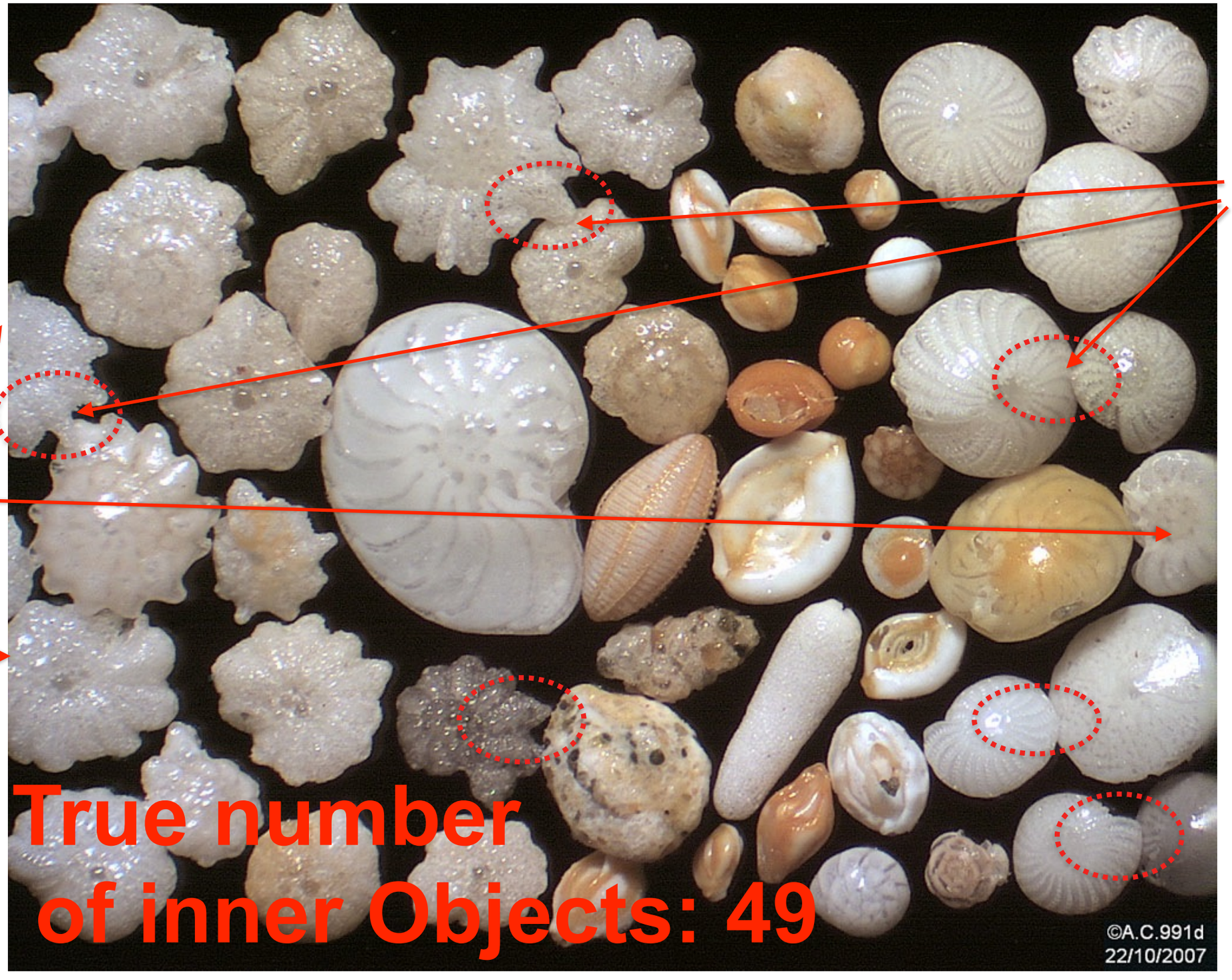
How can their shapes be extracted to vectors ?

Task to Solve and trailed paths

How many objects are there?

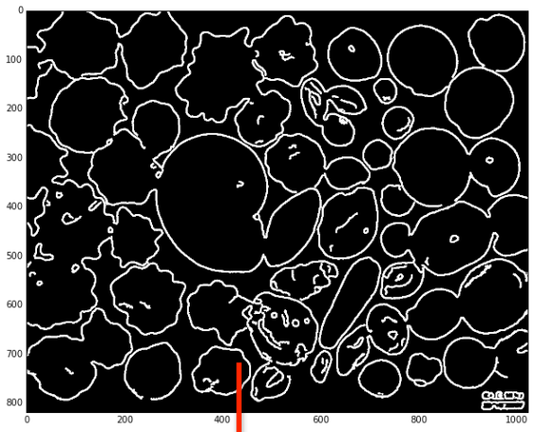
How can their shapes be extracted to vectors ?

Task to Solve and trailed paths

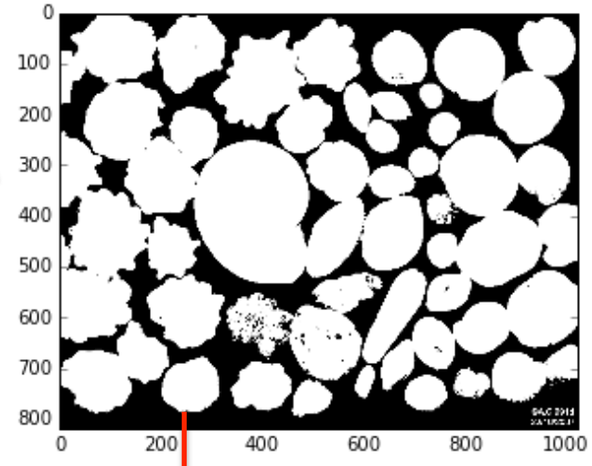


How to detect the objects and shapes?

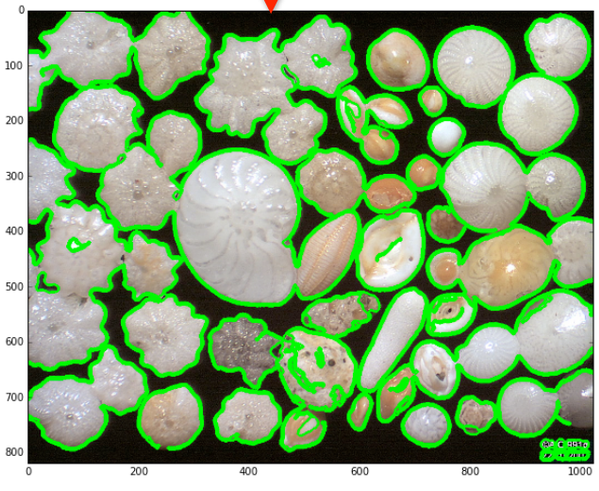
Edge Image



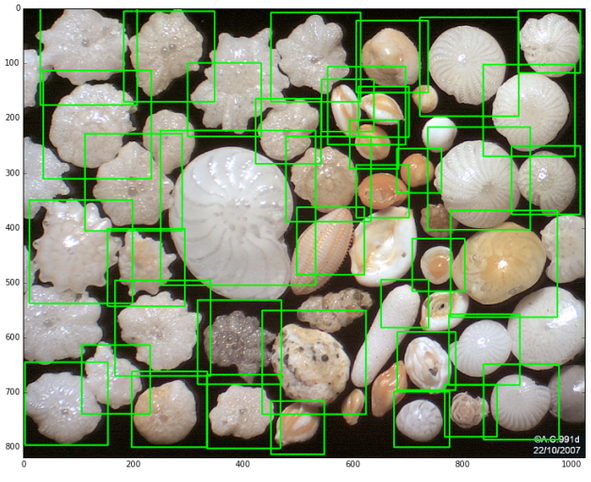
Threshhold Image



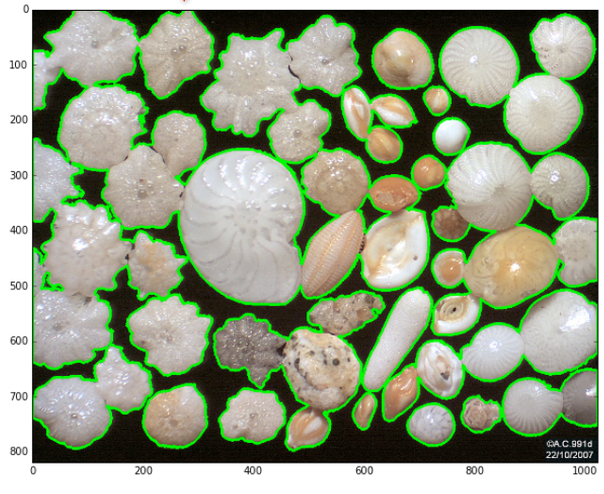
Find Contours



Find Objects directly



Find Contours



How to detect the objects and shapes?

basically 3 different Approaches tried

PST Function &
Hough -Transforms

OpenCV
Haar-Classifier

Region
Growing

Thresholding or Canny/LoG Filters
+
FindContours - Function

Agenda

Overview of Image Analysis and Task to Solve

Matlab: PST+Hough-Transforms

OpenCV: Thresholding, Edge-Images &
FindContours

OpenCV: Haar-Classifiers

Outlook

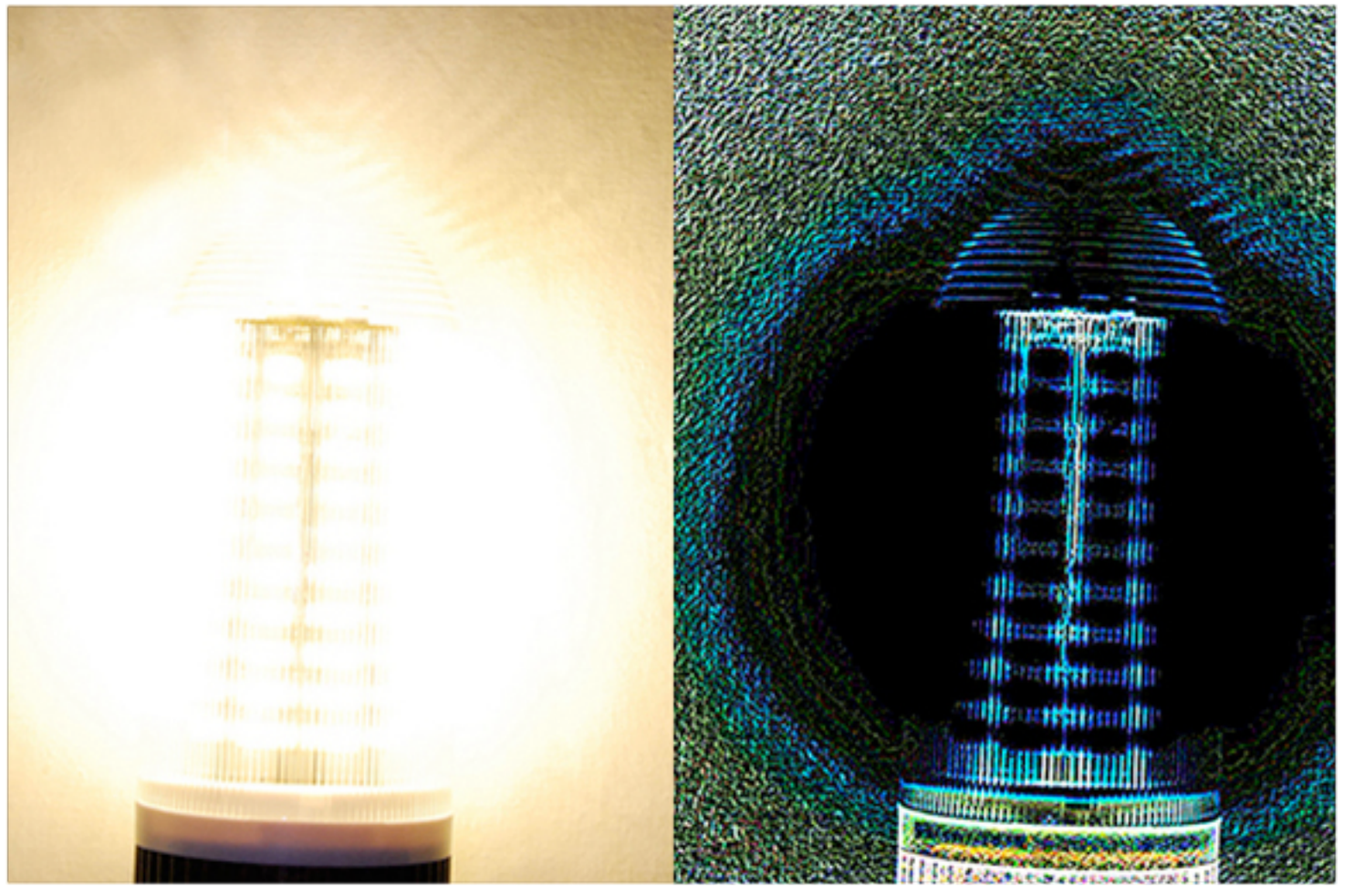
PST + Hough-Transforms

Phase stretch transform (PST)

- Frequency based Algorithm for feature detection and classification

- related to Fourier Transforms

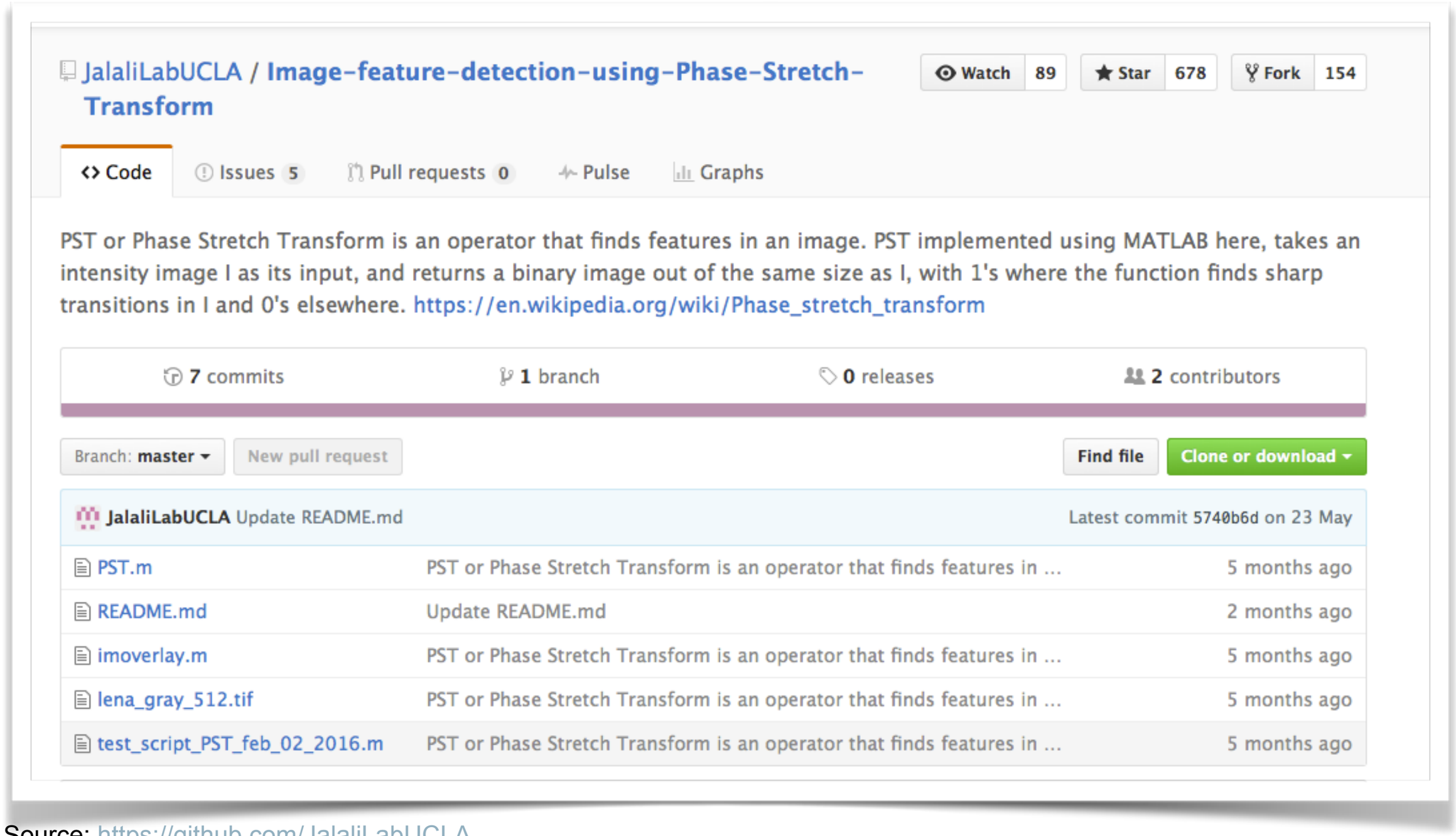
- can be used for Edge Detection but tuning the parameters is an art



Source: http://www.pcworld.com

PST + Hough-Transforms

github Archive:



Source: https://github.com/JalaliLabUCLA

PST + Hough-Transforms

Matlab Code :

```
handles.Phase_strength=0.48;
handles.Warp_strength=32.14;


handles.Thresh_min=-1;

handles.Thresh_max=0.02;


Morph_flag = 1 ;
```

```
%%  Tune PST parameters
% PST Kernel Phase Strength
% PST Kernel Warp Strength

% Thresholding parameters (for post processing)
% minimum Threshold ...
%(a number between 0 and -1)
% maximum Threshold ...
%(a number between 0 and 1)

% choose to compute the analog or digital edge
%  Morph_flag=0 to compute analog ...
%edge and Morph_flag=1 to compute digital edge.
```

```
%% Apply PST and find Features (sharp transitions)
[Edge PST_Kernel]= PST(Image_orig,handles,Morph_flag);
```


©A.C.991d

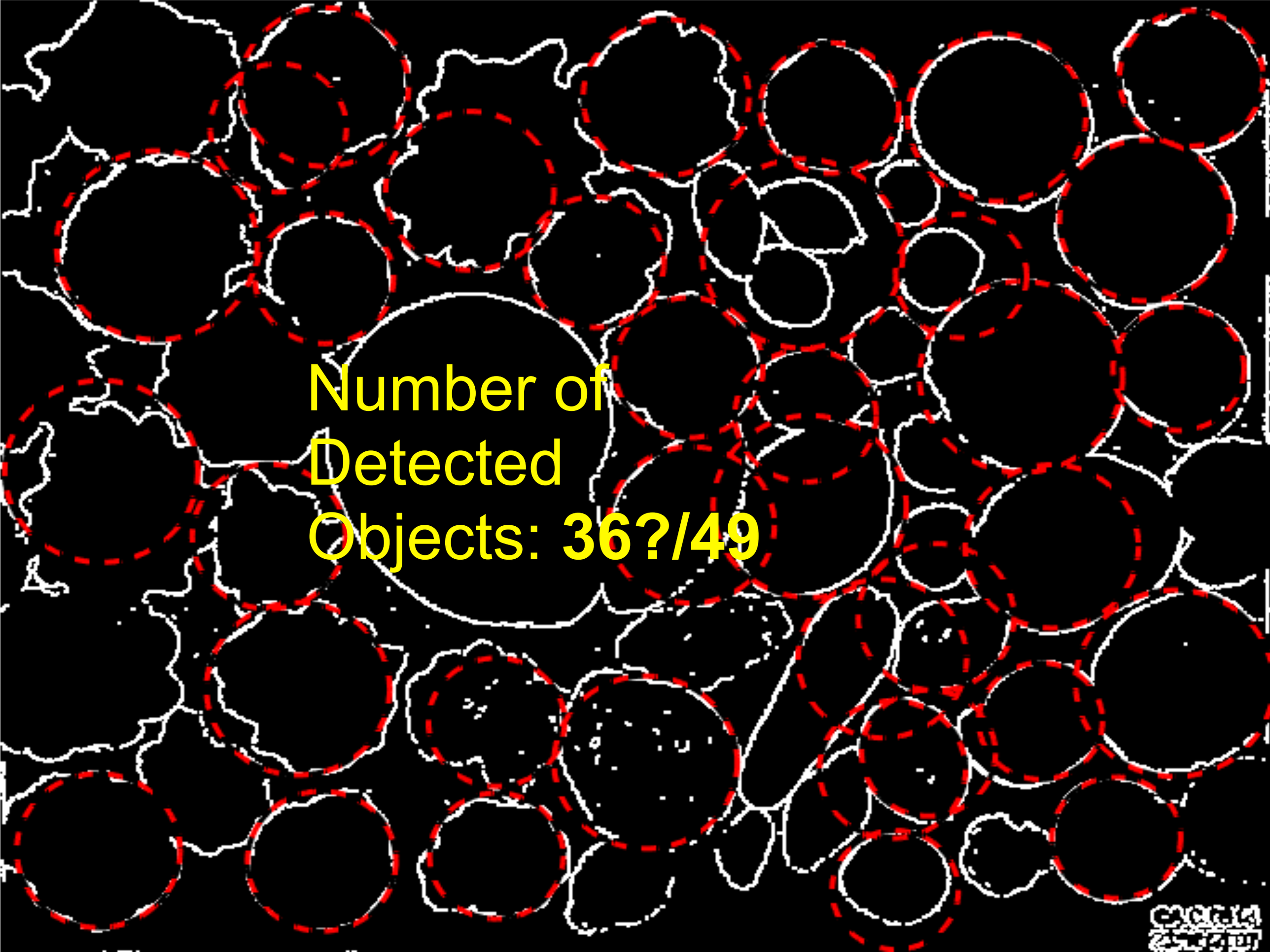
PST

Edge Image

PST + Hough-Transforms

Matlab Code :

```
%% Perform Hough-Tranform with binary Image and find Circular Structures

[centers,radii] = imfindcircles(Edge_enh,[50 150],...
                        'Method','TwoStage',...
                        'ObjectPolarity','Bright',...
                        'Sensitivity',0.92,...
                        'EdgeThreshold',0.20);

num = length(centers);
disp(['Number of Circles: ',num2str(num)]);
```


Number of
Detected
Objects: 36?/49

PST + Hough-Transforms

Fazit: ☹

- Number of Objects very inaccurate
- Problems with overlap and double counting
- No exact Shapes!
- possible Alternatives :

  —>Generalized Hough Transform, but did no better job

Agenda

Overview of Image Analysis and Task to Solve

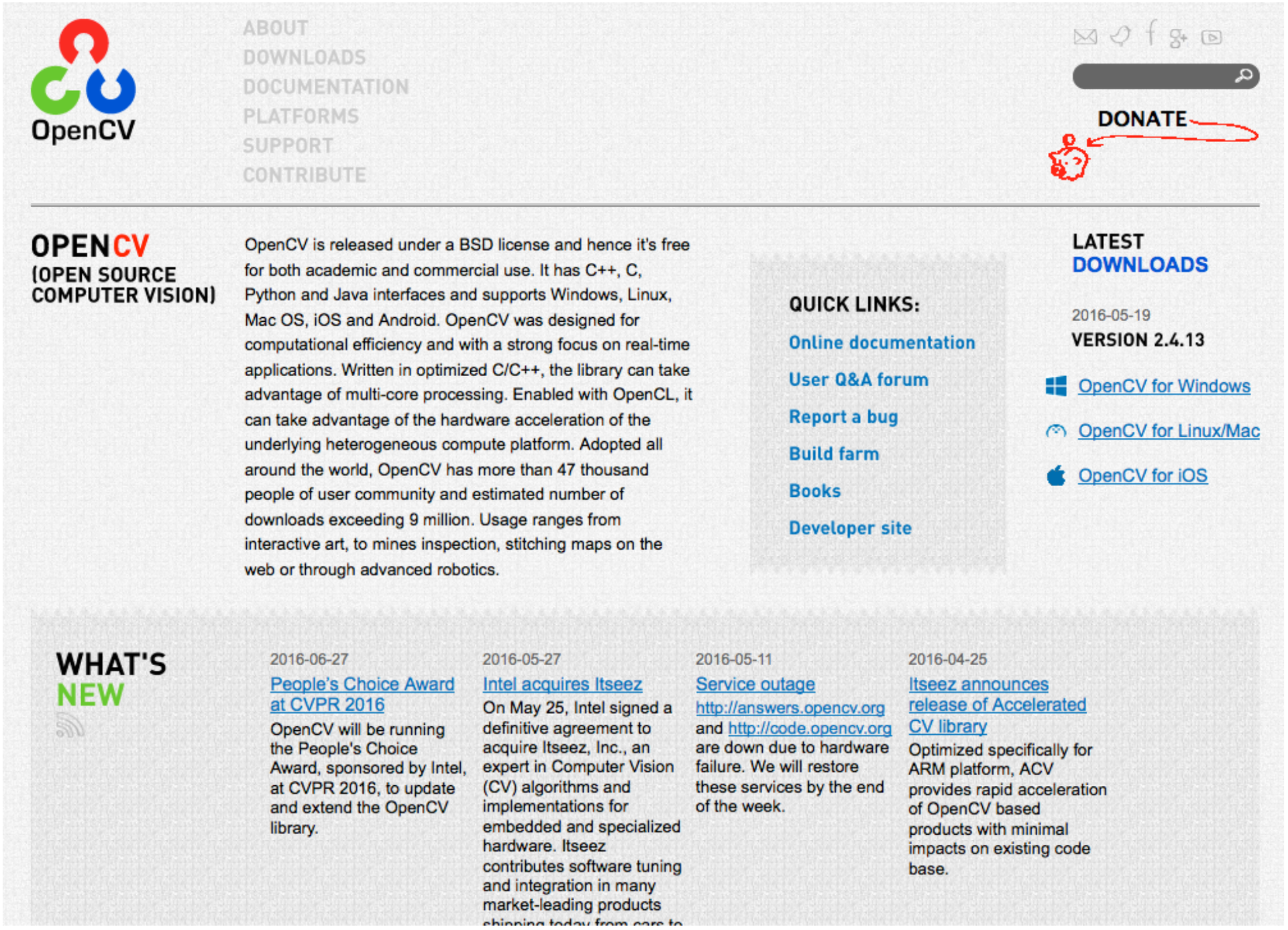
Matlab: PST+Hough-Transforms

OpenCV: Thresholding, Edge-Images & FindContours

OpenCV: Haar-Classifiers

Outlook

OpenCV - Open Computer Vision



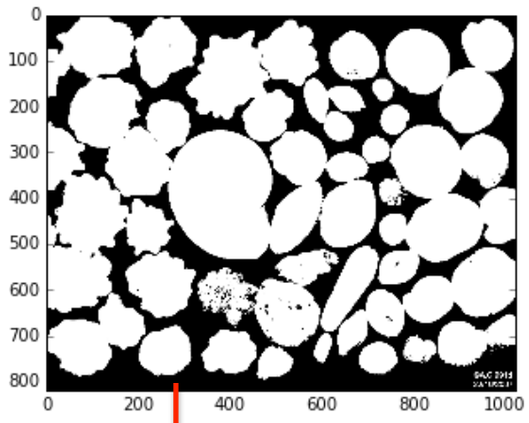
Source: http://opencv.org/

used with

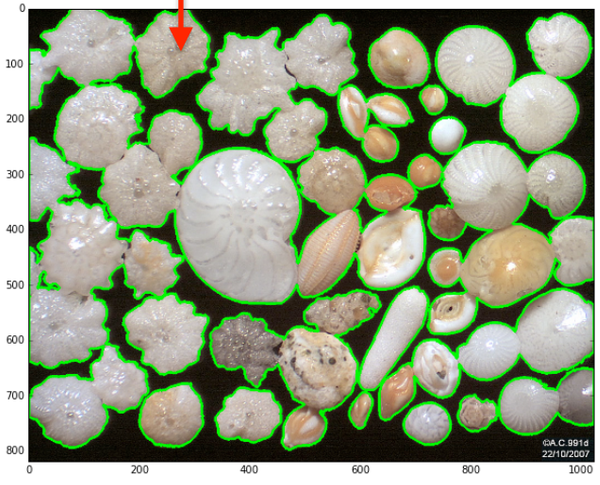
- C, C++
- Python
- Matlab (mex-files)

How to detect the shapes?

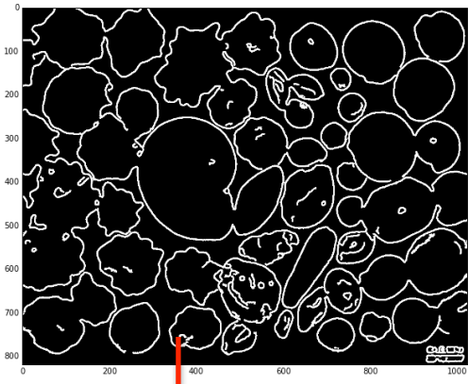
Threshhold Image



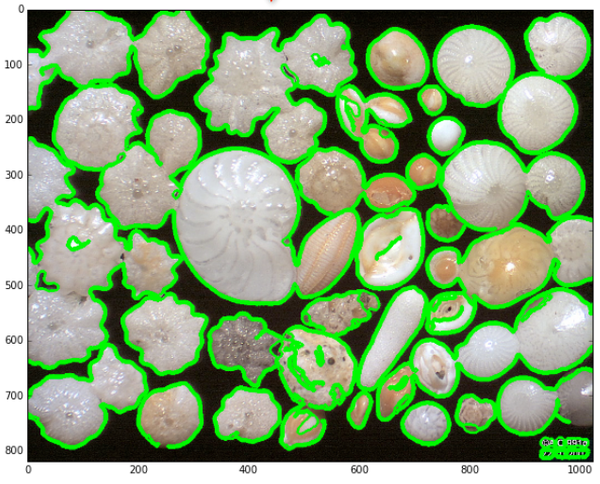
Find Contours



Edge Image



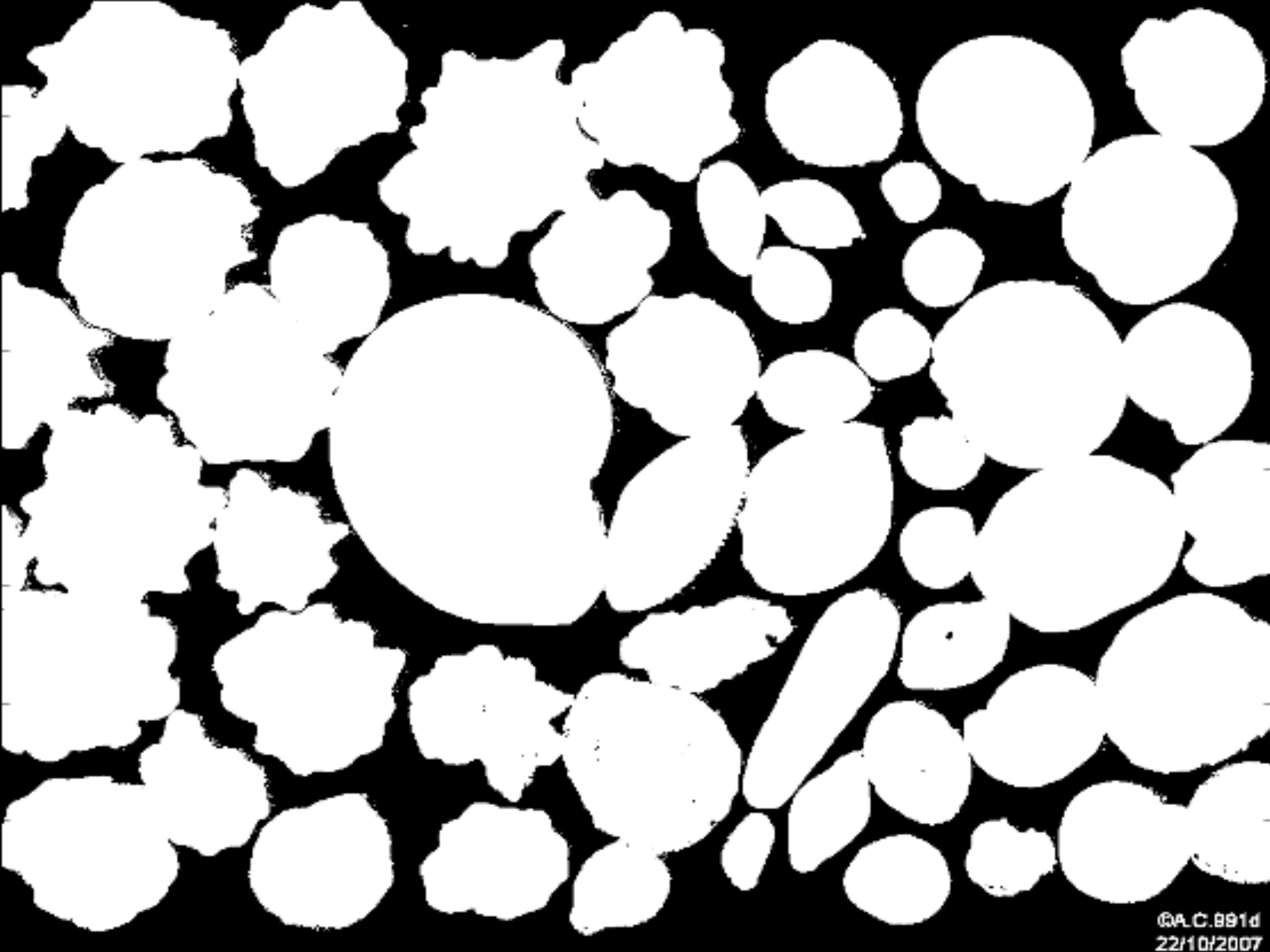
Find Contours



Thresholding & Edge-Images

Python Code :

```
import cv2
import matplotlib.pyplot as plt                          # Import modules
%matplotlib inline

IMG= cv2.imread('Ngapali.jpg')                           # read the image
imgray = cv2.cvtColor(IMG,cv2.COLOR_RGB2GRAY)            # convert to grayscale

ret,thresh = cv2.threshold(imgray,50,1,0)                # apply Threshold > 50

plt.figure(figsize=(10,10))                              # show the result
plt.imshow(thresh,cmap='gray')
plt.show()
```



Thresholding & Edge-Images

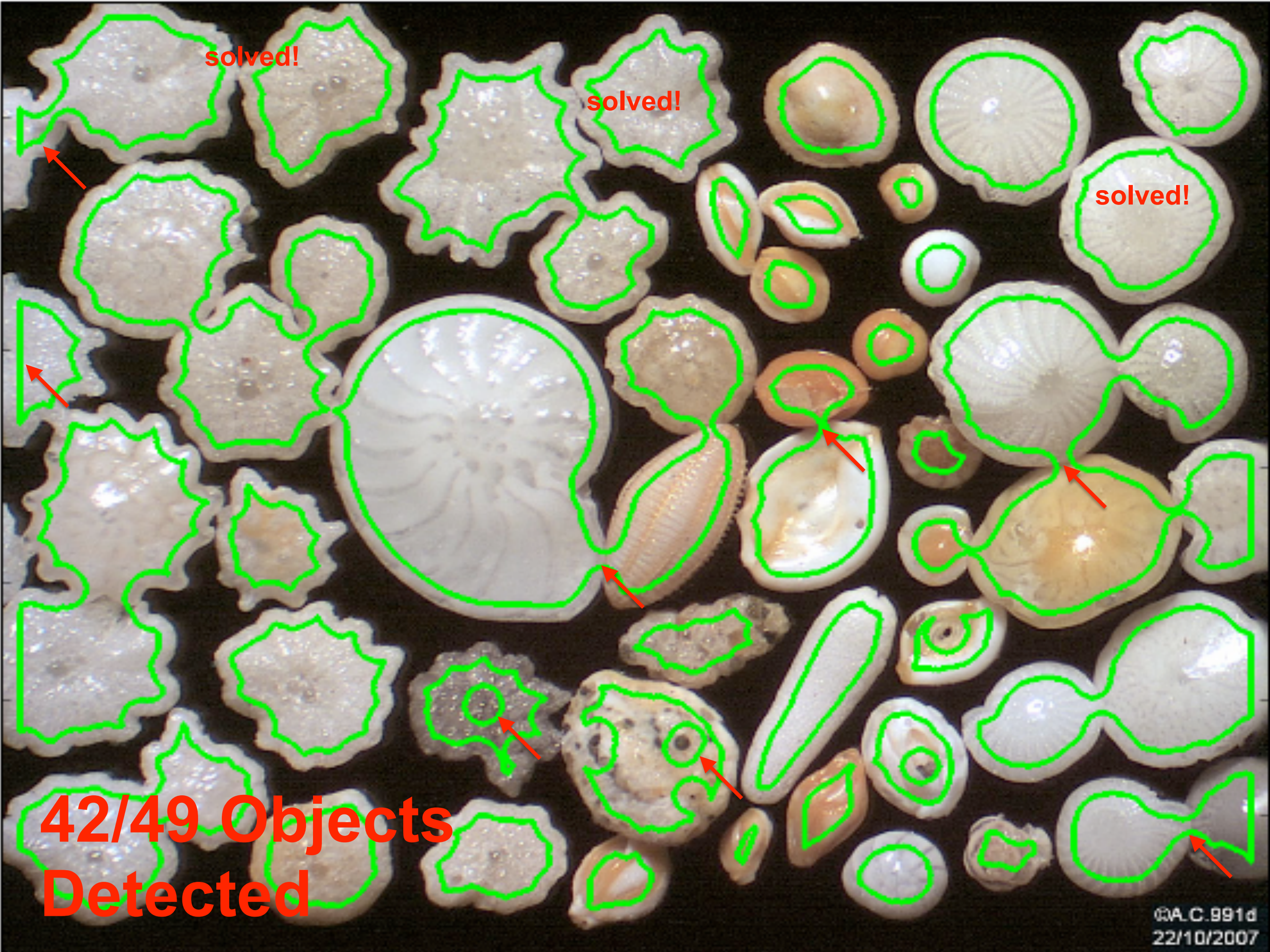
Python Code :

```
contours, hierarchy = cv2.findContours(thresh,cv2.RETR_CCOMP,cv2.CHAIN_APPROX_SIMPLE)

IMG2=skio.imread('Ngapali.jpg')                          # Read Image again
cv2.drawContours(IMG2, contours, -1, (0,255,0), 3)       # draw the contours over the image

plt.figure(figsize=(10,20))                              # show the result
plt.imshow(IMG2)
plt.show()
```


Overlap!
21/49 Objects Detected
©A.C.991d
22/10/2007

Thresholding & Edge-Images

… and after some morphologic enhancement
Erosion, Dilation, Opening, Closing…


solved!
solved!
solved!
42/49 Objects
Detected
©A.C.991d
22/10/2007

Thresholding & Edge-Images

Problems: 

- overlapping Objects become one big contour
- exact number of Objects not arbitrable
- contours consist of several parts or overlap
- small contours can be filtered out

Thresholding & Edge-Images

Python Code :

```
import cv2
from skimage import feature
import matplotlib.pyplot as plt                    # Import modules
%matplotlib inline

IMG= cv2.imread('Ngapali.jpg')                     # read the image
imgray = cv2.cvtColor(IMG,cv2.COLOR_RGB2GRAY)      # convert to grayscale

edges2 = feature.canny(imgray, sigma=3)            # Edge-filtering by canny operator

plt.figure(figsize=(10,10))                        # show results
plt.imshow(edges2,cmap='gray')
plt.show()
```

Canny Filter


solved!
Overlap
Mistake
Mistake
54/49 Objects Detected
Mistake

Threshholding & Edge-Images

Fazit: 

- Number of Objects still very inaccurate
- Morphologic Enhancement can help, but can also destroy smaller objects
- partially exact Shapes, but not in a single contours

possible Alternatives :

—>OpenCV Haar Classifier

Agenda

Overview of Image Analysis and Task to Solve

Matlab: PST+Hough-Transforms

OpenCV: Thresholding, Edge-Images &
FindContours

OpenCV: Haar-Classifiers

Outlook

OpenCV - Open Computer Vision

coding robin

TRAIN YOUR OWN OPENCV HAAR CLASSIFIER

22 July 2013, posted by Thorsten Ball

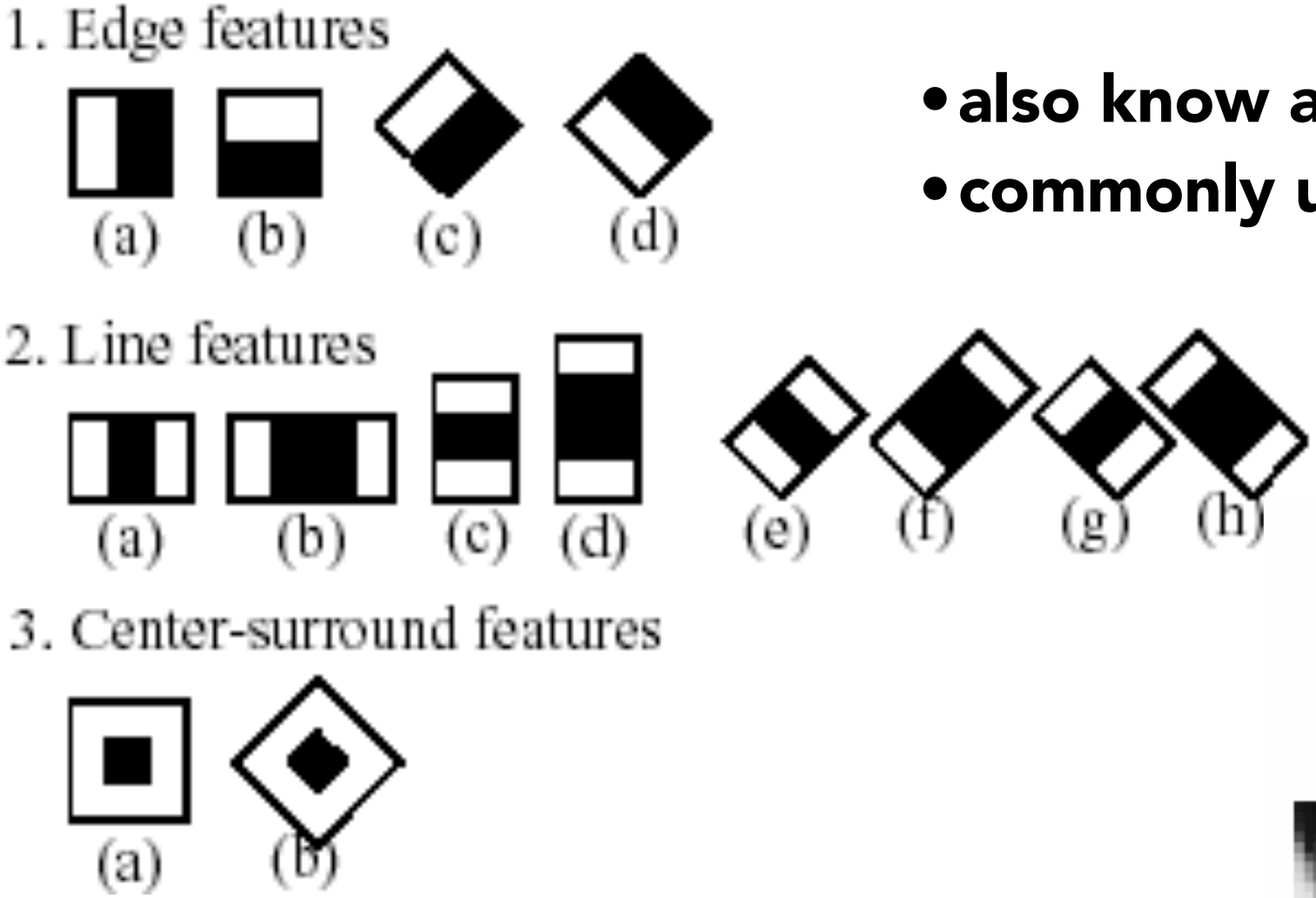
Open this page, allow it to access your webcam and see your face getting recognized by your browser using JavaScript and OpenCV, an "open source computer vision library". That's pretty cool! But recognizing faces in images is not something terribly new and exciting. Wouldn't it be great if we could tell OpenCV to recognize something of our choice, something that is not a face? Let's say... a banana?

That is totally possible! What we need in order to do that is called a "cascade classifier for Haar features" to point OpenCV at. A cascade classifier basically tells OpenCV what to look for in images. In the example above a classifier for face features was being used. There are a lot of cascade classifiers floating around on the internet and you can easily find a different one and use it. But most of them are for recognizing faces, eyes, ears and mouths though and it would be great if we could tell OpenCV to recognize an object of our choice. We need a cascade classifier that tells OpenCV how to recognize a banana.

Source: http://coding-robin.de

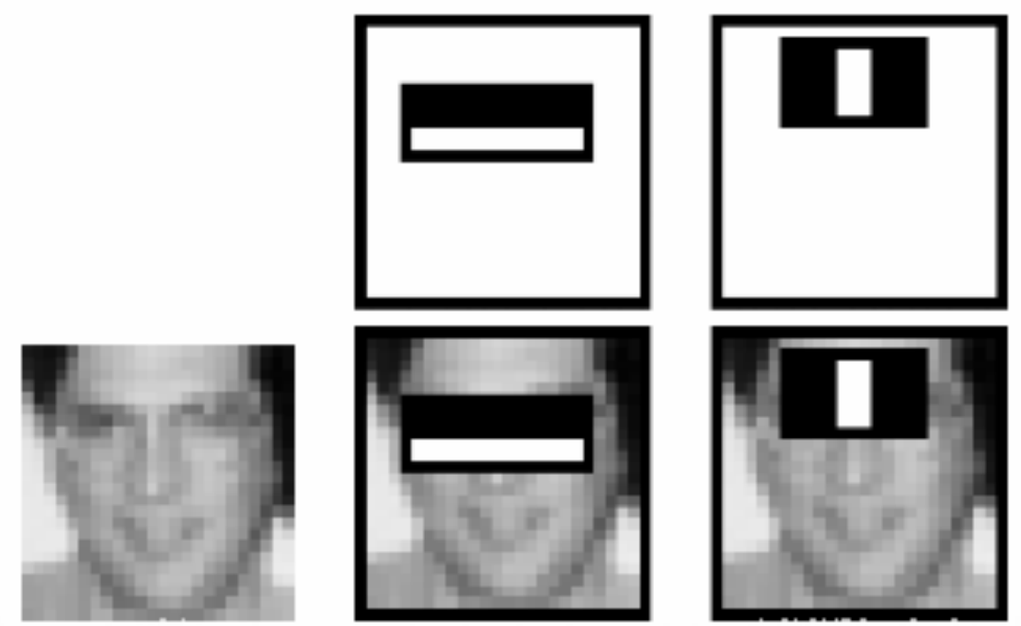
OpenCV - Open Computer Vision

How do Haar-Classifiers work?



- also know as Viola-Jones Algorithm
- commonly used for face detection

Source: https://kyamagu.github.io/mexopencv/



Source: http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0


40 Positiv
Images


20081228.jpg


20081229.jpg


20081230.jpg


20081231.jpg


82524190_8566
6cf553.jpg


3482524214_70fa
c76cc9.jpg


3482596154_f2f8e
f151c.jpg


3482599362_80dc
970c22.jpg


80901835_4e39
019df9.jpg


3480902247_feaf6
b2511.jpg


3481005279_3359
56fa01.jpg


3481087259_b2
b61784.jpg


82600480_763a
e8b7d1.jpg


3483467669_2f4e
4180cd.jpg


3483509611_105f
0905d8.jpg

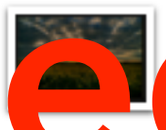

3483823415_29ea
986725.jpg

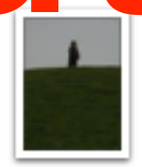

81133029_1174
153360.jpg

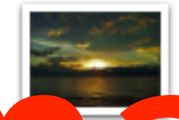

3481314527_54db
1e5ae3.jpg


3481316659_4489
bd4748.jpg


3481705642_a
70bfe2.jpg



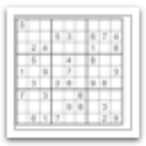



book_perspective.J
PG

boy_on_hill.jpg

calibration_setup.JP
G

calibration.JPG


sudoku15.JPG


sudoku16.JPG


sudoku17.JPG


sudoku18.JPG


climbing_1_small.j
pg

climbing_2_small.j
pg

crans_1_small.jpg

crans_2_small.jp

sudoku19.JPG

sudoku20.JPG

sudoku21.JPG

sudoku22.JPG

empire.jpg

fisherman.jpg

sudoku1.JPG

sudoku2.JPG

sudoku23.JPG

sudoku24.JPG

sunset_tree.jpg

turningtorso1.jpg

sudoku3.JPG

sudoku4.JPG

sudoku5.JPG

sudoku6.JPG

OpenCV - Open Computer Vision

And a few Steps to get your own Classifier…

OpenCV - Open Computer Vision

Commandline instructions:

image extension

Text File for output

- find ./positive_images -iname "*.jpg" > positives.txt
- find ./negative_images -iname "*.jpg" > negatives.txt

Linux Search-command

write Name/Path of Images in txt Files & generate Samples

Script request

Number of Samples

```
perl bin/createsamples.pl positives.txt negatives.txt samples 5000\
  "opencv_createsamples -bgcolor 0 -bgthresh 0 -maxxangle 1.1\
  -maxyangle 1.1 maxzangle 0.5 -maxidev 40 -w 40 -h 40"
```

OpenCV Function, that does the work

Size of Imageextracts in Pixels

OpenCV - Open Computer Vision

Output of script: binary files of samples

Now merge all the single *.vec binary files into one file…

OpenCV - Open Computer Vision

```
find ./samples -name '*.vec' > samples.txt
./mergevec samples.txt samples.vec
```

use mergence - application to merge binary *.vec files

Train the Classifier, multi-threaded

binary File of Samples

```
opencv_traincascade -data classifier -vec samples.vec -bg negatives.txt\
-numStages 20 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 4000\
-numNeg 2000 -w 40 -h 40 -mode ALL -precalcValBufSize 1024\
-precalcIdxBufSize 1024
```

20 Stages of details

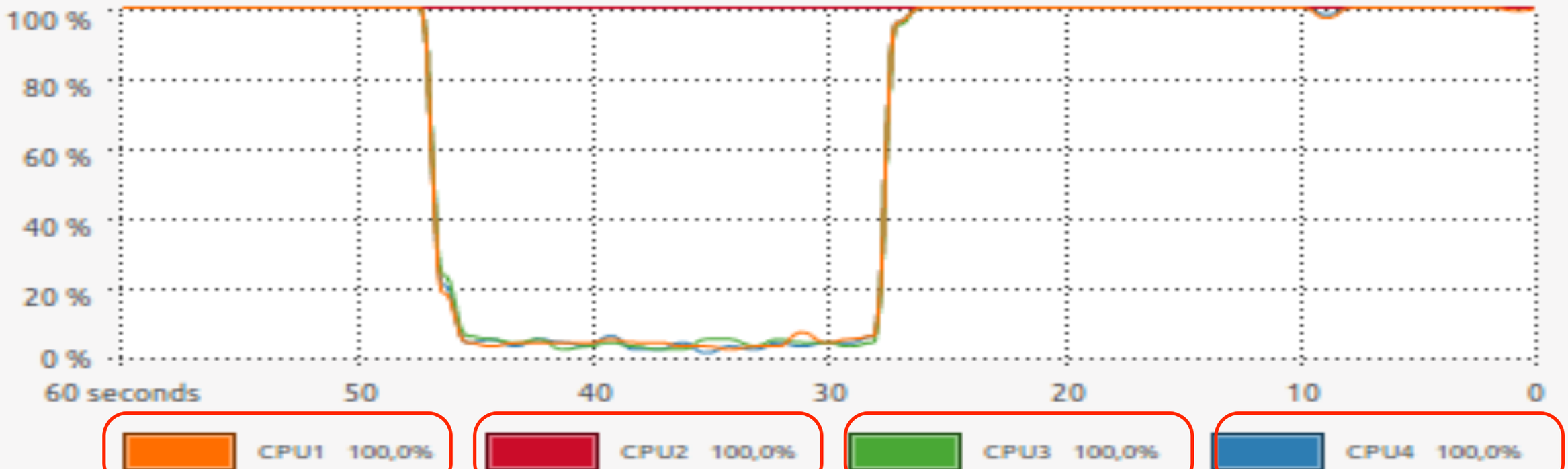
number of used Samples

4 days of Training

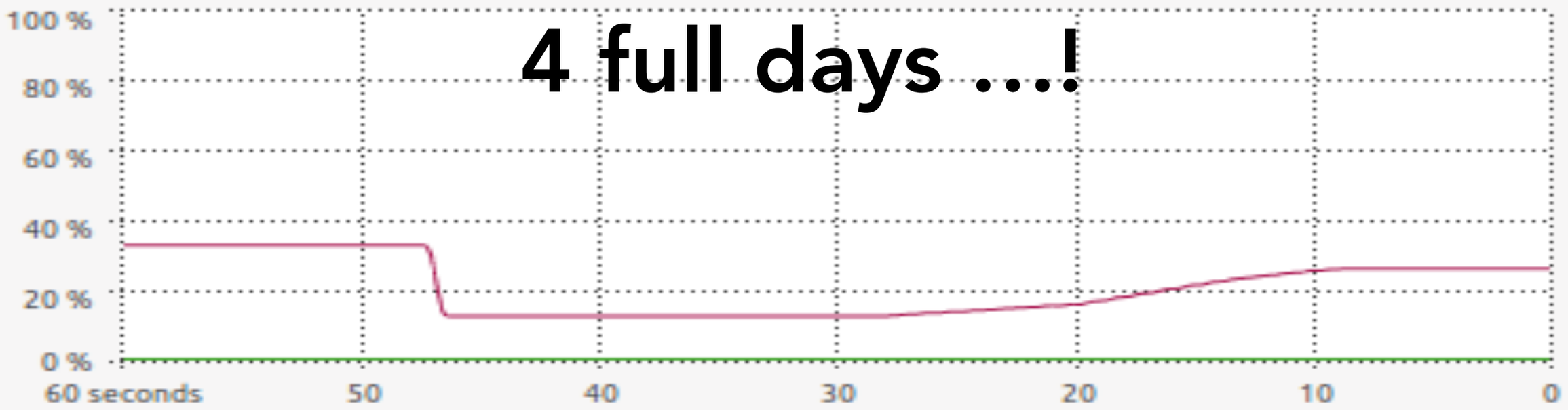
System Monitor

Processes | Resources | File Systems

CPU History



Memory and Swap History



Memory

3,8 GiB (25,7%) of 14,9 GiB

Swap

0 bytes (0,0%) of 15,7 GiB

```
+----+--------+--------+
|  36|  0.99925|   0.671|
+----+--------+--------+
|  37|  0.99925|  0.6235|
+----+--------+--------+
|  38|  0.99925|   0.595|
+----+--------+--------+
|  39|  0.99925|   0.654|
+----+--------+--------+
|  40|  0.99925|  0.6085|
+----+--------+--------+
|  41|  0.99925|  0.6215|
+----+--------+--------+
|  42|  0.99925|   0.581|
+----+--------+--------+
|  43|  0.99925|  0.5125|
+----+--------+--------+
|  44|  0.99925|  0.5435|
+----+--------+--------+
|  45|  0.99925|  0.4355|
+----+--------+--------+
END>
Training until now has taken 9 days 22 hours 39 minutes 3 seconds.
administrator@Mercury:~/C_Projects/OpenCV/opencv-haar-classifier-training$ 3~
```

…Very Time consuming and computationally intensive task

Time added up from all Processors: **~10 days**

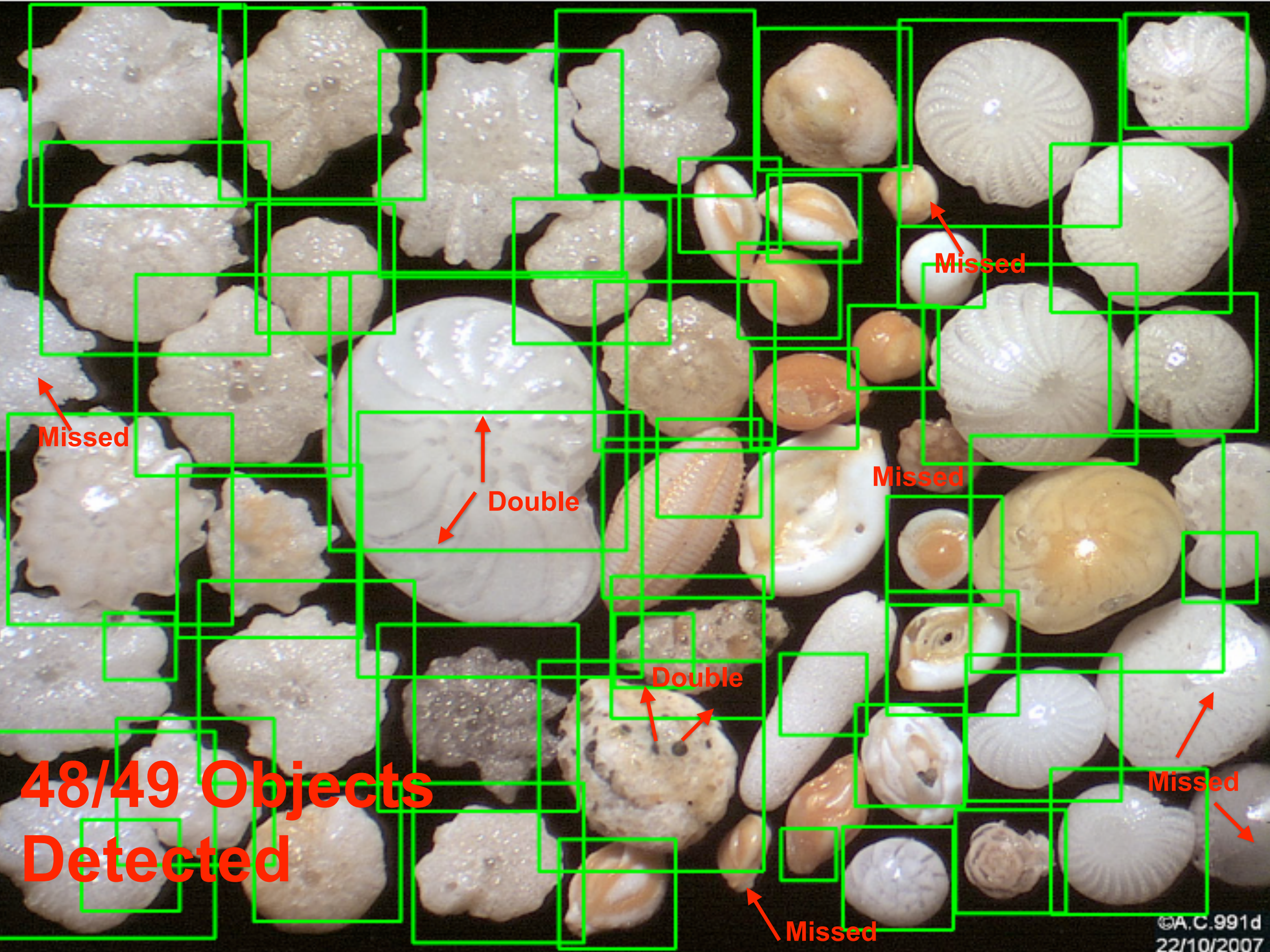
```
<?xml version="1.0"?>
<opencv_storage>
<cascade>
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>40</height>
  <width>40</width>
  <stageParams>
    <boostType>GAB</boostType>
    <minHitRate>9.9900001287460327e-01</minHitRate>
    <maxFalseAlarm>5.0000000000000000e-01</maxFalseAlarm>
    <weightTrimRate>9.4999999999999996e-01</weightTrimRate>
    <maxDepth>1</maxDepth>
    <maxWeakCount>100</maxWeakCount></stageParams>
<featureParams>
  <maxCatCount>0</maxCatCount>
  <featSize>1</featSize>
  <mode>ALL</mode></featureParams>
<stageNum>20</stageNum>
<stages>
  <!-- stage 0 -->
  <_>
    <maxWeakCount>5</maxWeakCount>
    <stageThreshold>-2.1124680042266846e+00</stageThreshold>
    <weakClassifiers>
      <_>
        <internalNodes>
          0 -1 12 2.8657460212707520e-01</internalNodes>
```

Result: an xml-File

OpenCV - Open Computer Vision

Python Code :

```
import cv2
import matplotlib.pyplot as plt                          # Import modules
%matplotlib inline
import numpy as np

IMG= cv2.imread('Ngapali.jpg')                           # read the image
IMG=cv2.cvtColor(IMG, cv2.COLOR_BGR2RGB)

                                                         # aply the classifier
Cascade = cv2.CascadeClassifier('_02_Classifiers/_02_cascade_05_1000_625_19stages.xml')
Objects = Cascade.detectMultiScale(IMG)

IMG2= cv2.imread('Ngapali.jpg')                          # read the image in second var
IMG2 = cv2.cvtColor(IMG2,cv2.COLOR_BGR2RGB)              # to mark the found objects on it

centres=[]
Radi=[]                                                  # draw a rectangle around detected Objects
for (x, y, w, h) in Objects:
    cv2.rectangle(IMG2, (x, y), (x+w, y+h), (0, 255, 0), 2)
    centres.append( (int(x+w/2.),int(y+h/2.) ) )         # calculate centerpoints
    Radi.append( np.sqrt((w/2.)**2+(h/2.)**2 ))          # calculate radius

plt.figure(figsize=(20,10))                              # show the result
plt.imshow(IMG2,cmap='gray')
plt.show()
```


Missed
Missed
Double
Missed
Double
Missed
48/49 Objects Detected
©A.C.991d
22/10/2007

OpenCV - Open Computer Vision

Get the Center-Points of all rectangles…

Missed

Double

Missed

Missed

Double

Double

Missed

Double

Missed

44/49 Objects Detected

©A.C.991d
22/10/2007

OpenCV - Haar Classifier

so far: 

- 1 Day of debugging to compile C++ Source Code of mergevec.cpp

- 10 Days of Trial and Error to train adequate Classifier including long calculation Times (10 to 60h for each run), even multithreaded

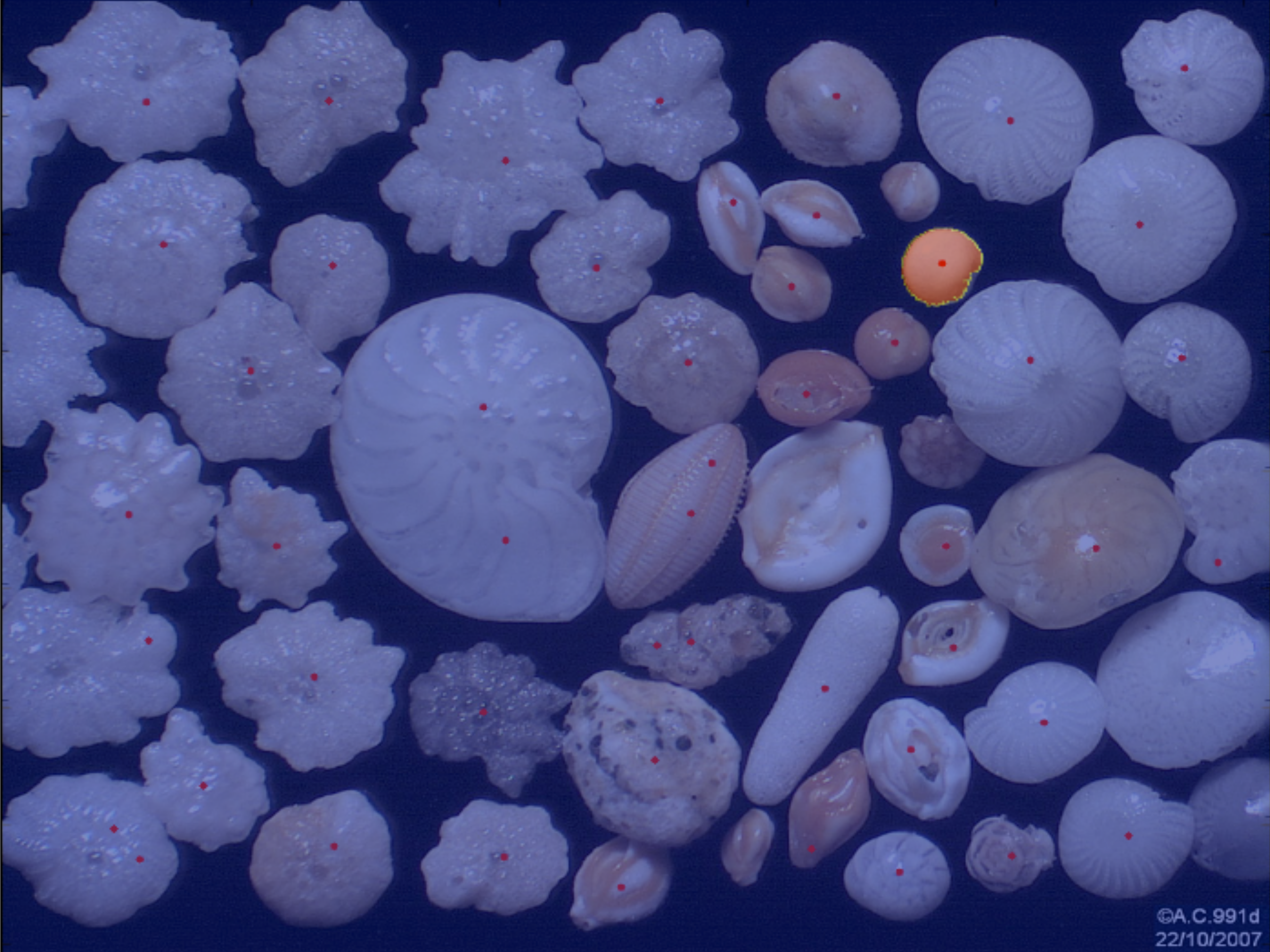
But: Just Object, no Shapes!

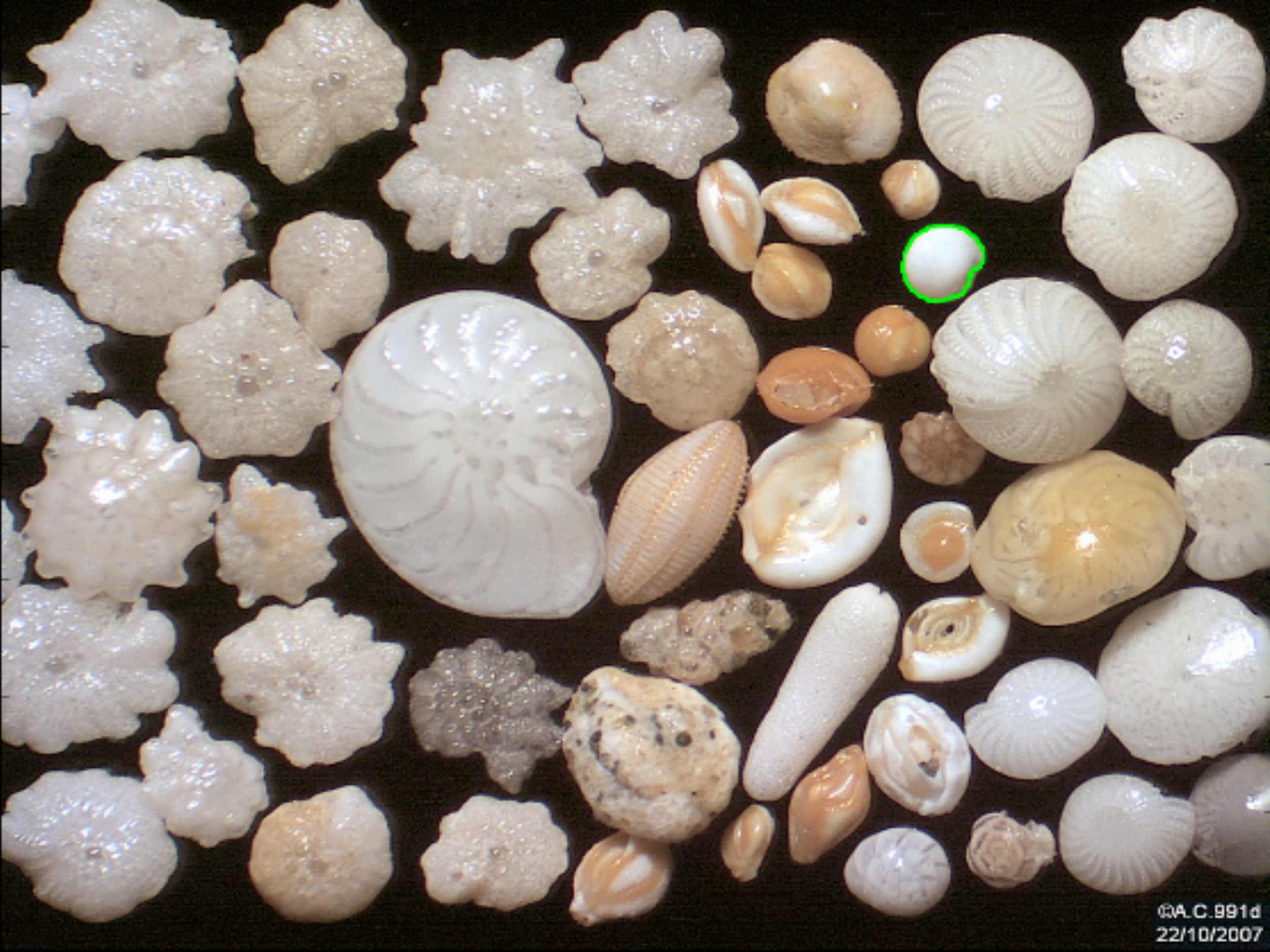
Region Growing

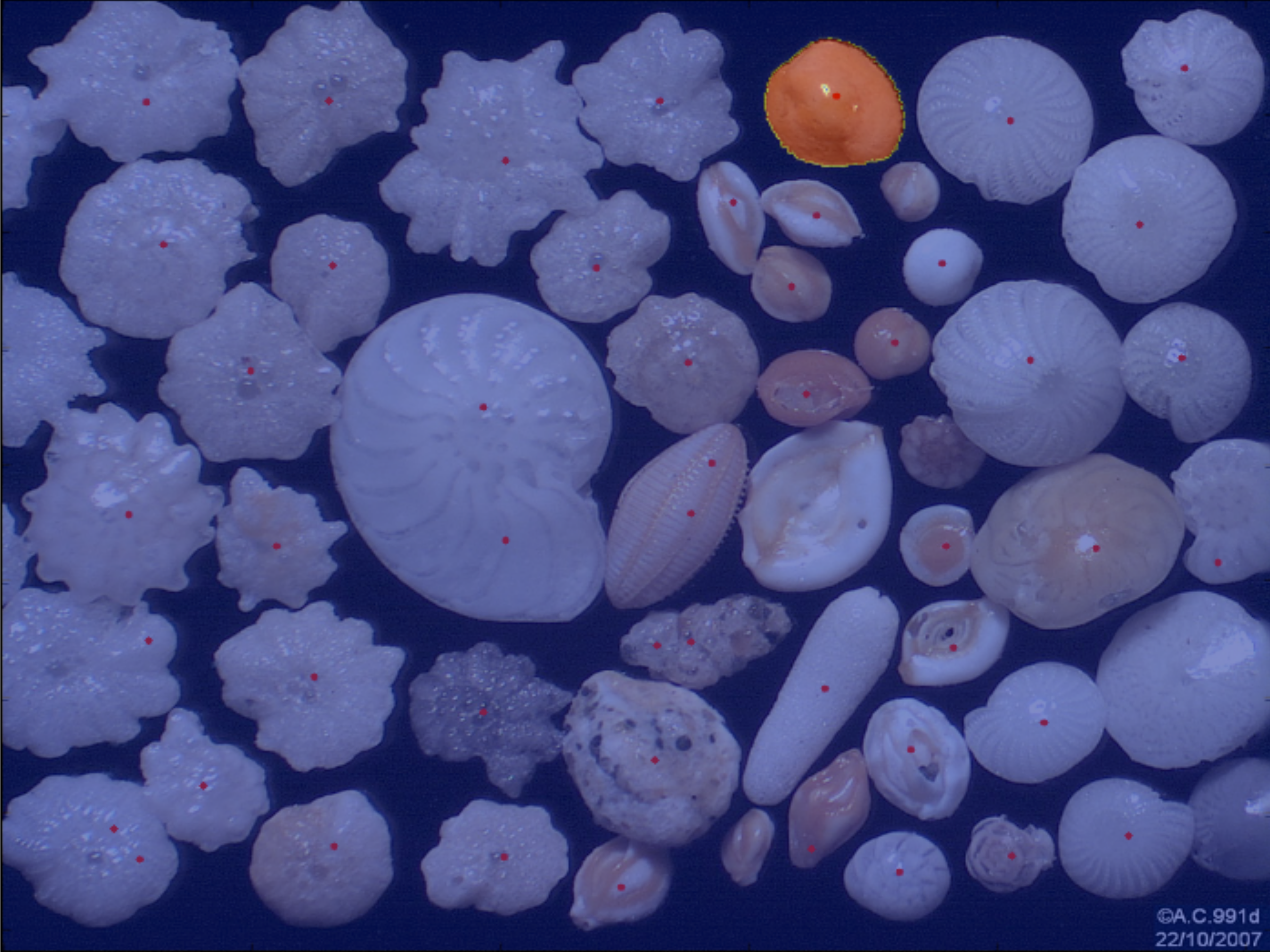
(at the center points)

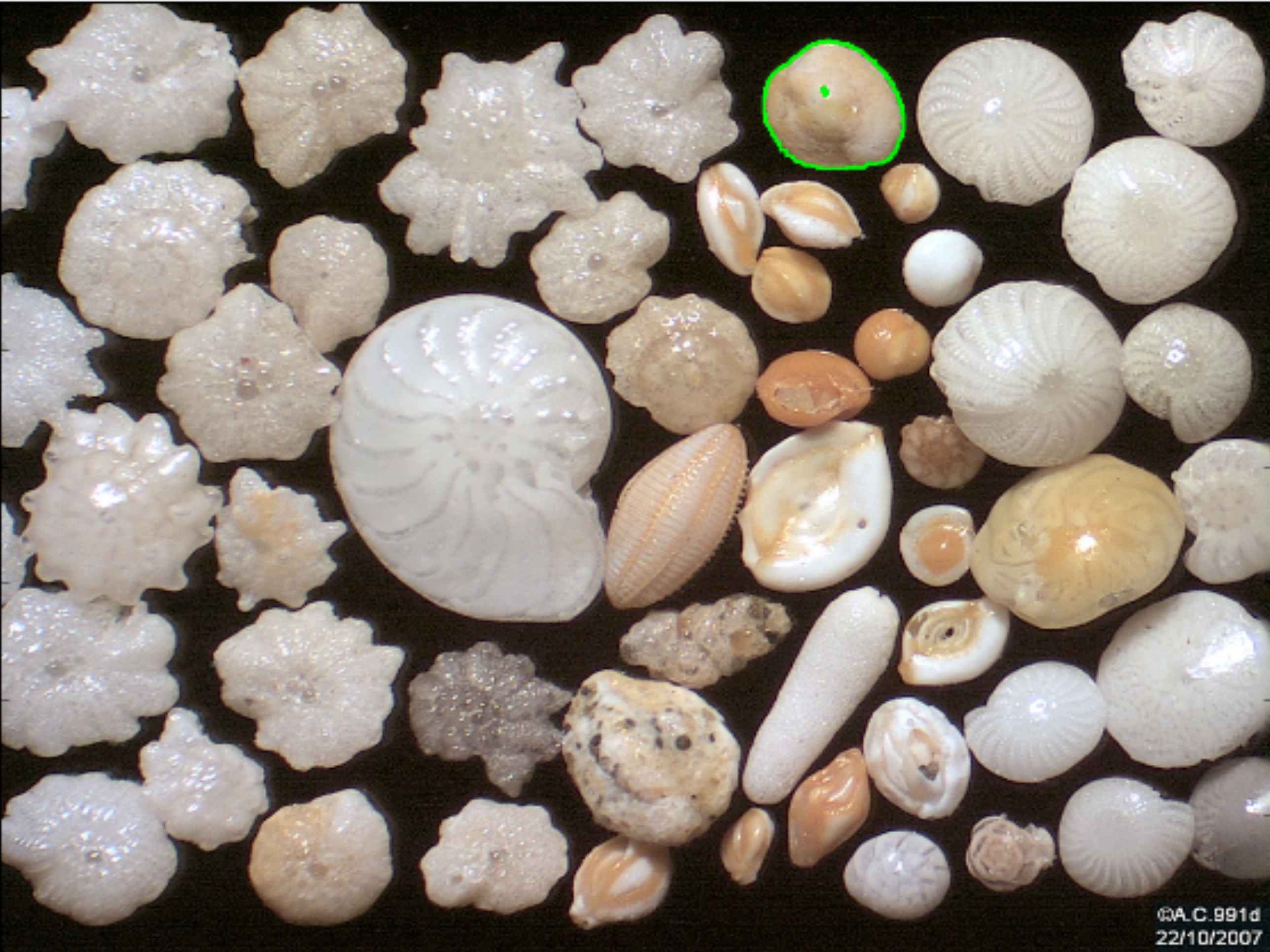
Than

findContours()


©A.C.991d
22/10/2007


©A.C.991d
22/10/2007


©A.C.991d
22/10/2007


©A.C.991d
22/10/2007

OpenCV - Haar Classifier

Fazit:

- works fine, but you need a lot experience to avoid mistakes and to tune the classifier adequate
- It Takes Time!!! Use more images for training to get more accurate results
- implement Region Growing with multiple seeds and max. distance to center point

Number of Detected Objects/Shapes: **44/49 (89.79%)**

Agenda

Into: Overview of Image Analysis and Task to Solve

Matlab: PST+Hough-Transforms

OpenCV: Threshholding, Edge-Images &
FindContours

OpenCV: HaarClassifiers

Outlook

Outlook

possible further Approaches:

- ➡standardize Technology (all Matlab,…)
- ➡Snakes - Active Contours
- ➡using SLIQ Superpixels Segmentation
- ➡Fourier Discriptors
- ➡CNN - Convolution Neural Networks
- ➡Skeleting
- ➡Watershed Transform
- ➡…

… How well does this Classifier perform on new Images?

Questions?

Thanks for your attention!