Library for organization of image recognition systems

Master's thesis

by

Oleksandr Shyrokov

http://www.ece.unh.edu/svpal/

Thesis adviser: Dr. Messner

May 2002

Outline

- Introduction
- Approach
- Automatic micro object recognition system
- Automatic human brain cell recognition
- Bacteria counting
- Library
- Conclusion

Where are we?

- **Introduction**
- Approach
- Automatic micro object recognition system
- Automatic human brain cell recognition
- Bacteria counting
- Library
- Conclusion

Introduction

- What is a Recognition System?
- Who needs it?
- What is done?
- What can be done?

Introduction (What)

- Acquire
  - Get the information from the sensors (camera)
- Processing
  - Enhance image; locate region of interests; get additional information (features)
- Make a decision
  - Label the situation
- React
  - Perform a corresponding action

Introduction (Who)

- Industry
  - High speed
  - Low error rate
- Science
  - Repetitive tasks (measurements,…)
- Hazardous environment
  - Radiation, distance

Introduction (What is done)

- Dedicated companies
  - Special hardware and software (high cost)
    - Datacube
    - Zeiss
- Software packages
  - Matlab, mathcad, etc…
- Absence of the common convention
  - Hard to reuse the work done by the others

Introduction (What can be done)

- Modularity
  - Acquisition
  - Object separation
  - Feature extraction
  - Classification
  - Displaying
- Standard
  - Predefined interface

Where are we?

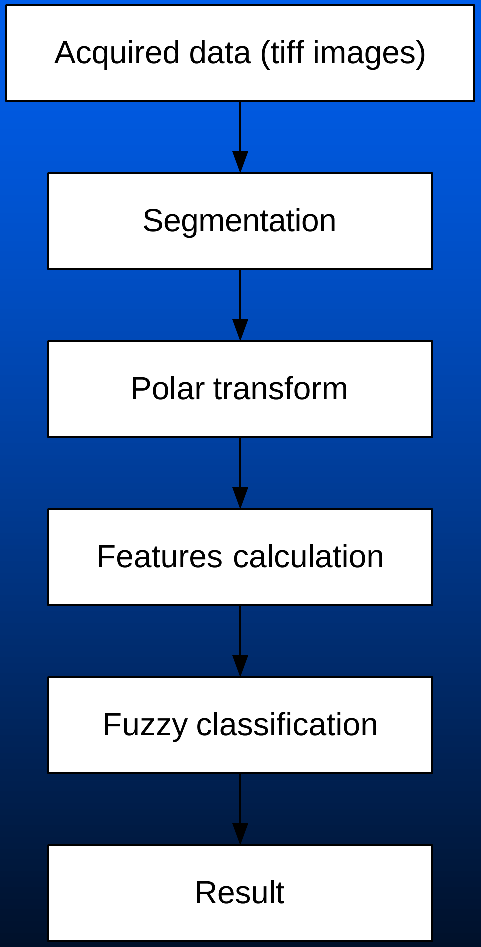
- *Introduction*
- ***Approach***
- Automatic micro object recognition system
- Automatic human brain cell recognition
- Bacteria counting
- Library
- Conclusion

Approach

- Human brain cell recognition research by Mr. Pawlak (HBCRR)
  - Matlab scripts
- Generalization
  - Bacteria counting
- Plug-in concept
  - Extendable system
- Interface definition
  - Exported functions and parameters

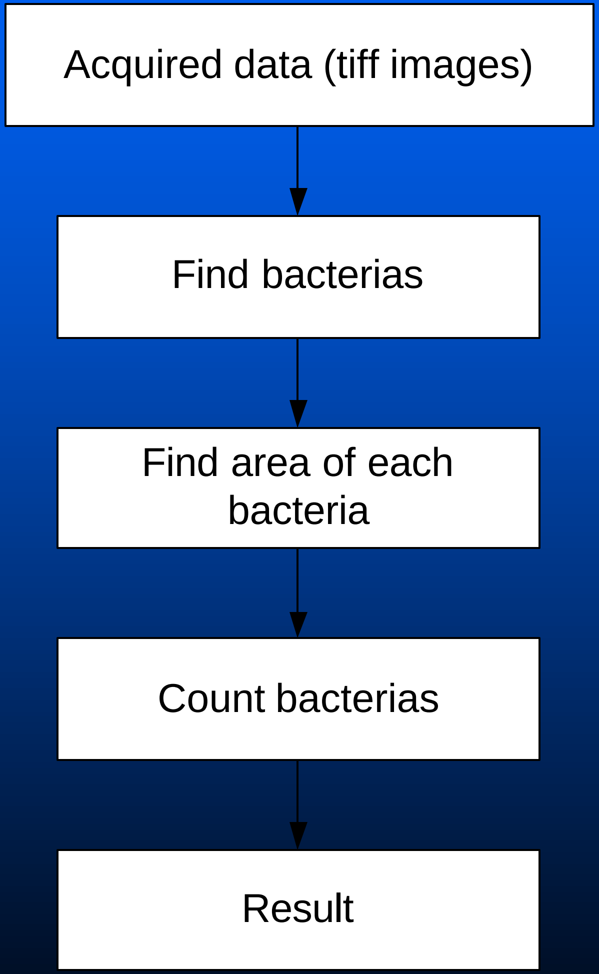
Approach (HBCRR)

- Human brain cell recognition research
  - Acquisition
    - » Acquire data
  - Object separation
    - » Locate cells
  - Feature extraction
    - » Make polar transform
    - » More features
  - Classification
    - » Classify each cell
  - Displaying
    - » Show the result



Approach (Generalization)

- Bacteria counting
  - Acquisition
    - » Acquire data
  - Object separation
    - » Locate bacteria
  - Feature extraction
    - » Find area of each bacteria
  - Classification
    - » Count bacteria
  - Displaying
    - » Show the result



Acquired data (tiff images)

Find bacterias

Find area of each bacteria

Count bacterias

Result

Approach (Plug-ins)

- What should be implemented as a plug-in?
  - Acquisition plug-in
  - Object Separation plug-in
  - Feature extraction plug-in
  - Database plug-in
  - Displaying plug-in

Approach (Interfacing)

- Acquisition
  - List of variables
  - List of images
- Segmentation
  - List of bounding rectangles
- Features
  - Add list of variables to the bounding rectangles
- Classification
  - Uses list of variables for each object
- Displaying
  - Access to all object data

Where are we?

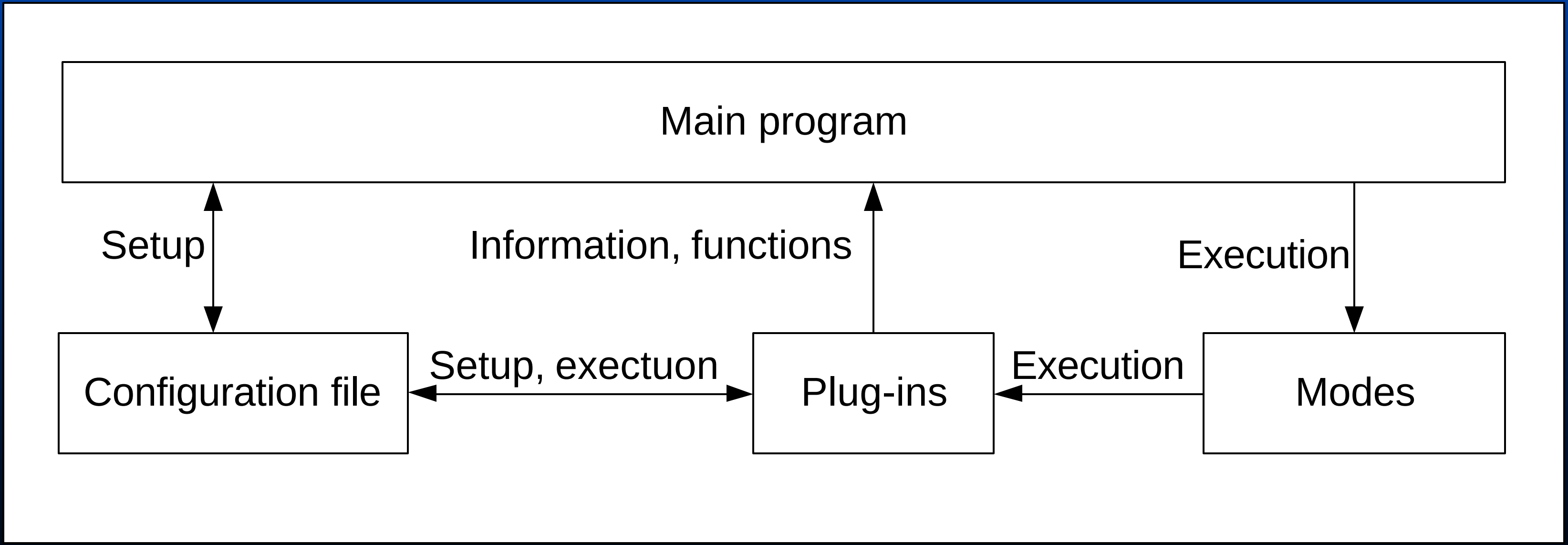
- *Introduction*
- *Approach*
- **Automatic micro object recognition system (AMORS)**
- Automatic human brain cell recognition
- Bacteria counting
- Library
- Conclusion

AMORS

- Main program
- System organization
- Modes of operation
  - Normal mode (run-time mode)
  - Research mode (active setup)
- Graphical User Interface (GUI)

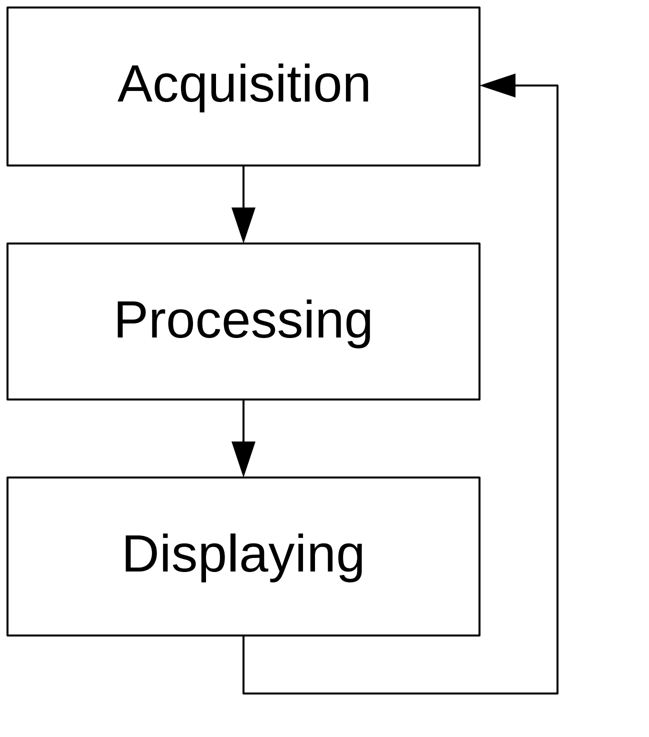
AMORS (Program)

- Main Program
  - Why AMORS
  - Plug-ins
  - GUI
  - Configuration file



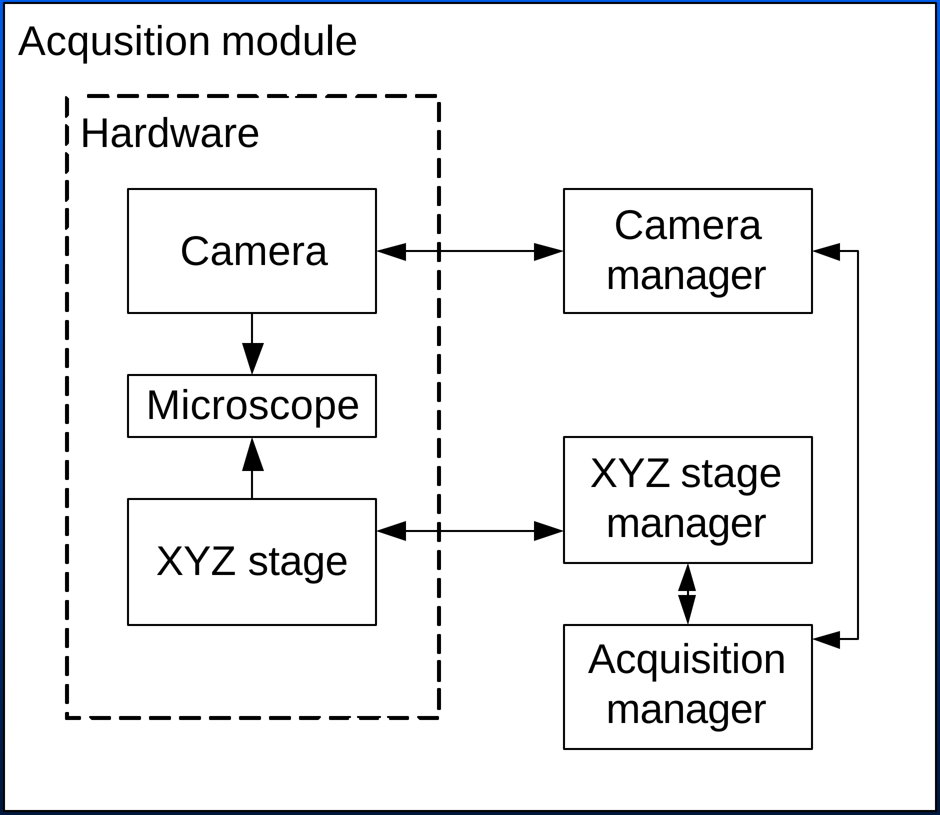
AMORS (System)

- Three main blocks
  - Acquisition
    - » Acquisition plug-in
  - Processing
    - » Object Separation plug-in
    - » Feature extraction plug-in
    - » Classification plug-in
  - Displaying
    - » Displaying plug-in



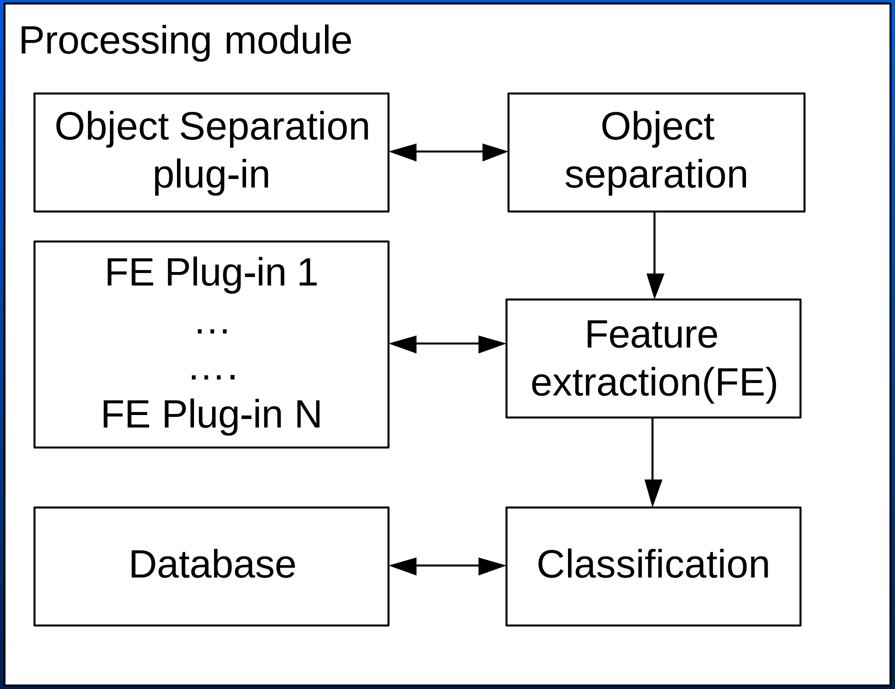
AMORS (Acquisition)

- Acquisition
  - Camera
  - Microscope
  - XYZ translation stage



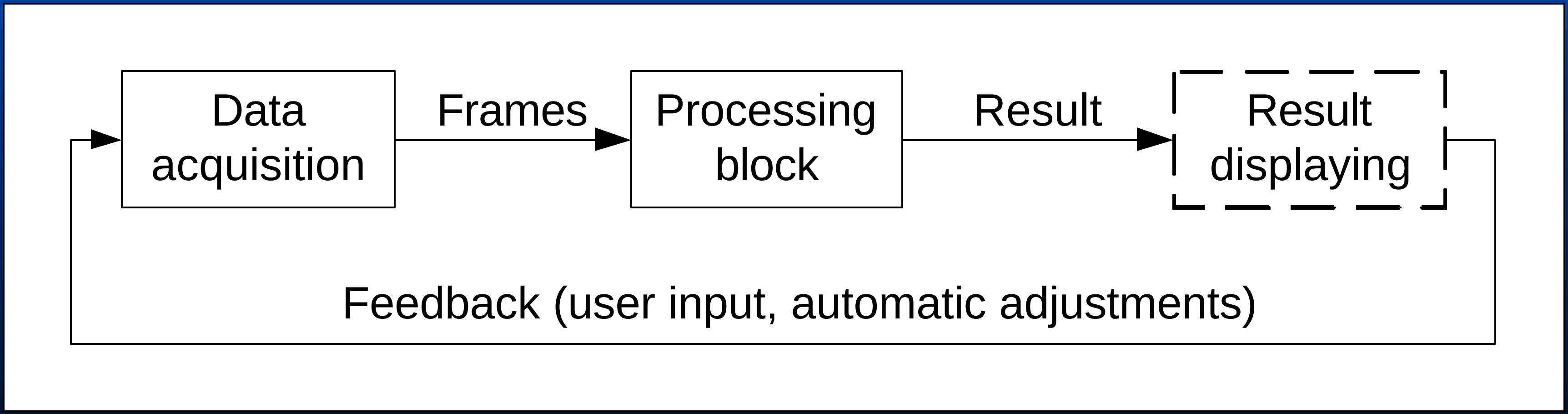
AMORS (Processing)

- Processing
  - Separation
    - » Find objects
  - Feature extraction
    - » Multiple features
  - Classification
    - » Database



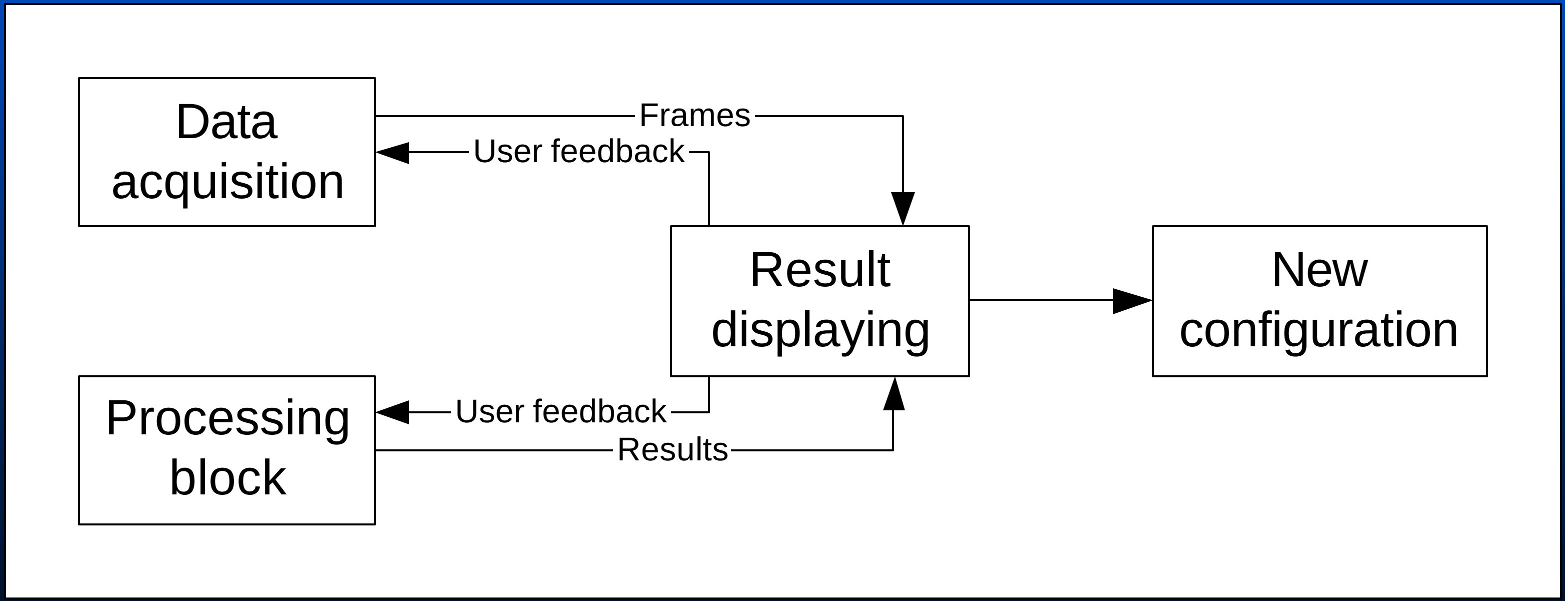
AMORS (Mode 1)

- Normal mode (Run-time)
  - – Minimum interaction with the user



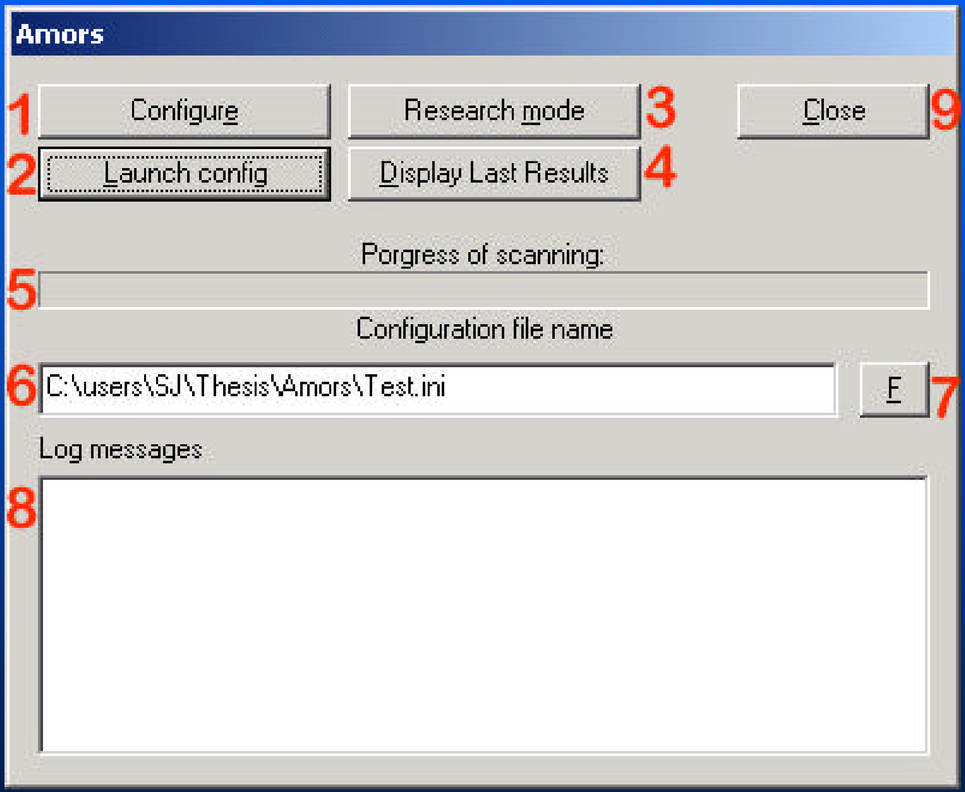
AMORS (Mode 2)

- Research mode (active setup)
  - A user can find parameters of the plug-ins



AMORS (GUI 1)

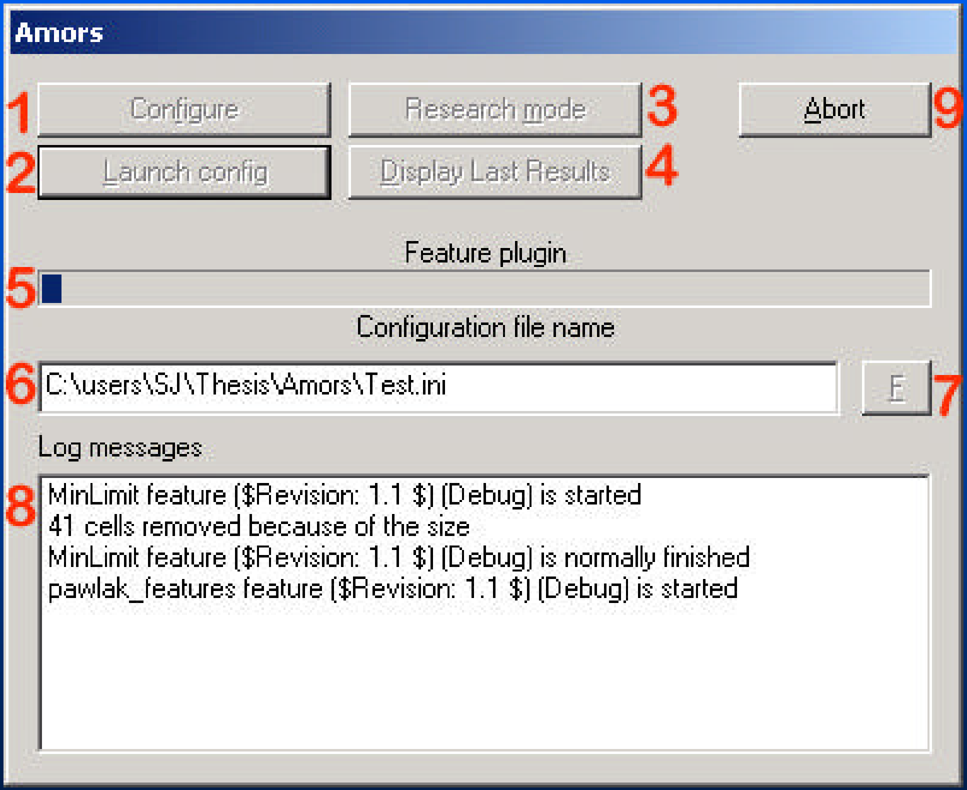
- Main window
  - Configure
  - Setup
  - Launch
  - Observe progress



Configuration file name: C:\users\SJ\Thesis\Amors\Test.ini

AMORS (GUI 1)

- Main window
  - Configure
  - Setup
  - Launch
  - Observe progress

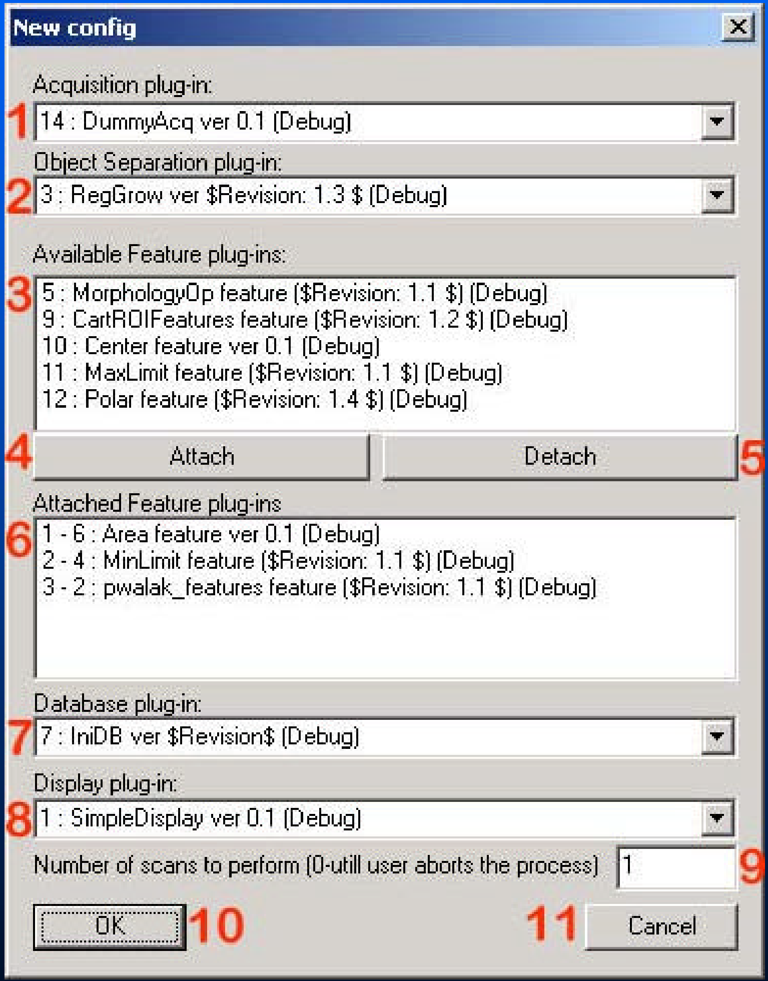


C:\users\SJ\Thesis\Amors\Test.ini

```
MinLimit feature ($Revision: 1.1 $) (Debug) is started
41 cells removed because of the size
MinLimit feature ($Revision: 1.1 $) (Debug) is normally finished
pawlak_features feature ($Revision: 1.1 $) (Debug) is started
```

AMORS (GUI 2)

- Configuration dialog box
  - Unique identifier for plug-ins
  - Plug-ins type is detected
  - Ordering of feature plug-ins



New config

Acquisition plug-in:

14 : DummyAcq ver 0.1 (Debug)

Object Separation plug-in:

3 : RegGrow ver $Revision: 1.3 $ (Debug)

Available Feature plug-ins:

5 : MorphologyOp feature ($Revision: 1.1 $) (Debug)
9 : CartROIFeatures feature ($Revision: 1.2 $) (Debug)
10 : Center feature ver 0.1 (Debug)
11 : MaxLimit feature ($Revision: 1.1 $) (Debug)
12 : Polar feature ($Revision: 1.4 $) (Debug)

Attached Feature plug-ins

1 - 6 : Area feature ver 0.1 (Debug)
2 - 4 : MinLimit feature ($Revision: 1.1 $) (Debug)
3 - 2 : pwalak_features feature ($Revision: 1.1 $) (Debug)

Database plug-in:

7 : IniDB ver $Revision$ (Debug)

Display plug-in:

1 : SimpleDisplay ver 0.1 (Debug)

Number of scans to perform (0-utill user aborts the process) 1

Where are we?

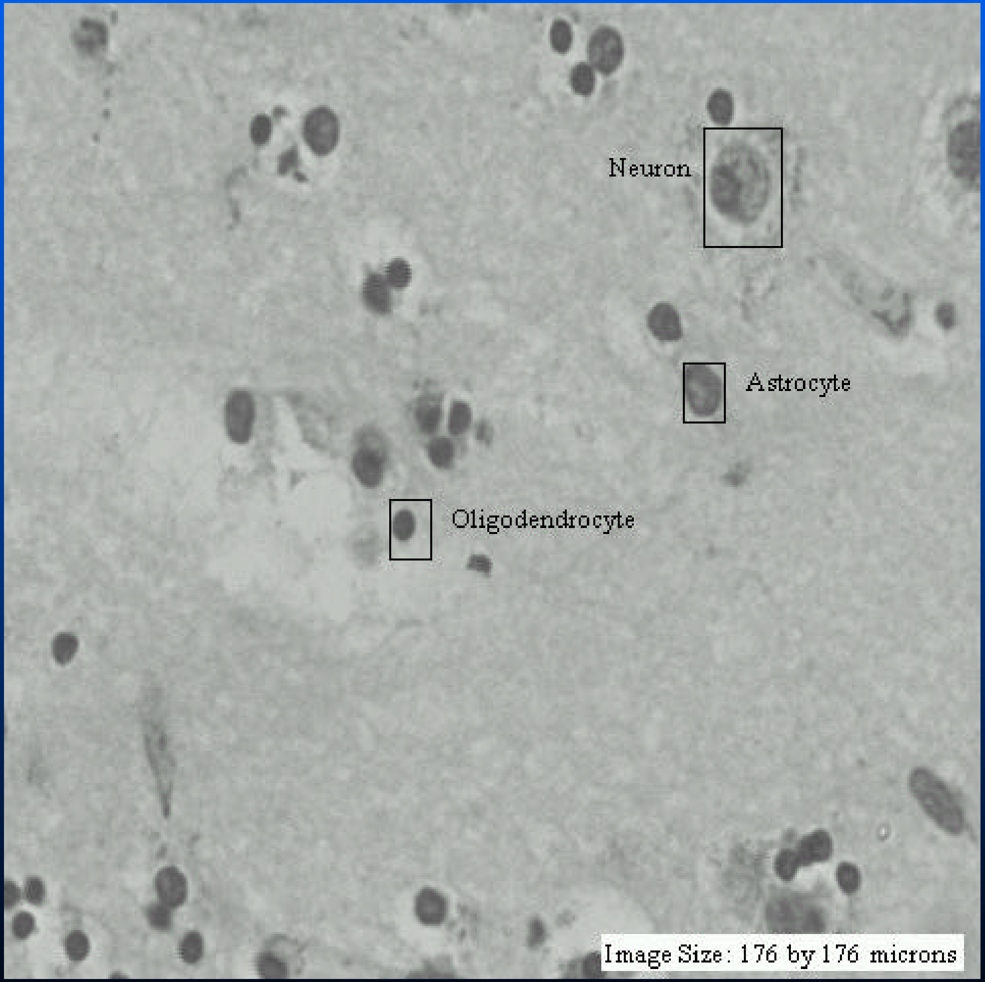
- *Introduction*
- *Approach*
- *Automatic micro object recognition system*
- **Automatic human brain cell recognition (HBCR)**
- Bacteria counting
- Library
- Conclusion

HBCR (Problem)

- Problem
  - Huntington’s Disease (HD)
  - Brain tissue
  - Tracing cells
  - Identification by experts
    - Oligodendrocyte (Oligo)
    - Astrocyte (Astro)
    - Neuron (Neuro)
    - Microgila
    - Endothelial

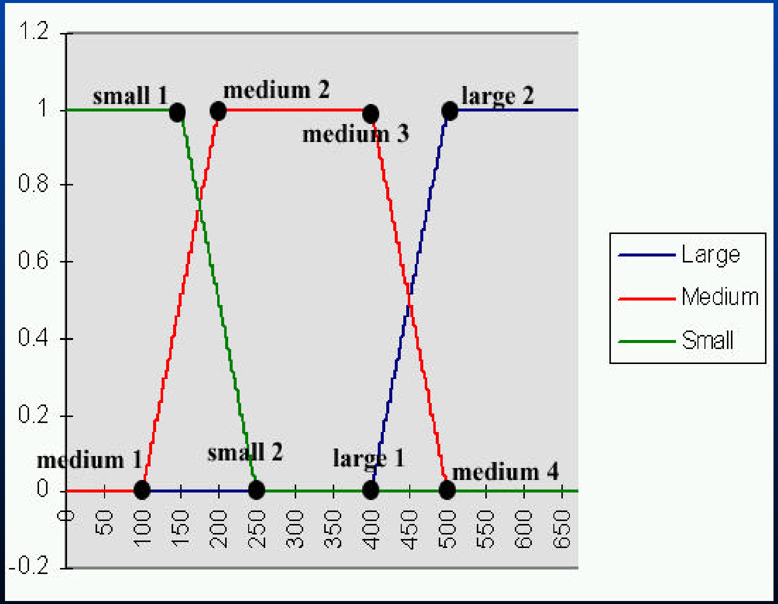
HBCR (Cells)

Examples of cells:



HBCR (Original)

- Mr. Pawlak’s research (Matlab scripts)
  - Image segmentation
    - » Region Growing method
  - Polar transform
  - Feature extraction
    - » Area, mean value, etc…
  - Fuzzy logic
    - » Membership functions

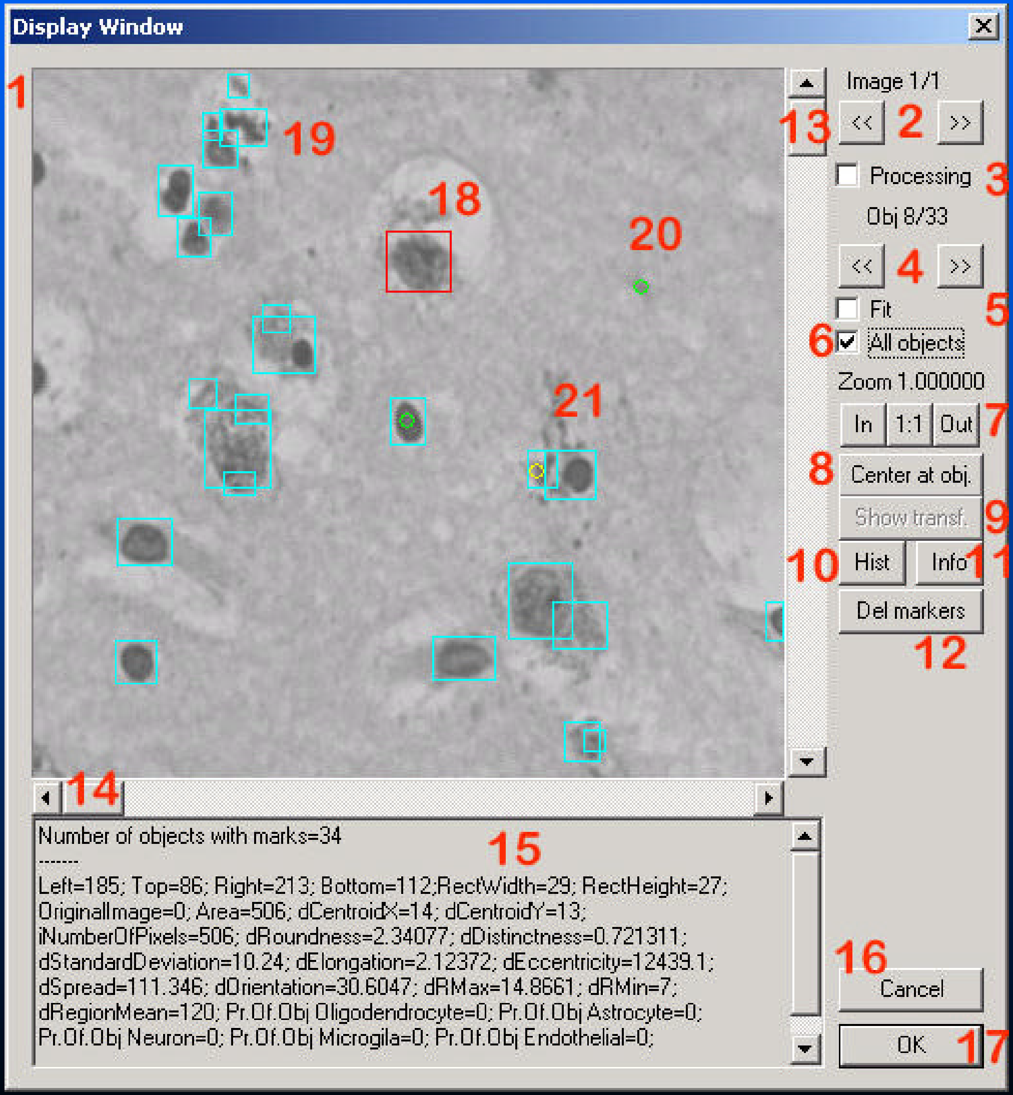


HBCR (AMORS)

- Application for AMORS
  - Acquisition
    - » DummyAcq plug-in for TIFF (Tagged Image File Format) files
  - Image segmentation
    - » RegGrow plug-in
  - Feature extraction
    - » Polar transform plug-in
    - » pawlak_features plug-in
  - Classification
    - » IniDB plug-in for fuzzy logic
  - Displaying
    - » SimpleDispaly plug-in

HBCR (Displaying 1)

- SimpleDisplay plug-in



Display Window

Image 1/1

☐ Processing

Obj 8/33

☐ Fit

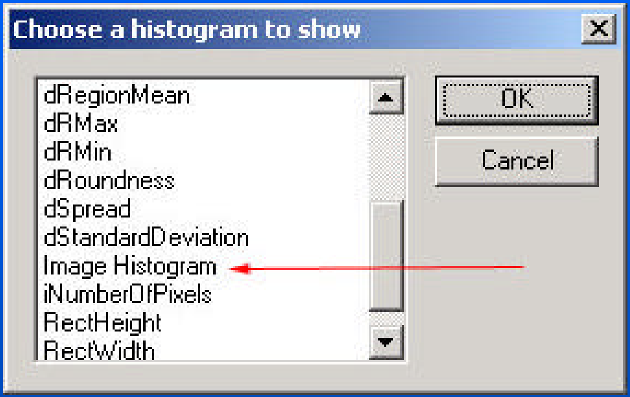
☑ All objects

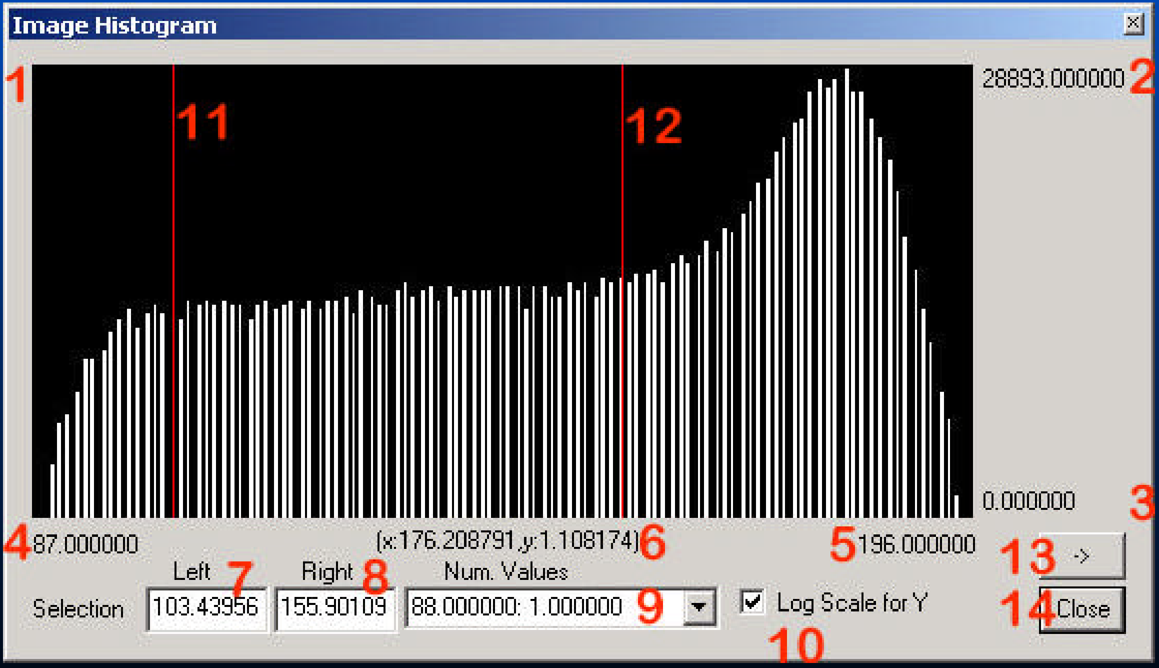
Zoom 1.000000

```
Number of objects with marks=34
-------
Left=185; Top=86; Right=213; Bottom=112;RectWidth=29; RectHeight=27;
OriginalImage=0; Area=506; dCentroidX=14; dCentroidY=13;
iNumberOfPixels=506; dRoundness=2.34077; dDistinctness=0.721311;
dStandardDeviation=10.24; dElongation=2.12372; dEccentricity=12439.1;
dSpread=111.346; dOrientation=30.6047; dRMax=14.8661; dRMin=7;
dRegionMean=120; Pr.Of.Obj Oligodendrocyte=0; Pr.Of.Obj Astrocyte=0;
Pr.Of.Obj Neuron=0; Pr.Of.Obj Microgila=0; Pr.Of.Obj Endothelial=0;
```

HBCR (Displaying 2)

- Histograms
  - Image histogram
  - Histogram for all numerical parameters





Selection Left 103.43956 Right 155.90109 Num. Values 88.000000: 1.000000

☑ Log Scale for Y

HBCR (Displaying 3)

- Histogram region selection

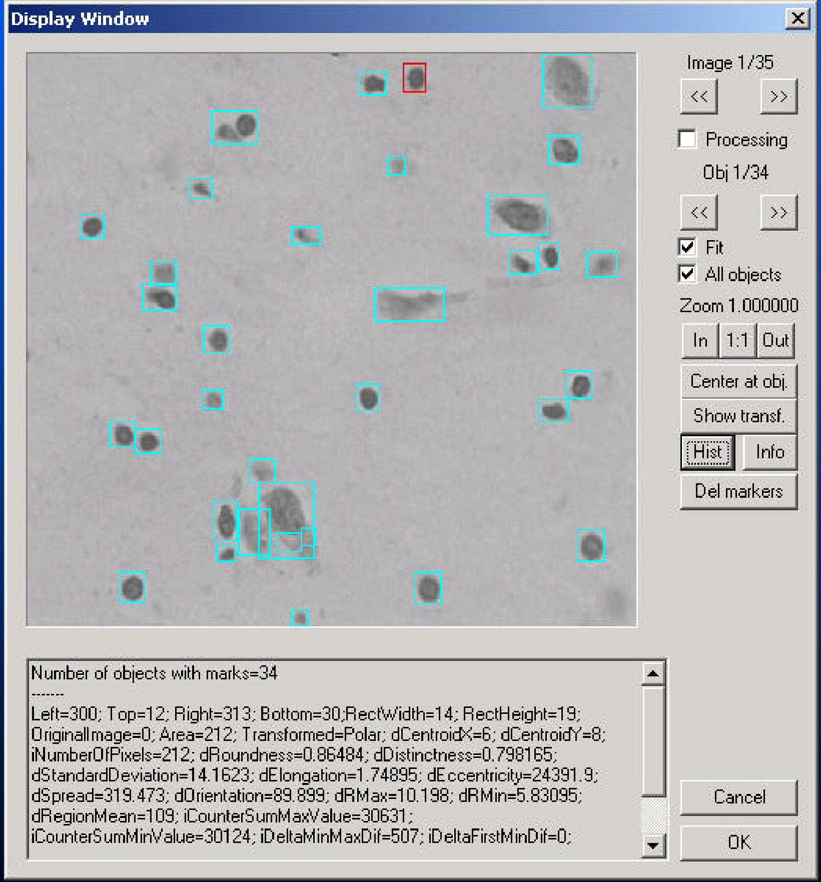


Image 1/35

☐ Processing

Obj 1/34

☑ Fit

☑ All objects

Zoom 1.000000

```
Number of objects with marks=34
-------
Left=300; Top=12; Right=313; Bottom=30;RectWidth=14; RectHeight=19;
OriginalImage=0; Area=212; Transformed=Polar; dCentroidX=6; dCentroidY=8;
iNumberOfPixels=212; dRoundness=0.86484; dDistinctness=0.798165;
dStandardDeviation=14.1623; dElongation=1.74895; dEccentricity=24391.9;
dSpread=319.473; dOrientation=89.899; dRMax=10.198; dRMin=5.83095;
dRegionMean=109; iCounterSumMaxValue=30631;
iCounterSumMinValue=30124; iDeltaMinMaxDif=507; iDeltaFirstMinDif=0;
```

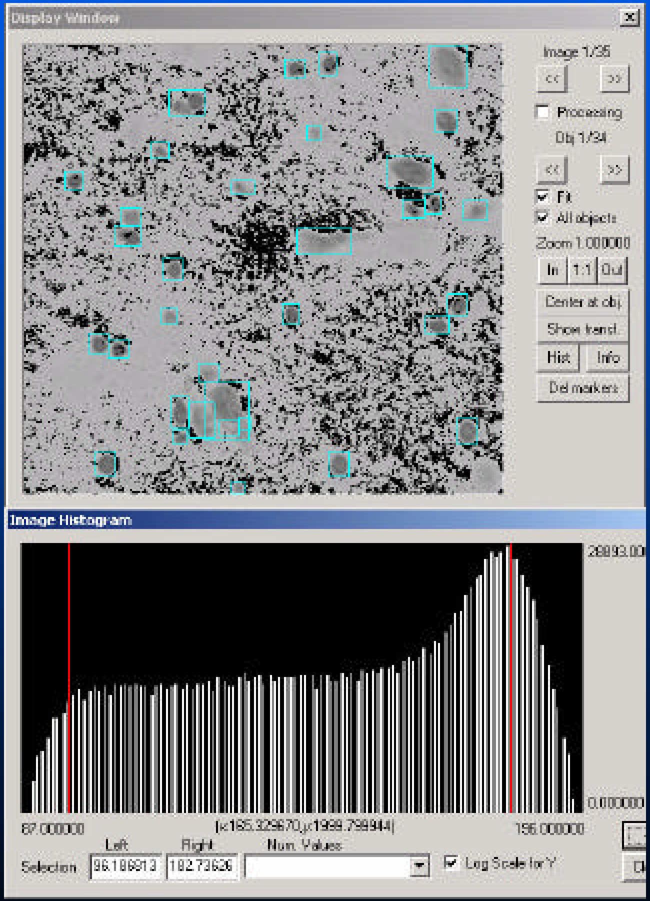


Image 1/35

☐ Processing

Obj 1/34

☑ Fit

☑ All objects

Zoom 1.000000

Selection Left 96.186913 Right 182.73626

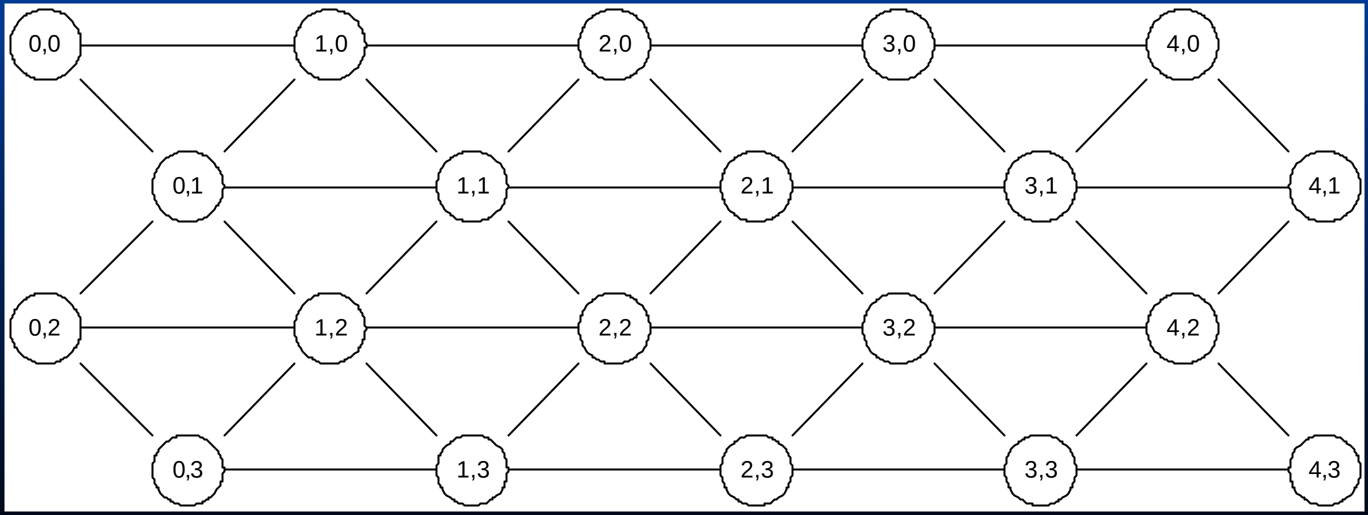
☑ Log Scale for Y

HBCR (Improvements)

- Improvements
  - – 3D Acquisition
    - » Analyze stack of images taken with different focal depths
      - Voting
  - – Self-Organizing Map (SOM)
    - » Solves problem of defining the membership functions

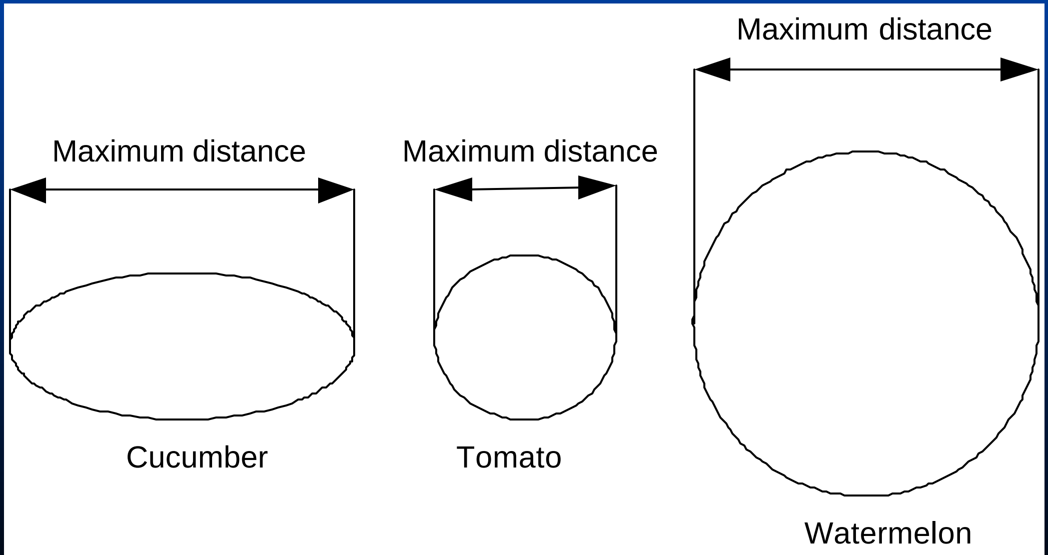
HBCR (SOM 1)

- Mapping from the input data space $\mathbf{R}^n$ onto a regular 2D array of nodes
  - Node **i**; parametric vector $\mathbf{m}_i$ Î $\mathbf{R}^n$ associated with node **i**; Node location is in form **(x,y)**;
  - This map has the size **5** by **4**



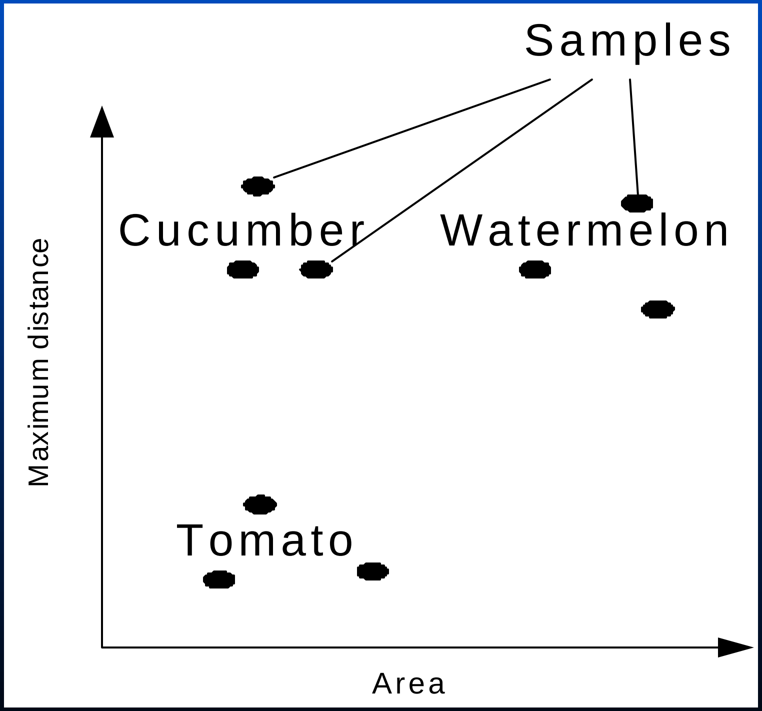
HBCR (SOM 2)

- Let me explain SOM using the following example
  - Need to separate cucumbers, tomatoes and watermelons
  - We are given the area occupied by the vegetable and the maximum distance, as shown below



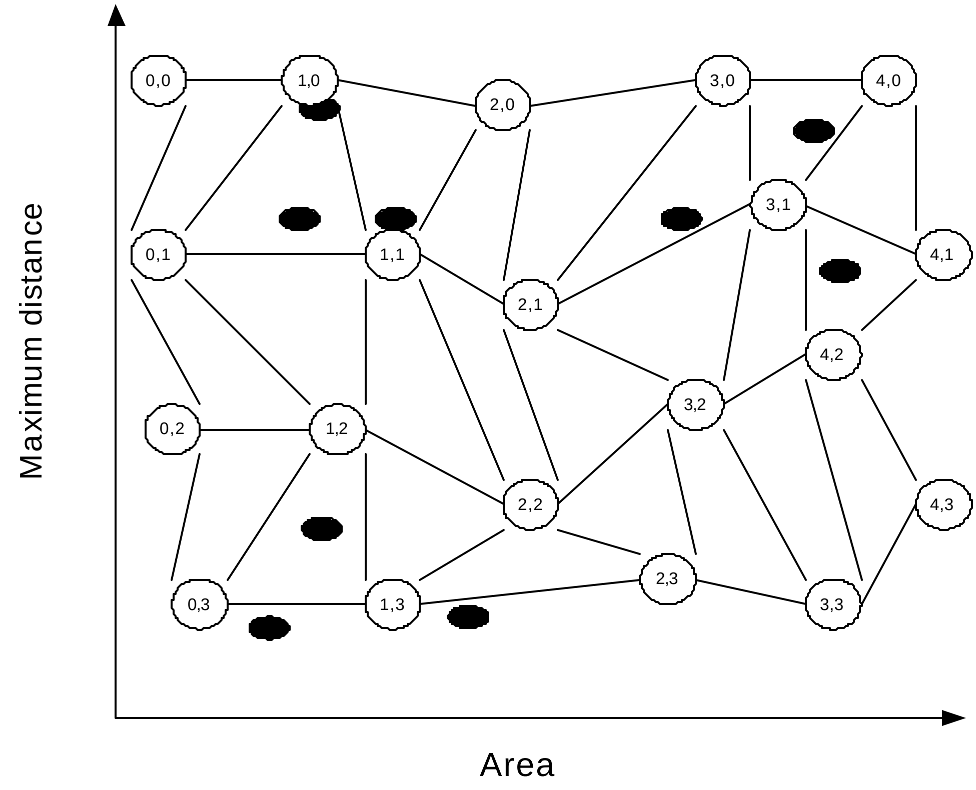
HBCR (SOM 3)

- Training data
  - – One need a predefined data to train the map



HBCR (SOM 4)

- Learning: Initializing
  - – $\mathbf{m}_i(\mathbf{0})$ is assigned randomly



HBCR (SOM 5)

- Learning: Adjusting the map
  - Shifting the closest node with its neighborhood closer to the input vector

$$m_i(t+1) = m_i(t) + h_{ci}(t) \cdot [x(t) - m_i(t)]$$

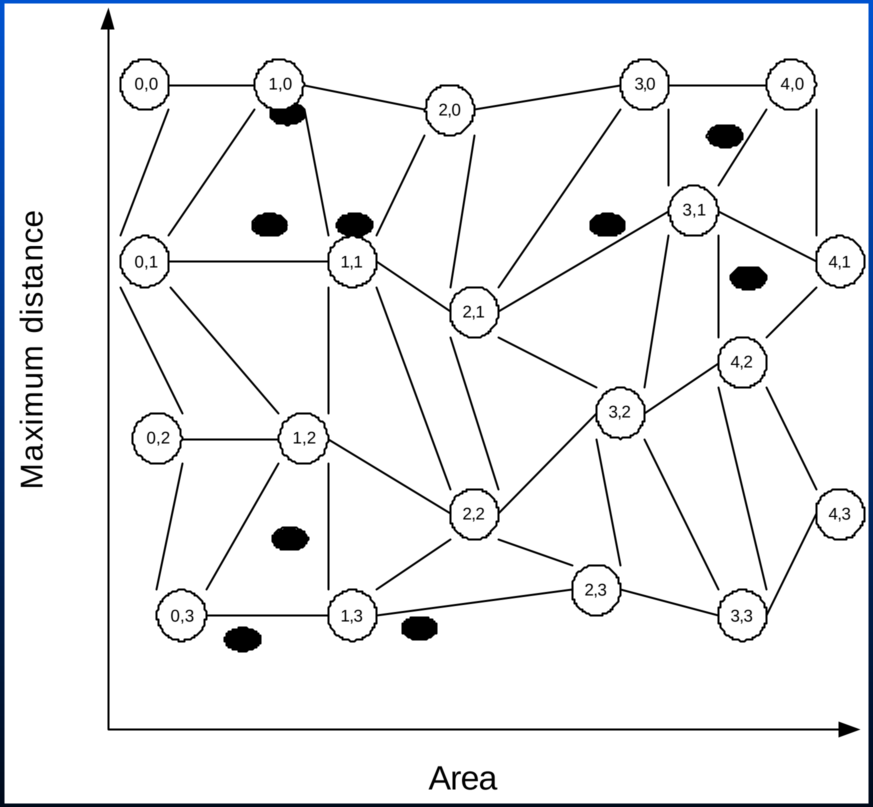
t – discrete-time coordinate; i – node index; m_i – node vector; x – input vector; h_{ci} – so-called neighborhood kernel

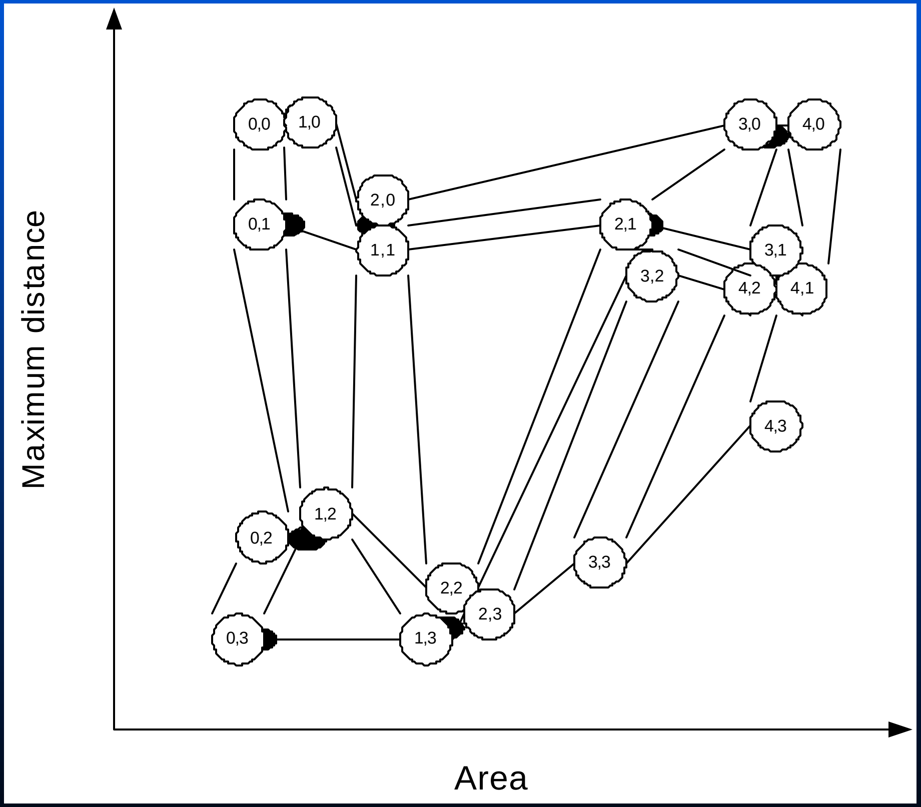
HBCR (SOM 6)

- Learning: Neighborhood kernel
  - Position dependence
    - $\mathbf{h}_{ci}=\mathbf{h}(||\mathbf{r}_c - \mathbf{r}_i||;\mathbf{t})$, where $\mathbf{r}_c$ Î $\mathbf{R}^2$ and $\mathbf{r}_i$ Î $\mathbf{R}^2$ – radius vectors of nodes **c** and **i**
    - $\mathbf{h}_{ci}$ goes to 0 with increasing $||\mathbf{r}_c - \mathbf{r}_i||$
    - defines the “stiffness” of the “elastic surface”
  - Time dependence
    - get smaller with time
    - learning rate
  - Two steps
    - Ordering
    - Fine tuning

HBCR (SOM 7)

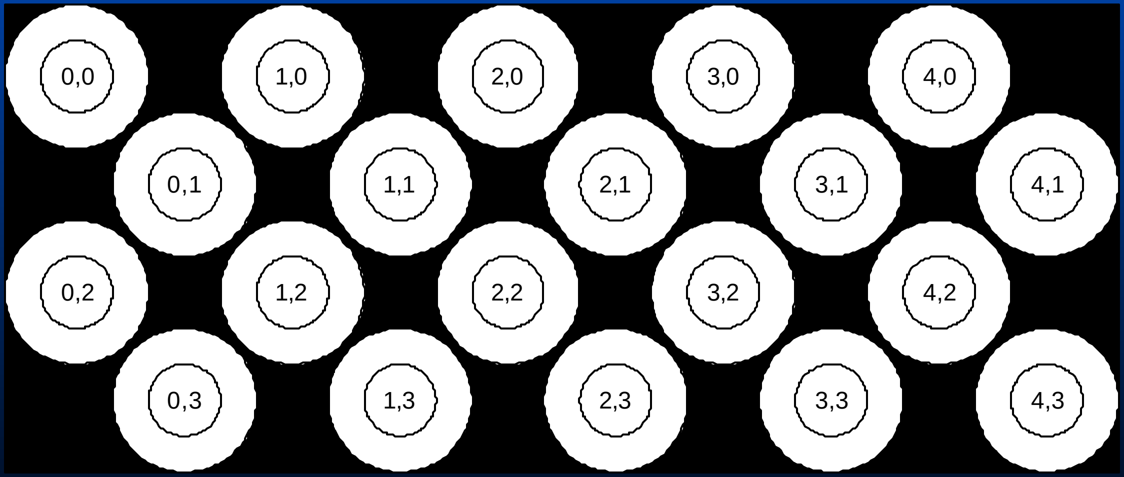
- Learning: Training result





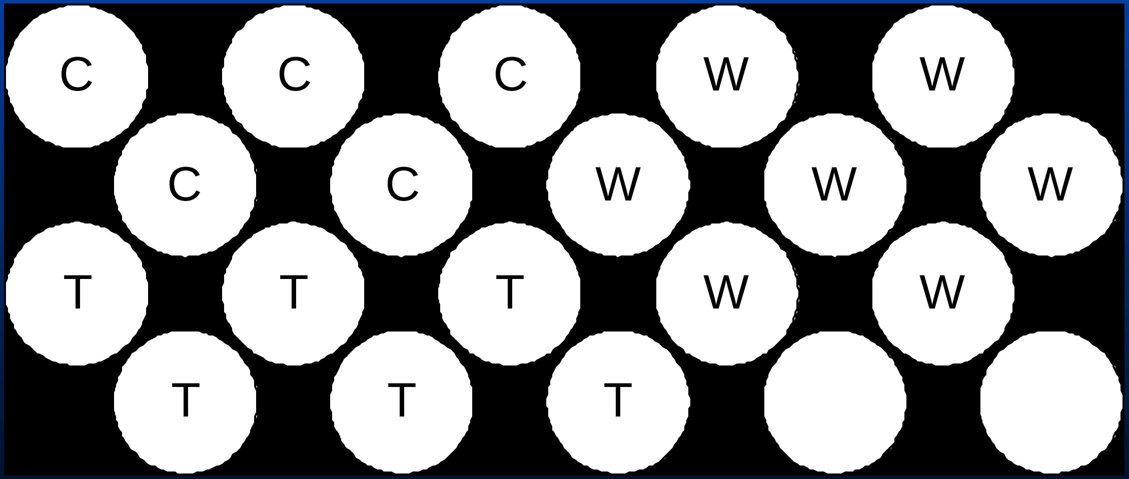
HBCR (SOM 8)

- Learning: Representation
  - high dimensions do not allow the direct presentation of mapping process
  - nodes are forming 2D grid



HBCR (SOM 9)

- Learning: Map calibration
  - Assigning labels of training samples to the nodes (C – cucumber, T – tomato, W – watermelon)



HBCR (SOM 10)

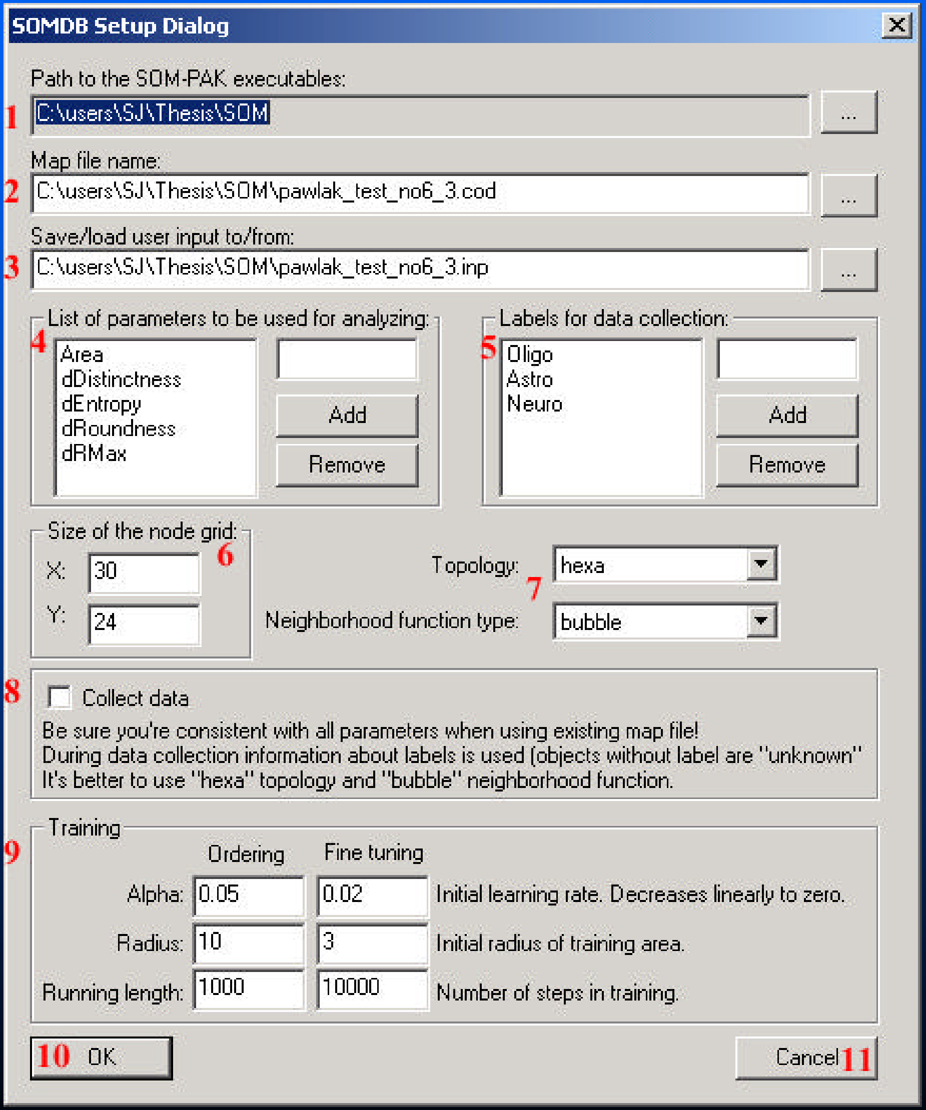
- Learning: Quantization error
  - finite number of nodes
  - approximation of sample distribution
  - quantization error – average difference between the input vector and the best-match vector

HBCR (SOM 11)

- Using SOM for cell recognition
  - SOM_PAK software package
    - randinit – initializes the map for training
    - vsom – trains the map
    - vcal – labels the map vectors
    - visual – finds best-matching nodes
    - qerror – quantization error
  - SOMDB plug-in
    - Parameter setup for the training

HBCR (SOM 12)

- SOMDB plug-in setup



SOMDB Setup Dialog

Path to the SOM-PAK executables: C:\users\SJ\Thesis\SOM

Map file name: C:\users\SJ\Thesis\SOM\pawlak_test_no6_3.cod

Save/load user input to/from: C:\users\SJ\Thesis\SOM\pawlak_test_no6_3.inp

List of parameters to be used for analyzing:

Area
dDistinctness
dEntropy
dRoundness
dRMax

Labels for data collection:

Oligo
Astro
Neuro

Size of the node grid:

X: 30

Y: 24

Topology: hexa

Neighborhood function type: bubble

☐ Collect data

Be sure you're consistent with all parameters when using existing map file!
During data collection information about labels is used (objects without label are "unknown"
It's better to use "hexa" topology and "bubble" neighborhood function.

Training

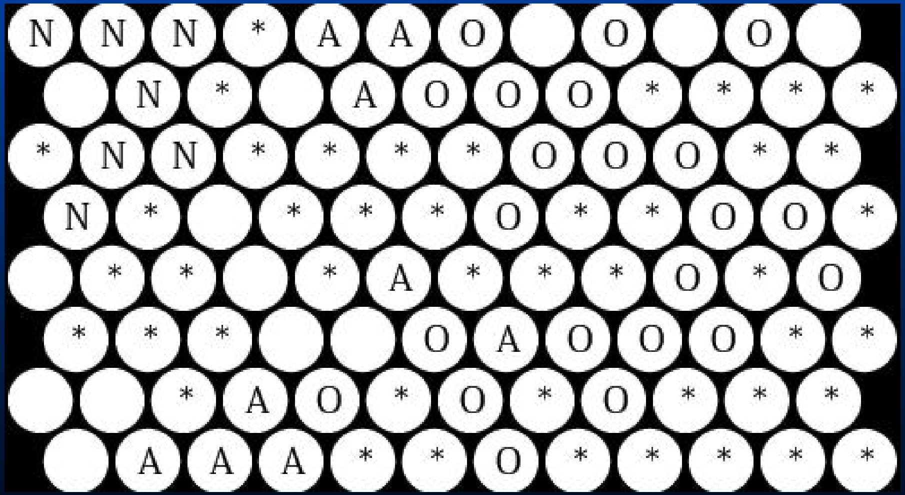
	Ordering	Fine tuning	
Alpha:	0.05	0.02	Initial learning rate. Decreases linearly to zero.
Radius:	10	3	Initial radius of training area.
Running length:	1000	10000	Number of steps in training.

HBCR (SOM 13)

- Parameters used for training

Name\Step	Ordering step	Fine tuning
Number of iterations	1,000	10,000
Learning rate	0.05	0.01
Radius of training area	10	3

- Example with the grid size 12x8 nodes



N	N	N	*	A	A	O		O		O	
	N	*		A	O	O	O	*	*	*	*
*	N	N	*	*	*	*	O	O	O	*	*
N	*		*	*	*	O	*	*	O	O	*
	*	*		*	A	*	*	*	O	*	O
*	*	*			O	A	O	O	O	*	*
		*	A	O	*	O	*	O	*	*	*
	A	A	A	*	*	O	*	*	*	*	*

N - Neuron cells, A - Astrocyte cells, O - Oligodendrocyte cells, * - unknown cells or clutter

HBCR (Future work)

- Work to be done
  - Testing of the different separation methods, by using different plug-ins
  - Make analysis to find the best learning parameters for SOM
  - Features, which give maximum separation of the cells

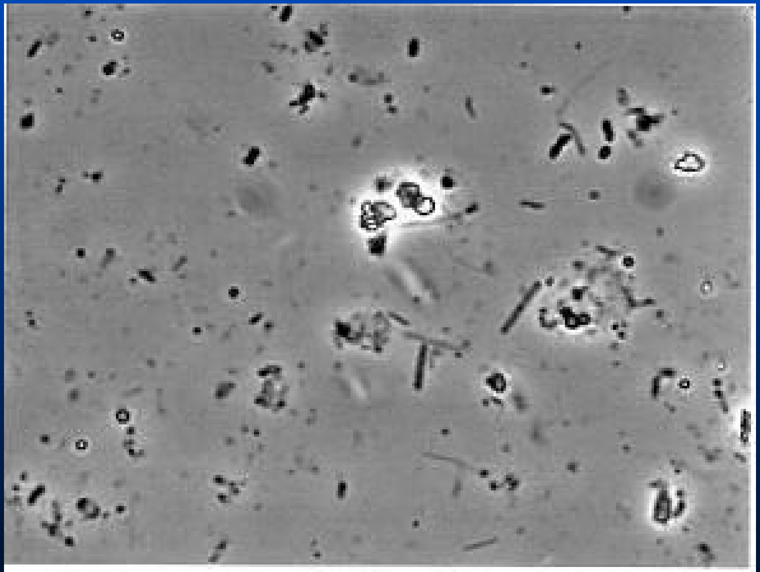
Where are we?

- *Introduction*
- *Approach*
- *Automatic micro object recognition system*
- *Automatic human brain cell recognition*
- **Bacteria counting (BC)**
- Library
- Conclusion

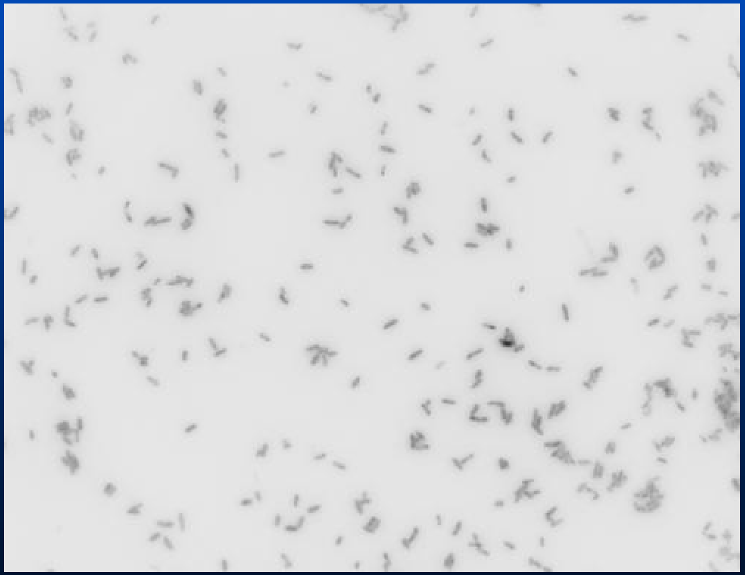
BC (Problem)

- Pure culture vs. industrial specimen

Industrial specimen



Pure culture (inverted)



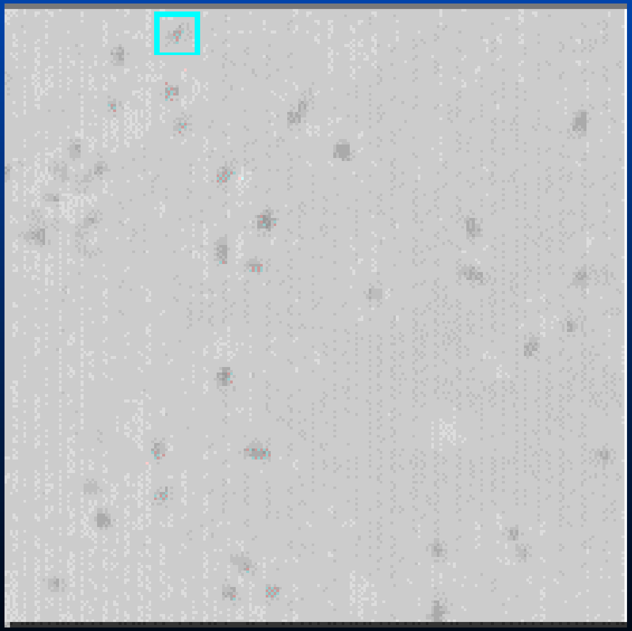
BC (Solution)

- Acquisition
  - MaxPci
  - Zeiss
  - FromFileAcq plug-in
- Methods
  - Threshold
  - Morphology operations
  - Clusters division

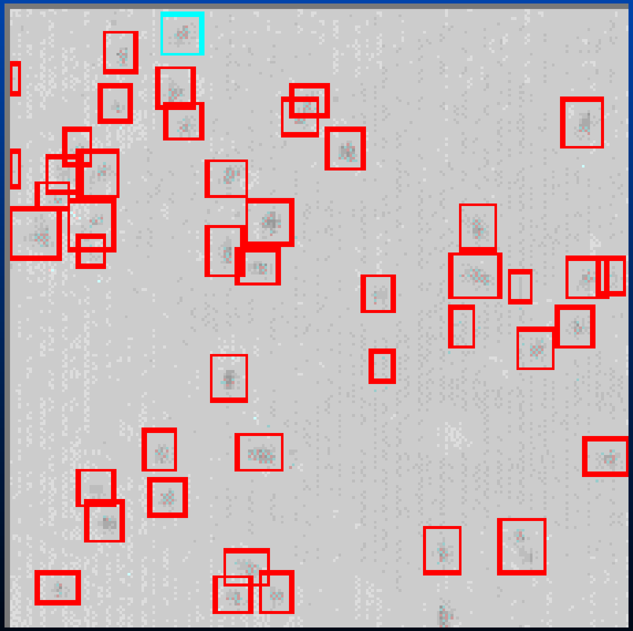
BC (Result)

- Accuracy (95 %) the same as for human
- Speed more then 5 times faster than for human

Input (inverted)

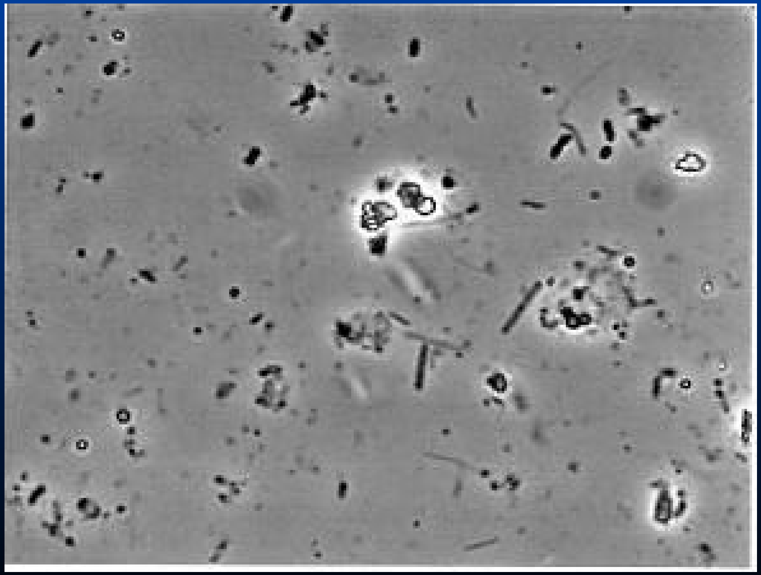


Counted (bacteria)



BC (Future work)

- Work to be done
  - Automatic acquisition
  - Automatic thresholding
  - Counting in the industrial specimens



Where are we?

- *Introduction*
- *Approach*
- *Automatic micro object recognition system*
- *Automatic human brain cell recognition*
- *Bacteria counting (BC)*
- **Library**
- Conclusion

Library

- Statically linked library
  - interfaces.lib
- Portability
  - Standard Template Library
- Classes
  - CSJIniFile (configuration files)
  - CSJBuffer (2D data)
  - CSJImage (Color images)
  - CSJImageList (List of images)
  - CSJObjectInfo (Object description)
  - CSJObjectInfoList (List of objects)
  - CSJSpecimen (Description of the specimen)

Library (CD)

- Open sources
- Templates
- Examples
- Documentation
- Applications

Library (Future work)

- Work to be done
  - Test for different operating systems
  - Non-linear connection of the plug-ins
  - Feedbacks
  - Standard
  - Code maintenance

Where are we?

- *Introduction*
- *Approach*
- *Automatic micro object recognition system*
- *Automatic human brain cell recognition*
- *Bacteria counting (BC)*
- *Library*
- **Conclusion**

Conclusion

- Ready-to-use procedure of creating the recognition system
- Free
- Examples
- Templates
- Documentation

References

- A. J. Pawlak, "Automatic human brain cell recognition", Master's Thesis, September 1998
- R.O. Duda, P.E. Hart, D.G. Stork, “Pattern Calssification”, 2nd edition, Wiley-Interscience Publications, 2000
- Full list online: http://www.ece.unh.edu/svpal/

The end

Questions, comments?

Maybe a little demo?

Thank you!