Multi-point Geostatistical Algorithm for Simulating the Subsurface using Non-linear Mapping function

Dr. Snehamoy Chatterjee

Asst. Professor

National Institute of Technology Rourkela
Department of Mining Engineering
Rourkela 769008

Outline

- Review of pattern-based simulation

- Mapping of pattern data using non-linear function

- Patterns classification

- Validation of the method

- Conclusions

Pattern-based simulation

- Generation of a pattern database

- Pattern database classification

- Similarity measures between conditional data and class prototypes

- Pattern selection and pattern pasting

Pattern generation

Pattern 1   Pattern 2

Pattern N-1   Pattern N
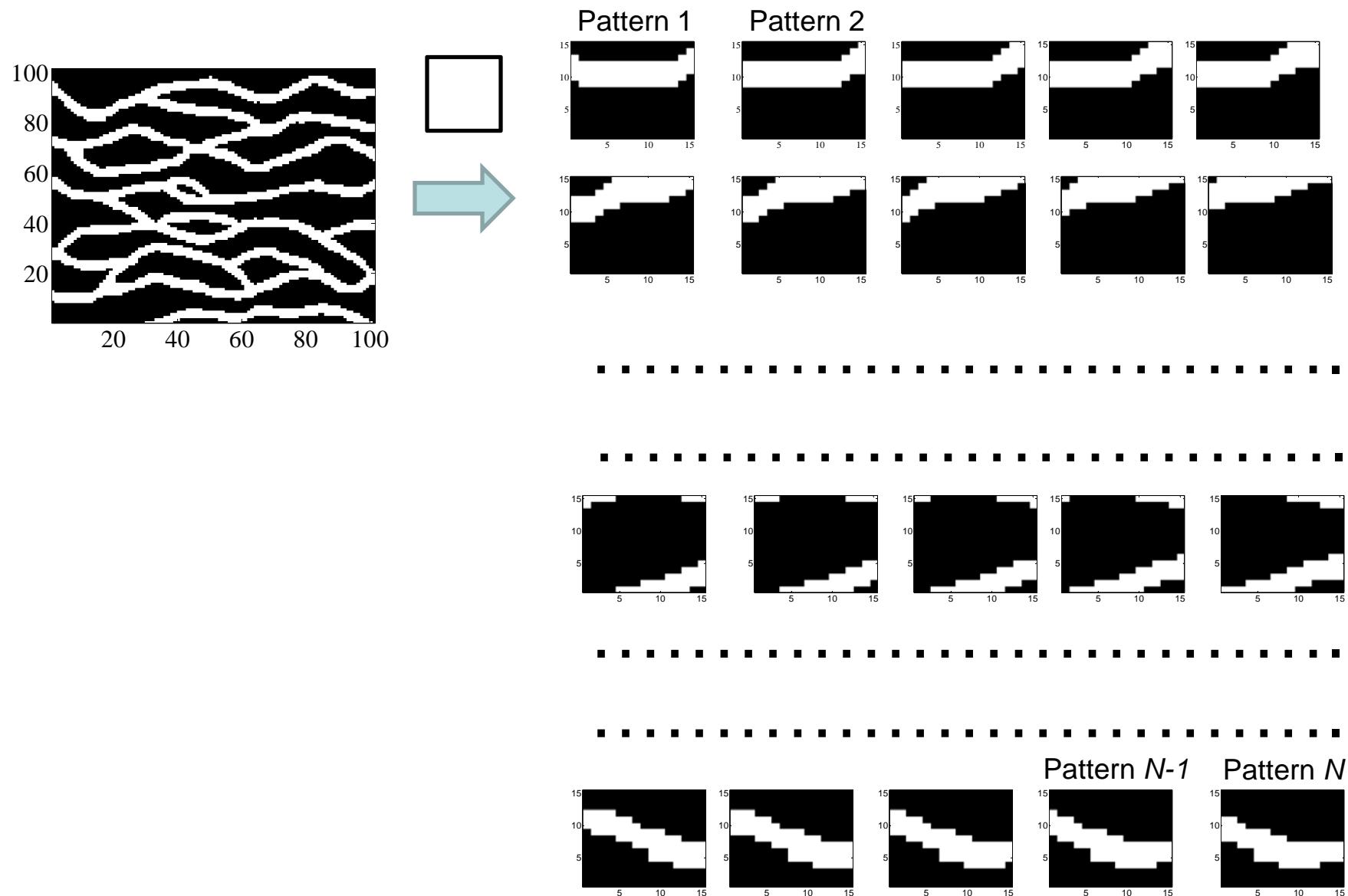
Pattern classification
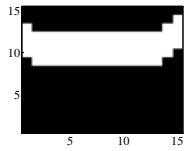
Pattern 1   Pattern 2
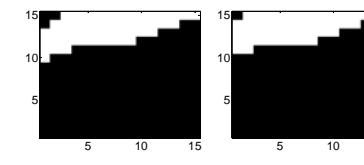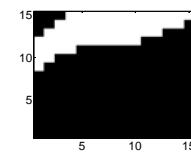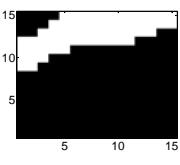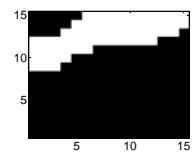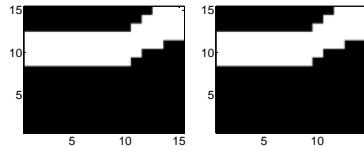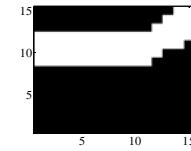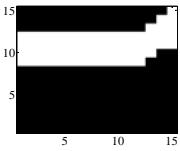
Class 1

Class 2
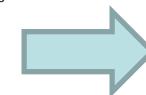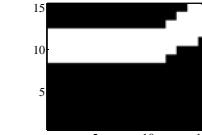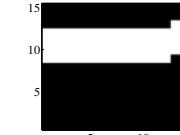
Class *n*
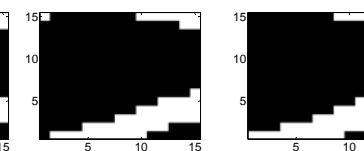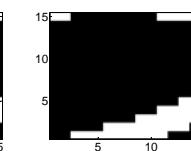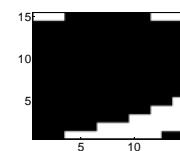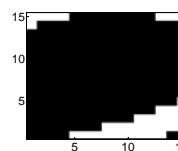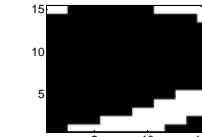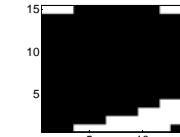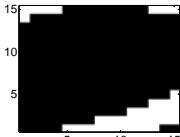
Pattern *N-1*   Pattern *N*
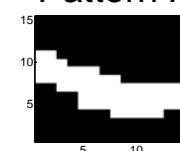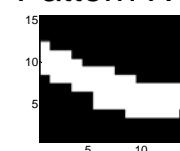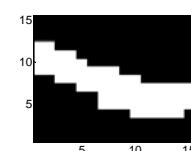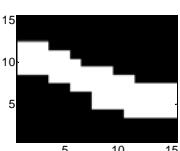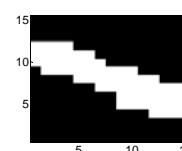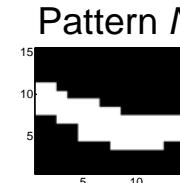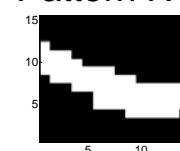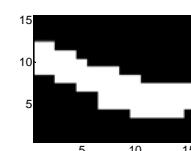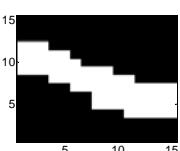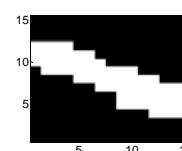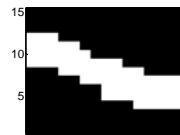
Prototype calculation

Distance calculation

Class 1 Class 2 Class *n*

Calculate minimum distance
of class prototype from condition
data; say, cluster 2 has minimum
distance.
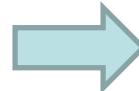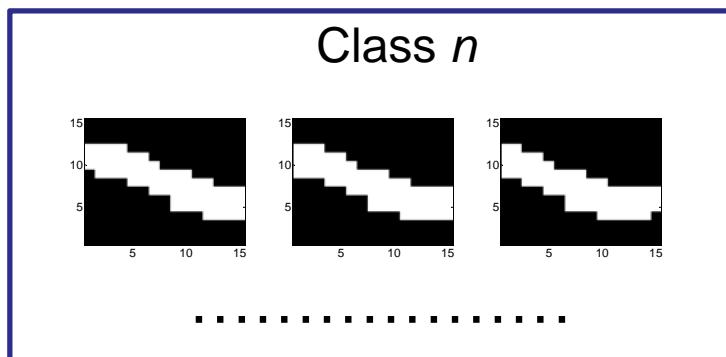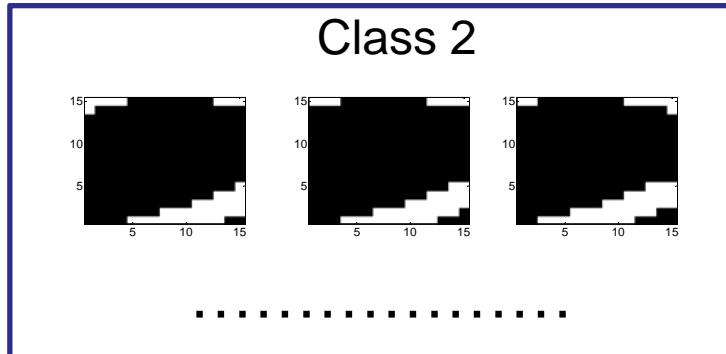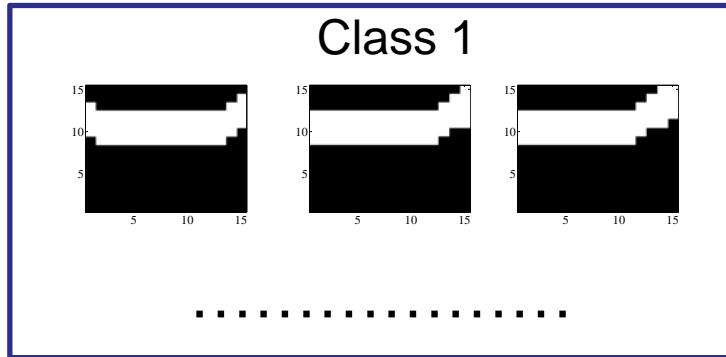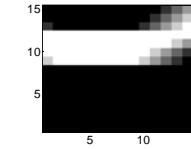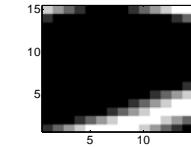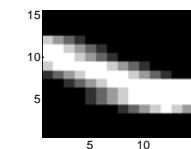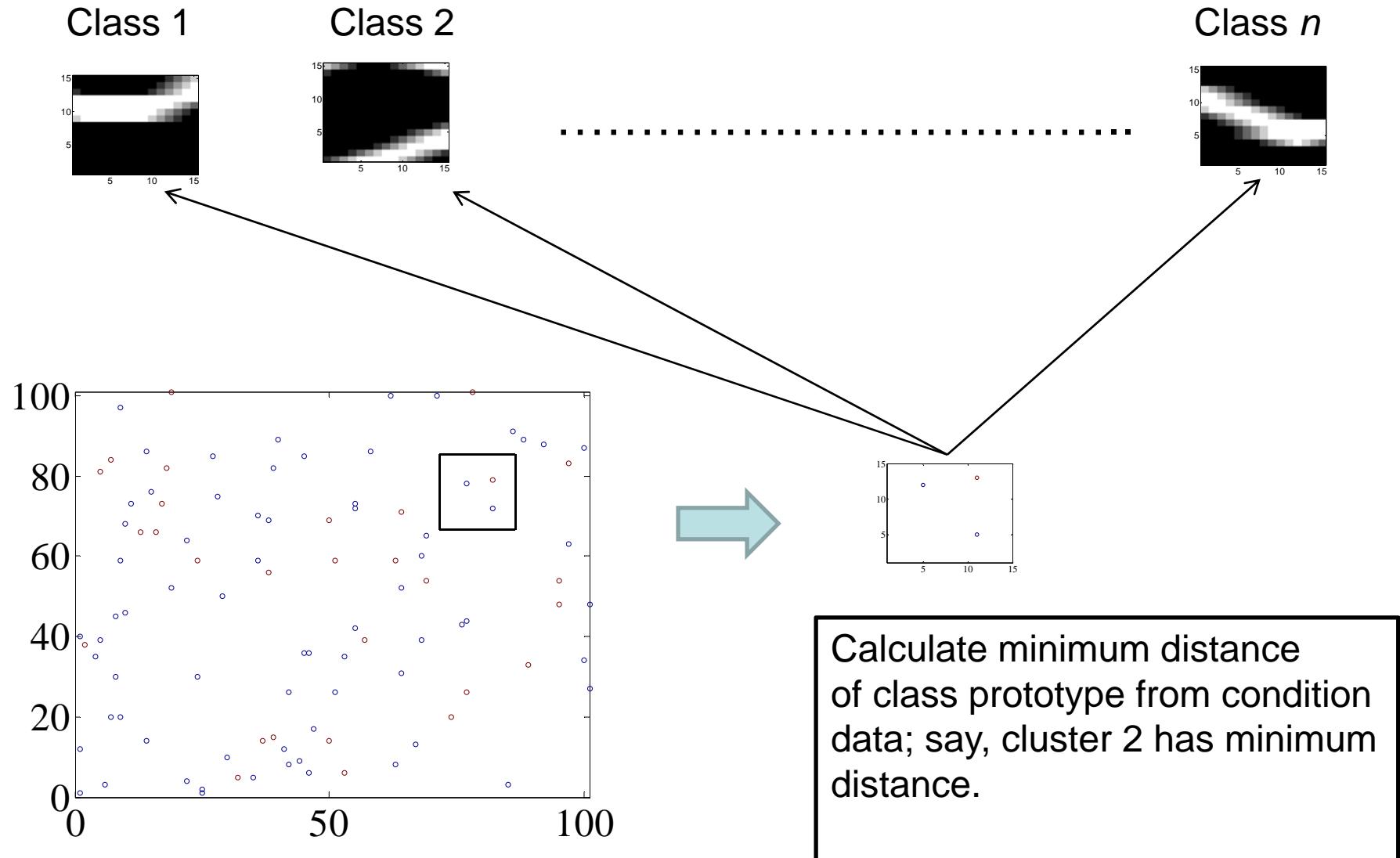
Pattern selection and pasting
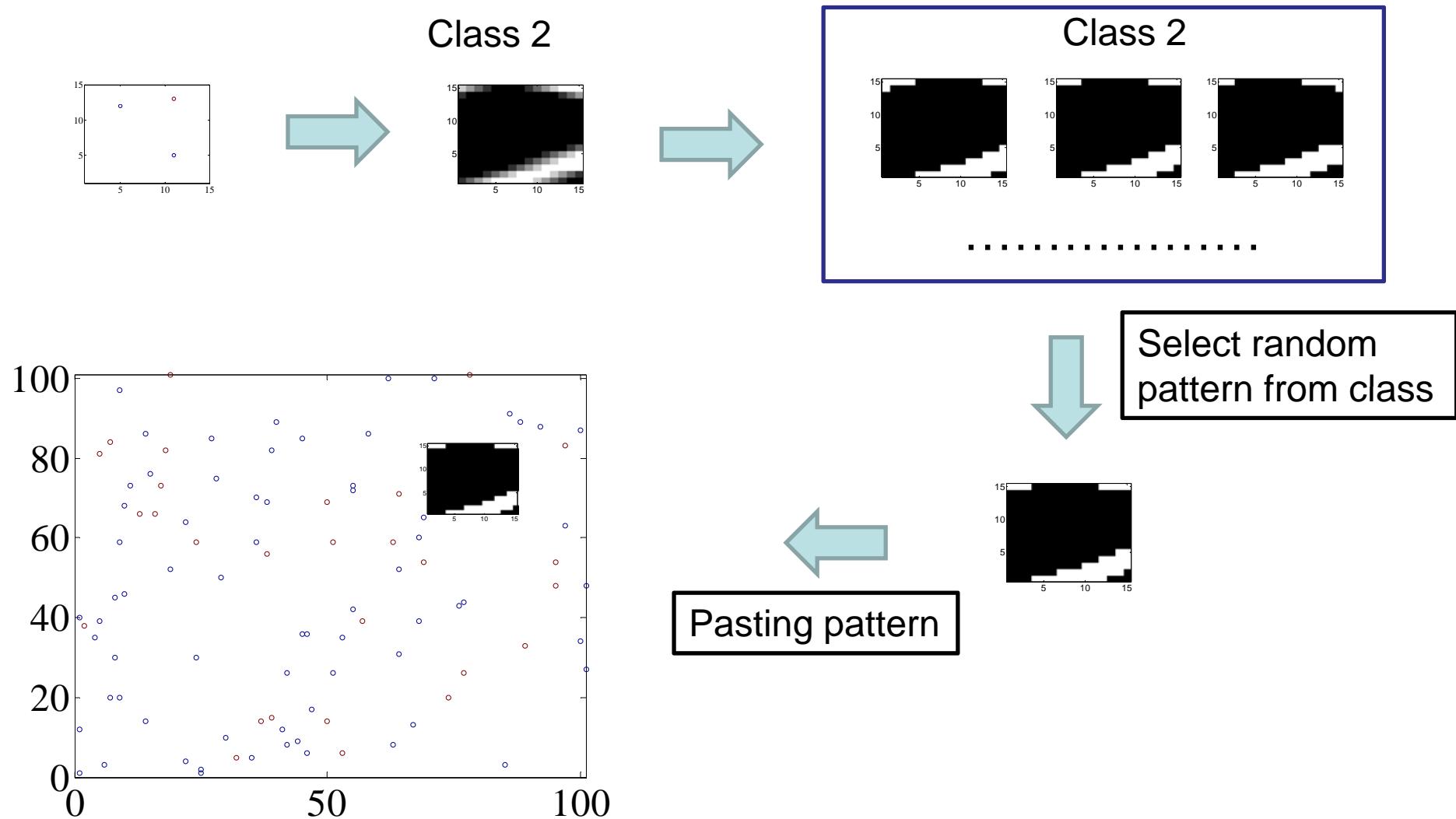
Pattern-based simulation algorithm

- Filtersim: Filter-based simulation

- Distsim: Distance-based simulation

- Wavesim: Wavelet-based simulation

How different algorithms are differ?

- Generation of a pattern database

- Pattern database classification

- Similarity measures between conditional data and class prototypes

- Pattern selection and pattern pasting

Filtersim algorithm
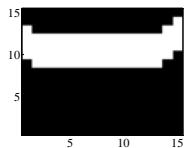
Pattern 1   Pattern 2
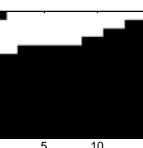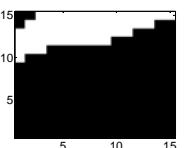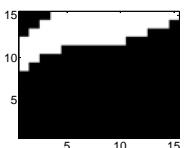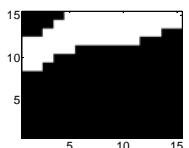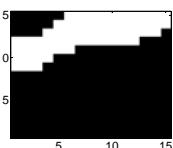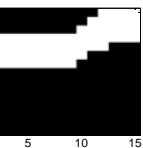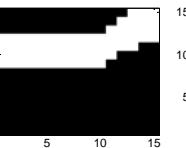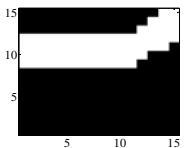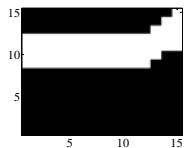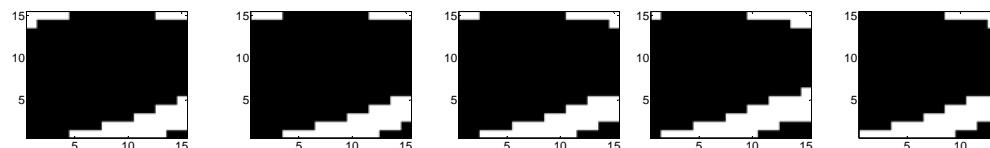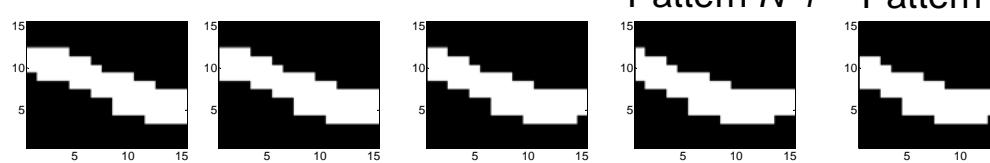
Pattern 1

F1	F2	F3	F4	F5	F6

Pattern 2

F1	F2	F3	F4	F5	F6

.
.
.
.
.
.

Pattern *N-1*

F1	F2	F3	F4	F5	F6

Pattern *N-1*   Pattern *N*

Pattern *N*

F1	F2	F3	F4	F5	F6

F is filter score

Distsim algorithm

Pattern 1   Pattern 2
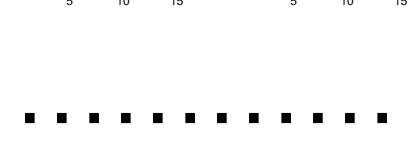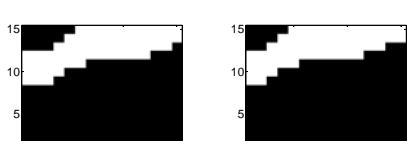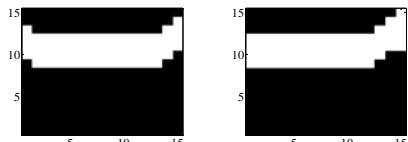
Distance Matrix

d11	d12	.	.	.	.	d1N
d21	d22	.	.	.	.	d2N
.	.		.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
dN1	dN2	.	.	.	.	dNN

Pattern *N-1*   Pattern *N*

dMN is distance from pattern M to pattern N

Wavesim algorithm

Pattern 1

a1	a2	.	.	.	ap

Pattern 2

a1	a2	.	.	.	ap

.

.

.

.

.

.

Pattern *N-1*

a1	a2	.	.	.	ap

Pattern *N*

a1	a2	.	.	.	ap

a is approximate coefficient
p is number of coefficients

Pattern classification using self organized maps

Pattern vector

Weight vector

2d array of neurons

$ti(u+h_1)$

$ti(u+h_2)$

$ti(u+h_3)$

$ti(u+h_{nT})$

w_{j1}

w_{j2}

w_{j3}
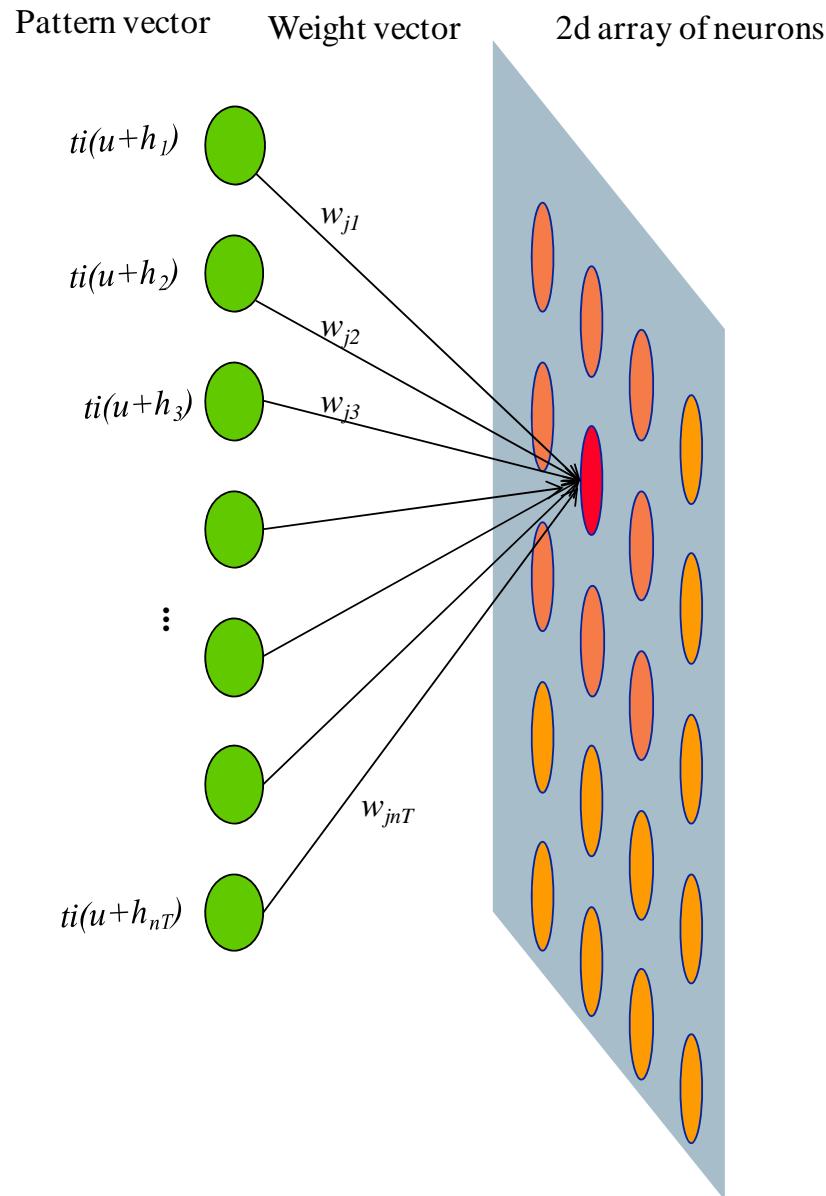
w_{jnT}

Ste 1 : Select self organized map topology with L neuron

Step 2: Initialize Random weight vectors for each neuron j

$$W_j = [w_{j1}, ..., w_{jn_T}]^T$$

Step 3: Determine best matching unit (BMU)

$$\left\| ti(u)_T - W_b \right\| = \min_j \left\| ti(u)_T - W_j \right\|$$

Step 4: Adjusts weight vectors of the BMU

$$W_j(k+1) = W_j(k) + \alpha(k)h_{bj}(k)[ti(u)_T - W_j(k)]$$

$$h_{bj}(k) = \exp(-\frac{\left\| r_b - r_j \right\|^2}{2\sigma^2(k)})$$

Step 5: Decrease $\alpha(k)$ and $\sigma(k)$

Step 6: Go to step 3 until training is done

Clustering of self organized maps

N patterns each with nT dimension   Mapped 2-D SOM with L (24) nodes   Classified SOM in P (4) clusters
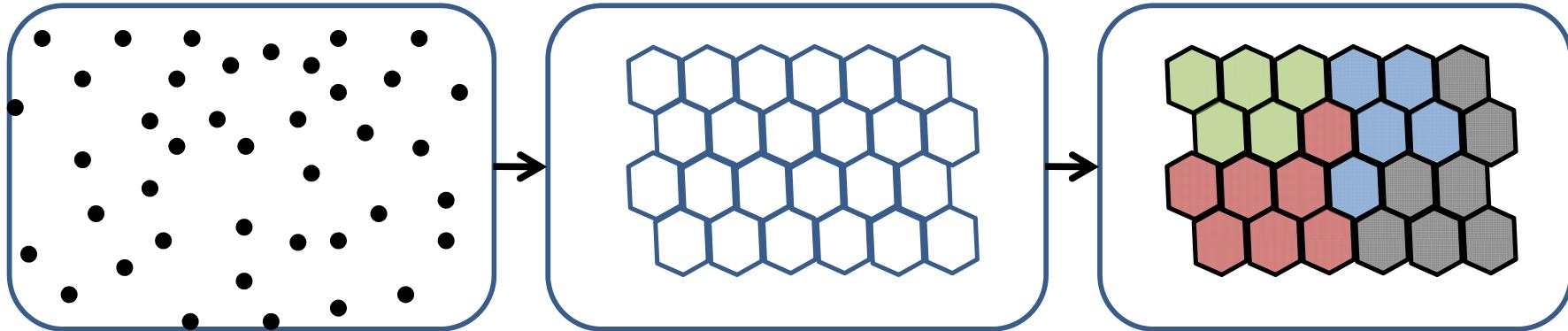
- K-means clustering algorithm is used
- The Davies–Bouldin index is calculated for selecting optimum cluster number

$$\frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left\{\frac{S\ (Q_i)+S\ (Q_j)}{D\ (Q_i,Q_j)}\right\}$$

S is within the cluster distance, D is between the cluster distance, and k is the number of clusters

The proposed Simulation algorithm

Step 1: Scan the training image ti using the given template T. Store the patterns in the pattern database patdbT.

Step 2: Map the high-dimensional patterns to a 2-dimensional space using a self-organized map.

Step 3: Classify the SOM's reference vectors (prototypes) using the k-means clustering algorithm. The patterns are assigned to the same cluster as their nearest prototype.

Step 4: Calculate class prototypes for all classes.

Step 5: Define a random path visiting all unsampled nodes once and only once.

The proposed Simulation algorithm

Step 6: The distance from the class prototype to the conditioning data is calculated. Select the most similar class. If no conditioning data are available, a random class is selected.

Step 7: Draw a random pattern from the most similar class and paste the pattern by centering the simulated location u.

Step 8: Add a simulated value at point u to a different file to use during distance calculation.

**Step 9: Repeat Steps 6 to 8 for the next points in the random path defined in Step (5).**

Step 10: Repeat Steps 5 to 9 to generate different realizations using different random paths.

Validation of the method

## Unconditional simulation with binary image

Cluster number selection

Cluster number 50
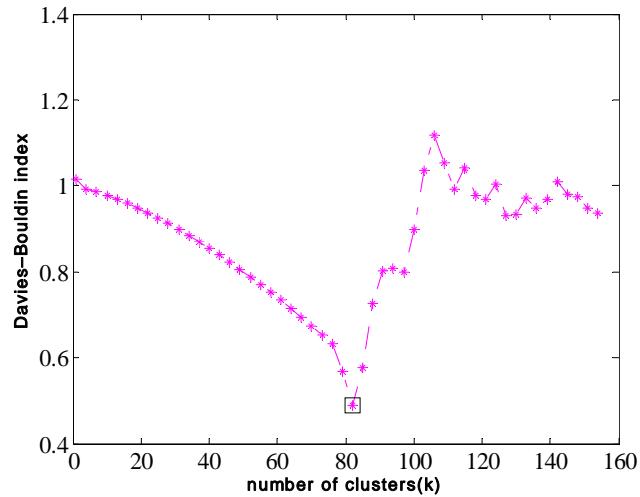
Cluster number 84

Cluster number 150
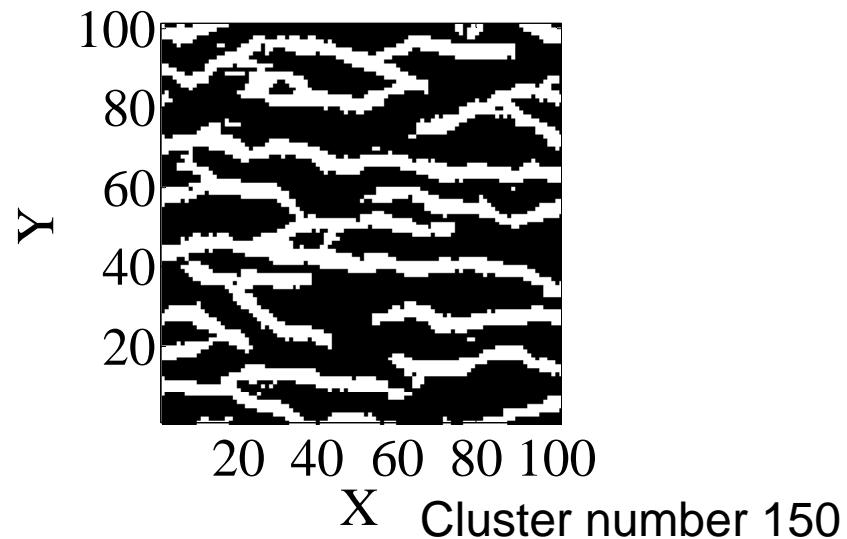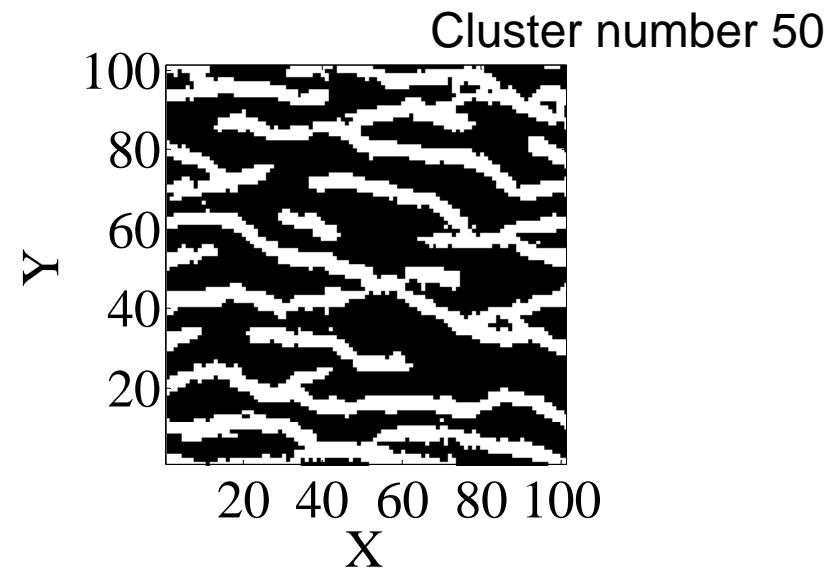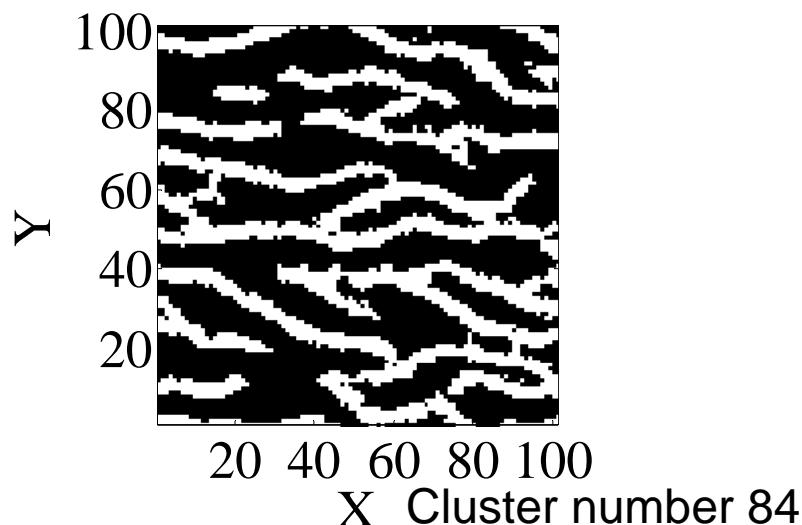
Validation of the method

Our proposed approach
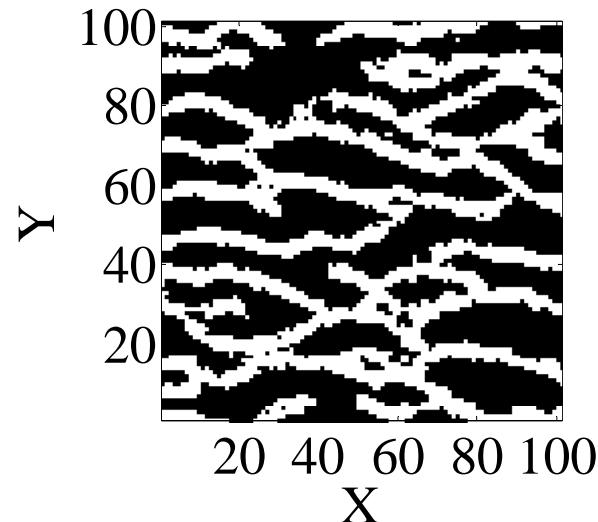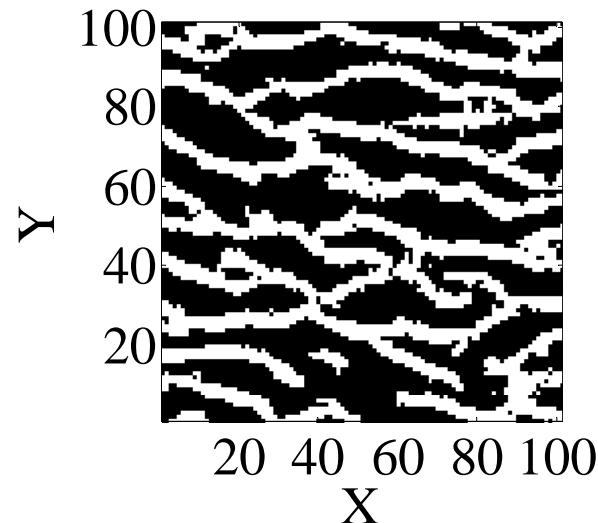
FILTERSIM
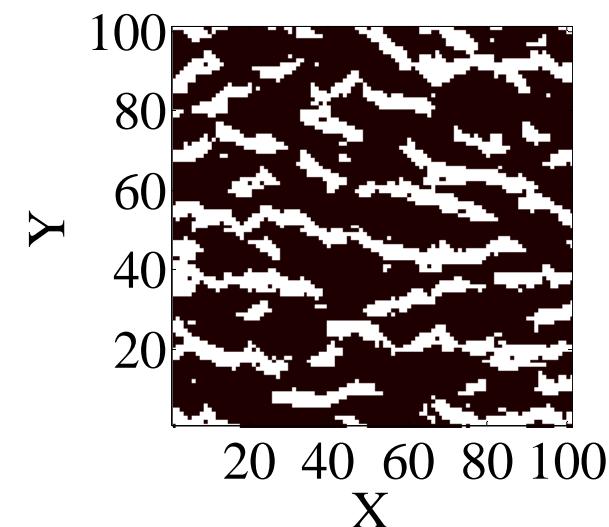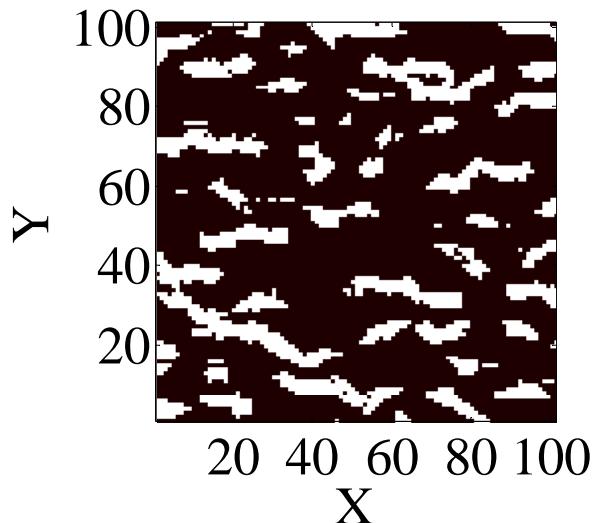
Validation of the method

Conditional simulation with categorical image

Training image

Reference image

Channel

Non-channel

Proposed method

Filtersim method
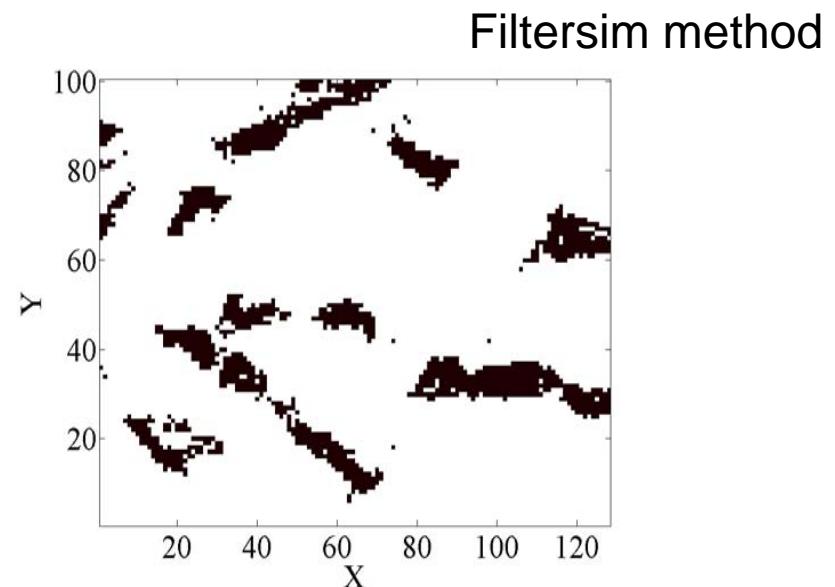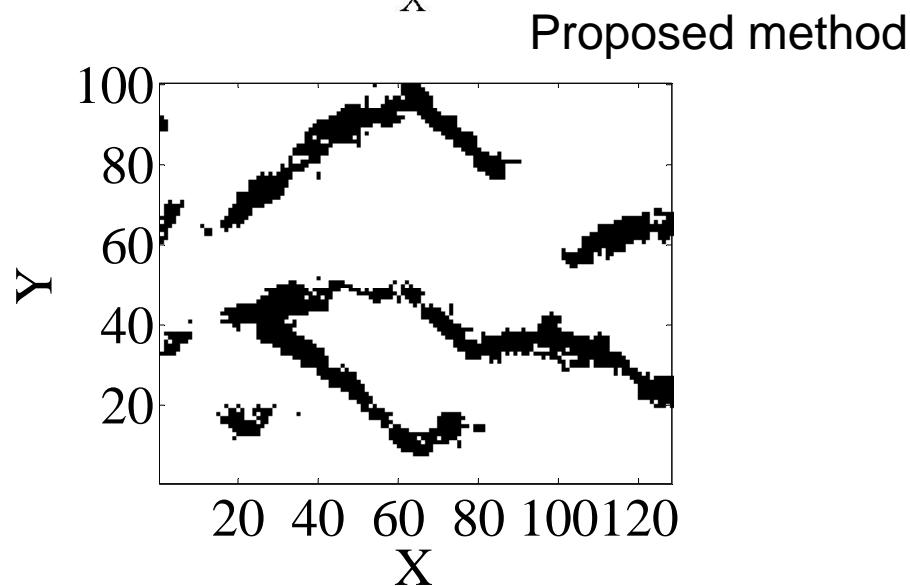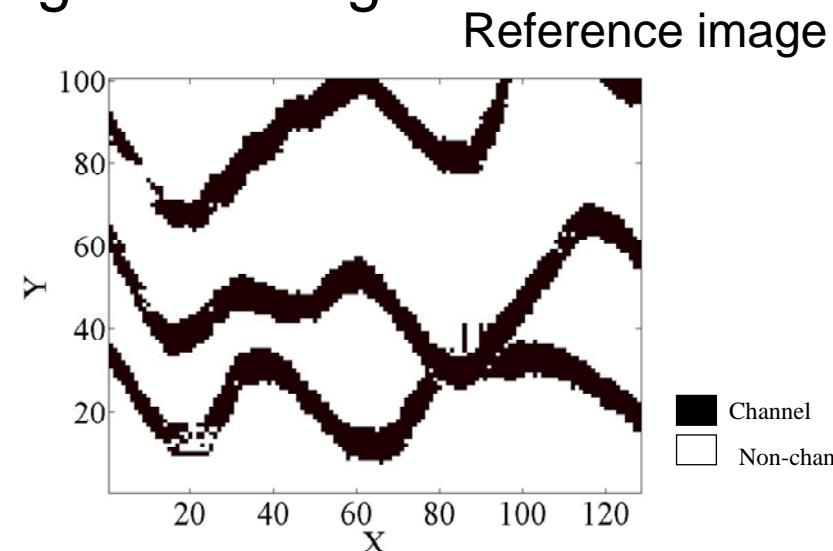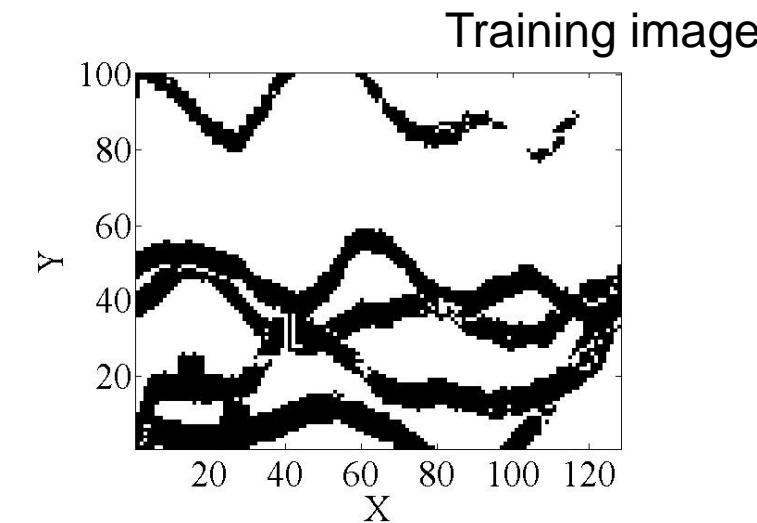
Conclusions

- The high dimensional patterns are projected in 2-dimension by non-linear mapping function

- Topology preservation of SOM helps to guide the pattern classification of the projected data.

- Automatic automatic cluster number selection helps to select the optimum parameters for the algorithm

- The algorithm is verified by different examples and compared with filtersim

Thank you