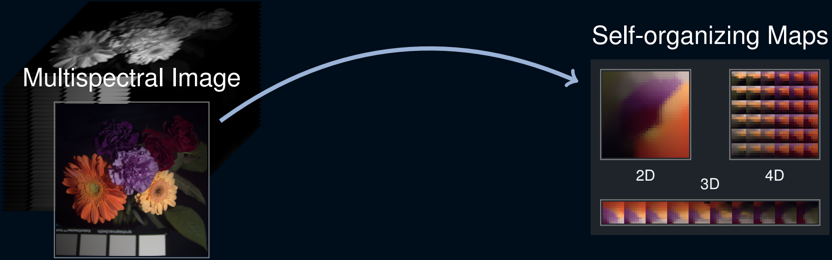
How I Learned to Stop Worrying and Love the SOM

Johannes Jordan
Feb 6, 2015
Pattern Recognition Lab, University of Erlangen-Nuremberg

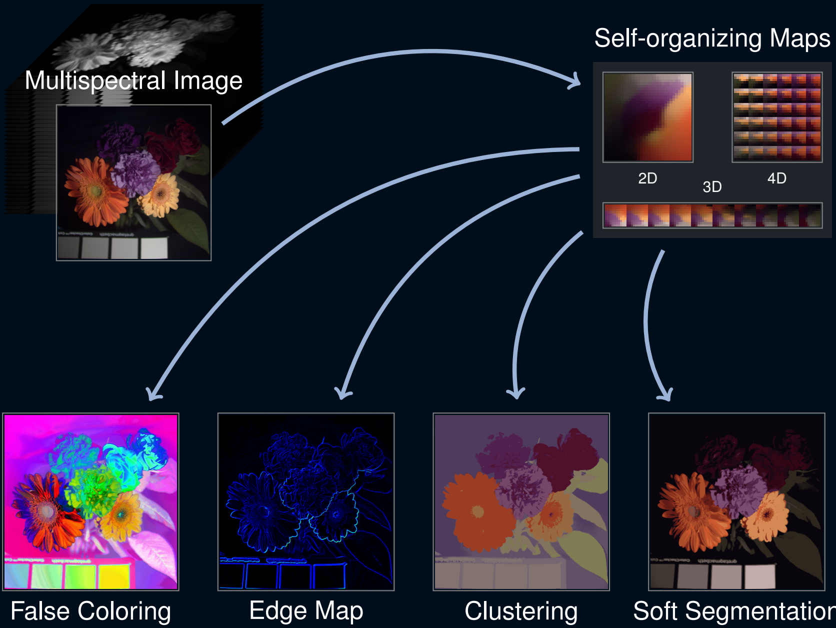
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Motivation

Motivation

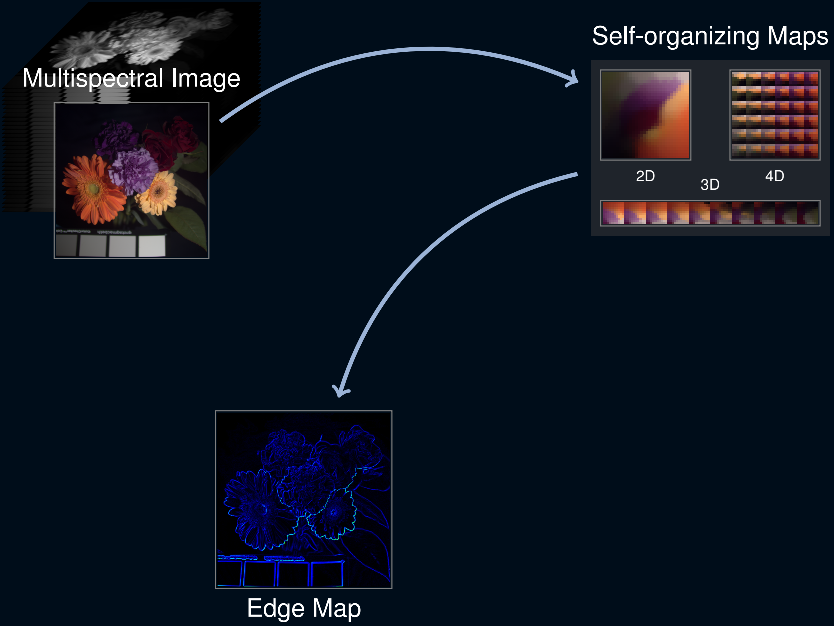


Motivation



Multispectral Image
Self-organizing Maps
2D
3D
4D
False Coloring
Edge Map
Clustering
Soft Segmentation

Motivation



Multispectral Gradient Calculation

True-color rendering

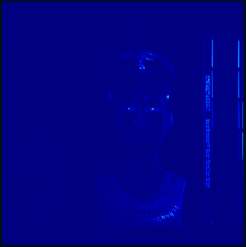
False-color rendering

Multispectral Gradient Calculation

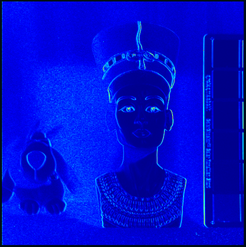
True-color rendering

False-color rendering

Established measures for spectral vector distance: ∇_x



Euclidean Distance

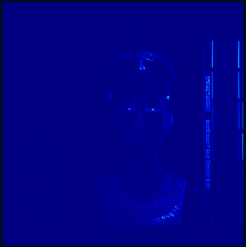


Spectral Angle

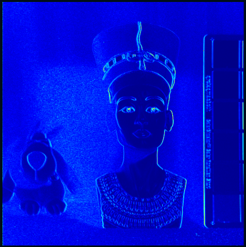
Spectral Information Divergence (SID)

Multispectral Gradient Calculation

Established measures for spectral vector distance: ∇_x



Euclidean Distance



Spectral Angle

Spectral Information Divergence (SID)

Observations:

1. Euclidean distance misses crucial details
2. Spectral Angle and SID are prone to noise in dark regions

Multispectral Edge Detection

Canny: 1986, Gradient-based

1. Sobel / Gaussian filter
2. Non-maximum suppression
3. Edge tracking with double threshold

Multispectral Edge Detection

Canny: 1986, Gradient-based

1. Sobel / Gaussian filter
2. Non-maximum suppression
3. Edge tracking with double threshold

RCMG: 2004, Ordering-based

1. Compute pair distances in pixel neighborhood
2. Reject outlier vector pairs
3. Take maximum distance

Multispectral Edge Detection

Canny: 1986, Gradient-based

1. Sobel / Gaussian filter
2. Non-maximum suppression
3. Edge tracking with double threshold

RCMG: 2004, Ordering-based

1. Compute pair distances in pixel neighborhood
2. Reject outlier vector pairs
3. Take maximum distance

$\rightarrow$ Both methods rely on spectral vector distance

Multispectral Edge Detection

Canny: 1986, Gradient-based

1. Sobel / Gaussian filter
2. Non-maximum suppression
3. Edge tracking with double threshold

RCMG: 2004, Ordering-based

1. Compute pair distances in pixel neighborhood
2. Reject outlier vector pairs
3. Take maximum distance

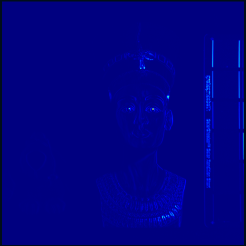
$\rightarrow$ Both methods rely on spectral vector distance

Vector Order Statistics: 1993, 2003

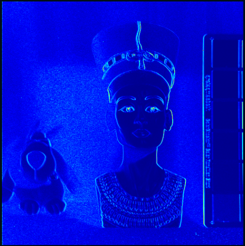
- Local R-ordering: *see RCMG*
- Global Ordering (SOM):
  Ordering is flawed
  False edges appear

Challenges

True color



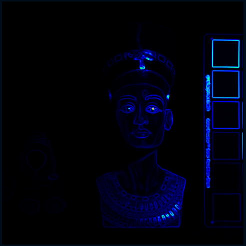
Euclidean Distance



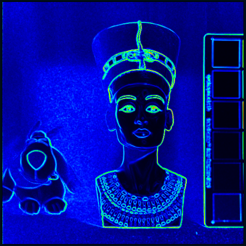
Spectral Angle

Challenges

True color



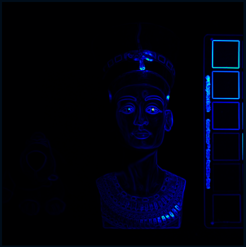
RCMG on L_2



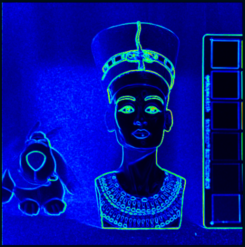
RCMG on Spectral Angle

Challenges

True color

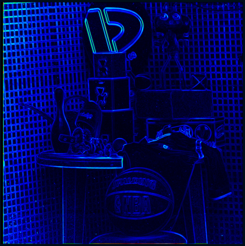


RCMG on L_2



RCMG on Spectral Angle

True color

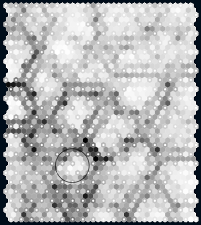


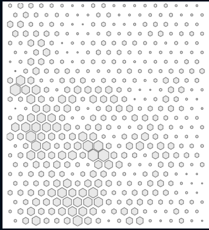
RCMG on L_2

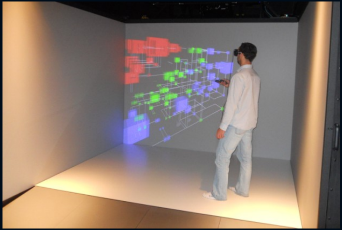
RCMG on Spectral Angle

Self-organizing Map

The Self-organizing Map (SOM) is a tool for data visualization:

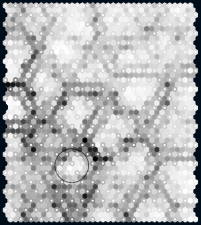
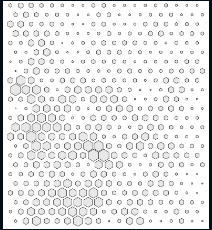
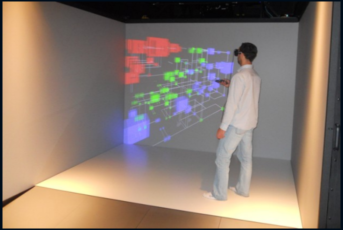






Self-organizing Map

The Self-organizing Map (SOM) is a tool for data visualization:

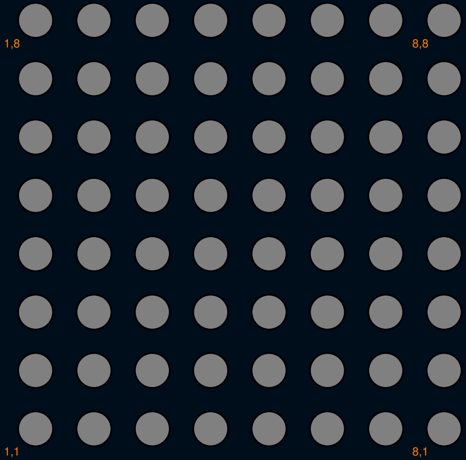
SOM design:

- n-dimensional mesh of *model vectors (neurons)*
- Mapping: spectral vector $\mathbf{x} \rightarrow$ model vector $\mathbf{m}_i$ (L_2 distance)
- Topological representation of original spectral distribution
  Typical topology: hexagon lattice

Learning

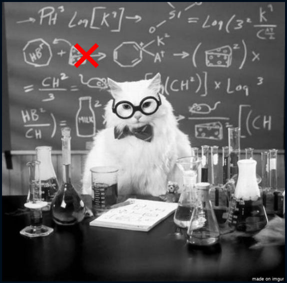
Input Image (Multispectral)

Initial State: Random



Self-organizing Map (8x8 Lattice)

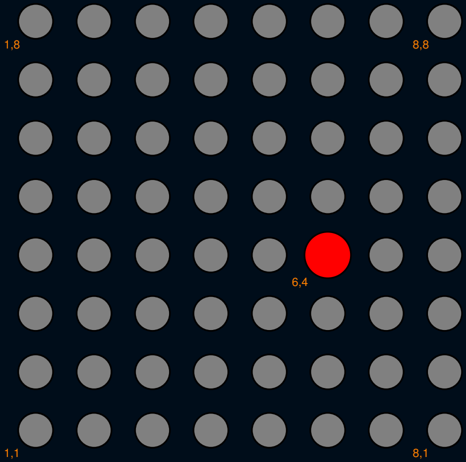
Learning



Input Image (Multispectral)

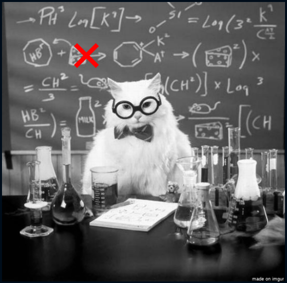
First Iteration

1. Find best match (BMU)



Self-organizing Map (8x8 Lattice)

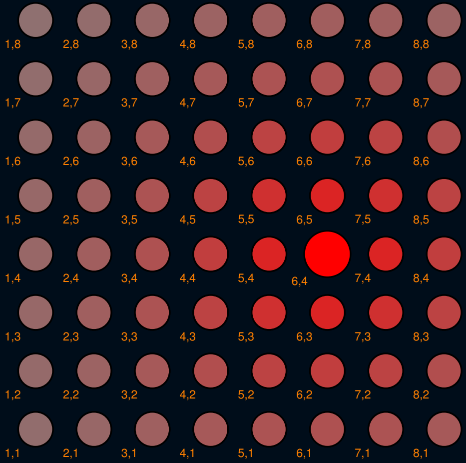
Learning



Input Image (Multispectral)

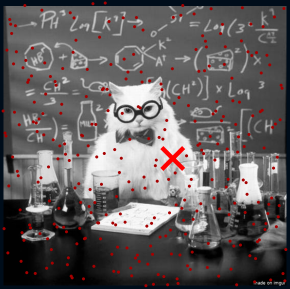
First Iteration

2. Update neighborhood



Self-organizing Map (8x8 Lattice)

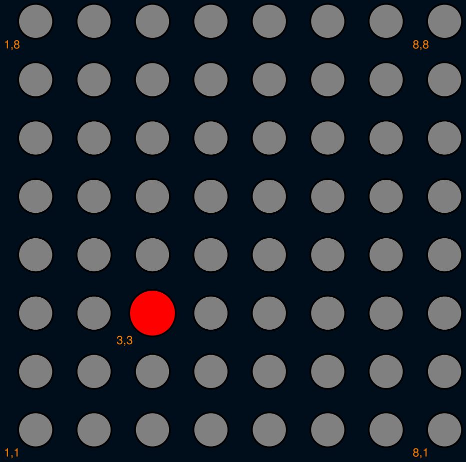
Learning



Input Image (Multispectral)

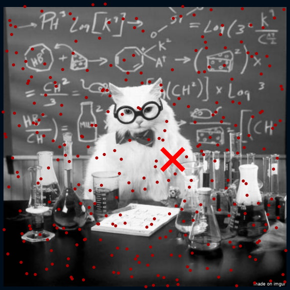
Iteration at 33%

1. Find best match (BMU)



Self-organizing Map (8x8 Lattice)

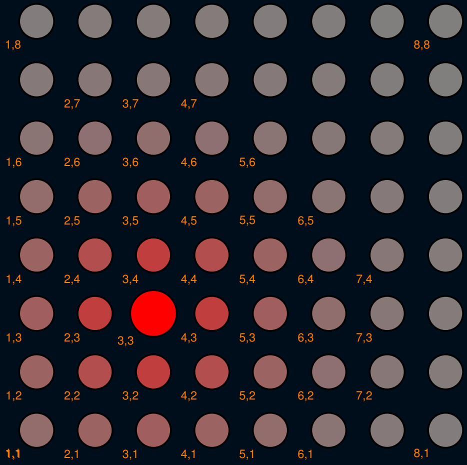
Learning



Input Image (Multispectral)

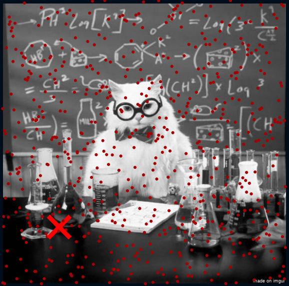
Iteration at 33%

2. Update neighborhood



Self-organizing Map (8x8 Lattice)

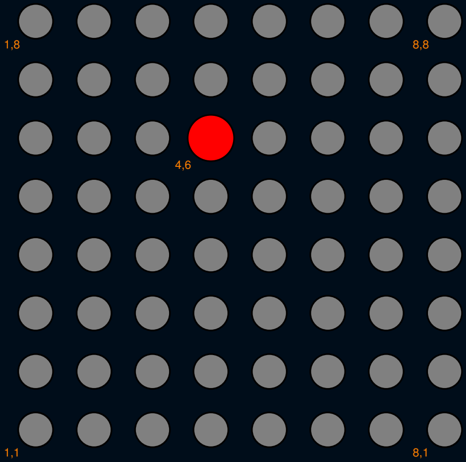
Learning



Input Image (Multispectral)

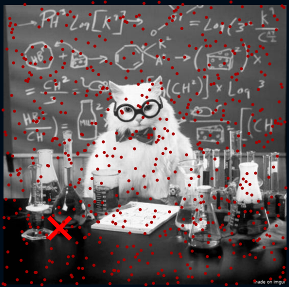
Iteration at 67%

1. Find best match (BMU)



Self-organizing Map (8x8 Lattice)

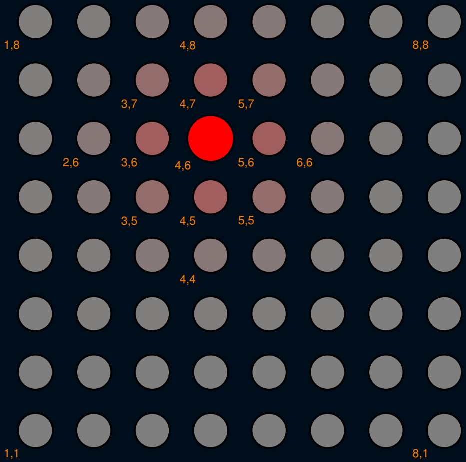
Learning



Input Image (Multispectral)

Iteration at 67%

2. Update neighborhood



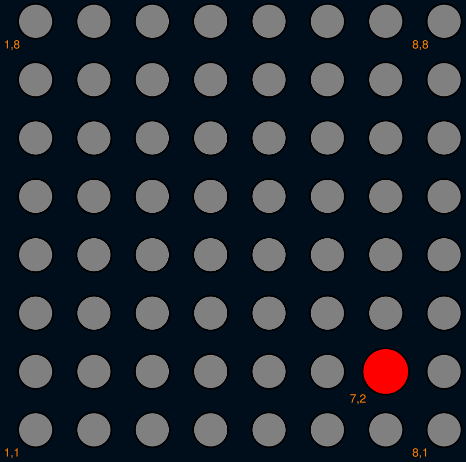
Self-organizing Map (8x8 Lattice)

Learning

Input Image (Multispectral)

Last Iteration

1. Find best match (BMU)



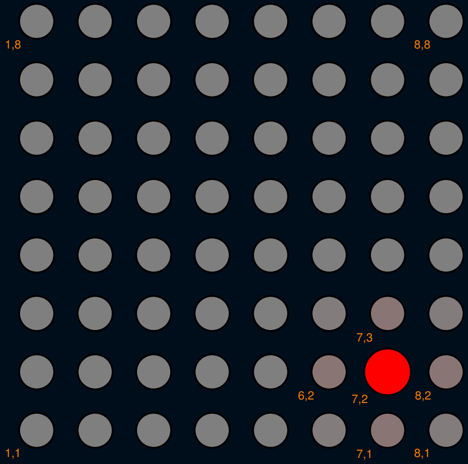
Self-organizing Map (8x8 Lattice)

Learning

Input Image (Multispectral)

Last Iteration

2. Update neighborhood



Self-organizing Map (8x8 Lattice)

Pushing the SOM further

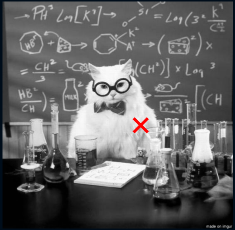
We adapt the SOM for new purposes:

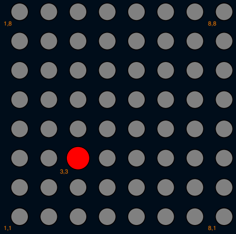
1. Increase size: typically 64 to 256 neurons, now 1000 to 4000 neurons
   $\rightarrow$ reduces quantization problem
2. Increase dimensionality: Train cubes and tesseracts
   $\rightarrow$ improves manifold representation

Pushing the SOM further

We adapt the SOM for new purposes:

1. Increase size: typically 64 to 256 neurons, now 1000 to 4000 neurons
   $\rightarrow$ reduces quantization problem
2. Increase dimensionality: Train cubes and tesseracts
   $\rightarrow$ improves manifold representation
3. Lift *winner takes it all* concept:

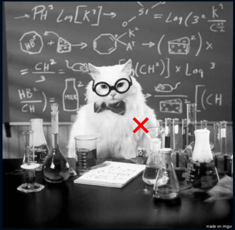


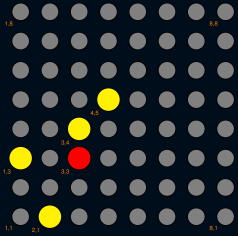


Pushing the SOM further

We adapt the SOM for new purposes:

1. Increase size: typically 64 to 256 neurons, now 1000 to 4000 neurons
   → reduces quantization problem
2. Increase dimensionality: Train cubes and tesseracts
   → improves manifold representation
3. Lift *winner takes it all* concept:





Pushing the SOM further

We adapt the SOM for new purposes:

1. Increase size: typically 64 to 256 neurons, now 1000 to 4000 neurons
   → reduces quantization problem
2. Increase dimensionality: Train cubes and tesseracts
   → improves manifold representation
3. Lift *winner takes it all* concept
   → extends usability

We implement it efficiently:

- Exploit radially symmetric kernel
- Parallelized, heap-based readout

Pushing the SOM further

We adapt the SOM for new purposes:

1. Increase size: typically 64 to 256 neurons, now 1000 to 4000 neurons
   → reduces quantization problem
2. Increase dimensionality: Train cubes and tesseracts
   → improves manifold representation
3. Lift *winner takes it all* concept
   → extends usability

We implement it efficiently:

- Exploit radially symmetric kernel
- Parallelized, heap-based readout

→ Training a 10x10x10 SOM takes less than 4 seconds

Pushing the SOM further

We adapt the SOM for new purposes:

1. Increase size: typically 64 to 256 neurons, now 1000 to 4000 neurons
   $\rightarrow$ reduces quantization problem
2. Increase dimensionality: Train cubes and tesseracts
   $\rightarrow$ improves manifold representation
3. Lift *winner takes it all* concept
   $\rightarrow$ extends usability

We implement it efficiently:

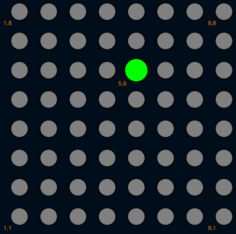
- Exploit radially symmetric kernel
- Parallelized, heap-based readout

$\rightarrow$ Training a 10x10x10 SOM takes less than 4 seconds

But how does it help edge detection?

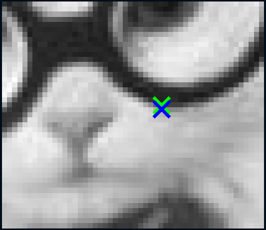
SOM Edge Detection – Sniper Approach

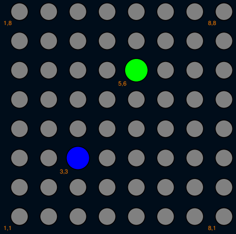
Case I: Gradient



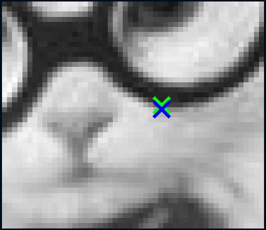
SOM Edge Detection – Sniper Approach

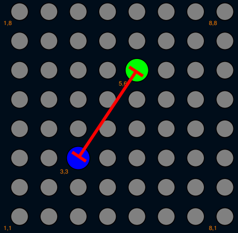
Case I: Gradient





SOM Edge Detection – Sniper Approach

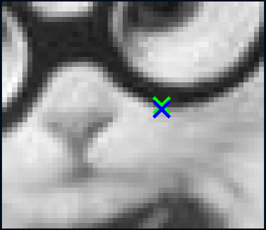


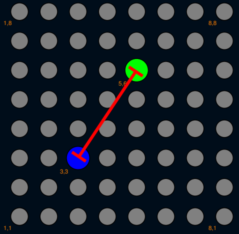


Case I: Gradient

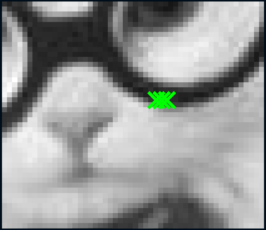
SOM Edge Detection – Sniper Approach

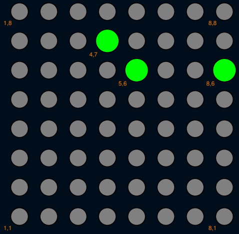
Case I: Gradient





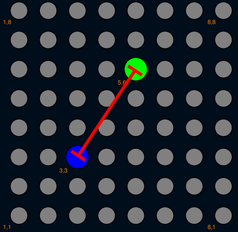
Case II: Sobel



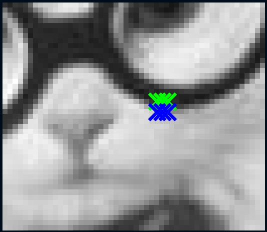


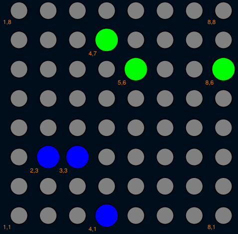
SOM Edge Detection – Sniper Approach

Case I: Gradient

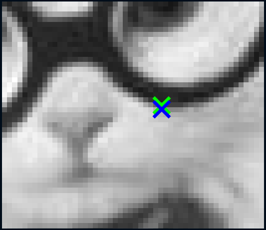


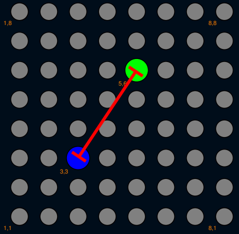
Case II: Sobel



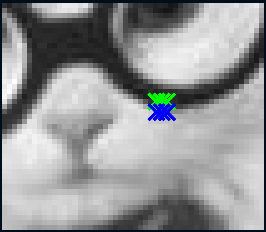


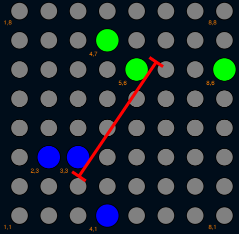
SOM Edge Detection – Sniper Approach





Case I: Gradient





Case II: Sobel

Problems with the sniper approach

Material representation in the SOM:

- depends on #pixels in the scene
- depends on material variation (noise!)

Problems with the sniper approach

Material representation in the SOM:

- depends on #pixels in the scene
- depends on material variation (noise!)

$\rightarrow$ Distances in the SOM are not robust

Gradient Magnitude

Canny Binary Map

Solution: Shotgun Approach

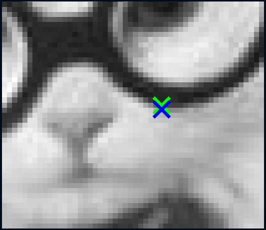
Exploit more information in the SOM!

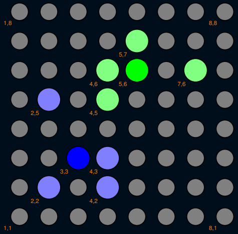
- Query for several BMUs for each pixels
- Compare the coordinate sets (not BMU distance)

Solution: Shotgun Approach

Exploit more information in the SOM!

- Query for several BMUs for each pixels
- Compare the coordinate sets (not BMU distance)



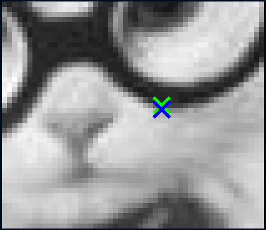


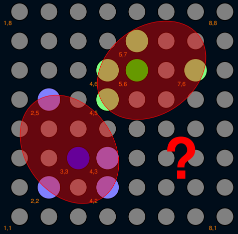
Case I: Gradient

Solution: Shotgun Approach

Exploit more information in the SOM!

- Query for several BMUs for each pixels
- Compare the coordinate sets (not BMU distance)





Case I: Gradient

Comparing Coordinate Sets

Idea: Coordinate set distance as a minimization problem

1. Pair coordinates from each set
2. Calculate sum of pair distances
3. Find optimal pairings w.r.t sum of pair distances

Comparing Coordinate Sets

Idea: Coordinate set distance as a minimization problem

1. Pair coordinates from each set
2. Calculate sum of pair distances
3. Find optimal pairings w.r.t sum of pair distances

The Lehrstuhlaufgaben problem:

- Each lab member needs a specific time for each task
- We need to assign each member to a task
- We minimize the total time spent on tasks

Comparing Coordinate Sets

Idea: Coordinate set distance as a minimization problem

1. Pair coordinates from each set
2. Calculate sum of pair distances
3. Find optimal pairings w.r.t sum of pair distances

The Lehrstuhlaufgaben problem:

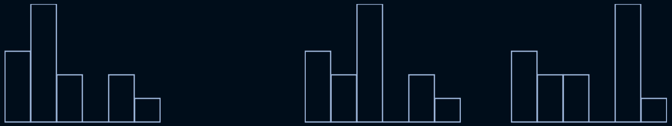
- Each lab member needs a specific time for each task
- We need to assign each member to a task
- We minimize the total time spent on tasks
- Matching in a bi-partite graph with weighted edges
- **Solution:** Hungarian Algorithm $O(n^3)$ ☹

Comparing Coordinate Sets (2)

The Earth Mover's Distance:

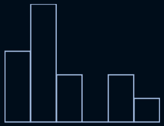
Comparing Coordinate Sets (2)

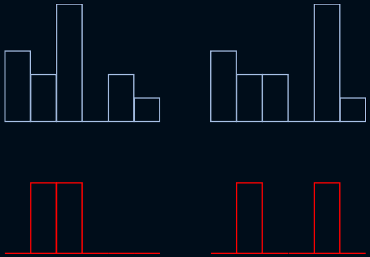
The Earth Mover's Distance:



Comparing Coordinate Sets (2)

The Earth Mover's Distance:

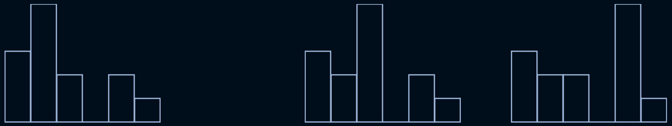




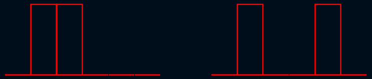
Element-wise comparison:

Comparing Coordinate Sets (2)

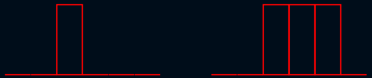
The Earth Mover's Distance:



Element-wise comparison:

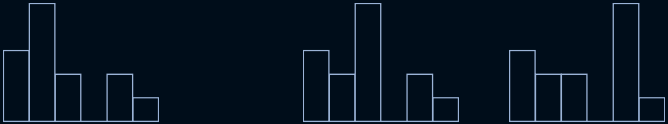


Earth Mover's Distance:

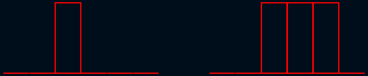


Comparing Coordinate Sets (2)

The Earth Mover's Distance:



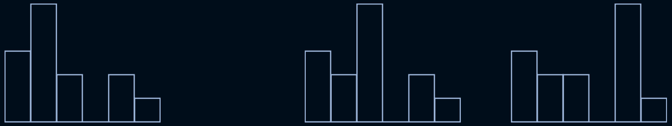
Earth Mover's Distance:



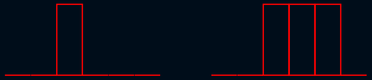
- **Solution:** Hungarian Algorithm

Comparing Coordinate Sets (2)

The Earth Mover's Distance:

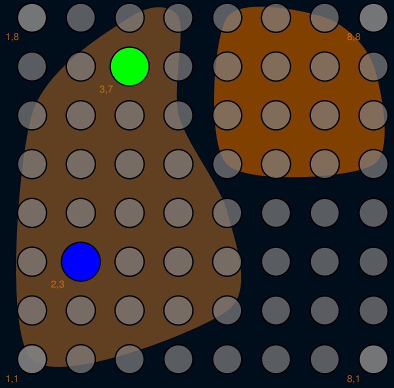


Earth Mover's Distance:



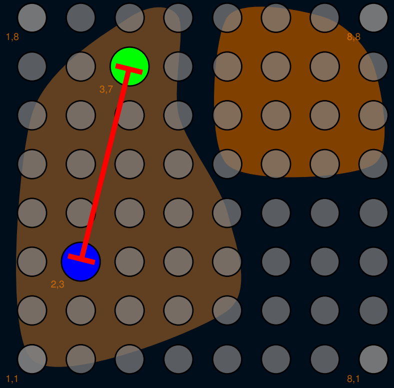
- **Solution:** Hungarian Algorithm
- Variant with $O(n^2)$
- Approximation using wavelets with $O(n)$

Sniper vs Shotgun



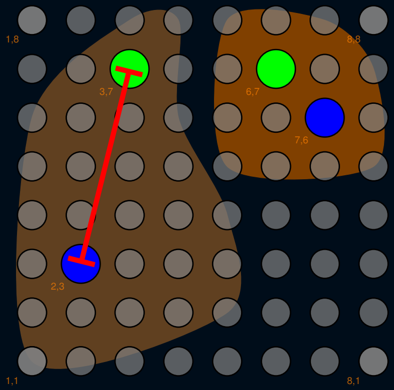
Case 1: No edge   Sniper

Sniper vs Shotgun



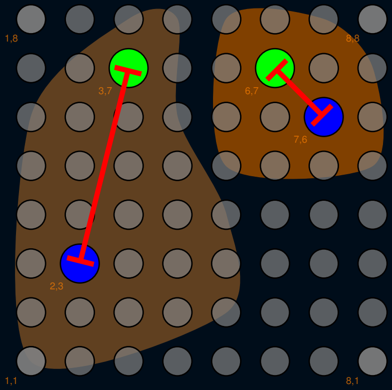
Case 1: No edge   Sniper

Sniper vs Shotgun



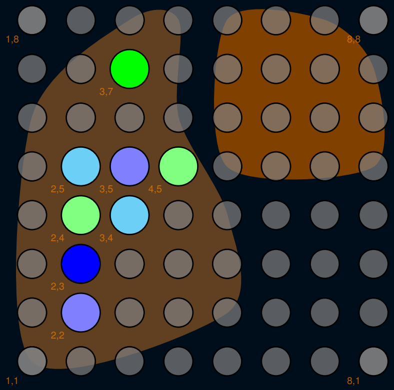
Case 1: No edge Sniper

Sniper vs Shotgun



Case 1: No edge   Sniper

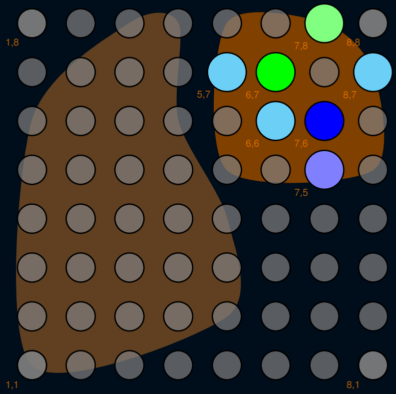
Sniper vs Shotgun



Case 1: No edge Shotgun

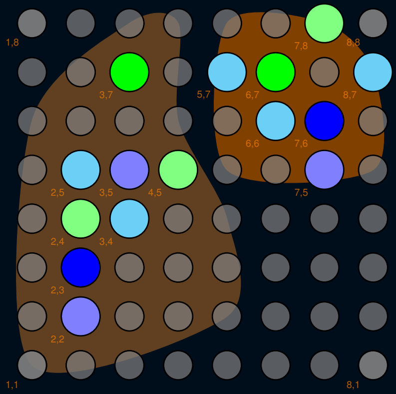
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Sniper vs Shotgun



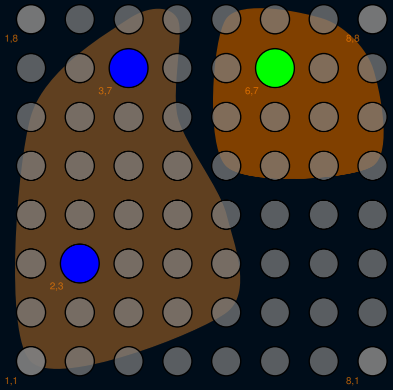
Case 1: No edge   Shotgun

Sniper vs Shotgun



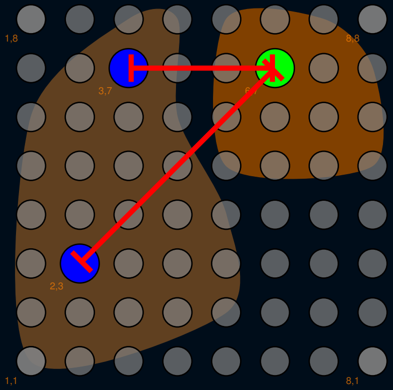
Case 1: No edge Shotgun

Sniper vs Shotgun



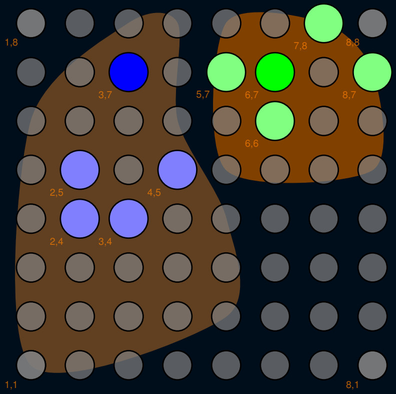
Case 2: Edge Sniper

Sniper vs Shotgun



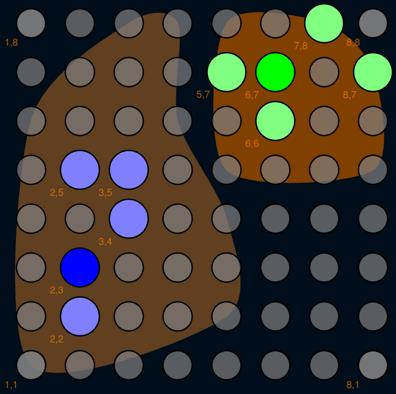
Case 2: Edge Sniper

Sniper vs Shotgun



Case 2: Edge Shotgun

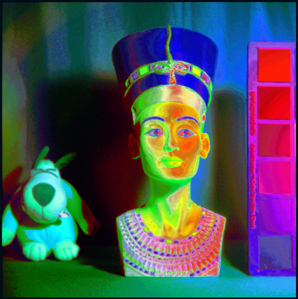
Sniper vs Shotgun

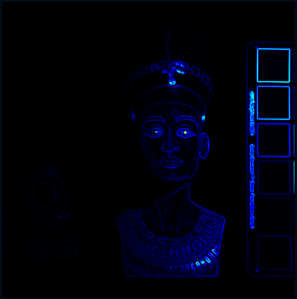


Case 2: Edge Shotgun

Evaluation: CAVE Egyptian Statue

False-color rendering





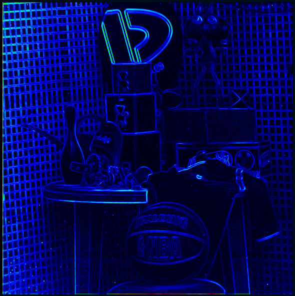
RCMG L_2

SOM Sniper

SOM Shotgun

Evaluation: Foster's Scene5 (2002)

True-color rendering

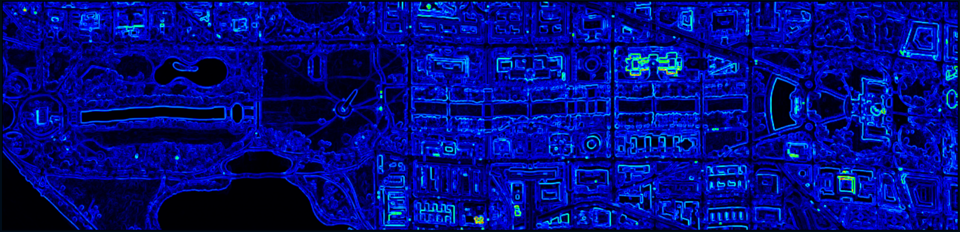


RCMG L_2

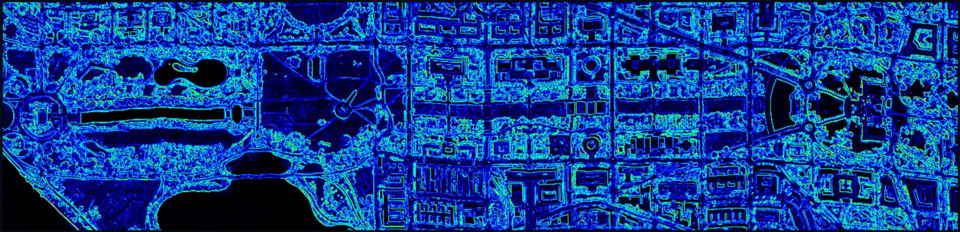
SOM Shotgun

SOM Shotgun RCMG

Evaluation: D.C. Mall



RCMG



SOM Shotgun 25

Conclusions

We investigated:

- Challenges of gradient calculation on multispectral images
- Multispectral and hyperspectral edge detection
- Adaptations of the Self-organizing map (SOM)

Conclusions

We investigated:

- Challenges of gradient calculation on multispectral images
- Multispectral and hyperspectral edge detection
- Adaptations of the Self-organizing map (SOM)

We found:

- Distance measures such as L_2 and SA are not generally useful
- With the SOM we can establish a data-driven distance measures
- Our measure is improved with larger SOMs and the use of several BMUs

Conclusions

We investigated:

- Challenges of gradient calculation on multispectral images
- Multispectral and hyperspectral edge detection
- Adaptations of the Self-organizing map (SOM)

We found:

- Distance measures such as L_2 and SA are not generally useful
- With the SOM we can establish a data-driven distance measures
- Our measure is improved with larger SOMs and the use of several BMUs

$\rightarrow$ **The SOM is simple, yet powerful**

Thank you!