Classification of Hyperspectral Data Using Spectral-Spatial Approaches

Yuliya Tarabalka, Jocelyn Chanussot and Jon Atli Benediktsson

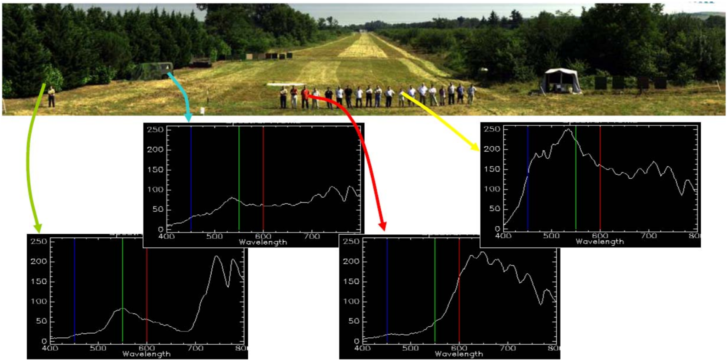
GIPSA-Lab, Grenoble Institute of Technologygy, France
Department of Electrical and Computer Engineering, University of Iceland, Iceland

May 13, 2010

Hyperspectral image

Every pixel contains a detailed spectrum (>100 spectral bands)

$+$ More information per pixel $\rightarrow$ increasing capability to distinguish objects



Hyperspectral image

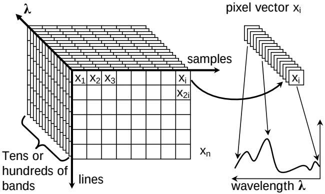
Every pixel contains a detailed spectrum (>100 spectral bands)

+ More information per pixel → increasing capability to distinguish objects

– Dimensionality increases → image analysis becomes more complex

⇓

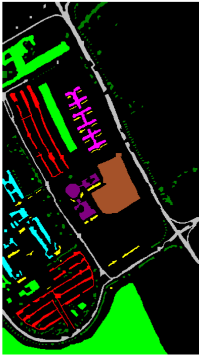
Efficient algorithms for automatic processing are required!



Classification problem

ROSIS image
Spatial resolution: 1.3m/pix
Spectral resolution: 103 bands

Ground-truth data



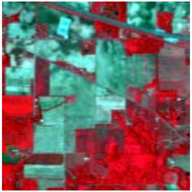
Task

Assign **every** pixel to **one** of the **nine** classes:

alphalt
meadows
gravel
trees
metal sheets
bare soil
bitumen
bricks
shadows

Classification problem

AVIRIS image
Spatial resolution: 20m/pix
Spectral resolution: 200 bands



Ground-truth data

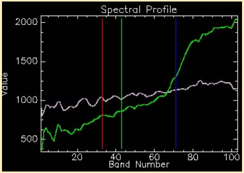
Task

Assign **every** pixel to **one** of the **16** classes:
corn-no till, corn-min till, corn, soybeans-no till, soybeans-min till, soybeans-clean till, alfalfa, grass/pasture, grass/trees, grass/pasture-mowed, hay-windrowed, oats, wheat, woods, bldg-grass-tree-drives, stone-steel towers

Classification approaches

Only spectral information

- Spectrum of each pixel is analyzed
- Directly accessible
- Variety of methods
  - Support Vector Machines (SVM) → good classification results [Camps-Valls05]



⇒

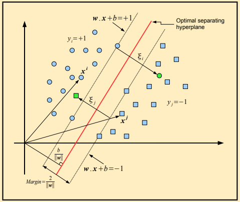
alphalt
meadows
gravel
trees
metal sheets
bare soil
bitumen
bricks
shadows

Overall accuracy = 81.01%

Classification approaches

Only spectral information

- Spectrum of each pixel is analyzed
- Directly accessible
- Variety of methods
    - Support Vector Machines (SVM) → good classification results [Camps-Valls05]



⇒

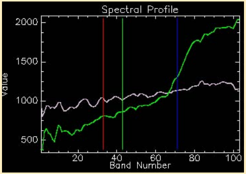
alphalt
meadows
gravel
trees
metal sheets
bare soil
bitumen
bricks
shadows

Overall accuracy = 81.01%

Classification approaches

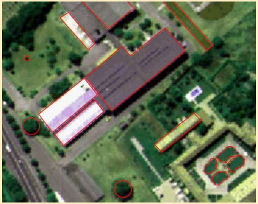
Only spectral information

- Spectrum of each pixel is analyzed
- Directly accessible
- Variety of methods
  - Support Vector Machines (SVM) $\rightarrow$ good classification results [Camps-Valls05]



Spectral + spatial information

- Info about spatial structures is included
  - Since neighboring pixels are related
- How to define spatial structures?
- How to combine spectral and spatial information?

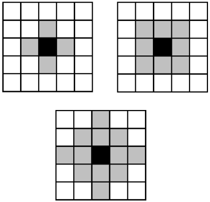


State of the art: Approaches for extracting spatial info

1. Closest fixed neighborhoods
   - Markov Random Field [Pony00, Jackson02, Farag05]
   - Contextual features [Camps-Valls06]

   \+ Simplicity

   − Imprecision at the border of regions

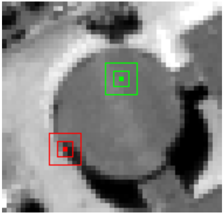


State of the art: Approaches for extracting spatial info

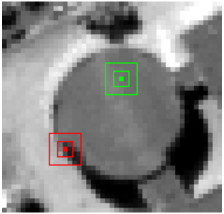
1. Closest fixed neighborhoods
   - Markov Random Field [Pony00, Jackson02, Farag05]
   - Contextual features [Camps-Valls06]

   \+ Simplicity

   − Imprecision at the border of regions



State of the art: Approaches for extracting spatial info



1. Closest fixed neighborhoods
   - Markov Random Field [Pony00, Jackson02, Farag05]
   - Contextual features [Camps-Valls06]

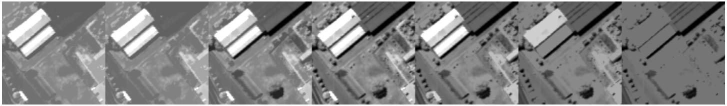
   \+ Simplicity

   – Imprecision at the border of regions

2. Morphological and area filtering
   - Morphological profiles [Dell'Acqua04, Benediktsson05, Fauvel07]
   - Area filtering [Fauvel07]

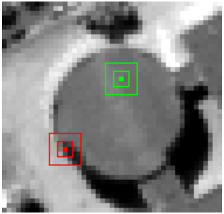
   \+ Neighborhoods are adapted to the structures

   – Neighborhoods are scale dependent ⇒ imprecision in the spatial info



Closing – Original – Opening

State of the art: Approaches for extracting spatial info



1. Closest fixed neighborhoods
   - Markov Random Field [Pony00, Jackson02, Farag05]
   - Contextual features [Camps-Valls06]

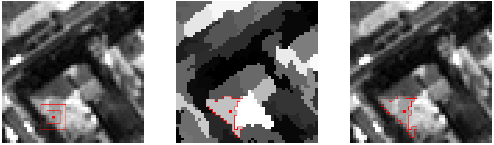
   \+ Simplicity

   – Imprecision at the border of regions

2. Morphological and area filtering
   - Morphological profiles [Dell'Acqua04, Benediktsson05, Fauvel07]
   - Area filtering [Fauvel07]

   \+ Neighborhoods are adapted to the structures

   – Neighborhoods are scale dependent ⇒ imprecision in the spatial info



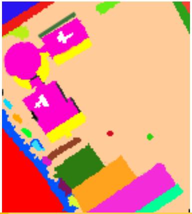
State of the art: Approaches for extracting spatial info

3. Segmentation map = exhaustive partitioning of the image into homogeneous regions

- Extraction and Classification of Homogeneous Objects [Kettig76]
  + Has become a standard spectral-spatial classification technique
  − Statistical approach ⇒ not well adapted for hyperspectral data

Recent works:

- Multiscale segmentation + SVM classification [Linden07, Huang09]
  − Computationally demanding
- Marker selection by morphological filtering + watershed [Noyel08]
  + Minimized dependence on the thresholds

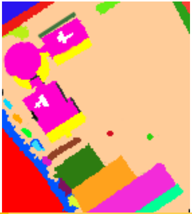


State of the art: Approaches for extracting spatial info

3. Segmentation map = exhaustive partitioning of the image into homogeneous regions
   - Extraction and Classification of Homogeneous Objects [Kettig76]
     + Has become a standard spectral-spatial classification technique
     – Statistical approach ⇒ not well adapted for hyperspectral data

   Recent works:

   - Multiscale segmentation + SVM classification [Linden07, Huang09]
     – Computationally demanding
   - Marker selection by morphological filtering + watershed [Noyel08]
     + Minimized dependence on the thresholds

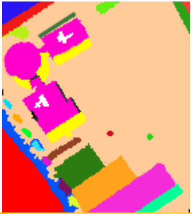


State of the art: Approaches for extracting spatial info

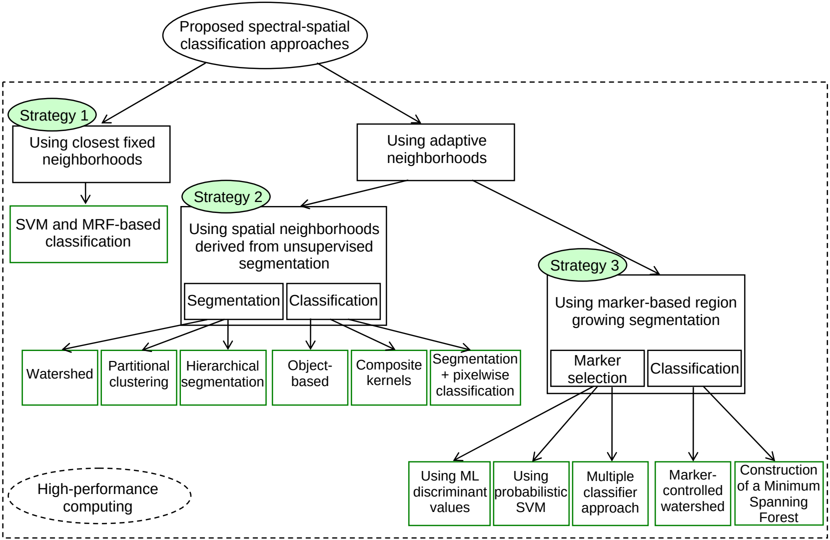
3. Segmentation map = exhaustive partitioning of the image into homogeneous regions
   - Extraction and Classification of Homogeneous Objects [Kettig76]
     + Has become a standard spectral-spatial classification technique
     – Statistical approach ⇒ not well adapted for hyperspectral data

   Recent works:

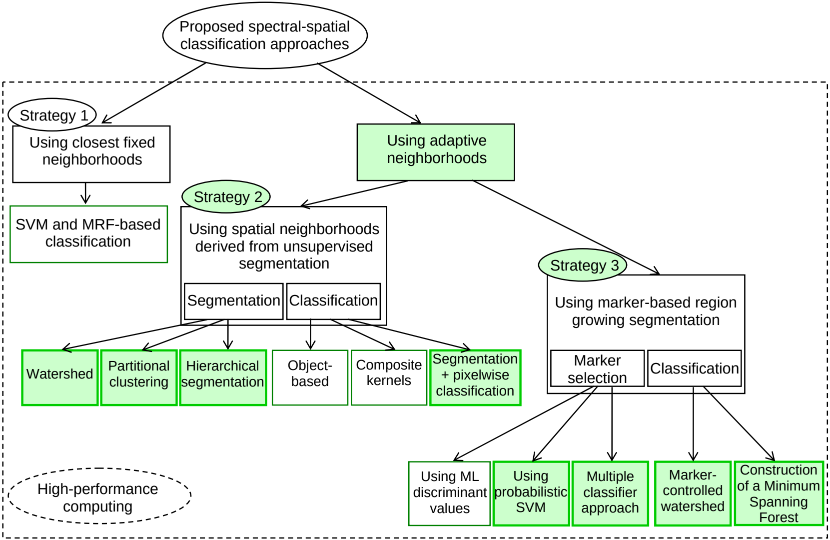
   - Multiscale segmentation + SVM classification [Linden07, Huang09]
     – Computationally demanding
   - Marker selection by morphological filtering + watershed [Noyel08]
     + Minimized dependence on the thresholds



combine spectral and spatial info



combine spectral and spatial info

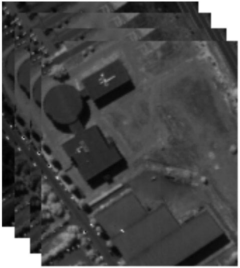


Outline

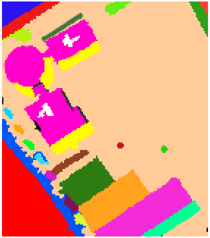
1. Introduction
2. Classification using segmentation-derived adaptive neighborhoods
   - Segmentation
   - Spectral-spatial classification
   - Concluding discussion
3. Segmentation and classification using automatically selected markers
   - Marker selection
   - Classification using marker-controlled region growing
   - Concluding discussion
4. Conclusions and perspectives

Objective

1. Automatically segment a hyperspectral image into homogeneous regions
2. Spectral info + spatial info → classify image



+



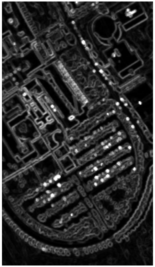
→ Classification map

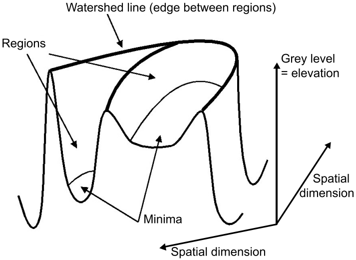
Outline

1. Introduction
2. Classification using segmentation-derived adaptive neighborhoods
   - Segmentation
   - Spectral-spatial classification
   - Concluding discussion
3. Segmentation and classification using automatically selected markers
   - Marker selection
   - Classification using marker-controlled region growing
   - Concluding discussion
4. Conclusions and perspectives

1. Watershed segmentation

gradient
⇒





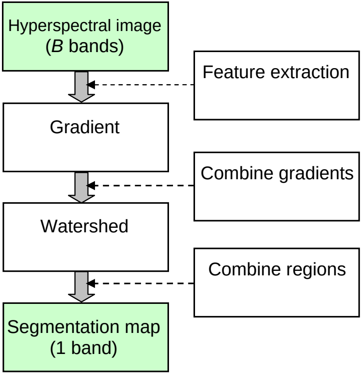
Region growing method:

- **Minimum** of a gradient = core of a homogeneous region
- **1 region** = set of pixels connected to 1 local minimum of the gradient
- **Watershed lines** = edges between adjacent regions

1. Watershed segmentation for hyperspectral image

From B-band image $\rightarrow$
1-band segmentation map:

- Feature extraction (PCA, ICA,...)?
- Vectorial gradient?
- Combine B gradients?
- Combine B watershed regions?

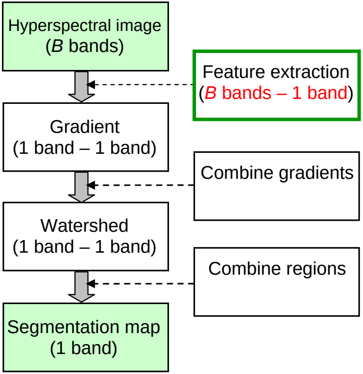


Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

1. Watershed segmentation for hyperspectral image

From B-band image $\rightarrow$
1-band segmentation map:

- Feature extraction (PCA, ICA,...)?
- Vectorial gradient?
- Combine B gradients?
- Combine B watershed regions?

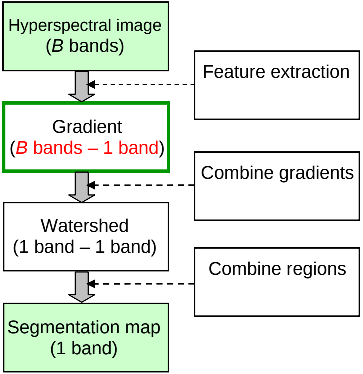


Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

1. Watershed segmentation for hyperspectral image

From B-band image $\rightarrow$ 1-band segmentation map:

- Feature extraction (PCA, ICA,...)?
- Vectorial gradient?
- Combine B gradients?
- Combine B watershed regions?

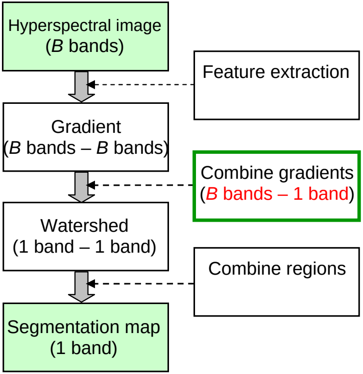


Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

1. Watershed segmentation for hyperspectral image

From B-band image $\rightarrow$ 1-band segmentation map:

- Feature extraction (PCA, ICA,...)?
- Vectorial gradient?
- Combine B gradients?
- Combine B watershed regions?

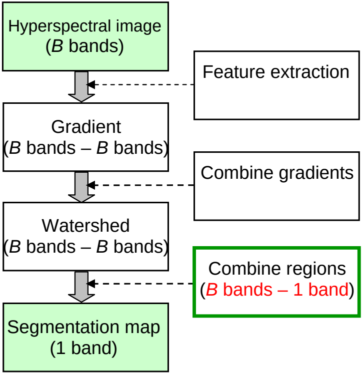


Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

1. Watershed segmentation for hyperspectral image

From B-band image $\rightarrow$
1-band segmentation map:

- Feature extraction (PCA, ICA,...)?
- Vectorial gradient?
- Combine B gradients?
- Combine B watershed regions?



Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

1. Watershed segmentation

Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

Original image

1. Watershed segmentation

Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

Original image

⇒

Robust Color Morpho Gradient

1. Watershed segmentation

Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

Original image

⇒

Robust Color Morpho Gradient

⇒

Watershed 11802 regions

1. Watershed segmentation

Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.

Original image

⇒

Robust Color Morpho Gradient

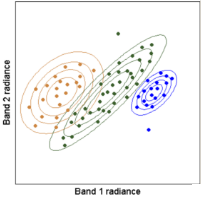
⇒

Watershed 11802 regions

⇒

Edges → to adjacent regions

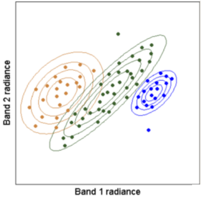
2. Partitional clustering (EM)



1. Clustering
   - pixels are grouped into C clusters
   - in each cluster $\rightarrow$ pixels drawn from a Gaussian distribution
   - distribution parameters $\rightarrow$ EM algorithm
2. Labeling of connected components

10 clusters
$\Rightarrow$

2. Partitional clustering (EM)



1. Clustering
   - pixels are grouped into C clusters
   - in each cluster → pixels drawn from a Gaussian distribution
   - distribution parameters → EM algorithm
2. Labeling of connected components

10 clusters
⇒
21450
regions

same cluster, but different regions!

3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

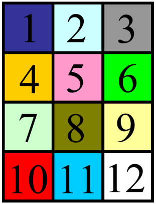
1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2

3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2

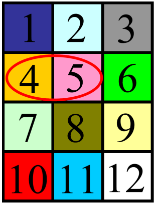


3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2

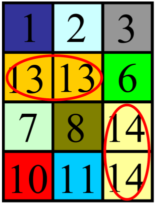


3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2})$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2

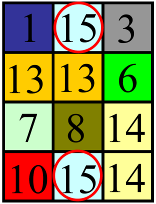


3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2

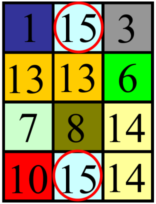


3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with
   $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2

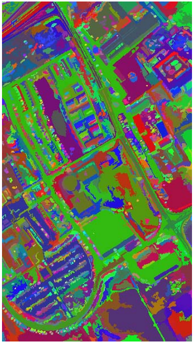


3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with
   $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2



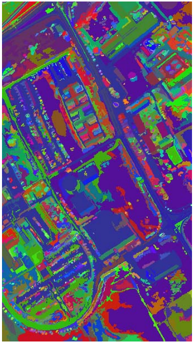
$SCW = 0.0$
7231 regions

3. Hierarchical SEGmentation (HSEG,[Tilton98])

- Region growing + Spectral Clustering
- Dissimilarity criterion (DC):
  Spectral Angle Mapper (SAM)
  between the region mean vectors u_i and u_j

$$SAM(u_i, u_j) = \arccos\left(\frac{u_i \cdot u_j}{\|u_i\|_2 \|u_j\|_2}\right)$$

1. Each pixel - one region
2. Find DC_{min} between adjacent regions
3. Merge adjacent regions with $DC = DC_{min}$
4. Merge non-adjacent regions with
   $DC \leq DC_{min} \cdot SpectralClusterWeight$
5. If not converge, go to 2



$SCW = 0.1$
7575 regions

Outline

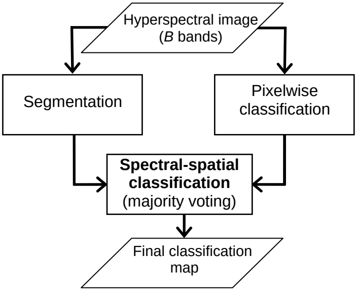
1 Introduction

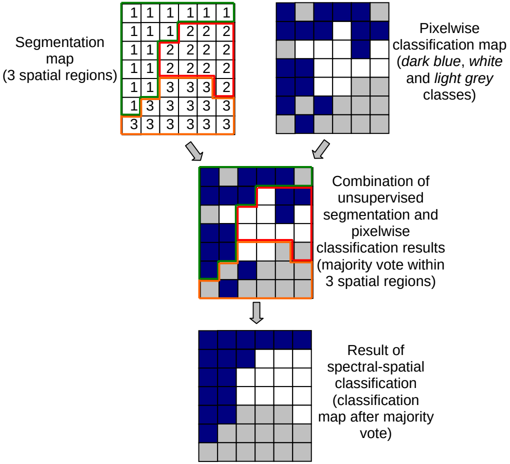
2 Classification using segmentation-derived adaptive neighborhoods
- Segmentation
- Spectral-spatial classification
- Concluding discussion

3 Segmentation and classification using automatically selected markers
- Marker selection
- Classification using marker-controlled region growing
- Concluding discussion

4 Conclusions and perspectives

Spectral-spatial classification: majority vote





Spectral-spatial classification

Original image

Spectral-spatial classification

Original image

⇒

SVM classification

OA = 81.01%
AA = 88.25%

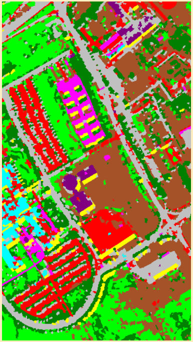
Spectral-spatial classification

SVM classification

OA = 81.01%
AA = 88.25%

⇒

SVM + Watershed

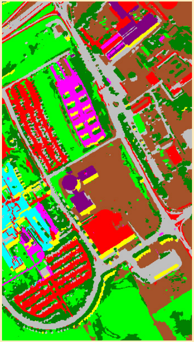


OA = 85.42%
AA = 91.31%

SVM + Partit.clustering

OA = 94.00%
AA = 93.13%

SVM + HSEG ($SCW = 0.1$)



OA = 93.85%
AA = 97.07%

Classification accuracies (%)

	SVM	+Watersh.	+Part.Cl.	+HSEG		EMP[1]	ECHO
SCW				0.0	0.1		
Overall Acc.	81.01	85.42	94.00	90.00	**93.85**	85.22	87.58
Average Acc.	88.25	91.31	93.13	94.15	**97.07**	90.76	92.16
Kappa Coef.κ	75.86	81.30	91.93	86.86	**91.89**	80.86	83.90
Asphalt	84.93	93.64	90.10	73.33	94.77	**95.36**	87.98
Meadows	70.79	75.09	**95.99**	88.73	89.32	80.33	81.64
Gravel	67.16	66.12	82.26	**97.47**	96.14	87.61	76.91
Trees	97.77	98.56	85.54	98.45	98.08	98.37	**99.31**
Metal sheets	99.46	99.91	**100**	99.10	99.82	99.48	99.91
Bare soil	92.83	97.35	96.72	98.43	**99.76**	63.72	93.96
Bitumen	90.42	96.23	91.85	95.92	**100**	98.87	92.97
Bricks	92.78	97.92	98.34	98.81	**99.29**	95.41	97.35
Shadows	98.11	96.98	97.36	97.11	96.48	97.68	**99.37**

[1] *A. Plaza et al., "Recent advances in techniques for hyperspectral image processing," Remote Sensing of Environment, vol. 113, Suppl. 1, 2009.*

Outline

1. Introduction
2. Classification using segmentation-derived adaptive neighborhoods
   - Segmentation
   - Spectral-spatial classification
   - Concluding discussion
3. Segmentation and classification using automatically selected markers
   - Marker selection
   - Classification using marker-controlled region growing
   - Concluding discussion
4. Conclusions and perspectives

1. Spectral-spatial classification improves accuracies when compared to pixel-wise classification
2. Several segmentation techniques are investigated
3. The HSEG segmentation map leads to the best classification
4. Obtained classification accuracies > all previous results

However...

Unsupervised segmentation

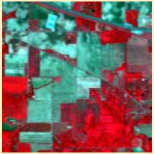
- Unsupervised segmentation = exhaustive partitioning into homogeneous regions
- How to define a measure of homogeneity?

Unsupervised segmentation

- Unsupervised segmentation = exhaustive partitioning into homogeneous regions
- How to define a measure of homogeneity?

Hierarchical SEGmentation (HSEG, [Tilton98])

Image



$\Rightarrow$

2381reg.	1389reg.	280reg.
pgflastimage	pgflastimage	pgflastimage

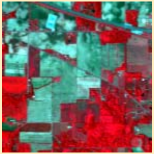
Oversegmentation! Undersegmentation?

Unsupervised segmentation

- Unsupervised segmentation = exhaustive partitioning into homogeneous regions
- How to define a measure of homogeneity?

Hierarchical SEGmentation (HSEG, [Tilton98])

Image



⇒

2381reg.	1389reg.	280reg.
pgflastimage	pgflastimage	pgflastimage

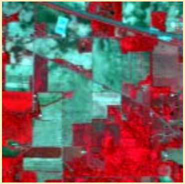
Oversegmentation Undersegmentation?

$\Downarrow$

Preferred in previous works $\rightarrow$ not to lose objects

Watershed segmentation

Original image



⇒

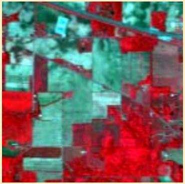
Robust Color Morpho Gradient

⇒

Watershed 1277 regions

Watershed segmentation

Original image

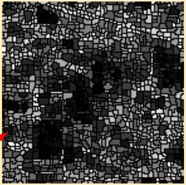


⇒

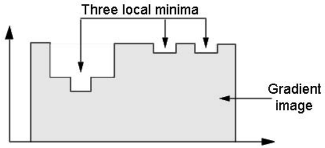
Robust Color Morpho Gradient

⇒

Watershed 1277 regions



Severe oversegmentation!



Every local minimum of the gradient

↓

one region

Marker-controlled segmentation

- Reduce oversegmentation $\Leftarrow$ incorporate an additional knowledge into segmentation
- We propose to use **markers**

Determine markers for each region of interest

$\Rightarrow$

Segment an image

One region
in a segmentation map
$\Uparrow$
One marker

Objective

- Determine markers automatically $\leftarrow$ using probabilistic classification results
- Marker-controlled region growing $\rightarrow$ segment and classify a hyperspectral image

Outline

1 Introduction

2 Classification using segmentation-derived adaptive neighborhoods
- Segmentation
- Spectral-spatial classification
- Concluding discussion

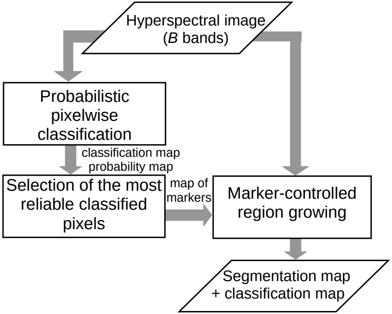
3 Segmentation and classification using automatically selected markers
- Marker selection
- Classification using marker-controlled region growing
- Concluding discussion

4 Conclusions and perspectives

Using probabilistic SVM

Marker selection using probabilistic SVM

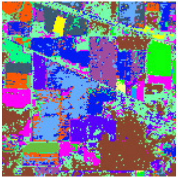
- B-band hyperspectral image $\mathbf{X} = \{\mathbf{x}_j \in \mathbb{R}^B, j = 1, 2, ..., n\}$
- $B \sim 100$



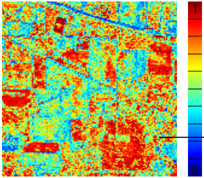
Marker selection using probabilistic SVM

- SVM classifier* → well suited for hyperspectral images
- Output:

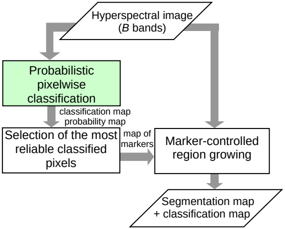
classification map



probability map



probability estimate for each pixel to belong to the assigned class

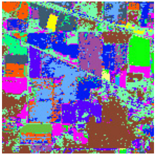


**C. Chang and C. Lin, "LIBSVM: A library for Support Vector Machines," Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.*

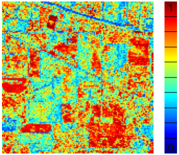
Marker selection using probabilistic SVM

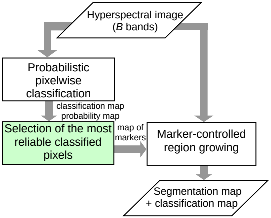
Analysis of classification and probability maps:

classification map



probability map



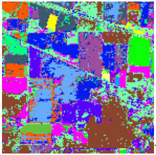


1. Perform connected components labeling of the classification map
2. Analyze each connected component:
   - If it is large ($>$ 20 pixels) $\rightarrow$ use $P\%$ (5%) of its pixels with the highest probabilities as a marker
   - If it is small $\rightarrow$ its pixels with probabilities $> T\%$ (90%) are used as a marker

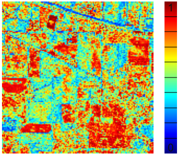
Marker selection using probabilistic SVM

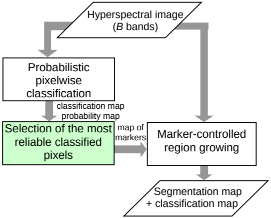
Analysis of classification and probability maps:

classification map



probability map



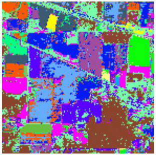


1. Perform connected components labeling of the classification map
2. Analyze each connected component:
   - If it is large (> 20 pixels) $\rightarrow$ use $P\%$ (5%) of its pixels with the highest probabilities as a marker
   - If it is small $\rightarrow$ its pixels with probabilities $> T\%$ (90%) are used as a marker

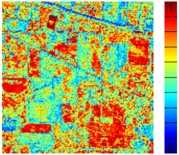
Marker selection using probabilistic SVM

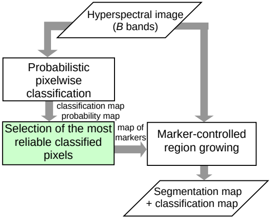
Analysis of classification and probability maps:

classification map



probability map



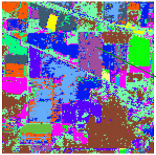


1. Perform connected components labeling of the classification map
2. Analyze each connected component:
   - If it is large (> 20 pixels) $\rightarrow$ use $P\%$ (5%) of its pixels with the highest probabilities as a marker
   - If it is small $\rightarrow$ its pixels with probabilities $> T\%$ (90%) are used as a marker

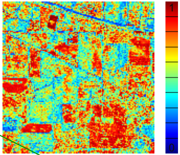
Marker selection using probabilistic SVM

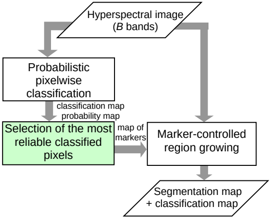
Analysis of classification and probability maps:

classification map



probability map





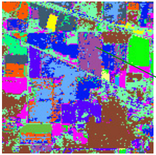
1. Perform connected components labeling of the classification map
2. Analyze each connected component:
   - If it is large ($>$ 20 pixels) $\rightarrow$ use $P\%$ (5%) of its pixels with the highest probabilities as a marker
   - If it is small $\rightarrow$ its pixels with probabilities $> T\%$ (90%) are used as a marker

Must contain a marker!

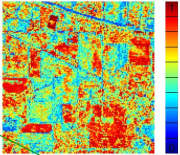
Marker selection using probabilistic SVM

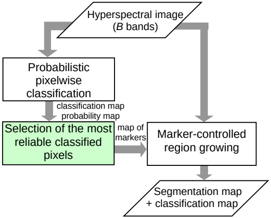
Analysis of classification and probability maps:

classification map



probability map





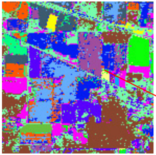
1. Perform connected components labeling of the classification map
2. Analyze each connected component:
   - If it is large (> 20 pixels) $\rightarrow$ use $P\%$ (5%) of its pixels with the highest probabilities as a marker
   - If it is small $\rightarrow$ its pixels with probabilities $> T\%$ (90%) are used as a marker

Must contain a marker!

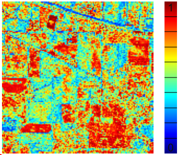
Marker selection using probabilistic SVM

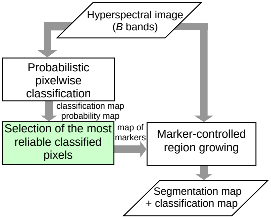
Analysis of classification and probability maps:

classification map



probability map





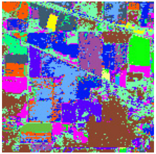
1. Perform connected components labeling of the classification map
2. Analyze each connected component:
   - If it is large ($>$ 20 pixels) $\rightarrow$ use $P\%$ (5%) of its pixels with the highest probabilities as a marker
   - If it is small $\rightarrow$ its pixels with probabilities $> T\%$ (90%) are used as a marker

Has a marker only if it is very reliable

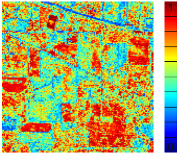
Marker selection using probabilistic SVM

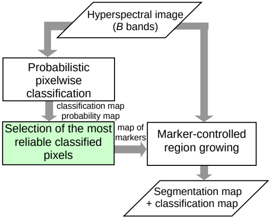
Analysis of classification and probability maps:

classification map



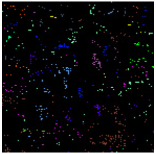
probability map





- Each connected component $\rightarrow$ 1 or 0 marker (2250 regions $\rightarrow$ 107 markers)
- Marker is not necessarily a connected set of pixels
- Each marker has a class label

map of 107 markers



Multiple classifier approach

Multiple classifier approach for marker selection

- Previous method: strong dependence on the performances of the selected probabilistic classifier
- Objective: mitigate this dependence
  $\rightarrow$ using multiple classifiers

Multiple classifier approach for marker selection

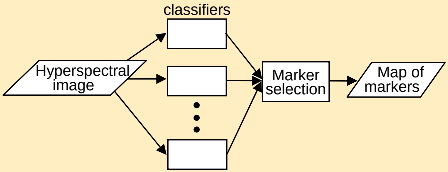
- Previous method: strong dependence on the performances of the selected probabilistic classifier
- Objective: mitigate this dependence
  $\rightarrow$ using multiple classifiers

Multiple classifier marker selection approach

1. Classify an image by several **independent** classifiers
2. Pixels assigned by **all** the classifiers to the same class

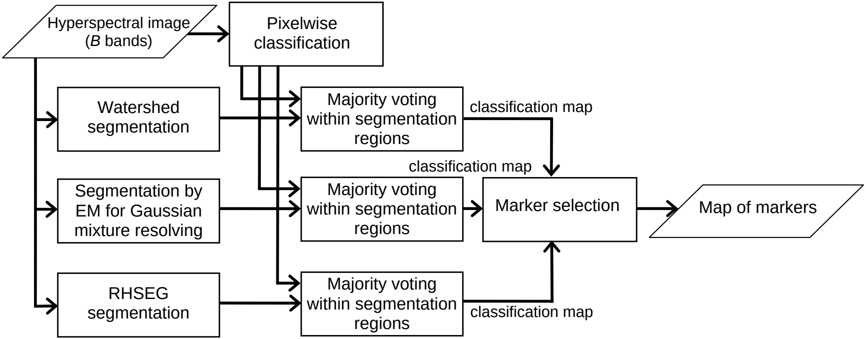
$\Downarrow$

Map of markers



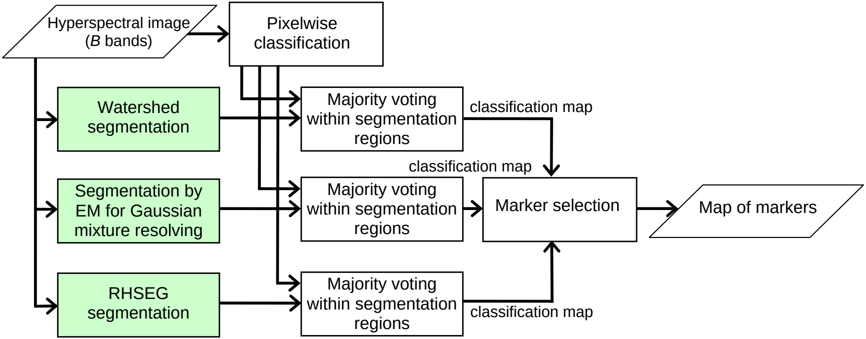
Multiple spectral-spatial classifier marker selection

- Take into account spatial context



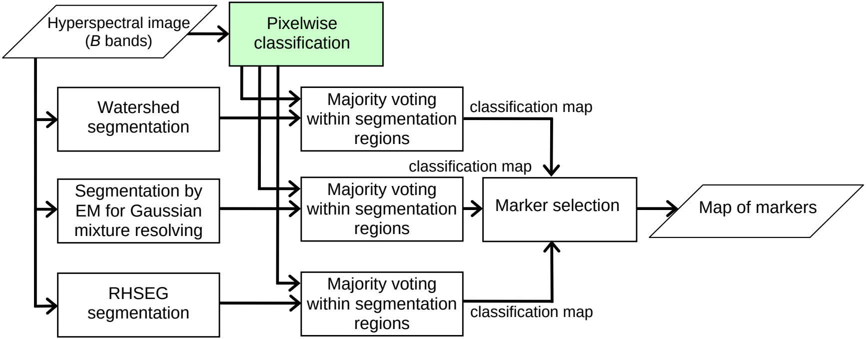
Multiple spectral-spatial classifier marker selection

- Take into account spatial context



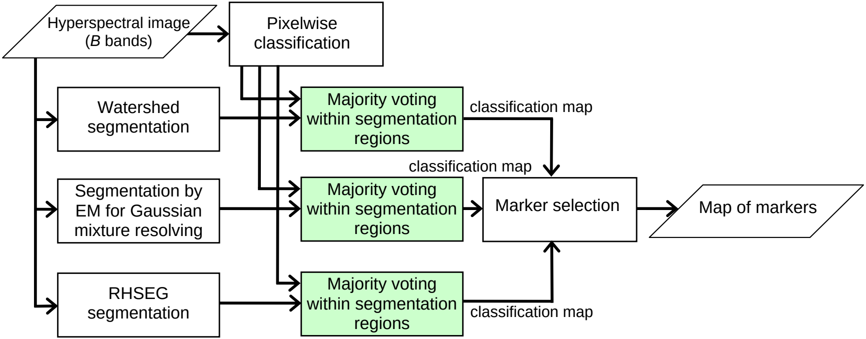
Multiple spectral-spatial classifier marker selection

- Take into account spatial context



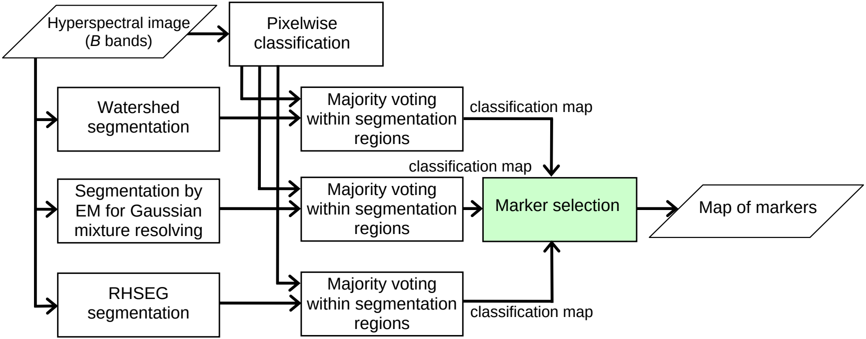
Multiple spectral-spatial classifier marker selection

- Take into account spatial context



Multiple spectral-spatial classifier marker selection

- Take into account spatial context



Outline

1 Introduction

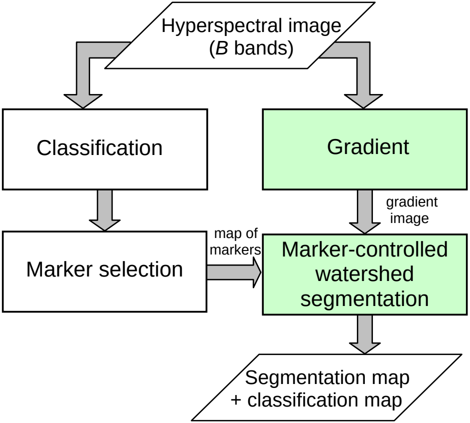
2 Classification using segmentation-derived adaptive neighborhoods
- Segmentation
- Spectral-spatial classification
- Concluding discussion

3 Segmentation and classification using automatically selected markers
- Marker selection
- Classification using marker-controlled region growing
- Concluding discussion

4 Conclusions and perspectives

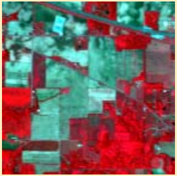
Marker-controlled watershed

1. Marker-controlled watershed



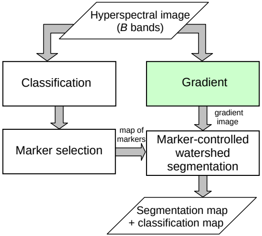
1. Marker-controlled watershed / Gradient

Original hyperspectral image



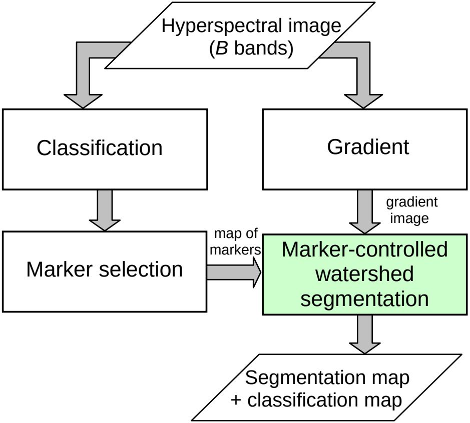
⇒

Robust Color Morphological Gradient*



**Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognition, vol. 43, no. 7, pp. 2367-2379, July 2010.*

1. Marker-controlled watershed



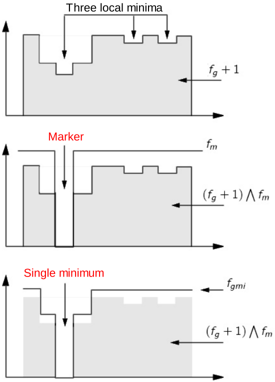
1. Marker-controlled watershed

1. Transform the gradient f_g $\rightarrow$ markers are the only minima
   - Create a marker image:

$$f_m(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ belongs to marker,} \\ t_{max}, & \text{otherwise} \end{cases}$$

   - Compute $(f_g + 1) \bigwedge f_m$
   - Perform minima imposition: morphological reconstruction by erosion of $(f_g + 1) \bigwedge f_m$ from f_m:

$$f_{gmi} = R^{\varepsilon}_{(f_g+1) \bigwedge f_m}(f_m)$$



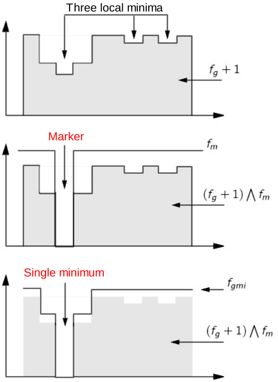
1. Marker-controlled watershed

1. Transform the gradient $f_g \rightarrow$ markers are the only minima
   - Create a marker image:

$$f_m(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ belongs to marker,} \\ t_{max}, & \text{otherwise} \end{cases}$$

   - Compute $(f_g + 1) \bigwedge f_m$
   - Perform minima imposition: morphological reconstruction by erosion of $(f_g + 1) \bigwedge f_m$ from f_m:

$$f_{gmi} = R^{\varepsilon}_{(f_g+1) \bigwedge f_m}(f_m)$$



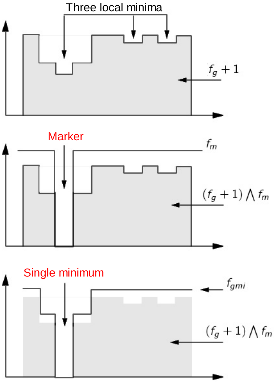
1. Marker-controlled watershed

1. Transform the gradient f_g → markers are the only minima
   - Create a marker image:

$$f_m(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ belongs to marker,} \\ t_{max}, & \text{otherwise} \end{cases}$$

   - Compute $(f_g + 1) \bigwedge f_m$
   - Perform minima imposition: morphological reconstruction by erosion of $(f_g + 1) \bigwedge f_m$ from f_m:

$$f_{gmi} = R^{\varepsilon}_{(f_g+1) \bigwedge f_m}(f_m)$$



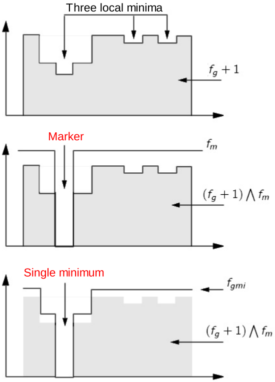
1. Marker-controlled watershed

1. Transform the gradient f_g → markers are the only minima
   - Create a marker image:

$$f_m(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ belongs to marker,} \\ t_{max}, & \text{otherwise} \end{cases}$$

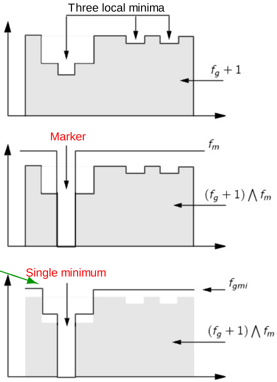
   - Compute $(f_g + 1) \bigwedge f_m$
   - Perform minima imposition: morphological reconstruction by erosion of $(f_g + 1) \bigwedge f_m$ from f_m:

$$f_{gmi} = R^{\varepsilon}_{(f_g+1) \bigwedge f_m}(f_m)$$



1. Marker-controlled watershed

1. Transform the gradient f_g $\rightarrow$ markers are the only minima
2. Apply watershed on the filtered gradient image f_{gmi} (Vincent and Soille, 1991)



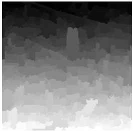
1. Marker-controlled watershed

1. Transform the gradient f_g $\rightarrow$ markers are the only minima
2. Apply watershed on the filtered gradient image f_{gmi} (Vincent and Soille, 1991)

1. Marker-controlled watershed

1. Transform the gradient f_g → markers are the only minima
2. Apply watershed on the filtered gradient image f_{gmi} (Vincent and Soille, 1991)
3. Assign every watershed pixel to the spectrally most similar neighboring region

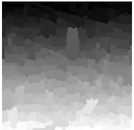
⇓



1. Marker-controlled watershed

1. Transform the gradient f_g $\rightarrow$ markers are the only minima
2. Apply watershed on the filtered gradient image f_{gmi} (Vincent and Soille, 1991)
3. Assign every watershed pixel to the spectrally most similar neighboring region

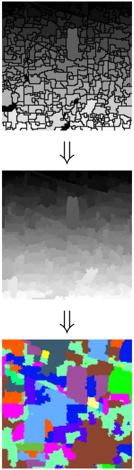
$\Downarrow$



$\rightarrow$ Several minima in the filtered gradient $\rightarrow$ Several regions in the segmentation map

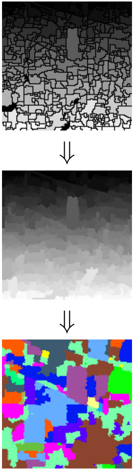
1. Marker-controlled watershed

1. Transform the gradient f_g → markers are the only minima
2. Apply watershed on the filtered gradient image f_{gmi} (Vincent and Soille, 1991)
3. Assign every watershed pixel to the spectrally most similar neighboring region
4. Merge regions belonging to the same marker



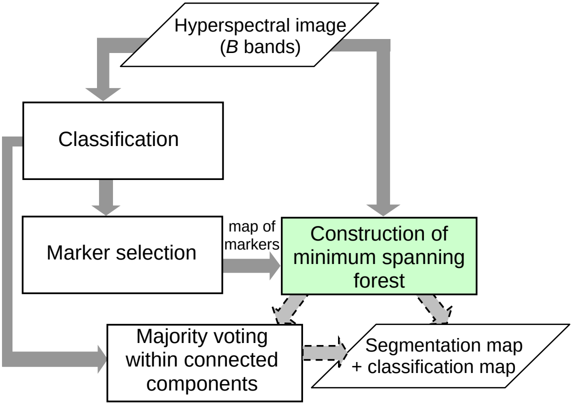
1. Marker-controlled watershed

1. Transform the gradient f_g $\rightarrow$ markers are the only minima
2. Apply watershed on the filtered gradient image f_{gmi} (Vincent and Soille, 1991)
3. Assign every watershed pixel to the spectrally most similar neighboring region
4. Merge regions belonging to the same marker
5. Class of each marker $\rightarrow$ class of the corresponding region

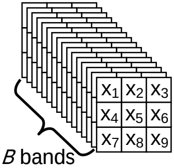


Construction of a Minimum Spanning Forest

2. Construction of a Minimum Spanning Forest (MSF)



2. Construction of a Minimum Spanning Forest (MSF)



map of markers

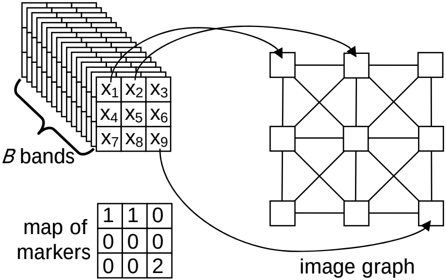
1	1	0
0	0	0
0	0	2

1) Map an image onto a graph

- Weight $w_{i,j}$ indicates the degree of dissimilarity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. Spectral Angle Mapper (SAM) distance can be used:

$$w_{i,j} = SAM(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\sum_{b=1}^{B} x_{ib}x_{jb}}{[\sum_{b=1}^{B} x_{ib}^2]^{1/2}[\sum_{b=1}^{B} x_{jb}^2]^{1/2}}\right)$$

2. Construction of a Minimum Spanning Forest (MSF)

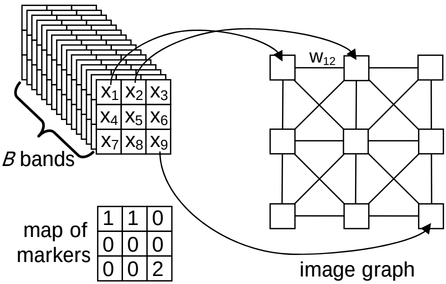


1) Map an image onto a graph

- Weight $w_{i,j}$ indicates the degree of dissimilarity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. Spectral Angle Mapper (SAM) distance can be used:

$$w_{i,j} = SAM(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\sum_{b=1}^{B} x_{ib}x_{jb}}{[\sum_{b=1}^{B} x_{ib}^2]^{1/2}[\sum_{b=1}^{B} x_{jb}^2]^{1/2}}\right)$$

2. Construction of a Minimum Spanning Forest (MSF)

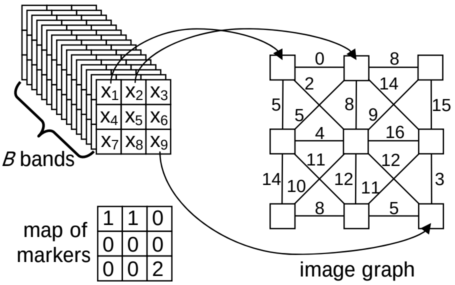


1) Map an image onto a graph

- Weight $w_{i,j}$ indicates the degree of dissimilarity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. Spectral Angle Mapper (SAM) distance can be used:

$$w_{i,j} = SAM(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\sum_{b=1}^{B} x_{ib}x_{jb}}{[\sum_{b=1}^{B} x_{ib}^2]^{1/2}[\sum_{b=1}^{B} x_{jb}^2]^{1/2}}\right)$$

2. Construction of a Minimum Spanning Forest (MSF)

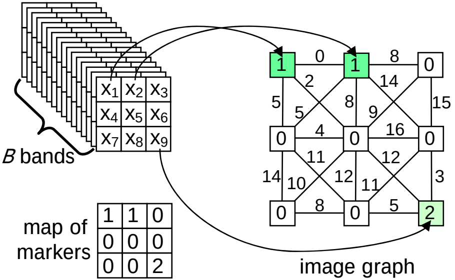


1) Map an image onto a graph

- Weight $w_{i,j}$ indicates the degree of dissimilarity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. Spectral Angle Mapper (SAM) distance can be used:

$$w_{i,j} = SAM(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\sum_{b=1}^{B} x_{ib}x_{jb}}{[\sum_{b=1}^{B} x_{ib}^2]^{1/2}[\sum_{b=1}^{B} x_{jb}^2]^{1/2}}\right)$$

2. Construction of a Minimum Spanning Forest (MSF)



1) Map an image onto a graph

- Weight $w_{i,j}$ indicates the degree of dissimilarity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. Spectral Angle Mapper (SAM) distance can be used:

$$w_{i,j} = SAM(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\sum_{b=1}^{B} x_{ib}x_{jb}}{[\sum_{b=1}^{B} x_{ib}^2]^{1/2}[\sum_{b=1}^{B} x_{jb}^2]^{1/2}}\right)$$

2. Construction of a Minimum Spanning Forest (MSF)

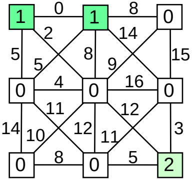
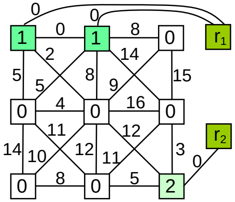


image graph

Given a graph G, a **MSF** F^* rooted on vertices $\{r_1, ..., r_m\}$ is:

- a non-connected graph without cycles
- contains all the vertices of G
- consists of connected subgraphs, each subgraph (tree) contains (is rooted on) one root r_i
- sum of the edges weights of F^* is minimal

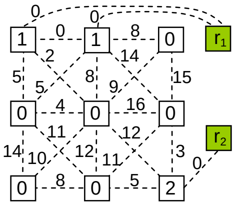
2. Construction of a Minimum Spanning Forest (MSF)



modified graph

2) Add m extra vertices $r_i, i = 1, ..., m$ corresponding to m markers

2. Construction of a Minimum Spanning Forest (MSF)

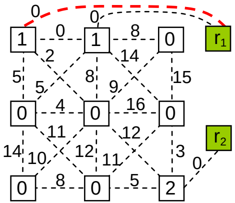


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

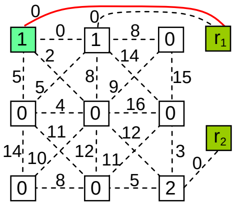


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

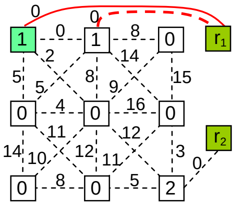


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

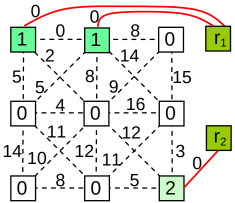


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

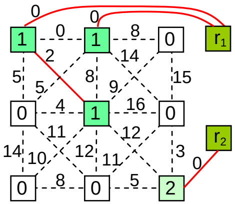


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

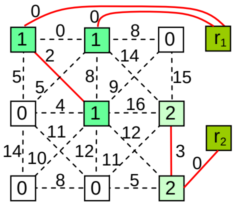


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

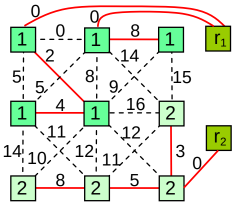


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

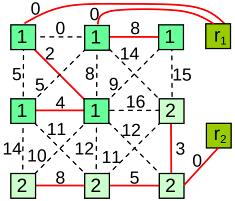


3) Construct a MSF $F^* = (V^*, E^*)$

Initialization: $V^* = \{r_1, r_2, ..., r_m\}$ (roots are in the forest)

1. Choose edge of the modified graph e_{ij} with minimal weight such that $i \in V^*$ and $j \notin V^*$
2. $V^* = V^* \cup \{j\}$, $E^* = E^* \cup \{e_{i,j}\}$
3. If $V^* \neq V$, go to 1

2. Construction of a Minimum Spanning Forest (MSF)

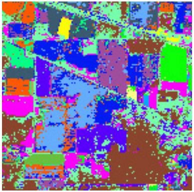


3) Construct a MSF $F^* = (V^*, E^*)$

4) Class of each marker $\rightarrow$ class of the corresponding region
(of all the pixels grown from this marker)

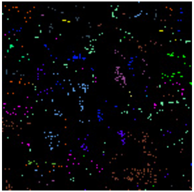
2. Construction of a Minimum Spanning Forest (MSF)

Pixel-wise classification map



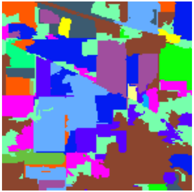
⇒

Map of 107 markers



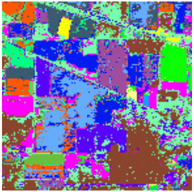
⇒

MSF-based classification map



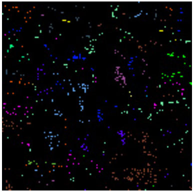
2. Construction of a Minimum Spanning Forest (MSF)

Pixel-wise classification map



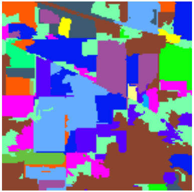
⇒

Map of 107 markers



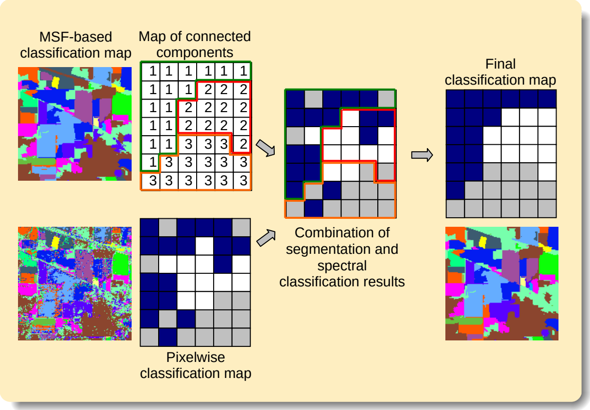
⇒

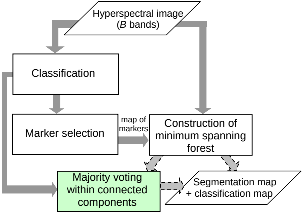
MSF-based classification map



If a marker is classified to the wrong class ⇒ The whole region grown from this marker risks to be wrongly classified!

2. Construction of a Minimum Spanning Forest / Post-processing





Classification accuracies for the Indian Pines data (%)

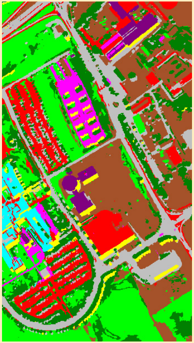
	SVM	ECHO	Strategy 2	Strategy 3: Marker-based classification			
			SVM+ Waters.	SVM- Waters.	SVM- MSF	SVM- MSF+MV	MSSC- MSF
Overall Accuracy	78.17	82.64	86.63	85.99	88.41	91.80	**92.32**
Average Accuracy	85.97	83.75	91.61	86.95	91.57	**94.28**	94.22
Kappa Coef. κ	75.33	80.38	84.83	83.98	86.71	90.64	**91.19**
Corn-no till	78.18	83.45	**94.22**	80.35	90.97	93.21	89.74
Corn-min till	69.64	75.13	78.06	71.94	69.52	**96.56**	86.99
Corn	91.85	92.39	88.59	73.37	**95.65**	**95.65**	95.11
Soybeans-no till	82.03	90.10	96.30	**98.91**	98.04	93.91	91.84
Soybeans-min till	58.95	64.14	68.82	80.48	81.97	81.97	**89.16**
Soybeans-clean till	87.94	89.89	90.78	84.75	85.99	97.16	**97.34**
Alfalfa	74.36	48.72	**94.87**	**94.87**	**94.87**	**94.87**	**94.87**
Grass/pasture	92.17	94.18	95.08	**95.30**	94.63	94.63	94.63
Grass/trees	91.68	96.27	**97.99**	92.97	92.40	97.27	97.85
Grass/pasture-mow	**100**	36.36	**100**	**100**	**100**	**100**	**100**
Hay-windrowed	97.72	97.72	99.54	99.54	**99.77**	**99.77**	**99.77**
Oats	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Wheat	98.77	98.15	**99.38**	**99.38**	**99.38**	**99.38**	**99.38**
Woods	93.01	94.21	97.11	99.36	97.59	**99.68**	99.44
Bldg-Grass-Tree-Dr	61.52	**81.52**	69.39	55.45	68.79	68.79	73.64
Stone-steel towers	**97.78**	**97.78**	95.56	64.44	95.56	95.56	**97.78**

Classification of the Pavia image

SVM classification

OA = 81.01%
AA = 88.25%

Strategy 2: SVM + HSEG

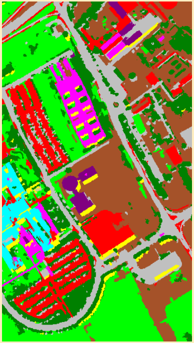


OA = 93.85%
AA = 97.07%

Strategy 3: SVM-MSF+MV

OA = 91.08%
AA = 94.76%

Strategy 3: MSSC - MSF



OA = 97.90%
AA = 98.59%

Classification accuracies for the Pavia image (%):

	SVM	ECHO	Strategy 2		Strategy 3: Marker-based classif.		
			SVM+ Waters.	SVM+ HSEG	SVM-MSF	SVM-MSF+MV	MSSC-MSF
Overall Accuracy	81.01	87.58	85.42	93.85	84.14	91.08	**97.90**
Average Accuracy	88.25	92.16	91.31	97.07	92.35	94.76	**98.59**
Kappa Coef. κ	75.86	83.90	81.30	91.89	79.71	88.30	**97.18**
Asphalt	84.93	87.98	93.64	94.77	93.05	93.16	**98.00**
Meadows	70.79	81.64	75.09	89.32	72.30	85.65	**96.67**
Gravel	67.16	76.91	66.12	96.14	89.15	89.15	**97.80**
Trees	97.77	**99.31**	98.56	98.08	87.02	91.24	98.83
Metal sheets	99.46	**99.91**	**99.91**	99.82	**99.91**	**99.91**	**99.91**
Bare soil	92.83	93.96	97.35	99.76	97.11	99.91	**100**
Bitumen	90.42	92.97	96.23	**100**	98.57	98.57	99.90
Bricks	92.78	97.35	97.92	99.29	95.66	99.05	**99.76**
Shadows	98.11	**99.37**	96.98	96.48	98.36	96.23	96.48

Outline

1 Introduction

2 Classification using segmentation-derived adaptive neighborhoods
- Segmentation
- Spectral-spatial classification
- Concluding discussion

3 Segmentation and classification using automatically selected markers
- Marker selection
- Classification using marker-controlled region growing
- Concluding discussion

4 Conclusions and perspectives

1. Classification using automatically selected markers:
    - significantly decreases oversegmentation
    - improves classification accuracies
    - provides classification maps with homogeneous regions
2. Marker selection: it is advantageous to use
    - SVM classifier
    - spatial information
    - multiple classifier approaches
3. Marker-controlled region growing
    - MSF-based method has proven to be efficient and robust

Contributions

- **Three spectral-spatial classification strategies:**
  1. Using SVM and MRF models
  2. Using adaptive neighborhoods derived from unsupervised segmentation
     - Segmentation techniques working both in the spatial and spectral domain → good performances
     - Pixelwise classification + majority voting within regions → simple and fast technique
  3. Using marker-based region growing segmentation
     - Analyzing probabilistic classification results for marker selection
     - Interest of using spatial information and multiple classifier approaches for marker selection
     - MSF-based marker-controlled region growing → efficient and robust
- Possibilities of high-performance parallel computing using commodity processors

Perspectives

1. Spectral-spatial image analysis
   - Develop new similarity measures
   - Automatically select results in segmentation hierarchies
   - Perform segmentation and classification concurrently
2. Further explore parallel strategies using commodity processors
3. Apply and adapt the proposed methods for other types of data/applications
   - medical imaging

Classification of Hyperspectral Data Using Spectral-Spatial Approaches

Yuliya Tarabalka, Jocelyn Chanussot and Jon Atli Benediktsson

GIPSA-Lab, Grenoble Institute of Technologygy, France
Department of Electrical and Computer Engineering, University of Iceland, Iceland

May 13, 2010