A distance geometry-based representation of hyperspectral imagery

Paul Scheunders, Rob Heylen

iMinds-Visionlab, University of Antwerp, Belgium

Universiteit Antwerpen

Outline

1. Introduction:
   1.1 The hyperspectral manifold
   1.2 Geodesic distances on the manifold
   1.3 The simplex paradigm

2. Geometric endmember extraction (NFindR)
   - Distance-geometry-based NFindR
   - Nonlinear (geodesic distance-based) NFindR

3. Geometric unmixing
   - Simplex Projection Unmixing (SPU)
   - Distance-based Simplex Projection Unmixing (DSPU)
   - Geodesic Distance Geometric Unmixing

4. Related, Applied and Future work

Introduction:
1. The hyperspectral manifold

Hyperspectral image, N bands
N-dimensional spectral space = Euclidean space

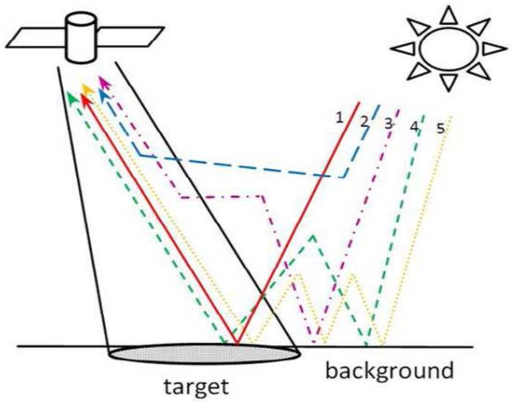
All traditional processing and analysis is performed in spectral space, using Euclidean distance geometry

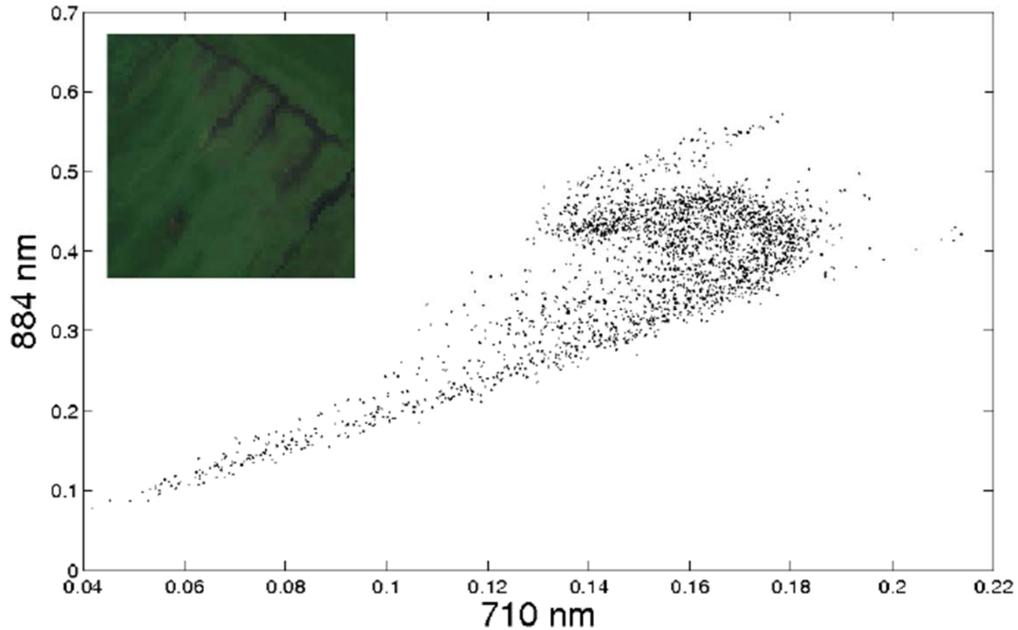
- Dimensionality reduction, feature extraction (PCA, Fisher Discriminant analysis, …)
- Classification (k-NN, LDA, …)
- Spectral unmixing (Nfindr, FCLSU, …)

Problem: Sparsity, Curse of dimensionality
Advantage: correlation: data manifold = very low dimensional subspace
Solution: dimensionality reduction

Introduction:
1. The hyperspectral manifold

- Secondary reflections
- Intricate mineral mixtures
- Shallow water environments
- ...





Scatter plot of band 10 (710 nm) and band 16 (884 nm) of partly submerged grassland. The data manifold has a highly nontrivial shape, indicating complex non-linear interactions are present.

Universiteit Antwerpen

Introduction:
1. The hyperspectral manifold

Estimating the data manifold

Many techniques are
- data-driven
- unsupervised
- geometrically oriented

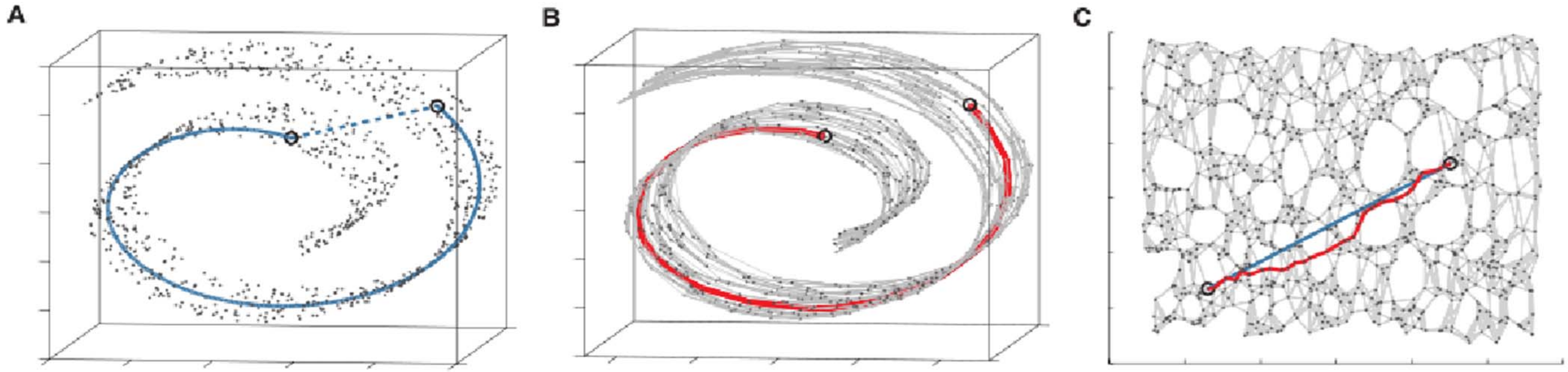
Graph-based methods (Isomap, LLE, …)
Kernel-based methods (kPCA, …)

Introduction:
2. Geodesic distances on the manifold

Graph-based methods:

The data cloud forms a curved manifold in spectral space.

Capture structure via shortest-path distances over nearest-neighbor graph.



J.B. Tenenbaum, V. de Silva, J.C. Langford: A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500): 2319-2323 (2000)

Introduction:
2. Geodesic distances on the manifold

ISOMAP Algorithm:
- Calculate Euclidean distance between all pairs of N points
- Construct a nearest-neighbor graph:
  - connect every point to K nearest points
  - weight of edge = Euclidean distance
  - symmetrized and connected
- Geodesic distance is approximated by shortest path along weighted graph (Dijkstra algorithm)
- NxN matrix of geodesic distances

- After which: dimensionality reduction (e.g. Multidimensional Scaling)
- requires calculation of eigenvectors of NxN distance matrix

Problems:
- computational cost, memory requirements
- which dimension?

Introduction:
2. Geodesic distances on the manifold

Our proposal: work directly on the nonlinear manifold, without having to unfold, or project on Euclidean space.

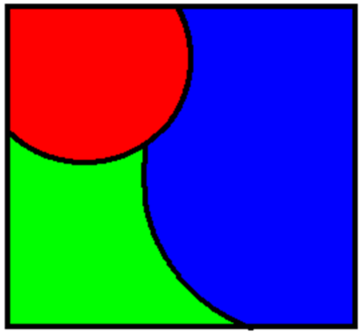
Prerequisites:
- Certain curvature conditions of the data manifold: “zero curvature” (folding without stretching)
- Processing technique can be written in terms of distance geometry

Advantages:
- data-driven and unsupervised
- Can handle nonlinear manifolds
- no projection required: cost and memory efficient
- No need to know intrinsic dimensionality
- No need to calculate and store whole manifold structure
- Geometric interpretation leads to new insights, even in linear case

Introduction:
3. The simplex paradigm

The linear mixing model: The observed spectrum is a linear combination of endmember spectra.

$$\boldsymbol{x}_i = \sum_{j=1}^{p} a_{ij}\boldsymbol{e}_j, \qquad \sum_{j=1}^{p} a_{ij} = 1, \quad \forall i,j : a_{ij} \geq 0$$



17%

23%

60%

Introduction:
3. The simplex paradigm

Unmixing: inversion of the mixing equation while respecting the constraints

- Easy without noise: Over-determined linear system
- Easy without constraints: Least-squares problem
- Difficult when noise and constraints present

- Many approaches exist: FCLSU, quadratic programming, Bayesian techniques, source separation, fuzzy set theory, ...

VISION LAB

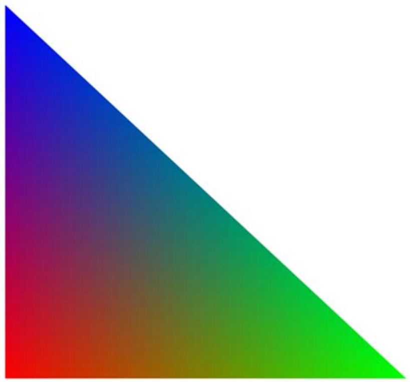
Introduction:
3. The simplex paradigm

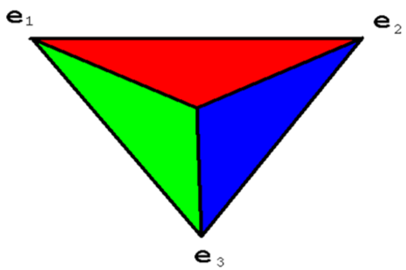
Geometric interpretation of linear unmixing:

Endmembers form linear basis for spectra with barycentric coordinates

Data manifold = simplex, spanned by p endmembers in spectral space

Many techniques exploit this geometric notion:
N-FINDR, PPI, Simplex growing, …





$$a_1 = \frac{V({\color{blue}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

$$a_2 = \frac{V({\color{green}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

$$a_3 = \frac{V({\color{red}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

Introduction:
3. The simplex paradigm

Nonlinear case: simplex notion fails?

Continuity conditions

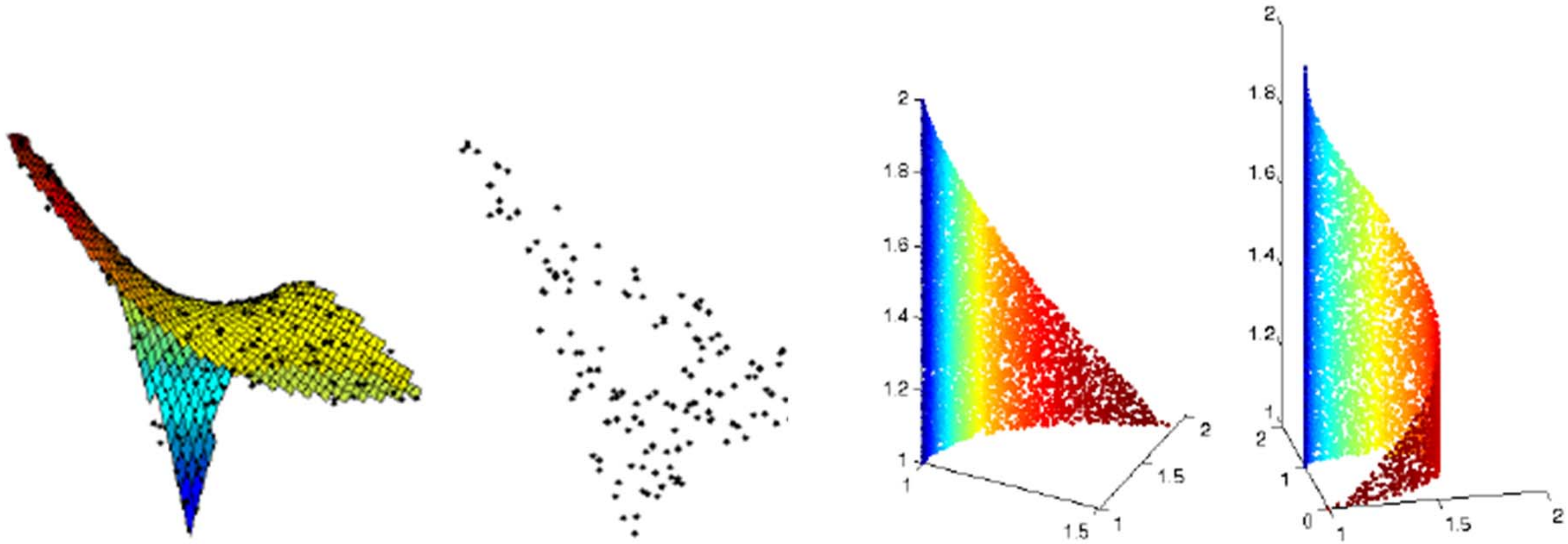
Assume non-linear, continuous bijective mapping F between linear space of abundance coefficients and spectral space:

$$\boldsymbol{x}_i = F\left(\sum_{j=1}^{p} a_{ij}\boldsymbol{e}_j\right)$$

F induces a manifold composed of the continuous projection of a linear simplex
Resembles nonlinearly transformed simplex.
Endmembers are vertices of data manifold

VISION LAB

Introduction:
3. The simplex paradigm



Universiteit Antwerpen

2. Geometric endmember extraction (NFindR)

2. Geometric endmember extraction (NFindR)

Traditional NfindR:

- Reduce dimensionality (potentially nonlinearly)
- Find simplex of largest volume
- [Calculate abundances]

Geometric NfindR:

- Work directly in spectral space. No dimensionality reduction !
- Transform the NfindR algorithm to work with distance geometry.
- Use geodesic distances in the resulting algorithm.

2.1 Distance geometry-based NFindR

Core of NfindR: Simplex volume calculation

Can be written in terms of inter-vertex distances using Cayley-Menger determinant

$$V_p^2 \sim \det(\boldsymbol{C}_p) = \det\begin{pmatrix} \boldsymbol{D}_p & \mathbf{1} \\ \mathbf{1} & 0 \end{pmatrix}, \qquad \boldsymbol{D}_p = \left[d_{ij}^2\right]_{i,j=1,\ldots,p}$$

Computationally interesting equivalent:

$$V_p^2 \sim \underbrace{\left(\boldsymbol{d}\boldsymbol{C}_{p-1}^{-1}\boldsymbol{d}^{\mathrm{T}}\right)}_{\text{orth. dist.}} \underbrace{\det(\boldsymbol{C}_{p-1})}_{\text{vol p-1 simplex}}, \quad \boldsymbol{d} = (d_{p1}^2, \ldots, d_{p,p-1}^2, 1)$$

Allows for very efficient way of searching for simplex of maximal volume

2.2 Nonlinear (geodesic distance-based) NFindR

- Replace Euclidean distance by geodesic distance
- Then: volume as measured along manifold
- Relation volume-distance valid if manifold can be covered by Euclidean space
- Flat manifold with zero curvature
- Assumption: works with small curvature

2.2 Nonlinear (geodesic distance-based) NFindR

The algorithm

- Construct weighted symmetrical and connected K-Nearest Neighbor graph
- Select p random points as initial vertices
- Calculate shortest-path distance from these points to all others: (pxN) distance matrix
- Determine simplex volume
- Replace 1 endmember by random point, if larger volume is found, recalculate row of (pXN)-distance matrix

Advantages

- Independent of spectral dimension.
- No need for dimensionality reduction
- Only parts of the geodesic distance matrix required
- Requires less memory (e.g. Cuprite dataset:

  (NxN)-distance matrix = 720 Gbyte

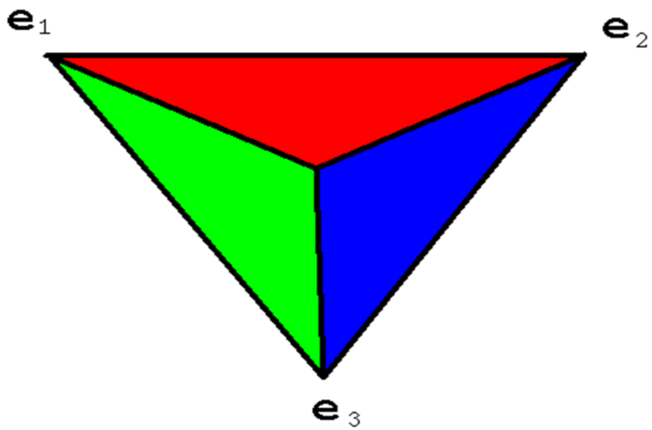
  (pxN)-distance matrix = 24 Mbyte)

Universiteit Antwerpen

2.2 Nonlinear (geodesic distance-based) NFindR

Abundance estimation

Abundance estimation via relative volumes



$$a_1 = \frac{V({\color{blue}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

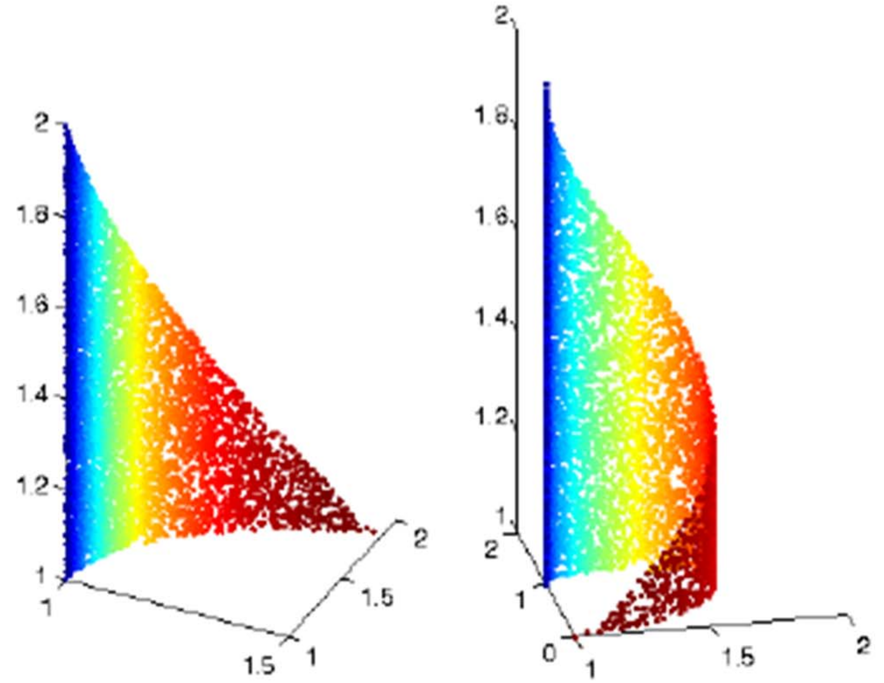
$$a_2 = \frac{V({\color{green}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

$$a_3 = \frac{V({\color{red}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

- All steps expressed in distance geometry.
- However: positivity constraint (points outside simplex)!

2.2 Nonlinear (geodesic distance-based) NFindR

Results on Artificial dataset



$$
\begin{aligned}
x_i &= a_{i1}\sin(\sigma a_{i1}) + 1 \\
y_i &= a_{i1}\cos(\sigma a_{i1}) + 1 \\
z_i &= a_{i2} + 1
\end{aligned}
$$

Fig. 3. The artificial data set for $\sigma = 0.5$ (left) and $\sigma = \pi$ (right), for 5000 randomly generated abundances, and color-coded by the value of a_1.

Universiteit Antwerpen

2.2 Nonlinear (geodesic distance-based) NFindR

Results on Artificial dataset

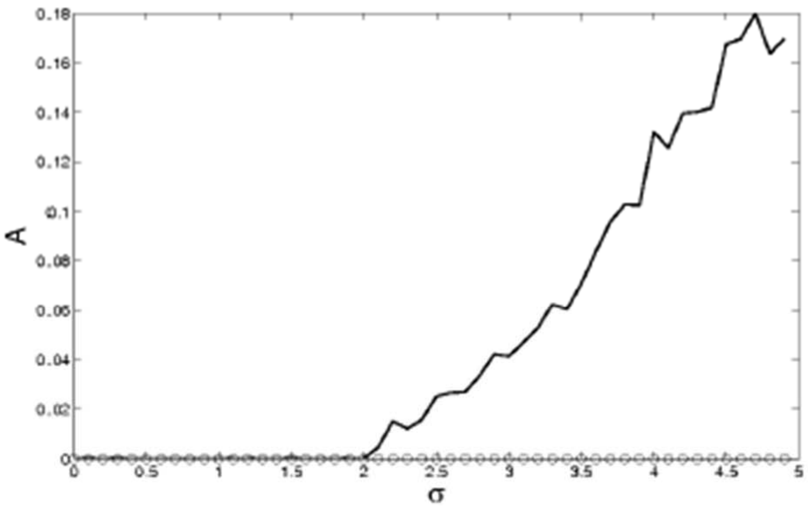


Fig. 4. The average minimum spectral angle A as a function of σ, with the N-FINDR algorithm (solid line), and the non-linear algorithm with $K = 20$ (circles).

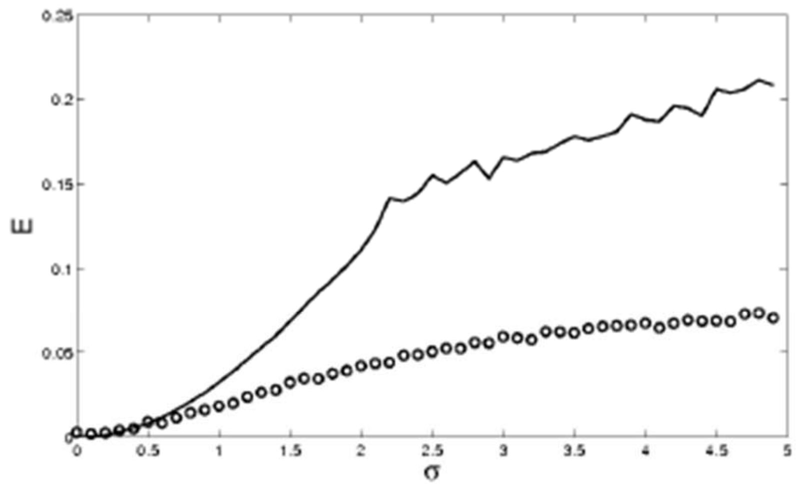
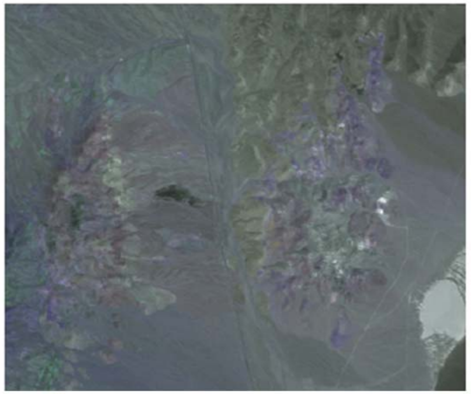


Fig. 5. The averaged absolute error E on the abundances as a function of σ, with the N-FINDR algorithm (solid line), and the non-linear algorithm with $K = 20$ (circles).

Universiteit Antwerpen

2.2 Nonlinear (geodesic distance-based) NFindR

Results on real dataset: Cuprite



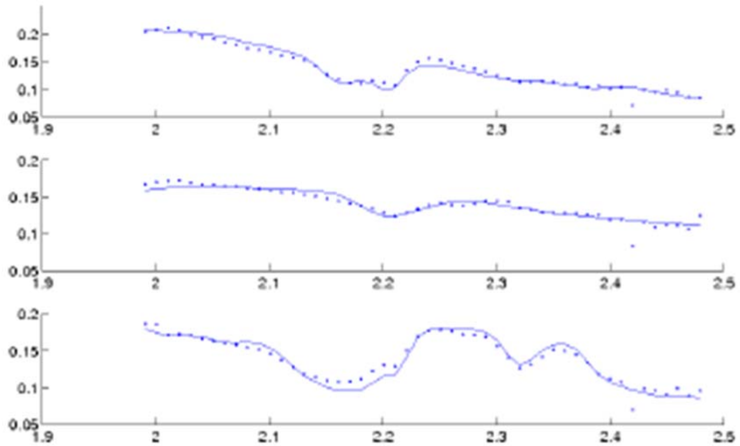


Fig. 7. Three (out of $p = 14$) extracted endmembers (dots) as found with the N-FINDR algorithm, and the library spectra of smallest spectral angle (solid line). Top: Kaolinite. Middle: Montmorrilonite. Bottom: Alunite. The spectral angles are 0.056, 0.048 and 0.043 respectively.

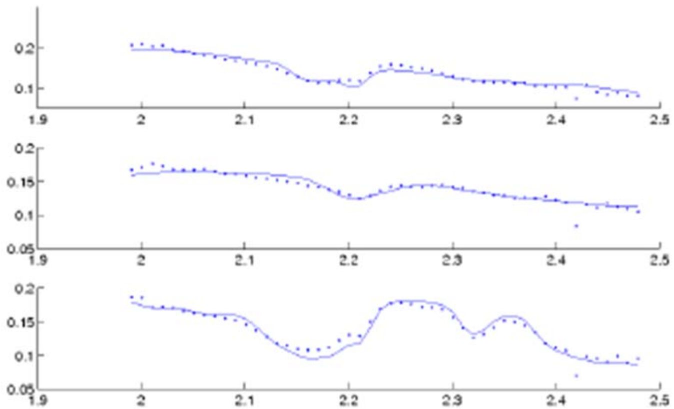
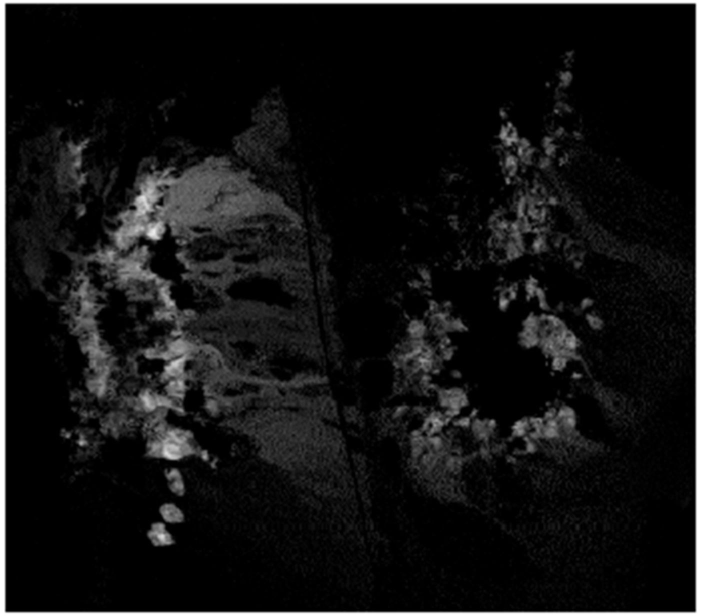


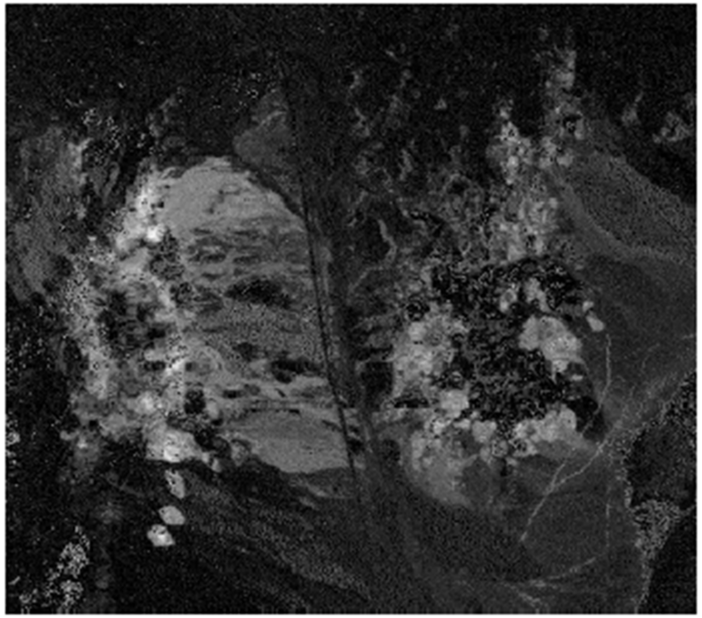
Fig. 8. Three (out of $p = 16$) extracted endmembers (dots) as found with the non-linear algorithm with $K = 20$, and the library spectra of smallest spectral angle (solid line). Top: Kaolinite. Middle: Montmorrilonite. Bottom: Alunite. The spectral angles are 0.070, 0.049 and 0.056 respectively.

Universiteit Antwerpen

2.2 Nonlinear (geodesic distance-based) NFindR

Results on real dataset: Cuprite (Alunite)

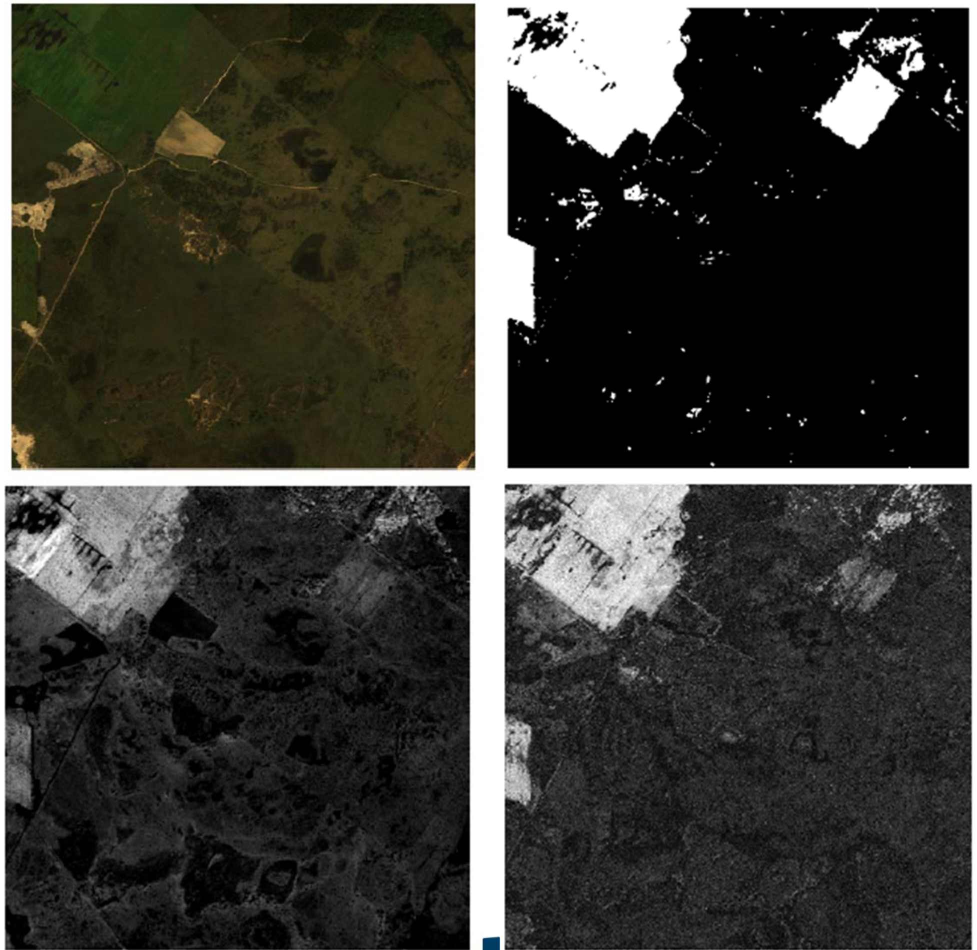




Universiteit Antwerpen

2.2 Nonlinear (geodesic distance-based) NFindR

Results on real dataset: Heathland



3. Geometric Unmixing

Universiteit Antwerpen

3.1 Simplex Projection Unmixing (SPU)

Spectral unmixing, viewed as a minimization problem

$$\{\hat{a}_i\} = \underset{\{\hat{a}_i\}_i}{\operatorname{argmin}} \left\| \sum_{i=1}^{p} \hat{a}_i \boldsymbol{e}_i - \boldsymbol{x} \right\|_2$$

Without constraints: Classical LS-problem

$$\hat{\boldsymbol{a}} = (\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{x} \quad \Rightarrow \quad \boldsymbol{x}' = \boldsymbol{E}\hat{\boldsymbol{a}} = \boldsymbol{E}(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{x}$$

Geometric interpretation: projection operator

$$P_{\mathrm{LSU}} = \boldsymbol{E}(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}} \quad \Rightarrow \quad \boldsymbol{x}' = P(\boldsymbol{x})$$

3.1 Simplex Projection Unmixing (SPU)

Geometric interpretation

- LS-solution corresponds to plane projection
- Including the constraints: Simplex projection

Simplex projection:

$$\boldsymbol{x}' \in S : \boldsymbol{x}' = P(\boldsymbol{x}) \iff \forall \boldsymbol{y} \in S : \|\boldsymbol{x} - \boldsymbol{y}\|_2 \geq \|\boldsymbol{x} - \boldsymbol{x}'\|_2$$

? P ?

Universiteit Antwerpen

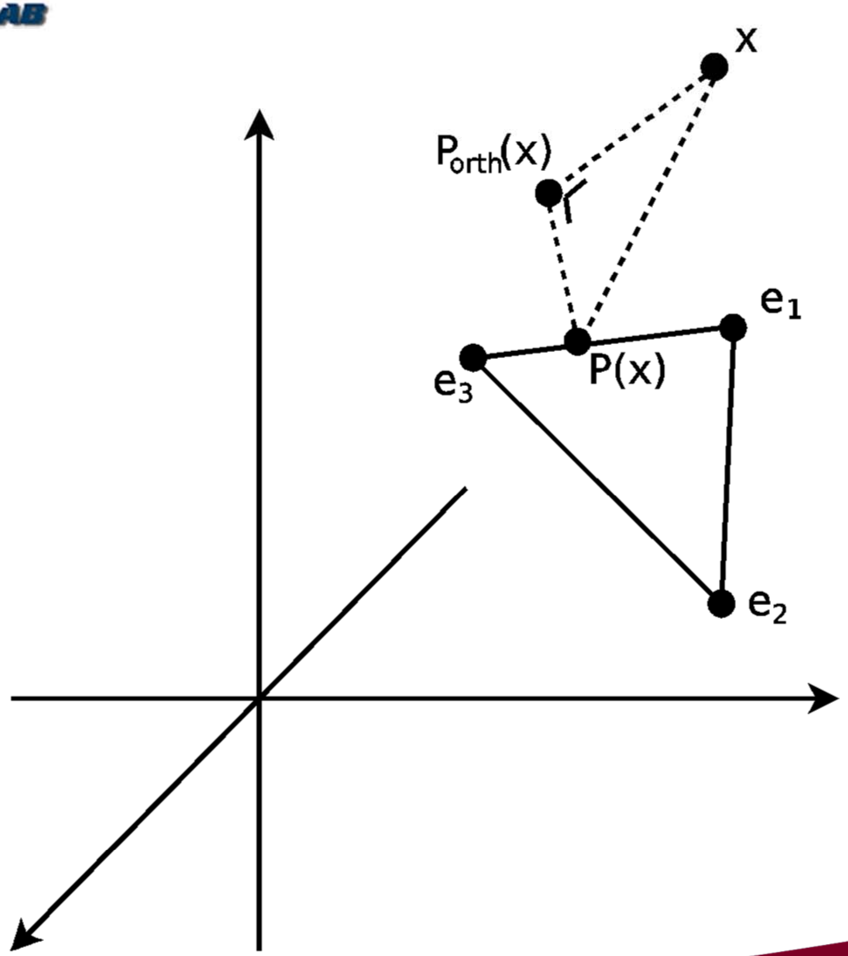
3.1 Simplex Projection Unmixing (SPU)

Some observations:

- Orthogonal projection onto the simplex plane leaves the simplex projection invariant

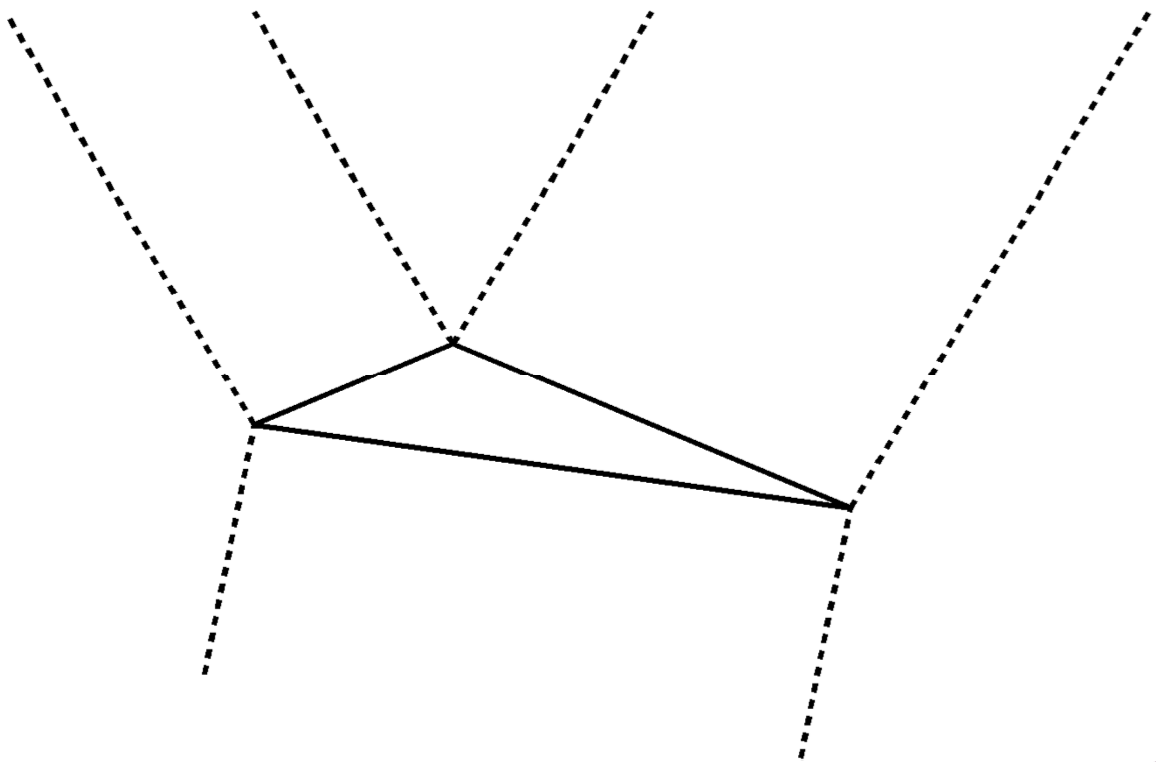
- The simplex projection of a point outside the simplex, but in the simplex plane, always lies on the boundary of the simplex

3.1 Simplex Projection Unmixing (SPU)



$$P\left(P_{\text{orth}}(x)\right) = P(x)$$

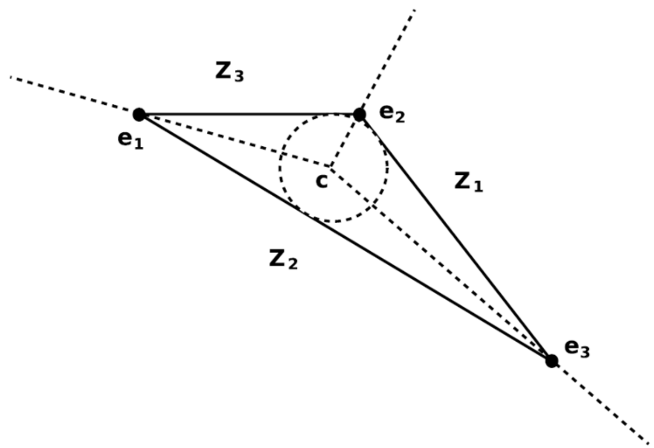
3.1 Simplex Projection Unmixing (SPU)



3.1 Simplex Projection Unmixing (SPU)

Incenter:

- Intersection of all (p-2)-dimensional planes that bisect the dihedral angles between the simplex faces
- Center of largest hyperspere within simplex

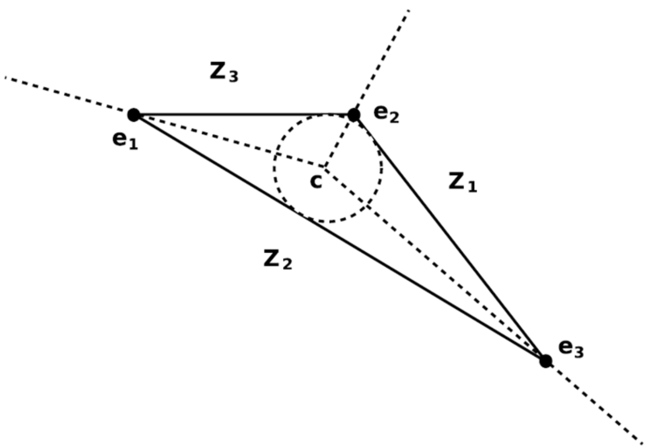


Universiteit Antwerpen

3.1 Simplex Projection Unmixing (SPU)

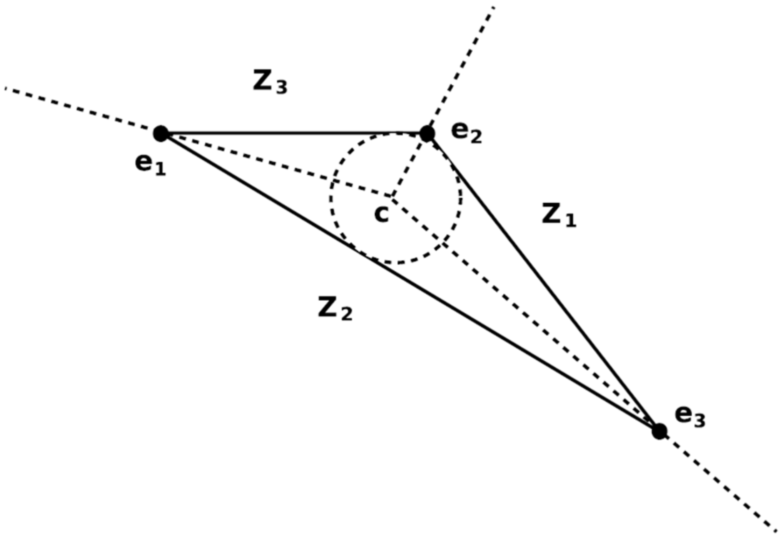
Bissective cones: set of all points intersecting a simplex face

$$\boldsymbol{x} \in Z_i \Leftrightarrow \exists b_1, \ldots, b_p \geq 0 : \begin{cases} \boldsymbol{x} = \boldsymbol{c} + \sum_{j=1}^{p} b_j(\boldsymbol{e}_j - \boldsymbol{c}) \\ b_i = 0 \end{cases}$$



Universiteit Antwerpen

3.1 Simplex Projection Unmixing (SPU)



Can be used to estimate the abundance that has to be zero:

$$\boldsymbol{x} \notin S, \boldsymbol{x} \in Z_i \quad \Rightarrow \quad \hat{a}_i = 0$$

Not always correct (for p > 3), but can be used in practice

Universiteit Antwerpen

3.1 Simplex Projection Unmixing (SPU)

Recursive simplex projection unmixing (SPU) algorithm:

1. Project the point onto the simplex plane.
2. If the point lies inside the simplex, finish.
3. Else, find which abundance has to be zero.
4. Remove the endmember from the set of endmembers and go to step 1.

Finally, the projected point is a linear combination of the remaining endmembers, which is an exactly solvable system of linear equations.

3.1 Simplex Projection Unmixing (SPU)

Properties of the algorithm

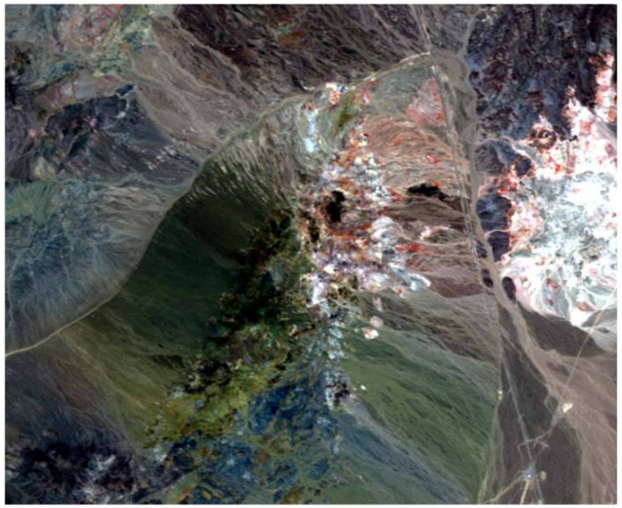
- Highly parallelizable. Very fast compared to e.g. FCLSU.
- No optimization steps required.
- Can be written completely in distance geometry (DSPU)

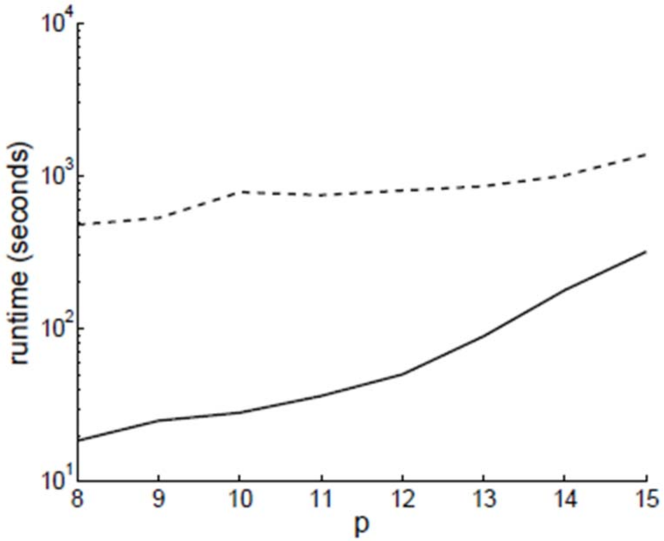
3.1 Simplex Projection Unmixing (SPU)

Results: Cuprite data set

NfindR to extract endmembers

Unmixing via FCLSU (as reference) and SPU



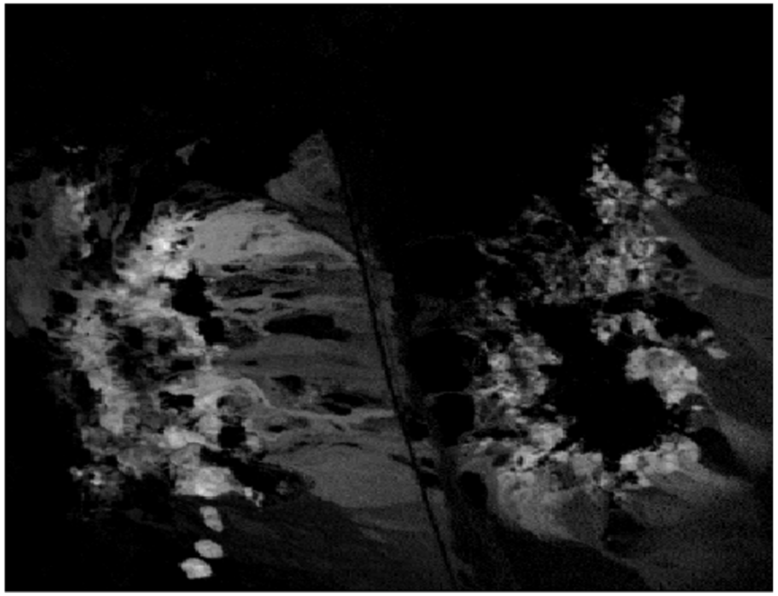


3.1 Simplex Projection Unmixing (SPU)

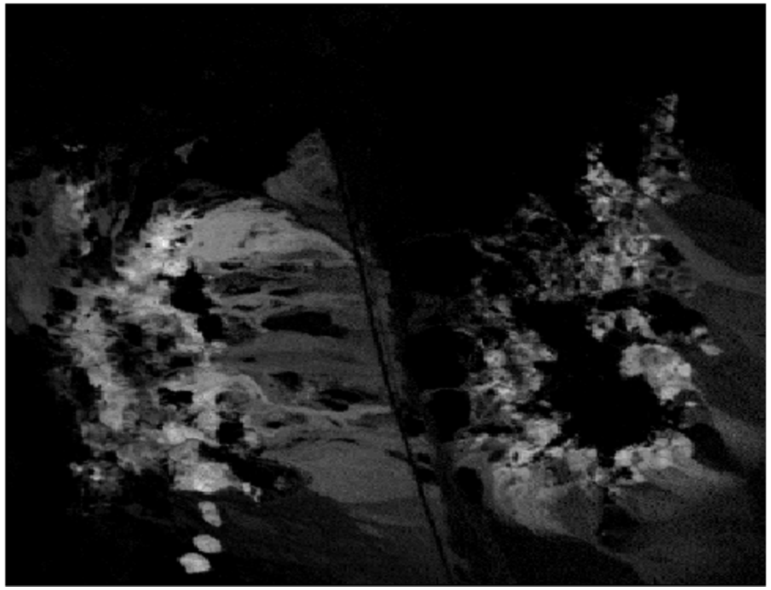
Results: Cuprite data set

Typical situation: 99.7% of abundances differ by less than 10^{-7}.

E.g. for the alunite endmember:

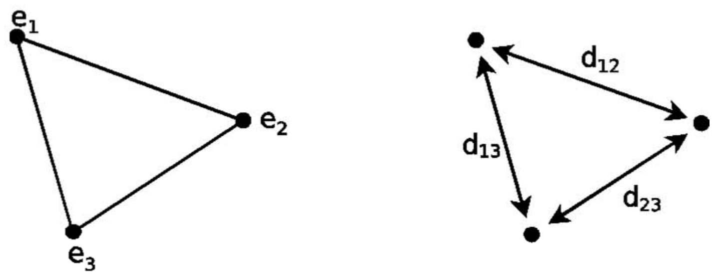


FCLSU



SPU

3.2 Distance-based SPU



$$V_p^2 \sim \underbrace{\left(\boldsymbol{d}\boldsymbol{C}_{p-1}^{-1}\boldsymbol{d}^{\mathrm{T}}\right)}_{\text{orth. dist.}} \underbrace{\det\left(\boldsymbol{C}_{p-1}\right)}_{\text{vol p-1 simplex}}, \quad \boldsymbol{d} = (d_{p1}^2, \ldots, d_{p,p-1}^2, 1)$$

$$d_\perp^2(\boldsymbol{e}_1; \boldsymbol{e}_2, \ldots, \boldsymbol{e}_p) = \frac{\boldsymbol{d}_1^{\mathrm{T}} \boldsymbol{C}_{2,\ldots,p}^{-1} \boldsymbol{d}_1}{2}$$

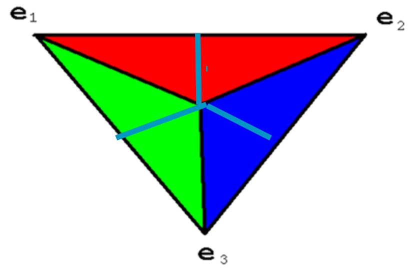
Projected point in simplex in terms of distance to endmembers

$$d^2(\boldsymbol{x}_\perp, \boldsymbol{e}_i) = d^2(\boldsymbol{x}, \boldsymbol{e}_i) - d_\perp^2(\boldsymbol{x}; \boldsymbol{e}_1, \ldots, \boldsymbol{e}_p)$$

Universiteit Antwerpen

3.2 Distance-based SPU

- **The incenter**



$$a_i^c = \frac{V_i}{\sum_{i=1}^{p} V_i}$$

Distance from incenter to endmembers:

$$d^2(\boldsymbol{x}, \boldsymbol{y}) \quad = \quad (\boldsymbol{a}_x - \boldsymbol{a}_y)^{\mathrm{T}} \left(-\frac{1}{2} \boldsymbol{J} \boldsymbol{D} \boldsymbol{J} \right) (\boldsymbol{a}_x - \boldsymbol{a}_y)$$

$$J_{ij} = \delta_{ij} - \frac{1}{p}$$

Universiteit Antwerpen

3.2 Distance-based SPU

- **The bissective cones**

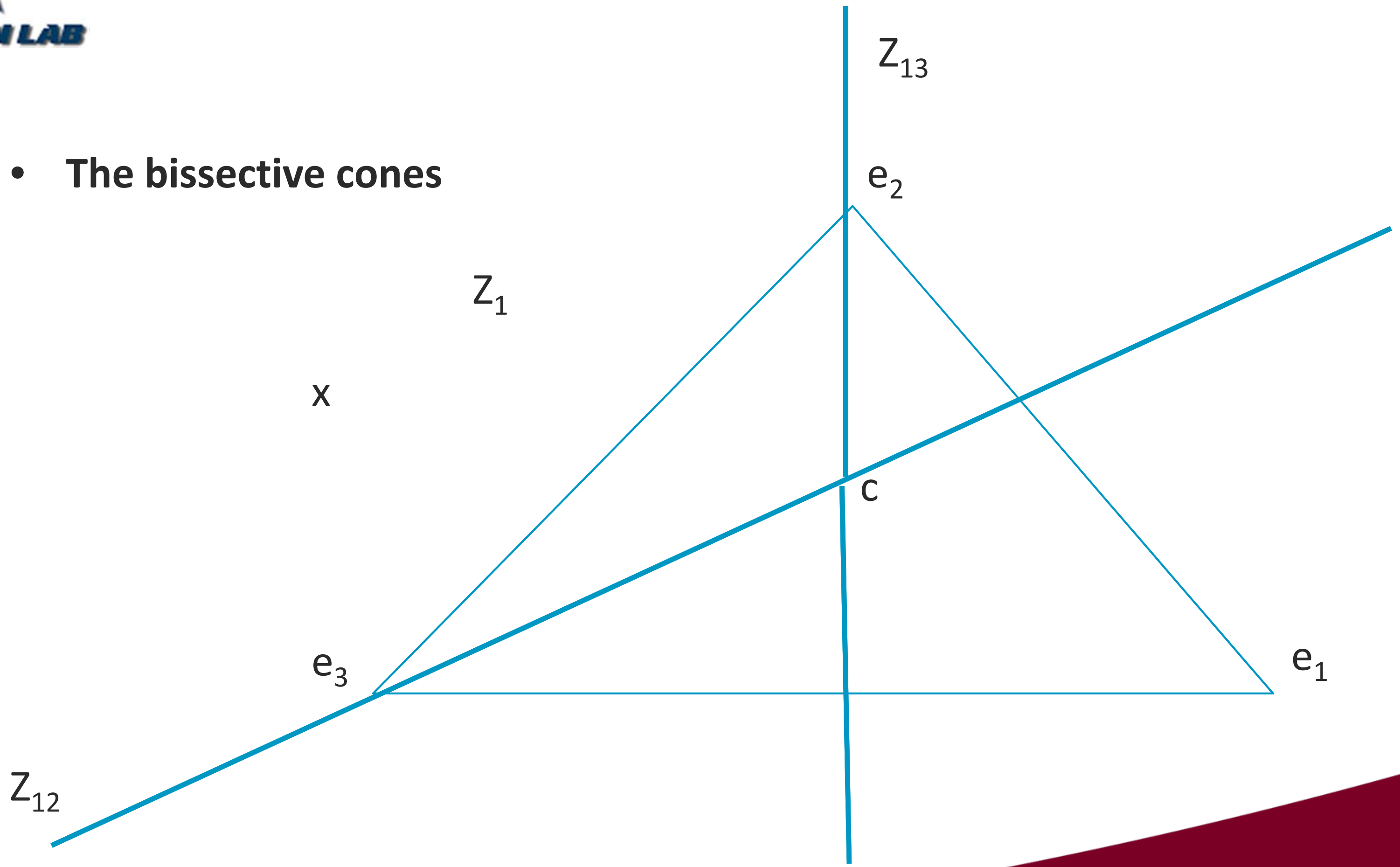
General property: given a p-dimensional simplex and 2 points.

- They are either on the same or opposite side of the simplex plane.
- Distance x between 2 points can be calculated by matrix completion.

$$\left|\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & x & d_1^x & \dots & d_p^x \\ 1 & x & 0 & d_1^y & \dots & d_p^y \\ \hline 1 & d_1^x & d_1^y & D_{11} & \dots & D_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & d_p^x & d_p^y & D_{p1} & \dots & D_{pp} \end{array}\right| = 0 \qquad \left|\begin{array}{c|c} \boldsymbol{X} & \boldsymbol{Y} \\ \hline \boldsymbol{Y}^T & \boldsymbol{D} \end{array}\right| = |\boldsymbol{D}| \left|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{D}^{-1}\boldsymbol{Y}^T\right|$$

3.2 Distance-based SPU

- **The bissective cones**

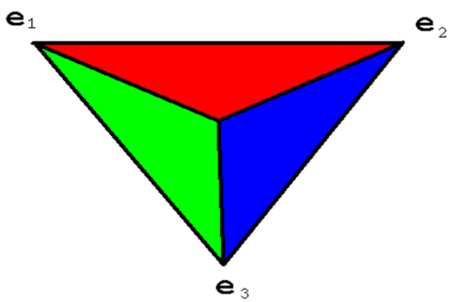


3.2 Distance-based SPU

- **The abundance coefficients**

Point x is inside simplex if for all i, both points x and e_i are on same side of (p-1)-dimensional simplex plane.

Abundance obtained by:



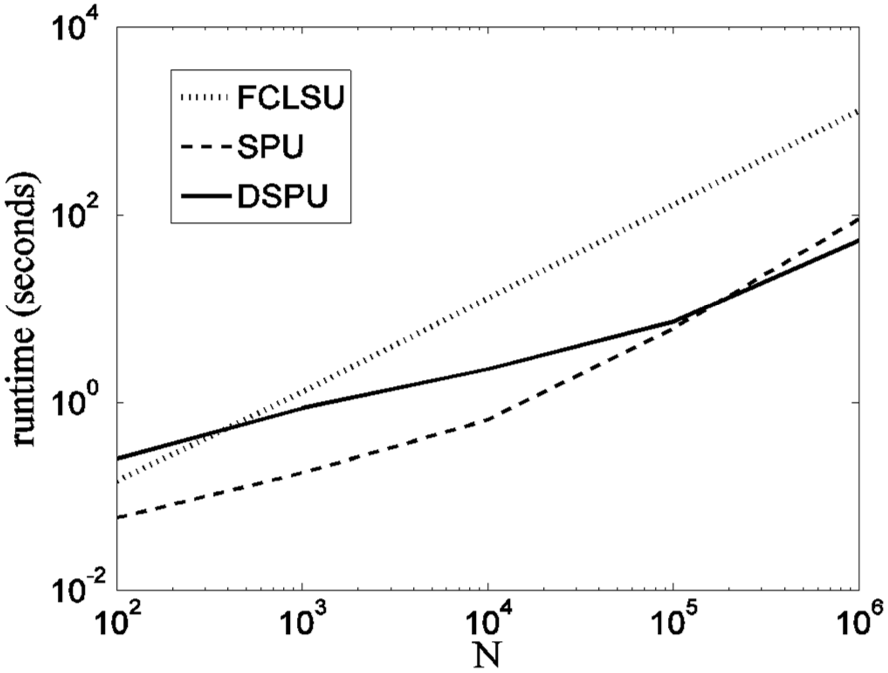
$$a_1 = \frac{V({\color{blue}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

$$a_2 = \frac{V({\color{green}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

$$a_3 = \frac{V({\color{red}\blacktriangle})}{V({\color{red}\blacktriangle}{\color{green}\blacktriangle}{\color{blue}\blacktriangle})}$$

3.2 Distance-based SPU

Runtimes on artificial data set (USGS library)



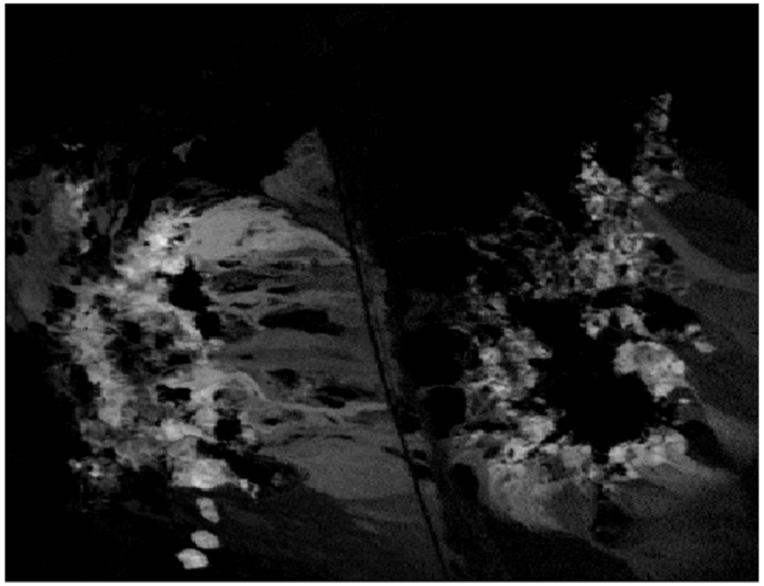
Universiteit Antwerpen

3.2 Distance-based SPU

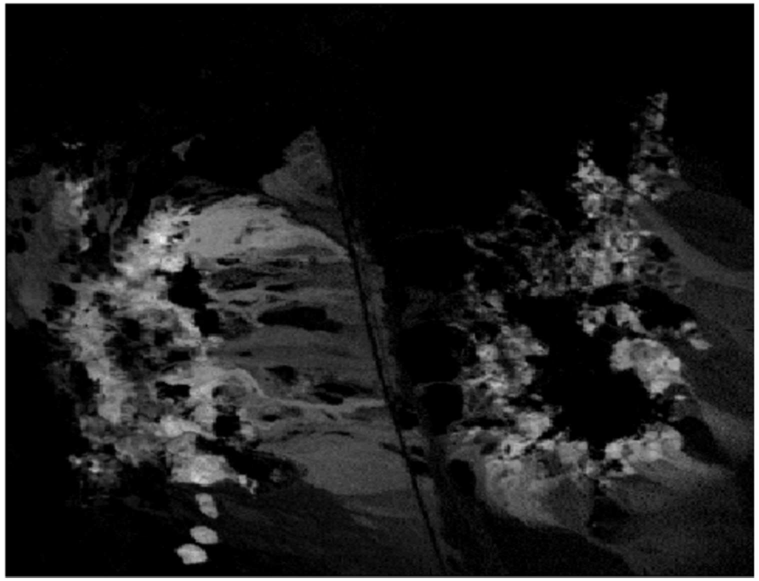
Cuprite: Linear unmixing

Typical situation: 99.7% of abundances differ by less than 10^{-7}.

E.g. for the alunite endmember:



FCLSU



DSPU

Universiteit Antwerpen

3.3 Geodesic Distance Geometric Unmixing

A data driven, fully-constrained non-linear unmixing method:

- Endmember extraction by geodesic distance-based NFindR.
- Unmixing via DSPU, applied to the geodesic distances.

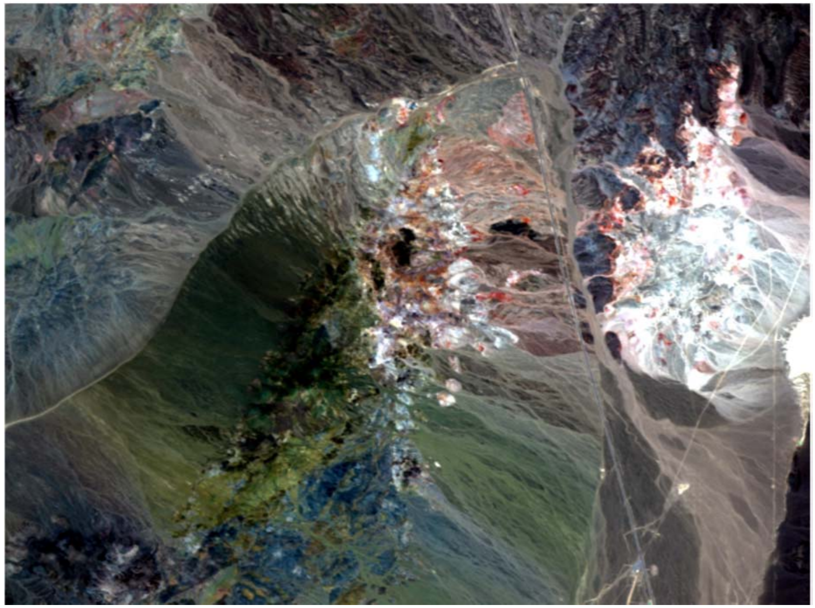
Universiteit Antwerpen

3.3 Geodesic Distance Geometric Unmixing

Results: Cuprite data set

Linear unmixing via NFindR and FCLSU

Non-linear unmixing via the proposed method

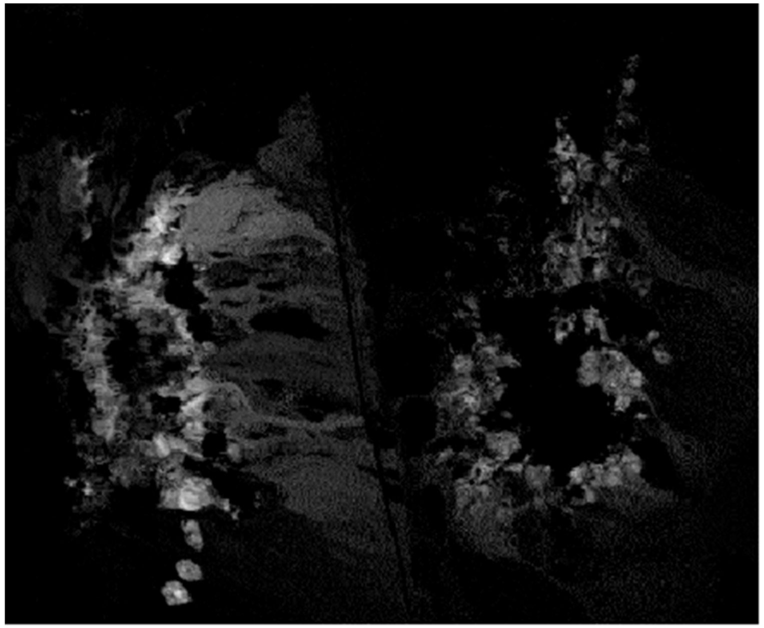


Universiteit Antwerpen

3.3 Geodesic Distance Geometric Unmixing

Cuprite: Non-linear unmixing

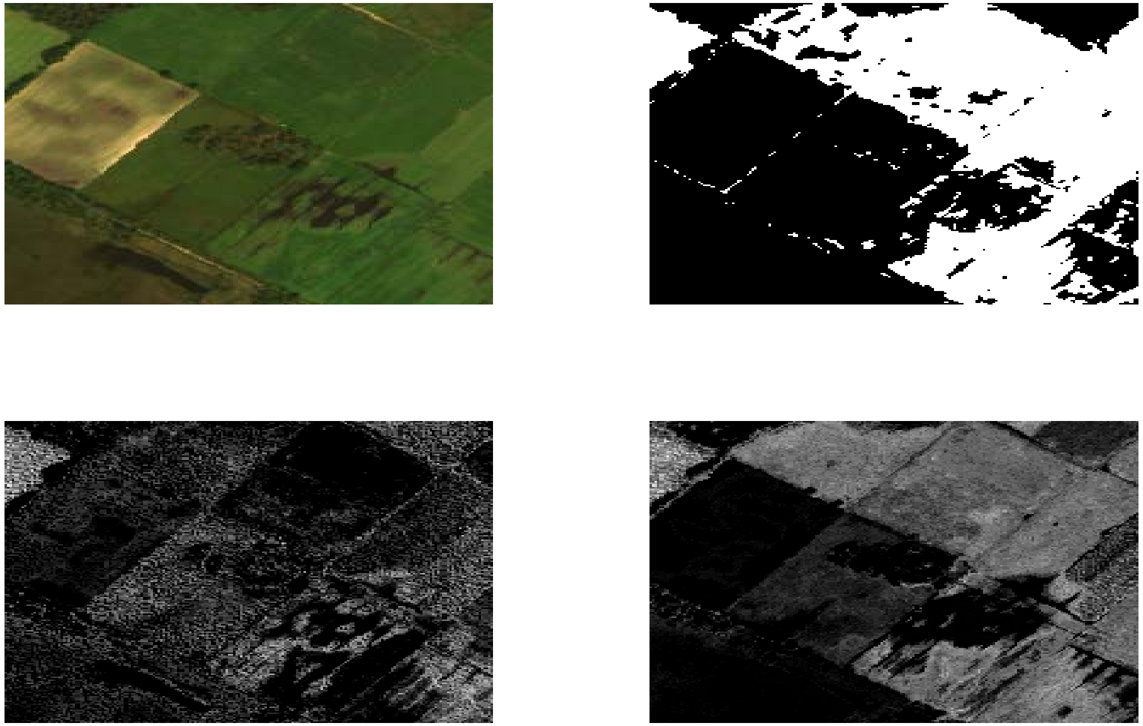
Alunite endmember:



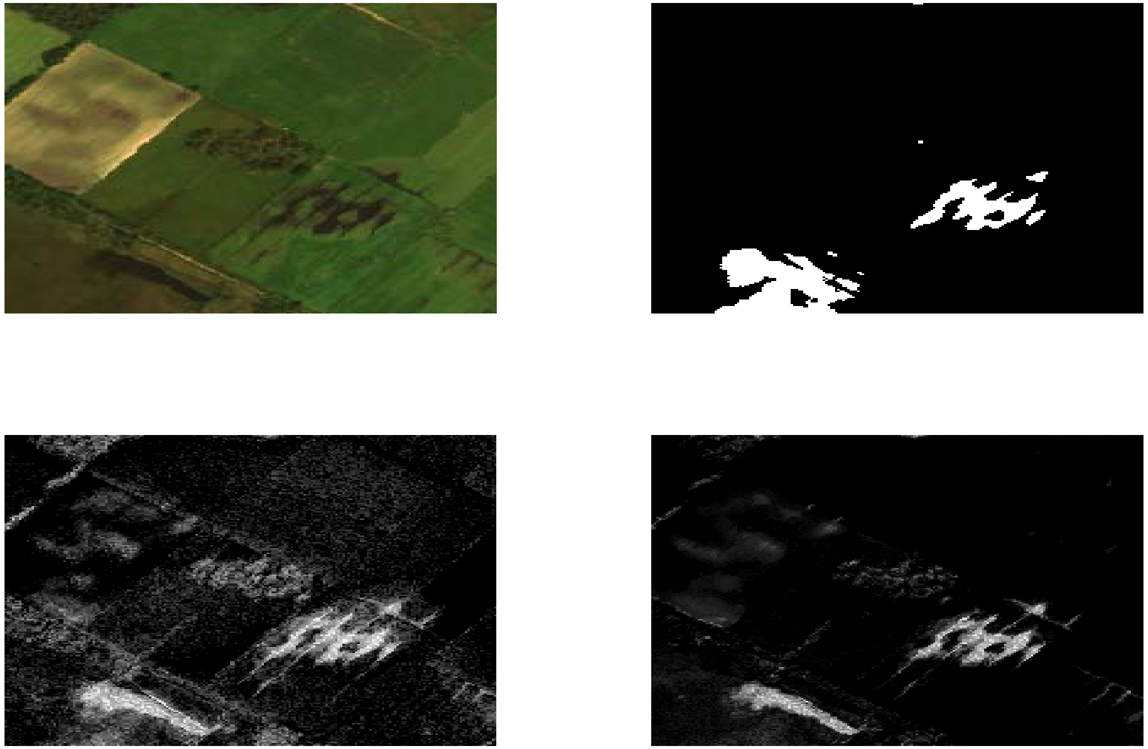
NFindR + FCLSU

Non-lin. NFindR + DSPU

3.3 Geodesic Distance Geometric Unmixing



3.3 Geodesic Distance Geometric Unmixing



4. Related, Applied and Future work

Universiteit Antwerpen

4.1 Geometric intrinsic dimensionality estimation

Popular ID estimation techniques:

- Virtual Dimensionality
- HySime

Manifold techniques:

Correlation dimension: count number of points inside small balls around each data point as function of the radius.

Q. Du, "Virtual dimensionality estimation for hyperspectral imagery with a fractal-based method," Proc. WHISPERS, pp. 1-4, 2010

Inverse approach: Use nearest neighbor distances as estimators for the radius.

A. M. Farahmand, C. Szepesvári and J.-Y. Audibert, "Manifold-adaptive dimension estimation," Proc. 24th intl. conf. Machine learning, pp. 265-272, 2007

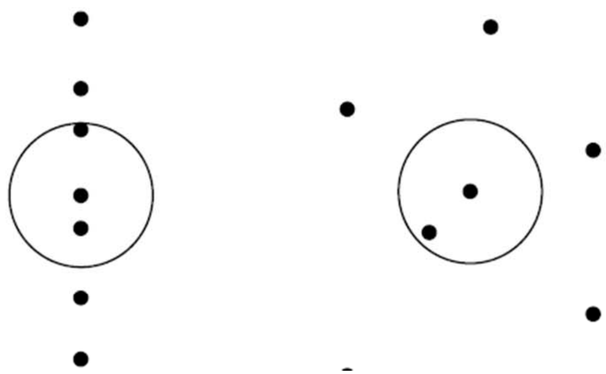
4.1 Geometric intrinsic dimensionality estimation

Consider random data set of N points with known ID q
Probability to find a point in a ball with radius ϵ around $\boldsymbol{x}$:

$$P(\boldsymbol{y} \in B(\boldsymbol{x}, \epsilon)) \sim \rho(\boldsymbol{x})\epsilon^q$$

Number of points expected inside this ball:

$$k = C(\rho) N \rho(\boldsymbol{x}) \epsilon^q$$



4.1 Geometric intrinsic dimensionality estimation

The HIDENN algorithm

Let $r_k(\boldsymbol{x})$ be the distance to the k'th nearest neighbor of $\boldsymbol{x}$. There are then k data points in $B(\boldsymbol{x}, r_k(\boldsymbol{x}))$. On average:

$$k = [C(\rho) N \rho(\boldsymbol{x})] \, (r_k(\boldsymbol{x}))^q$$

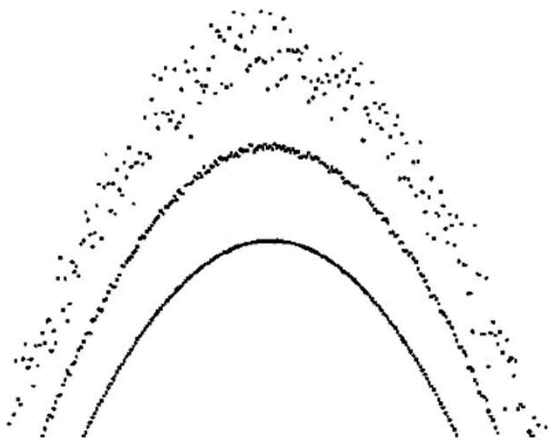
For two different values k and k', we find the relation

$$q(\boldsymbol{x}) = \frac{\log(k) - \log(k')}{\log(r_k(\boldsymbol{x})) - \log(r_{k'}(\boldsymbol{x}))}$$

Hence for every data point $\boldsymbol{x}$, we can generate an ID estimate, dependent on two parameters k and k'

4.1 Geometric intrinsic dimensionality estimation

ID estimation depends strongly on noise



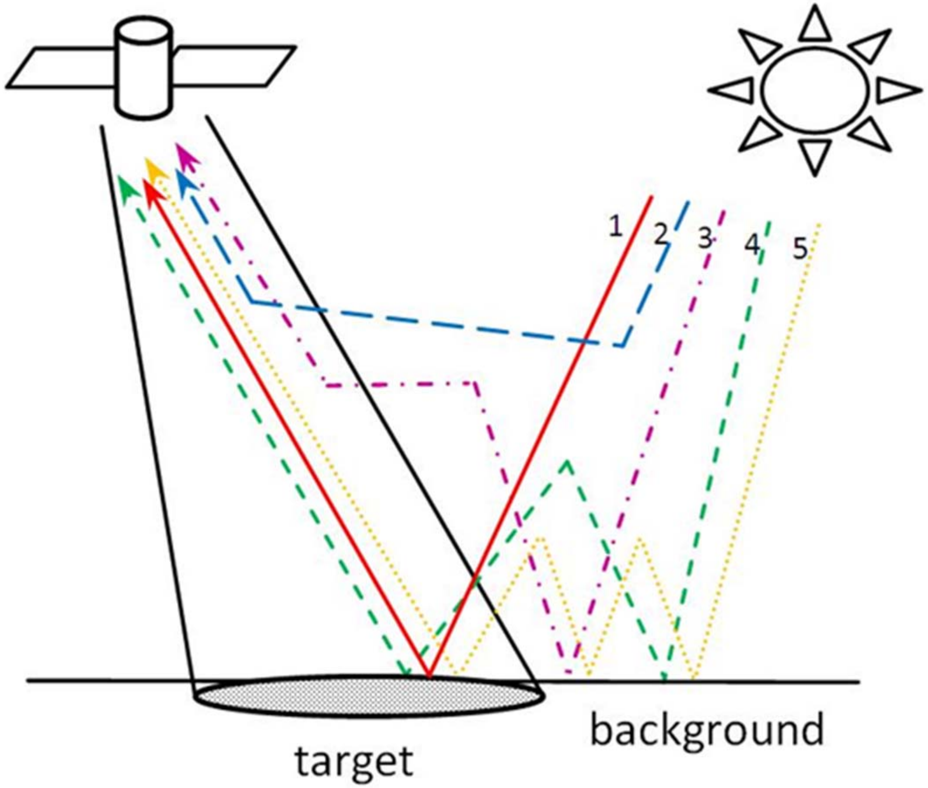
A preliminary denoising step will improve the results drastically

4.1 Geometric intrinsic dimensionality estimation

Advantages:

- Simple
- Consistent results over different data sets
- Bot linearly and nonlinearly mixed datasets
- Independent on spectral dimensionality
- NN distances are required anyway in geometric unmixing framework

4.2 Application: Spectral unmixing of Adjacency effect



1,3: $\boldsymbol{r} = \alpha_t(\mathbf{tgt}) + \alpha_b(\mathbf{bkgd})$

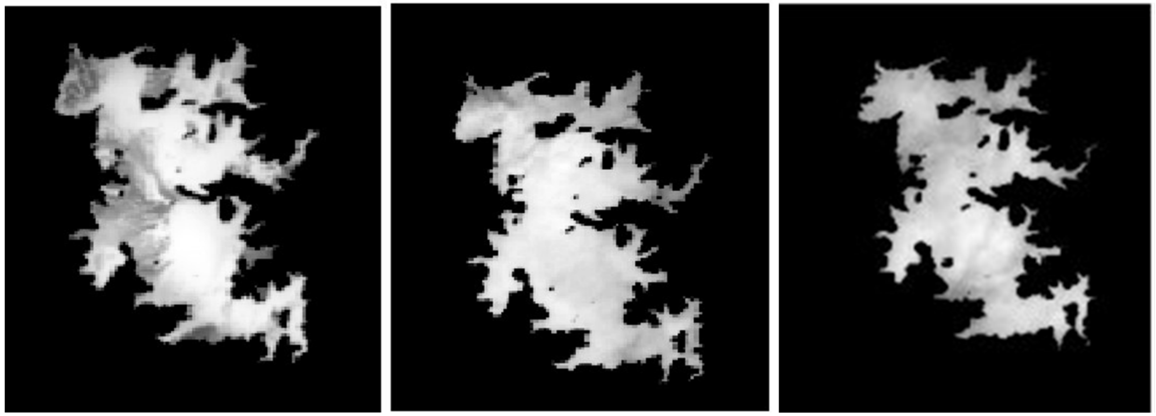
1,3,4: $\boldsymbol{r} = \beta_t(\mathbf{tgt}) + \beta_b(\mathbf{bkgd}) + \gamma_{tb}(\mathbf{tgt} \cdot \mathbf{bkgd})$

$\vdots$

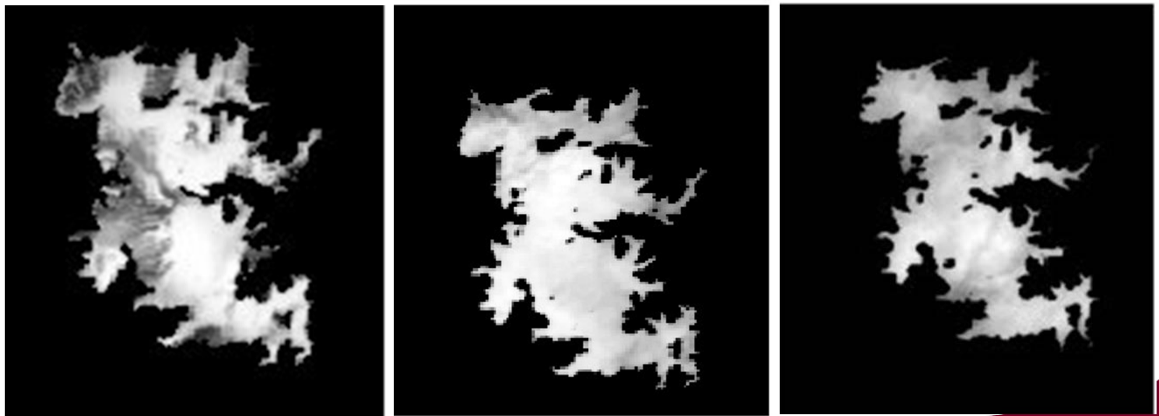
Universiteit Antwerpen

4.2 Application: Spectral unmixing of Adjacency effect

SIMEC:



FCLSU:



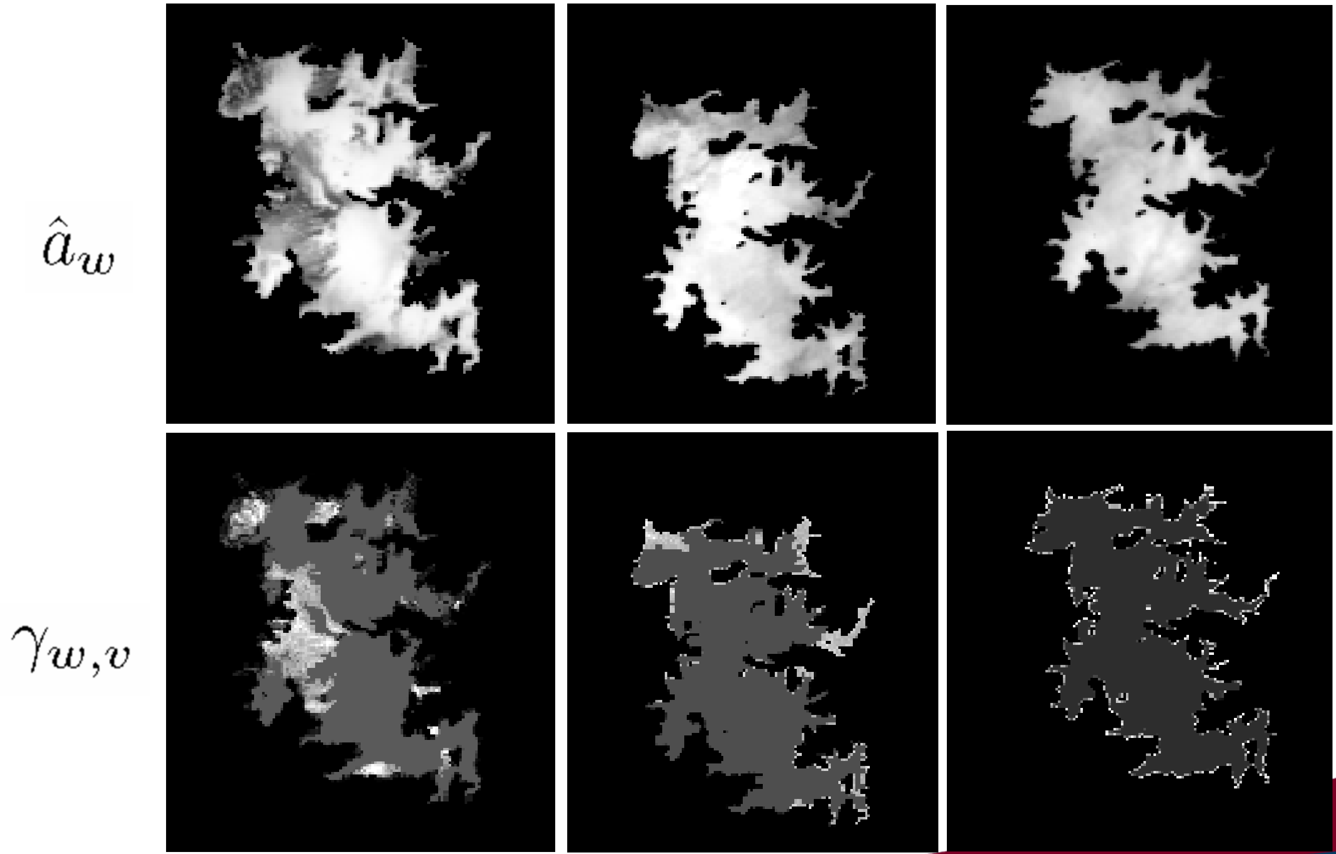
4.2 Application: Spectral unmixing of Adjacency effect

GBM: $$\boldsymbol{r} = \sum_{j=1}^{P} \hat{a}_j \boldsymbol{e}_j + \sum_{j=1}^{P-1} \sum_{k=j+1}^{P} \gamma_{jk} \hat{a}_j \hat{a}_k \boldsymbol{e}_j \odot \boldsymbol{e}_k + \boldsymbol{n}$$

$$\hat{a}_j \geq 0, \sum_{j=1}^{P} \hat{a}_j = 1, \gamma_{jk} \in [0, 1]$$

Universiteit Antwerpen

4.2 Application: Spectral unmixing of Adjacency effect



Universiteit Antwerpen

4.3 Future work: Towards Streaming Unmixing

- No dimensionality reduction required
- Only local information of manifold required
- Distance-base formulation: time and memory efficient

- Challenge: streaming endmember extraction
  - Initialization
  - Simplex growing/shrinking

- Possible applications:
  - On-board unmixing, avoids downlink of image data
  - Real-time spectral unmixing
  - Domain adaption

Universiteit Antwerpen