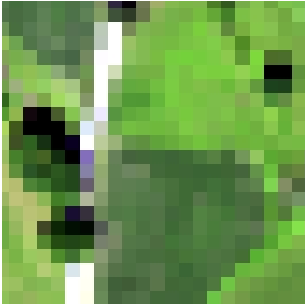
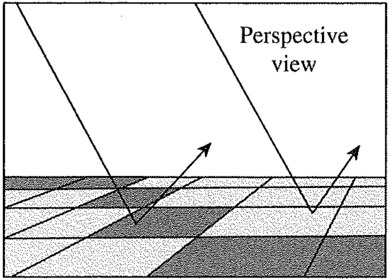
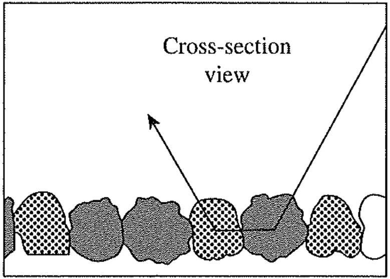
>>> **Spectral Unmixing**
>>> **GRSS Summer School**

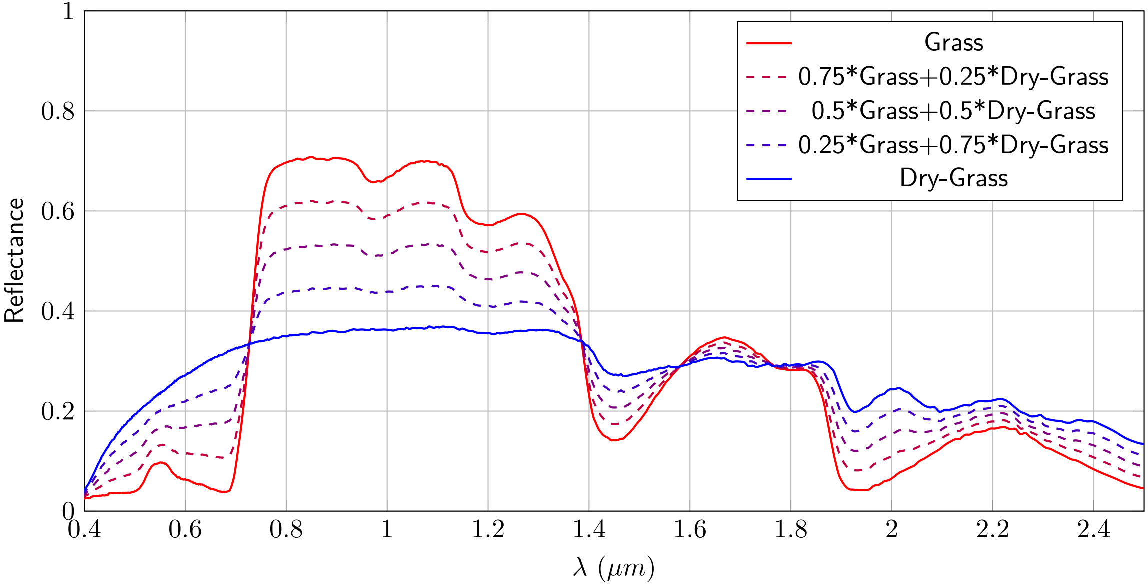
Name: Mathieu Fauvel (UMR Dynafor)
Date: *[2017-04-26 Wed 13:30]–[2017-04-26 Wed 16:30]*

- *When a pixel contains several materials: all these materials contribute to the collected reflectance* [MLC16]
- It is called a *mixel* or *mixed spectra* and pure contributions are termed *endmembers*








Grass
0.75*Grass+0.25*Dry-Grass
0.5*Grass+0.5*Dry-Grass
0.25*Grass+0.75*Dry-Grass
Dry-Grass
Reflectance
1
0.8
0.6
0.4
0.2
0
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
λ (μm)

>>> Applications

- Conventional mapping applications [Bio+12]:
  - Mineral proportions
  - Vegetation covers
- Multisource fusion:
  - Sources of various spatial resolution
  - Sources of various spectral resolution
- Misc:
  - Post-improvement of classification maps
  - Clouds reconstruction of multi-temporal HI

>>> Unmixing processing chain

1. Athmospheric correction
2. Dimensionality reduction
3. Find endmembers
   - ⋆ Hyperspectral library
   - ⋆ Unsupervised
4. Find abundances

- Each pixel is a convex linear combination of endmenbers (pure spectra):

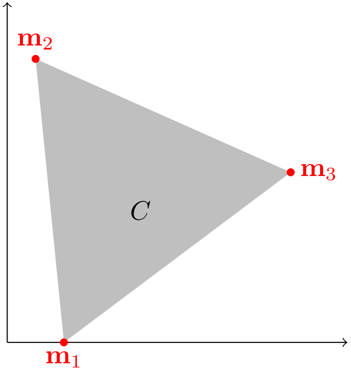
$$\mathbf{x} = \sum_{j=1}^{p} \alpha_j \mathbf{m}_j + \mathbf{e}$$

  - $\mathbf{m}_j$ is the j^{th} endmember,
  - α_j is the abundance of endmember j,
  - $\mathbf{e}$ is the modelling error.
- The abundances are subject to the following constraints:
  - Non negativity: $\alpha_j \geq 0, \forall j = 1, \ldots, p$
  - Sum to one: $\sum_{j=1}^{p} \alpha_j = 1$.
- Matricial formulation:

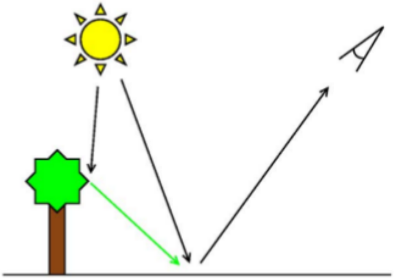
$$\mathbf{x} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{e}$$

where $\mathbf{M} = \left[\mathbf{m}_1, \ldots, \mathbf{m}_p\right]$.

- ⋆ If endmembers are affinely independent, hyperspectral pixels live in the convex hull C generated by the endmembers
- ⋆ $C := \left\{ \mathbf{x} = \mathbf{M}\boldsymbol{\alpha} \middle| \sum_{j=1}^{p} \alpha_j = 1,\ \alpha_j \geq 0, \forall j = 1, \ldots, p \right\}$



⋆ Consider multiple reflections [HPG14]



⋆ The model

$$\mathbf{x} = \sum_{j=1}^{p} \alpha_j \mathbf{m}_j + \sum_{j,k=1}^{p} \beta_{j,k} \mathbf{m}_j \odot \mathbf{m}_j$$

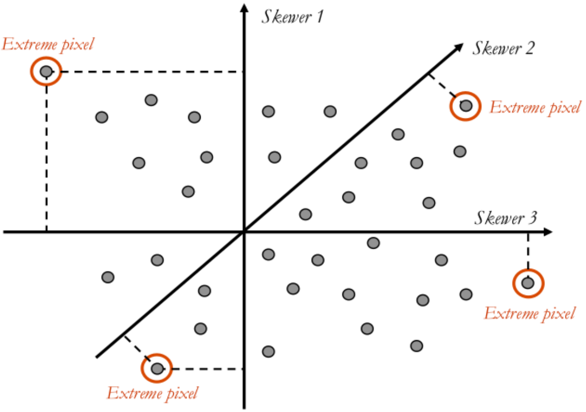
⋆ Several possible constraints on α and β

- Physics based model: Intimate mixture (e.g., Hapke model)
- Computer graphics: ray tracing
- Kernel methods:
  - "kernelized" linear models
  - Construct "smart" kernels

Find the subspace signal in order to search the endmembers:

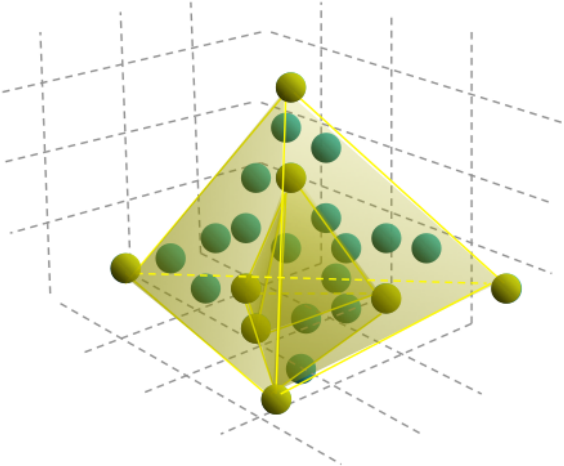
- Principal component analysis: *find the subspace that minimizes reconstruction error*
- Minimum noise fraction: *find the subspace that maximizes the SNR*
- HySime (hyperspectral subspace identification)
  - *Estimation of the signal and noise subspaces*
  - *Find the subpace that best represents the signal subspace*

1. Apply MNF
2. Project pixels onto random vector and find *extreme* projected pixels and store them
3. Pixels with the highest score are identified as the *purest one*



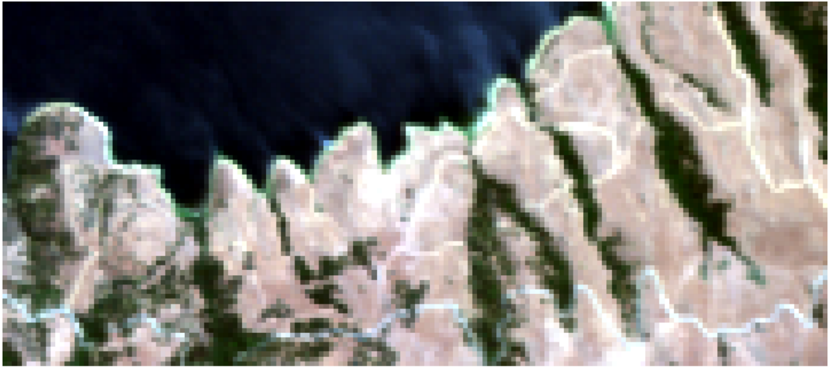
>>> N-FINDR [Win99]

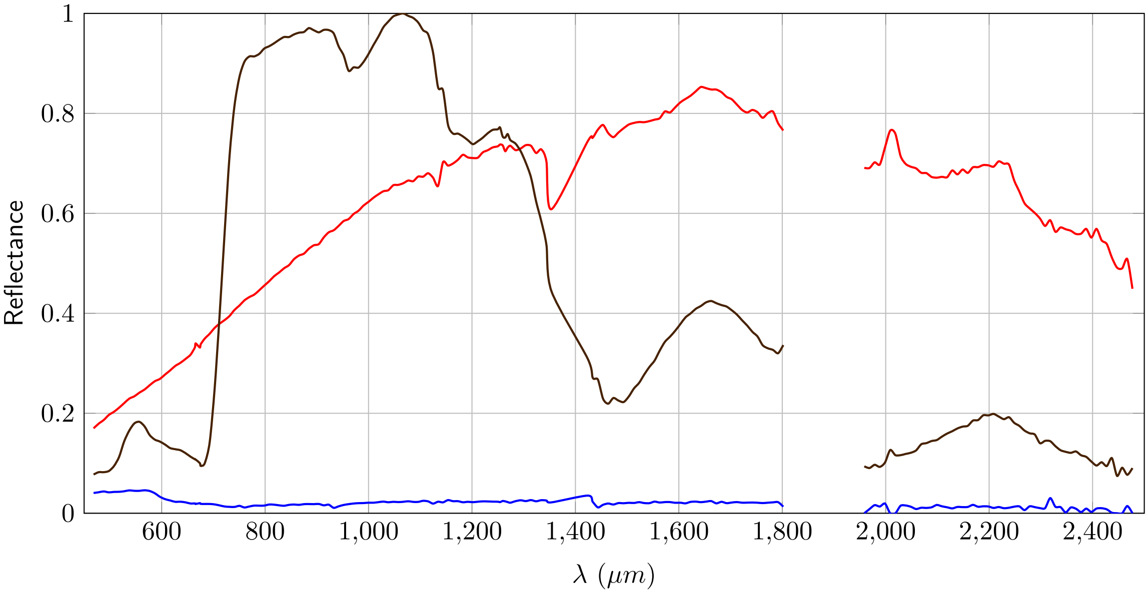
- ⋆ Assumes pure pixels are present
- ⋆ Find the pixels that maximizes the volume of the simplex



- Iteratively project pixels on *orthogonal direction* to the subspace spanned by previously selected endmembers
- New endmember correpond to the extreme of the projection
- Works similarly to *orthogonal subspace projection* but accounting for the noise

>>> Moffett data




1
0.8
0.6
0.4
0.2
0
Reflectance
600
800
1,000
1,200
1,400
1,600
1,800
2,000
2,200
2,400
λ (μm)

- One the endmembers are estimated, abundances can be estimated by minimizing a reconstruction error Re:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \left\{ Re(\mathbf{x}, \mathbf{M}\boldsymbol{\alpha}) \right\}$$

- Conventionally Re is chosen as the mean square square error:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{M}\boldsymbol{\alpha}\|^2$$

- Without any constraint the solution is given by

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{M}^\top \mathbf{M}\right)^{-1} \mathbf{M}^\top \mathbf{x}$$

- Positive constraints:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{M}\boldsymbol{\alpha}\|^2$$
$$\text{Subject to } \alpha_j \geq 0, \forall j = 1, \dots, p$$

- Full set of constraints

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{M}\boldsymbol{\alpha}\|^2$$
$$\text{Subject to } \sum_{j=1}^{p} \alpha_j = 1, \alpha_j \geq 0, \forall j = 1, \dots, p$$

- Not explicit solution ! Needs to optimize a quadratic function under convex constraints.
- This is known as *non negative least squares unmixing* or *fully constraints least square unmixing*
- Easy trick: use only positive constraints that add manually the *sum to one constraint* !

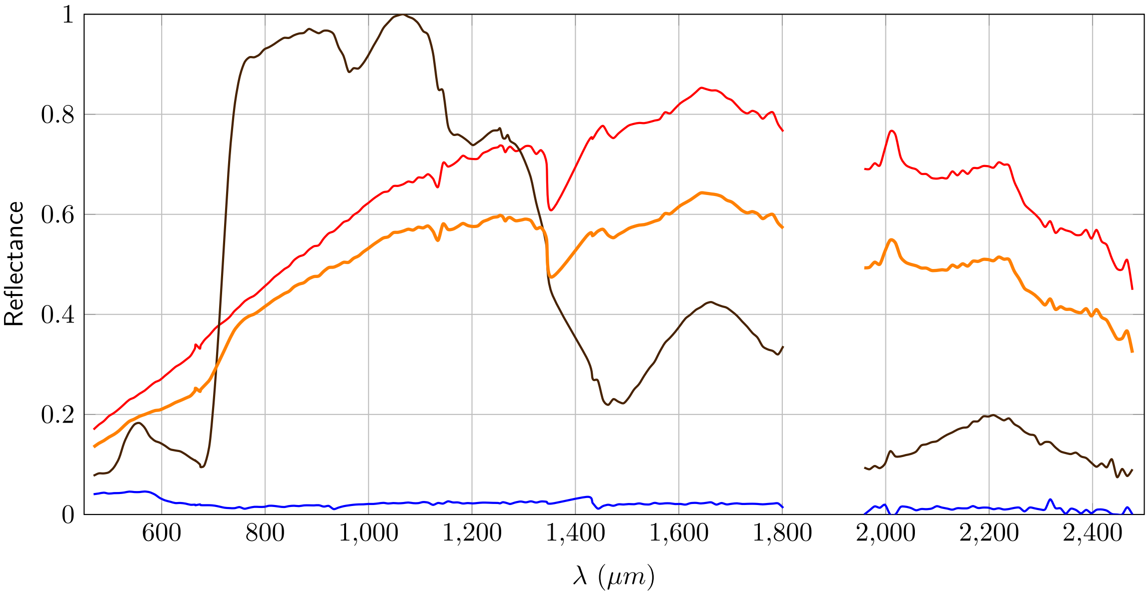
```
from scipy import optimize
import scipy as sp

# Load endmembers
M = sp.loadtxt("../Unmixing/figures/endmembers.csv",delimiter=',')[:,1:]
x = 0.2*M[:,0] + 0.7*M[:,1] + 0.1*M[:,2]

# NNLS
res = optimize.nnls(M,x)
print res[0]
```

```
[ 0.2  0.7  0.1]
```

>>> FCLS 1/2

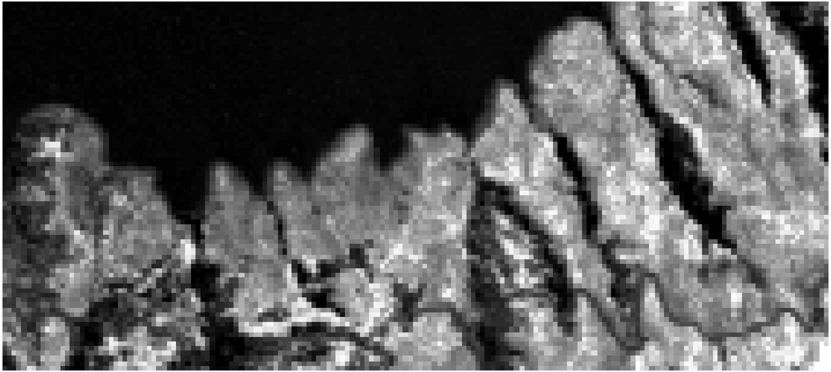
```
from scipy import optimize
import scipy as sp

# Loss
def loss(alpha,x,M):
    e = x-sp.dot(M,alpha)
    return (e**2).sum()

def jac(alpha,x,M):
    e = x-sp.dot(M,alpha)
    return -2*sp.dot(M.T,e)

cons = {'type':'eq','fun':lambda alpha: 1-alpha.sum(),'jac':lambda alpha: -alpha}
bnds = ((0, None), (0, None), (0, None,))

# Load endmembers
M = sp.loadtxt("../Unmixing/figures/endmembers.csv",delimiter=',')[:,1:]
x = 0.2*M[:,0] + 0.7*M[:,1] + 0.1*M[:,2]

# Optimize
alpha0 = sp.ones((3,))/3.0
res = optimize.minimize(loss, alpha0, args=(x,M,), jac=jac,constraints=cons, method='SLSQP',
                        bounds=bnds,)
print res['x']
```

```
[ 0.16441937  0.67116125  0.16441937]
```


Reflectance
1
0.8
0.6
0.4
0.2
0
600
800
1,000
1,200
1,400
1,600
1,800
2,000
2,200
2,400
λ (μm)

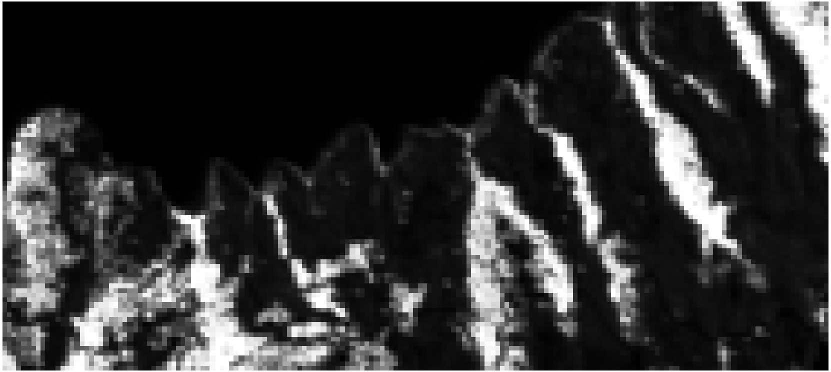
```python
from scipy import optimize
import scipy as sp
import rasterTools as rt
import pysptools.eea as eea

# Number of endmembers
NE = 3

# Load images
im,GeoT,Proj = rt.open_data('../Data/Moffett_full.tif')
[h,w,b]=im.shape
wave = sp.loadtxt('../Data/wave_moffett.csv',delimiter=',')

# Compute endmenbers
nfindr = eea.NFINDR()
M = nfindr.extract(im.astype(float), NE, normalize=False).T

abundances = sp.empty((h,w,NE))
for h_ in xrange(h):
    for w_ in xrange(w):
        x = im[h_,w_,:]
        res = optimize.nnls(M,x)
        a = res[0]
        abundances[h_,w_,:]= (a/a.sum())

# Write the image
rt.write_data("../Data/Moffett_abundances.tif",abundances,GeoT,Proj)
```

>>> Abundances 2/4



>>> Abundances 3/4

>>> Abundances 4/4



- When the number of endmembers are large (e.g. selected from spectral library), the observed spectral vector is usually a combination of few ones.
- Add *sparsity* constraints in the optimization problem such as

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0$$
$$\text{Subject to } \|\mathbf{x} - \mathbf{M}\boldsymbol{\alpha}\|^2 \leq \delta, \alpha_j \geq 0, \forall j = 1, \ldots, p$$

- However, this problem is NP-hard: no straightforward solution
- Use convex formulation from *sparse regression* (LASSO):

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{M}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1$$
$$\text{Subject to } \alpha_j \geq 0, \forall j = 1, \ldots, p$$

>>> Outline

Bioucas-Dias, J. M. et al. "Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches". In: *IEEE J. Sel. Topics Appl. Earth Observations Remote Sensing* 5.2 (Apr. 2012), pp. 354–379.

Boardman, Joseph W., Fred A. Kruse, and Robert O. Green. *Mapping Target Signatures via Partial Unmixing of AVIRIS data*. 1995.

Heylen, R., M. Parente, and P. Gader. "A Review of Nonlinear Hyperspectral Unmixing Methods". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.6 (June 2014), pp. 1844–1868. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2014.2320576.

Manolakis, D., R. Lockwood, and T. Cooley. *Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms*. Cambridge University Press, 2016. ISBN: 9781316033401. URL: https://books.google.fr/books?id=Zeg8DQAAQBAJ.

Nascimento, J. M. P. and J. M. B. Dias. "Vertex component analysis: a fast algorithm to unmix hyperspectral data". In: *IEEE Transactions on Geoscience and Remote Sensing* 43.4 (Apr. 2005), pp. 898–910. ISSN: 0196-2892. DOI: 10.1109/TGRS.2005.844293.

Winter, M. E. "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data". In: *Image Spectrometry*. Vol. 3753. V. SPIE, 1999.