Maël GUIHENEUF
Fry 2nd June 2010

Traineeship – Summer 2009

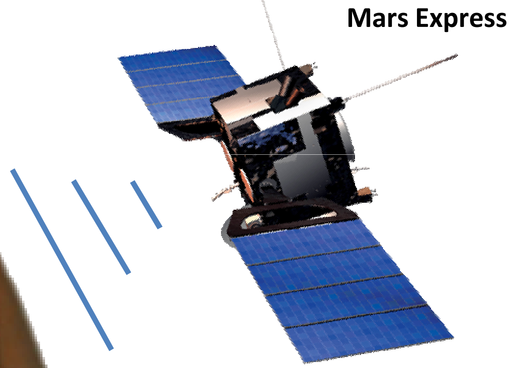
Advisors:
Frédéric Schmidt
Albrecht Schmidt

Deployment of BPSS algorithm (Bayesian positive Source Separation) on Matlab

I. **Why this project?**
- OMEGA
- The projet
- The algorithm - BPSS

II. **What have I done?**
- Profiling
- Optimizations
- Deployment

III. **Use cases / Issues**
- Input data
- The algorithm
- Interpretation of the results



Centrale Nantes

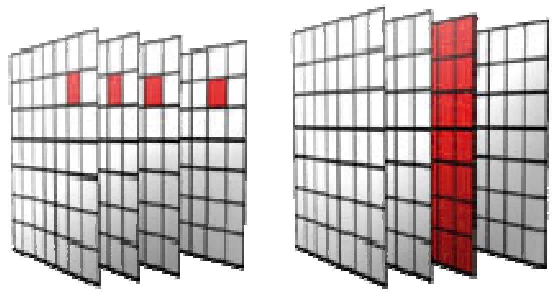
esa

Why this project?

Mars Express – OMEGA

OMEGA = **O**bservatoire pour la **M**ineralogie, l'**E**au, les **G**laces et l'**A**ctivité

spectro-imager onboard **Mars Express**



Spectral

Spatial

~~**Channel V : 0.38 → 1.05 µm**~~
Channel C : 0.93 → 2.73 µm
Channel L : 2.55 → 5.1 µm

256 wavelengths

Resolution : 350 m → 4 km
Image : 128 x 800 pixels
(> 100 000 px)

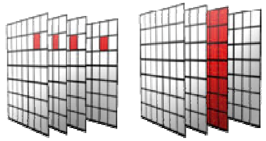
Why this project?

Supervised methods

Goal → to study components on the surface

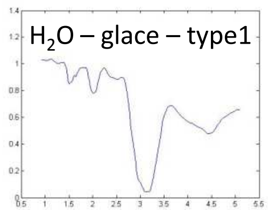
1. Supervized methods :

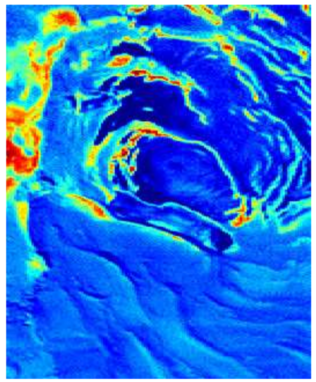
Cube OMEGA



+

laboratory





Map of this component

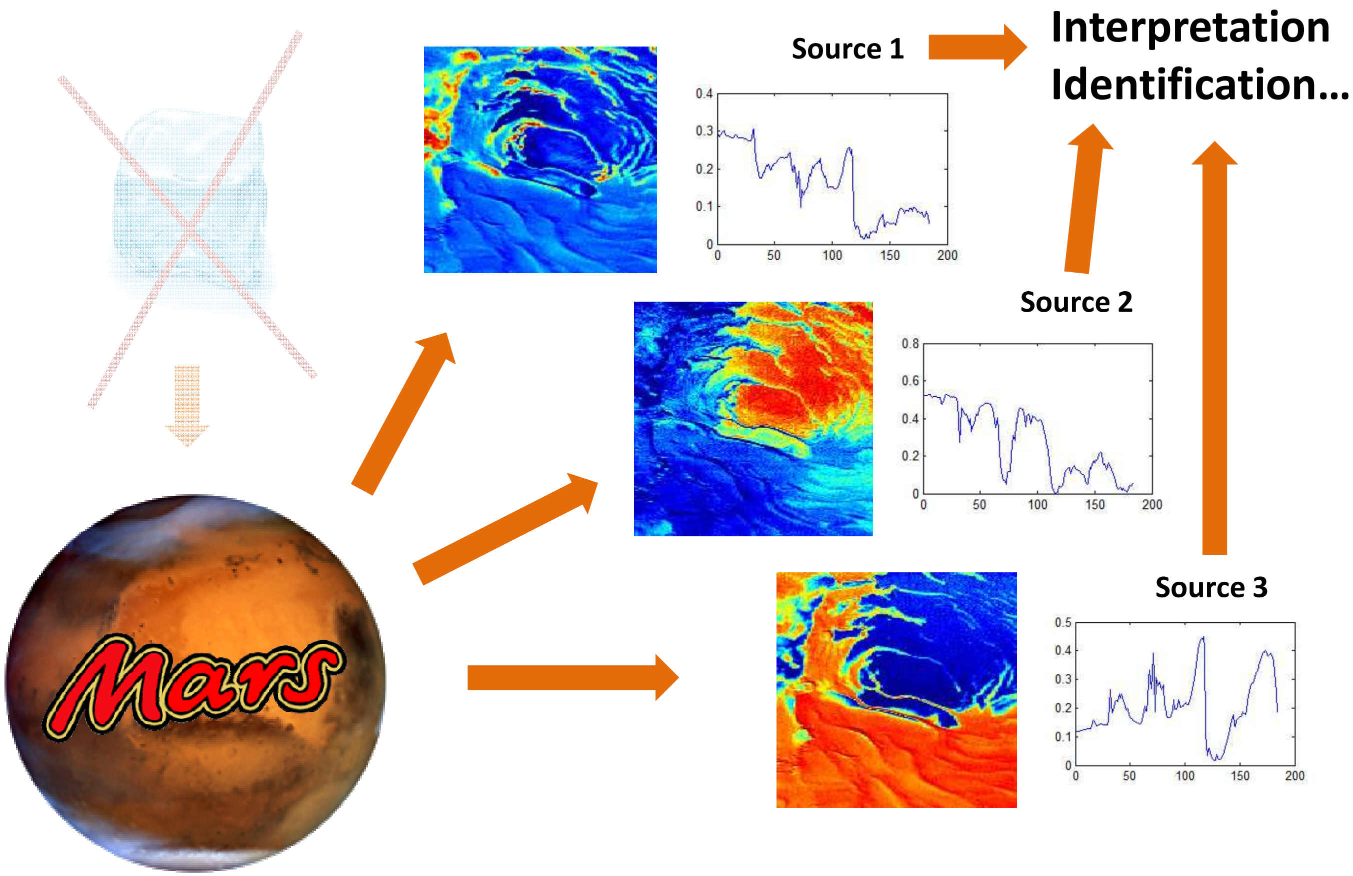
Maël GUIHENEUF
Fry 2nd June 2010

Why this project?

Blind methods

Goal → to study components on the surface

2. Blind methods :



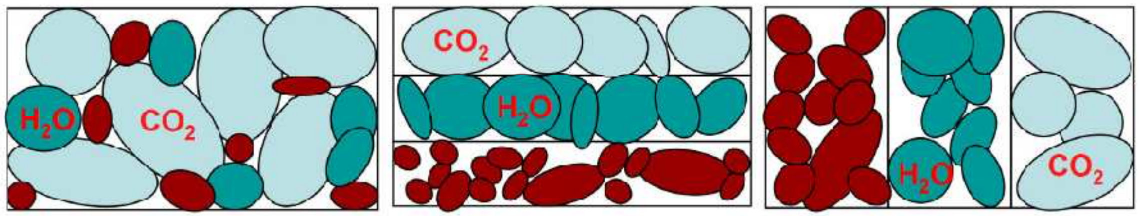
Why this project?

Limitations in those methods

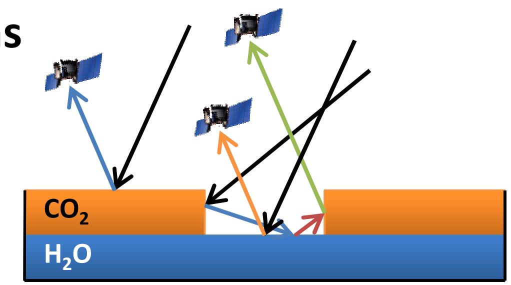
Sources of error:

The reflectance of the surface depends on a lot of factories

- **For a single component→ different granulations**
- **Mixes of componants**



- **Reflections**



→ Many spectrum possibilities

+ Unexpected components

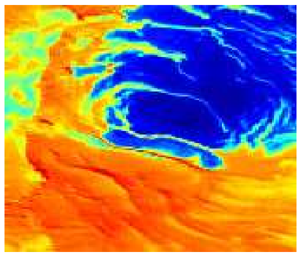
Why this project?

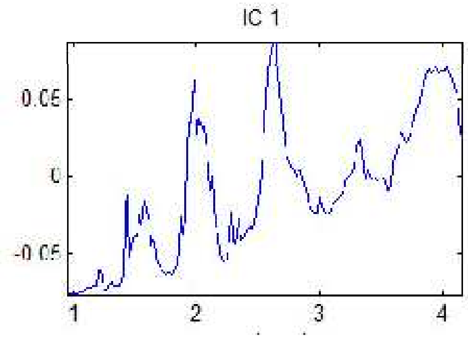
Choice of this algorithm

Goal → to obtain the sources statistically the more different as possible

Classic methods→ we don't impose the non-negativity of sources

Exemple: (JADE)





→ Negative spectrum
- **No physic**
- **No credibility**
- **No relevant**

Our choice: BPSS (Bayesian **P**ositive **S**ource **S**eparation**)**

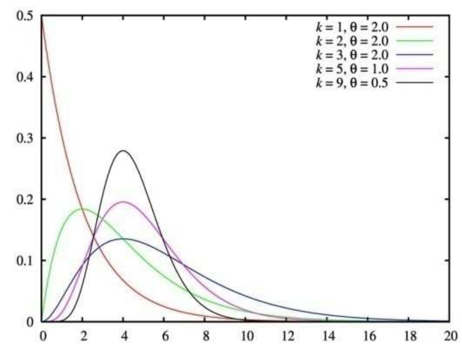
Why this project?

Presentation of the algorithm

BPSS, in a few words…

Linear mixing: X = A.S + E

noise (E), **sources** (S), **data** (X), **Mixing matrix** (A)



→ Estimate : $P(A,S \mid X) \propto P(X|A,S) \,.\, P(A) \,.\, P(S)$

Positivity: each matrix element follows a **Gamma distribution** (2 hyper parameters : $\theta = [\, \alpha , \beta \,]$)

→ Estimate : $P(A,S,\theta \mid X) \propto P(X \mid A,S,\theta) \,.\, P(A) \,.\, P(S) \,.\, P(\theta)$

Using of **Gibbs Sampler** = calculate itratively $P(x_i)$ using only the multivariate conditional distribution $P(x_i \mid x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$

For us: $\mathbf{S^{(r+1)}}$ generated randomly using $\mathbf{P(S^{(r)} \mid X, A^{(r)}, \theta^{(r)})}$
$\mathbf{A^{(r+1)}}$ generated randomly using $\mathbf{P(A^{(r)} \mid X, S^{(r+1)}, \theta^{(r)})}$
$\mathbf{\theta^{(r+1)}}$ generated randomly using $\mathbf{P(A^{(r)} \mid X, A^{(r+1)}, S^{(r+1)})}$

Variant : BPSS2 → constraint of $\Sigma = 1$

Why this project?

Limitations on previous code

Memory problems:

X = A.S + E

Dimensions :
X = nb_pixels x nb_wavelengths = E
A = nb_pixels x nb_sources
S = nb_sources x nb_wavelengths

→ **Pixel selection** (before using BPSS) :

spatial selection among the most “energetic” points
considering the results of classical ICA method (JADE)
100 000 pixels → 300 pixels

? **Is the selection relevant?**

Problems with computation time – Many iterations:

- **burn-in iterations** (convergence)
- **other iterations** (statistic iterpretation)

? **Is the computation time reasonable?** (<< a few weeks**)**

Why this project?

Project planning

To divelop a complete application

- Flexible (different datasets)
- As fast as possible
- Calculations chain possible

MATLAB®

To authomatize the algorithm

- To stop the burn-in step
- To modificate parameters easily

To address issues

- Effet of pixel selection
- Robustness of the algorithm?
- Shall we "clean" input datas ?

Development

Profiling

Used memory : fuction "*memory*"

→ Study of the biggest array possible

Computing time: *Matlab Profiler*

→ Study of critical points

Profile Summary

Generated 18-May-2009 10:07:46 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
bpssdemo	1	179.608 s	0.162 s	
bpss2	1	179.027 s	17.504 s	
sample_abundances	30	142.608 s	19.443 s	
dtrandnmult	300000	123.158 s	87.390 s	
truncrandn	1200000	31.672 s	31.672 s	
sample_sigma2e	30	16.964 s	2.781 s	
gamrnd	340	14.202 s	0.979 s	
gammar	300310	13.223 s	3.513 s	
gammar1	300310	9.710 s	9.710 s	
randperm	300030	4.103 s	4.103 s	

Algorithm			Features		
Profiling Understanding	Optimizations Tests	1st Deployment	1st results	modifications convergence	2nd deployment

Development

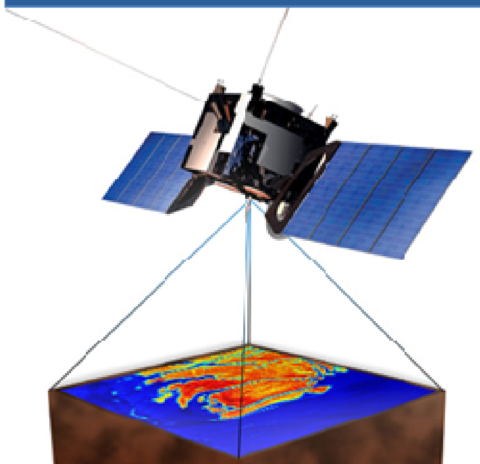
Optimizations

On the code:

- **Vectorizing loops**
- **Preallocating arrays for large data sets**
- **Study the influence of the data storage**
- **Rewrite some functions (MEX)**

On the machine :

- **Work on 64 bit processing**
- **Disable Java Environment on Matlab when it is not used**
- **Using multiprocessoring**
- **parallelize computing (on a grid)**



Originally planned: to use the GRID

With Matlab: need a lot of licenses
very expansive → Impossible

With MCR (Matlab **C**omponent **R**untime**):**
need a single toolbox
→ creation of a standalone

→ I wrote a documentation

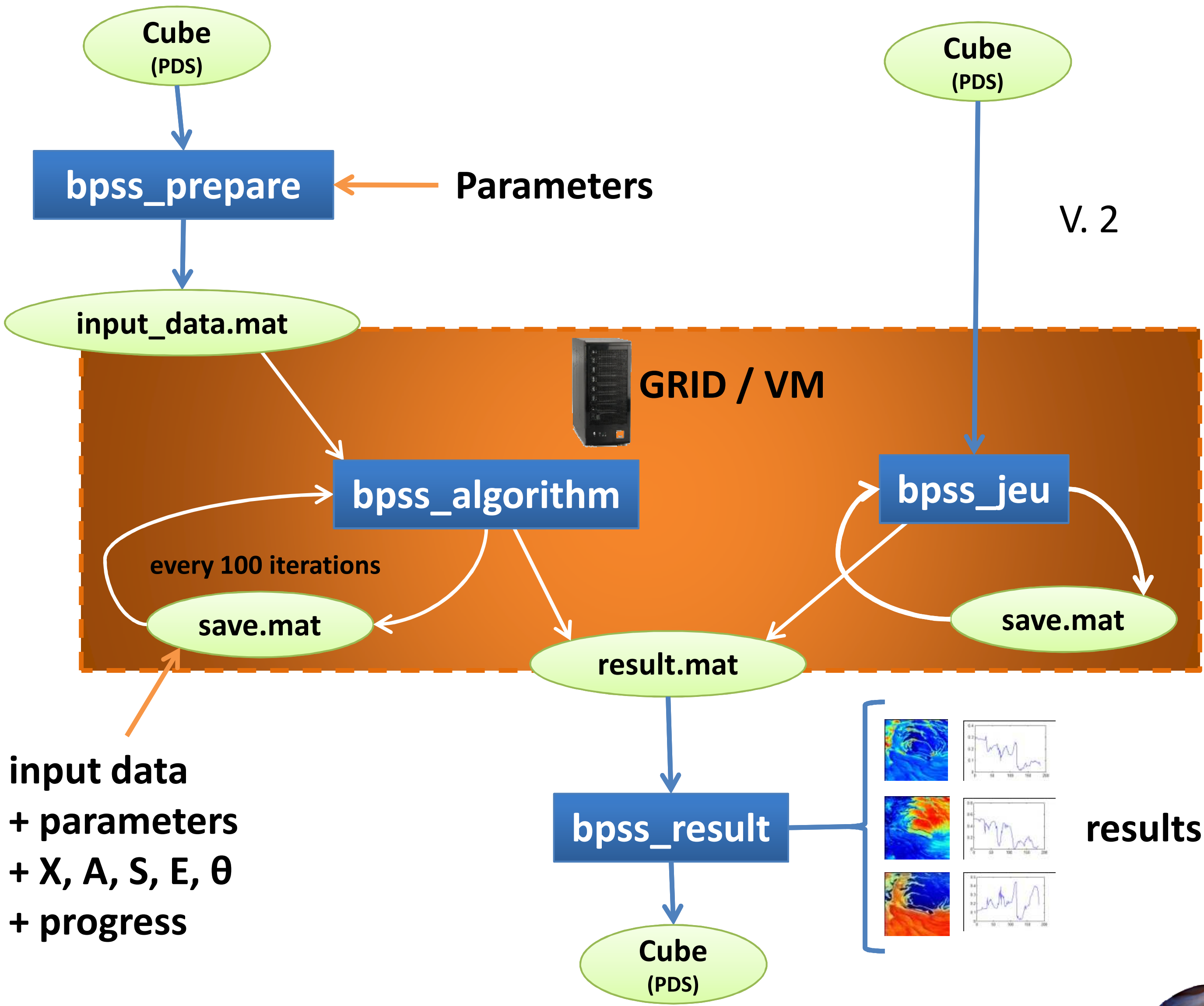
« Current » solution:

2 virtual machines

Mean computation time:
from a few hours → 1 day

Development

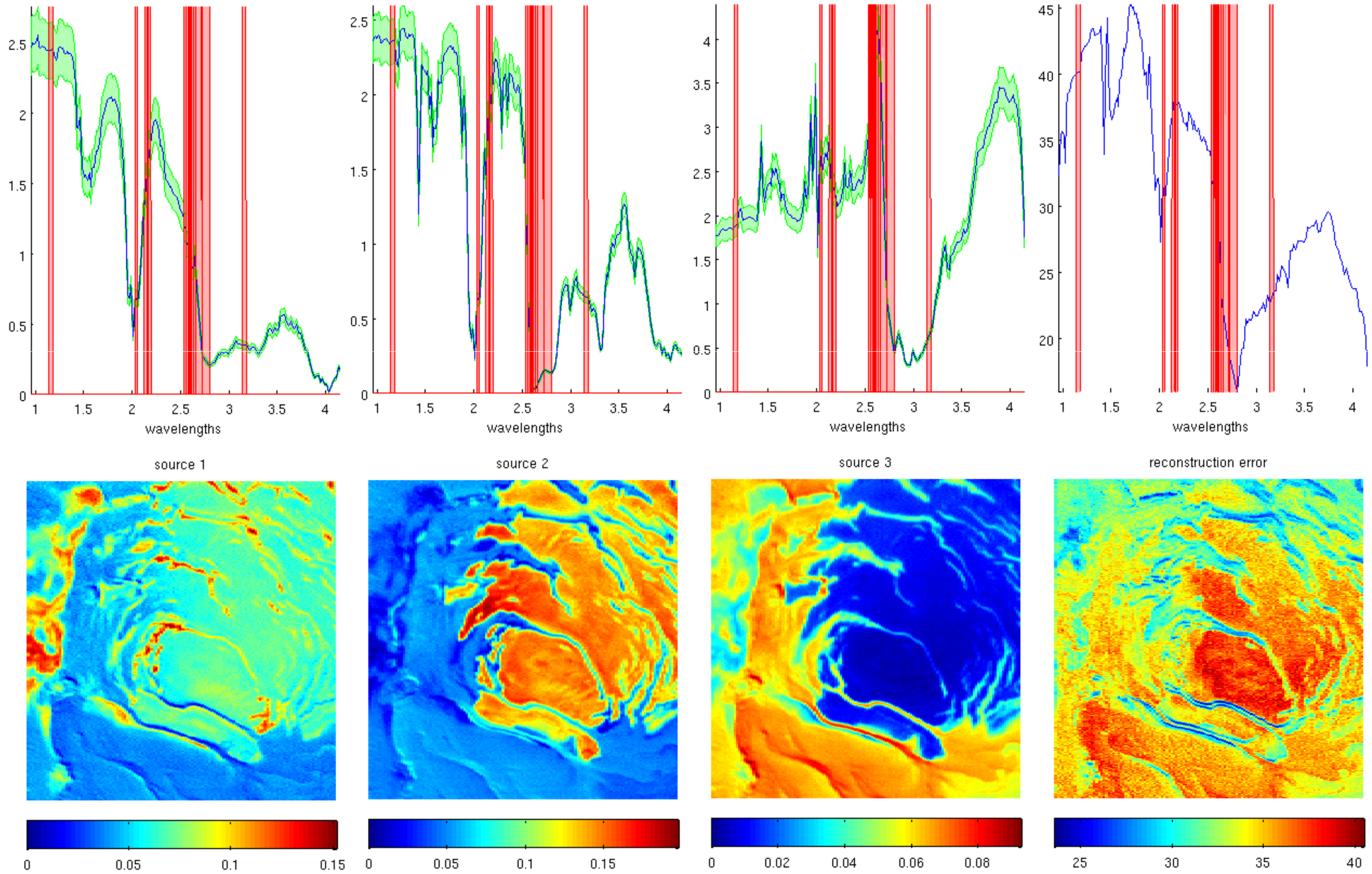
Deployment



Issues

Results

Image ORB0041_REF (300 1st lines), BPSS 3 sources



Spatial error :

$$s_N(n) = 10 \cdot log_{10}\left(\frac{\sum_{m=1}^{N_f} \mathbf{I}_{\lambda_m}(n)^2}{\sum_{m=1}^{N_f}\left(\mathbf{I}_{\lambda_m}(n) - \sum_{p=1}^{N} a_{\lambda_m,p} II_p(n)\right)^2}\right)$$

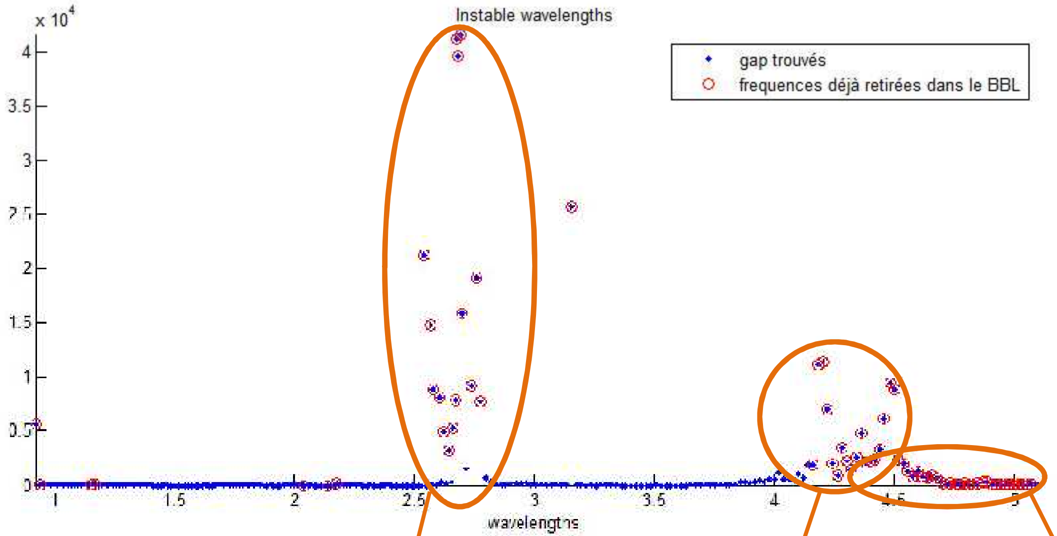
Issues

Data – Bad spectra

BPSS : no spatial or spectral correlation
→ the algorithm is sensitive to the noise

vs

From a statistic point of view we should keep as wavelengths as possible



The most noisy wavelengths

Overlapping of C and L channels

Aatmosphere absorption

Thermical emmissions

? Is it possible to order differently this issue?

Issues

Data – Selection of pixels

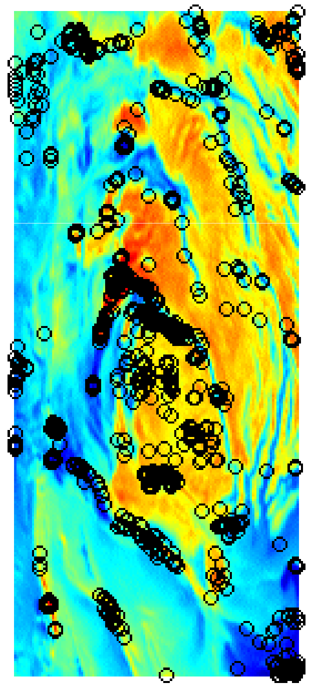
Originally made to solve the computing time issue:
→ How to make the application being more robust?

Differents methods :

- **simple JADE**
- **« convex » method**
- **the most differents**

Default method:
« convex » method

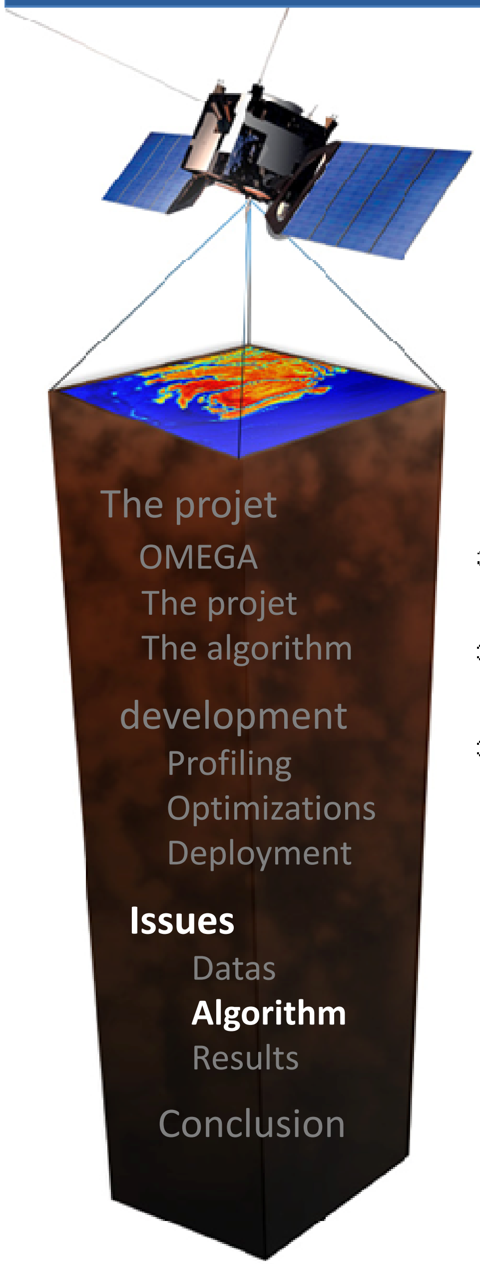
selection of 699 pixels



method of selection : using convex envelope

Issues

Convergence

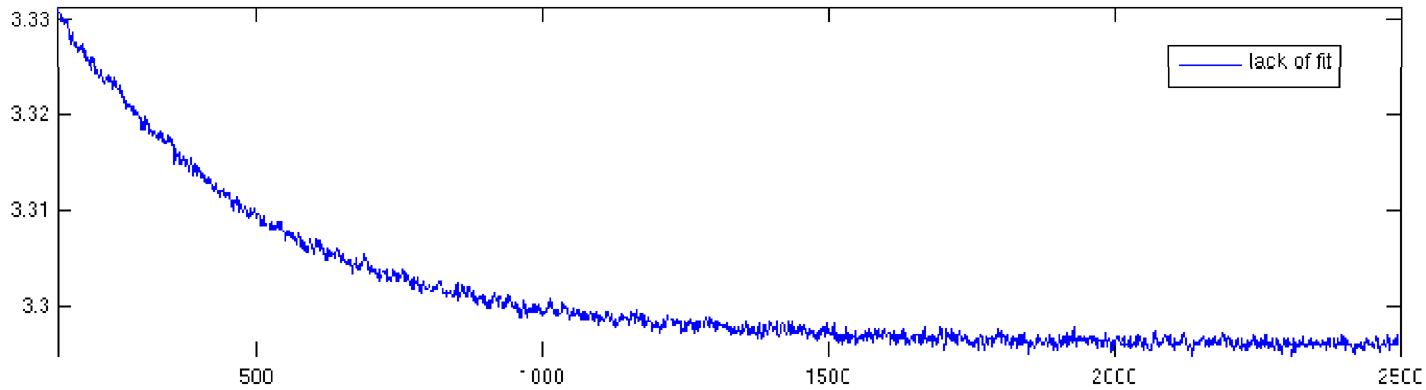


Remind: X = A.S + E

? We want the stationnarity, how to define it?

Choice : stationnarity of « lack of fit » :

$$lf = \sum_{all\ the\ coefficients} E = \sum_{all\ the\ coefficients} (X - A \times S)$$



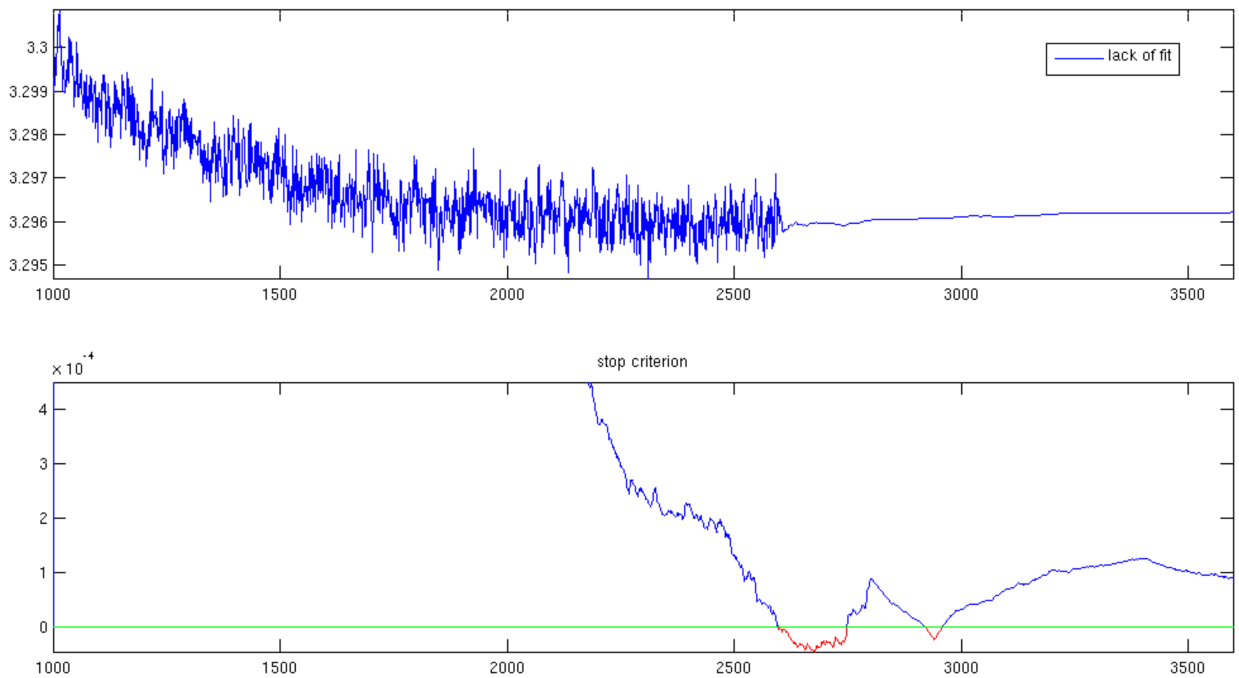
Remind: X = A.S + E

? We want the stationnarity, how to define it?

Choice : stationnarity of « lack of fit » :

$$lf = \sum_{\text{all the coefficients}} E = \sum_{\text{all the coefficients}} (X - A \times S)$$

Stop criterion :



Issues

Convergence

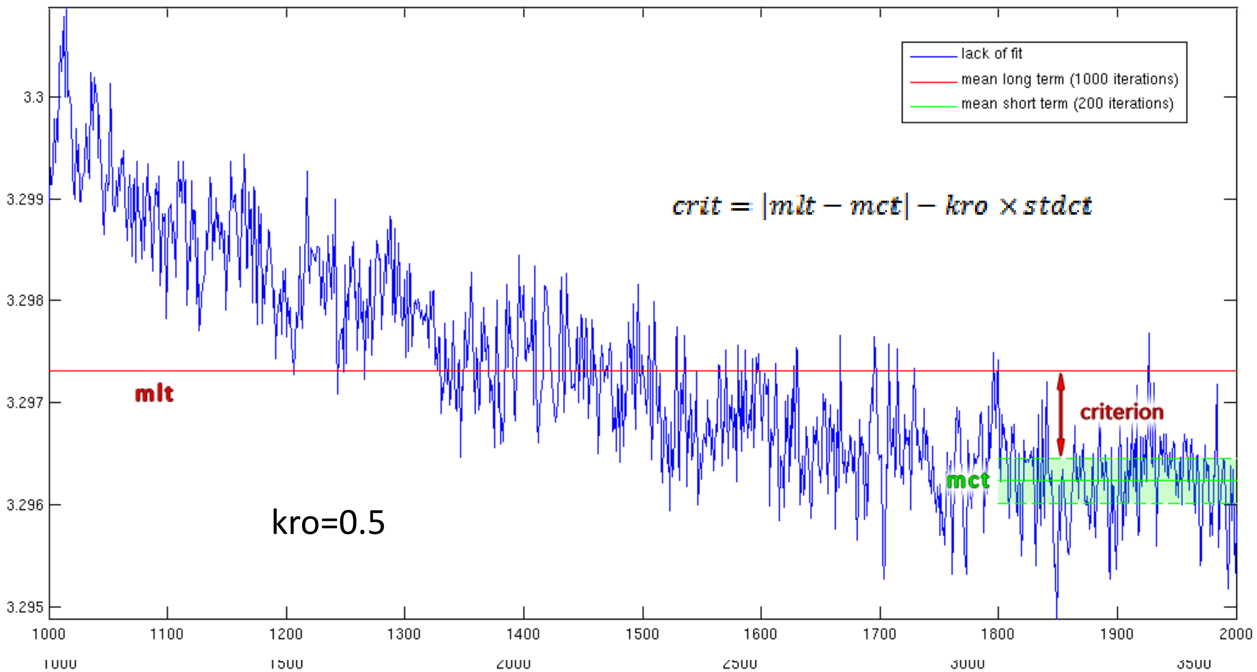
Remind: X = A.S + E

? We want the stationnarity, how to define it?

Choice : stationnarity of « lack of fit » :

$$lf = \sum_{all\ the\ coefficients} E = \sum_{all\ the\ coefficients} (X - A \times S)$$

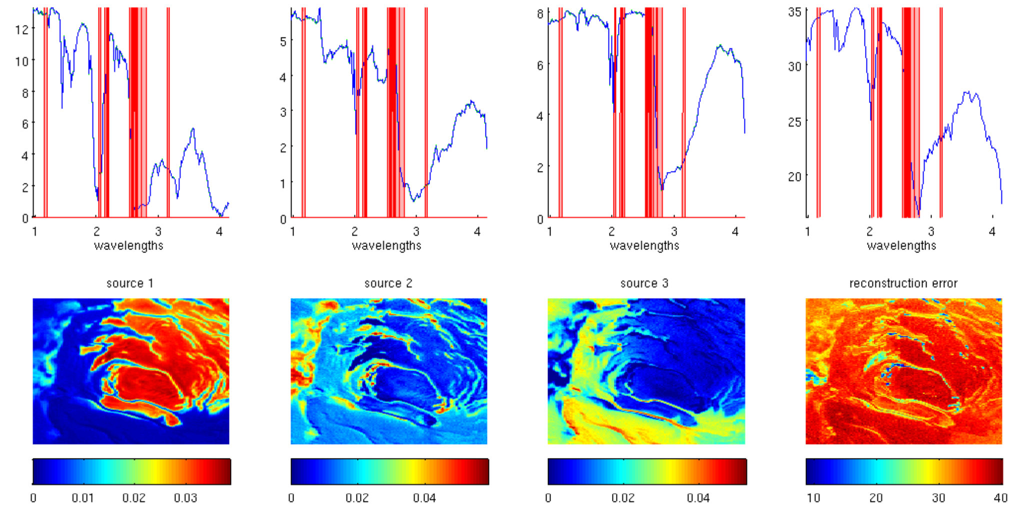
Stop criterion :



Issues

Number of sources

Image ORB0041
BPSS2 – 3 sources



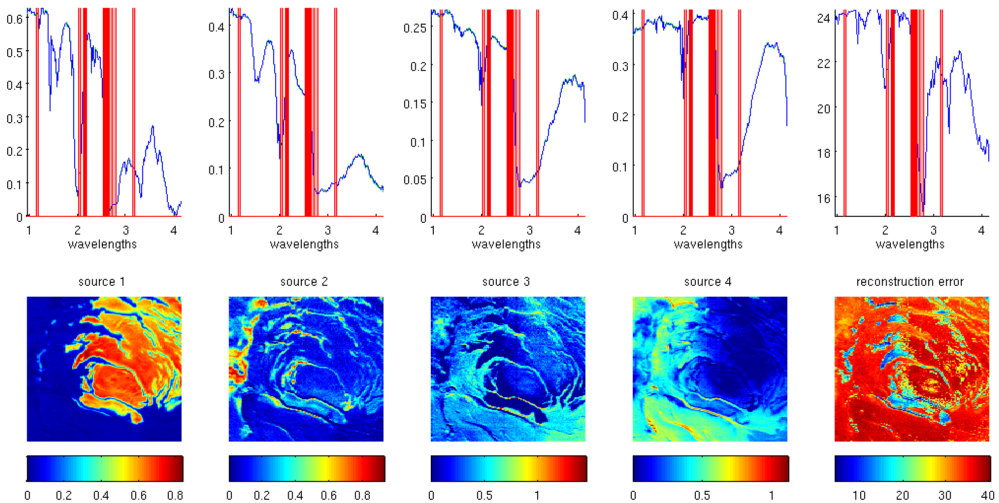
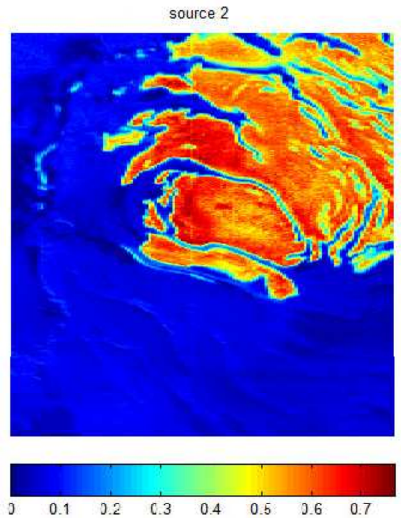
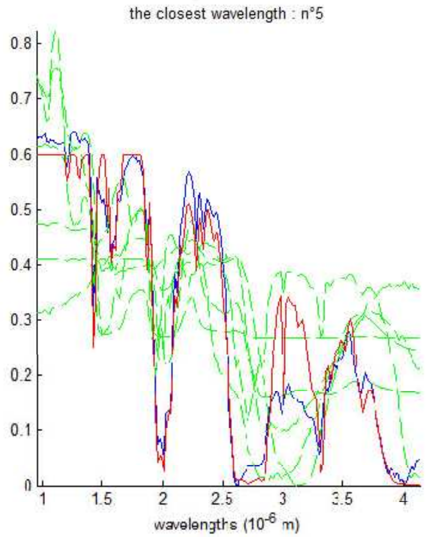


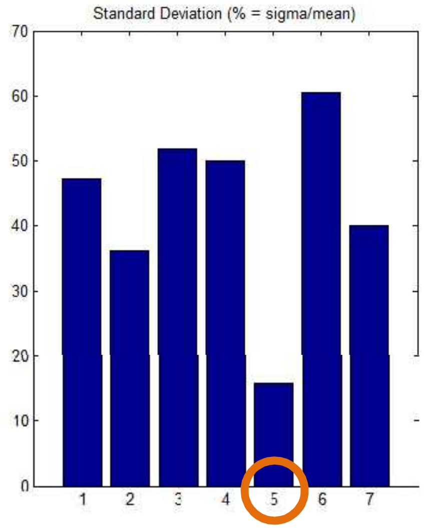
Image ORB0041
BPSS2 – 4 sources

SNR : $s = 10 \cdot log_{10}\left(\frac{\sum_{m=1}^{N_f}\sum_{n=1}^{N_x N_y} \mathbf{I}_{\lambda_m}(n)^2}{\sum_{m=1}^{N_f}\sum_{n=1}^{N_x N_y}\left(\mathbf{I}_{\lambda_m}(n) - \sum_{p=1}^{N} a_{\lambda_m,p} II_p(n)\right)^2}\right)$

Comparison with spectrum libraries (from laboratories)







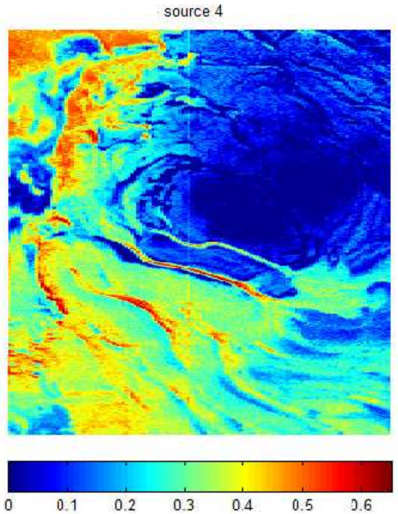
A few tests:

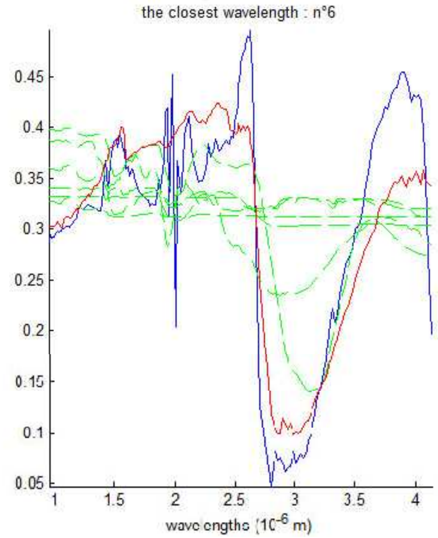
1. H_2O, grain 1
2. H_2O, grain 100
3. H_2O, grain 1000
4. CO_2,grain 100
5. CO_2,grain 10,000
6. BASALTE LPG – AP
7. GYPSUM – AP

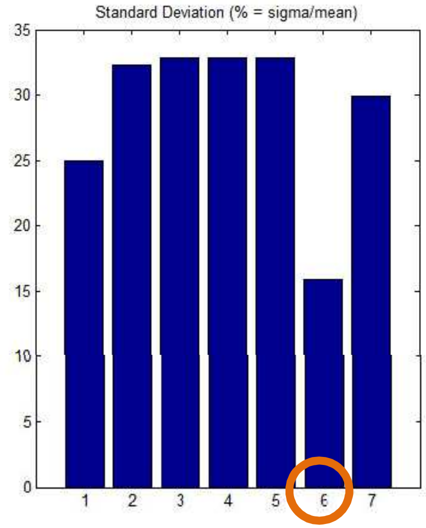
Issues

Result analysis

Comparison with spectrum libraries (from laboratories)



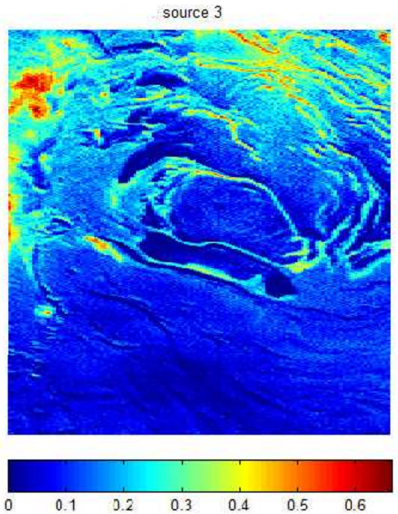


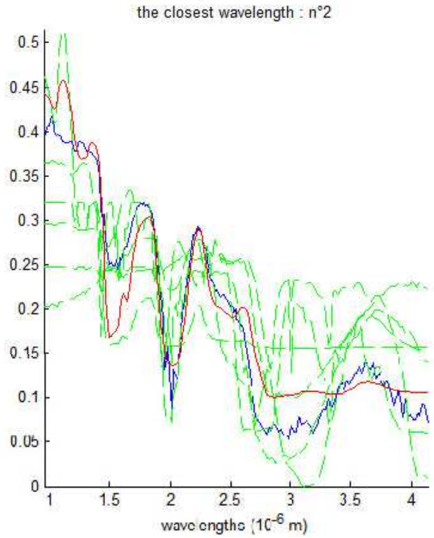


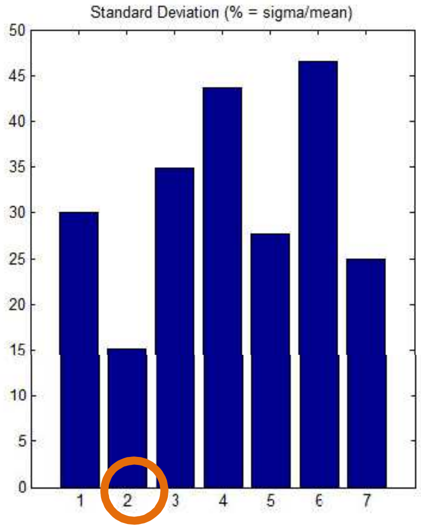
A few tests:

1. H_2O, grain 1
2. H_2O, grain 100
3. H_2O, grain 1000
4. CO_2,grain 100
5. CO_2,grain 10,000
6. BASALTE LPG – AP
7. GYPSUM – AP

Comparison with spectrum libraries (from laboratories)



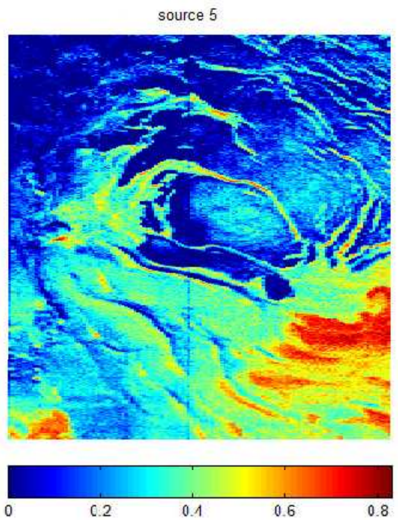


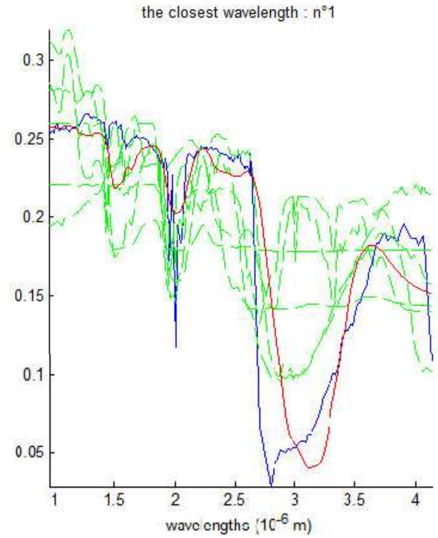


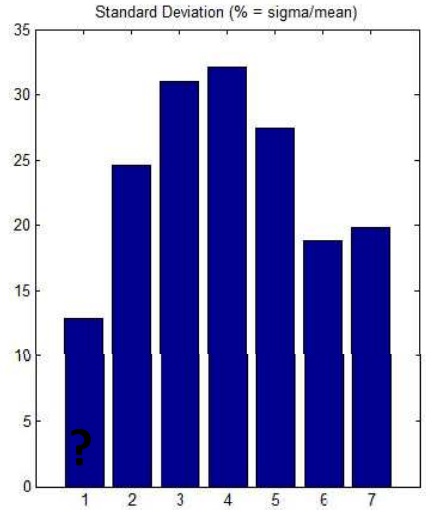
A few tests:

1. H_2O, grain 1
2. H_2O, grain 100
3. H_2O, grain 1000
4. CO_2,grain 100
5. CO_2,grain 10,000
6. BASALTE LPG – AP
7. GYPSUM – AP

Comparison with spectrum libraries (from laboratories)







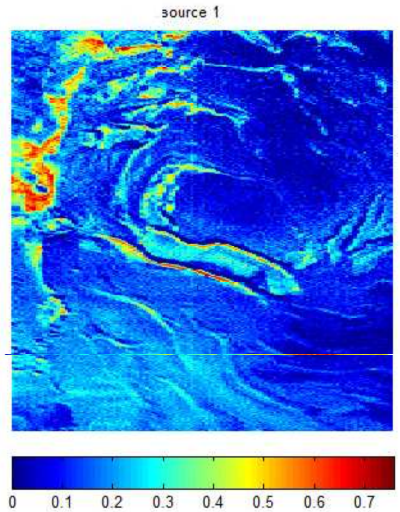
A few tests:

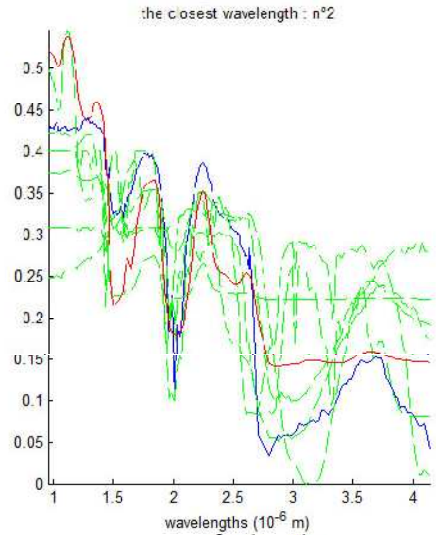
? 1. H_2O, grain 1
2. H_2O, grain 100
3. H_2O, grain 1000
4. CO_2,grain 100
5. CO_2,grain 10,000
6. BASALTE LPG – AP
7. GYPSUM – AP

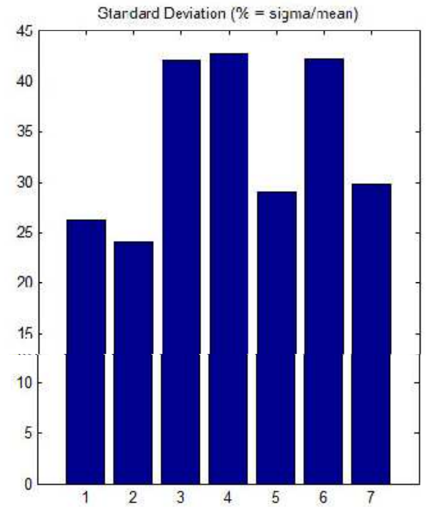
Issues

Result analysis

Comparison with spectrum libraries (from laboratories)







A few tests:

1. H_2O, grain 1
2. H_2O, grain 100
3. H_2O, grain 1000
4. CO_2,grain 100
5. CO_2,grain 10,000
6. BASALTE LPG – AP
7. GYPSUM – AP

?

?

→ **Many doubts**

Next steps:

- identification of linear mixings
- increase library size

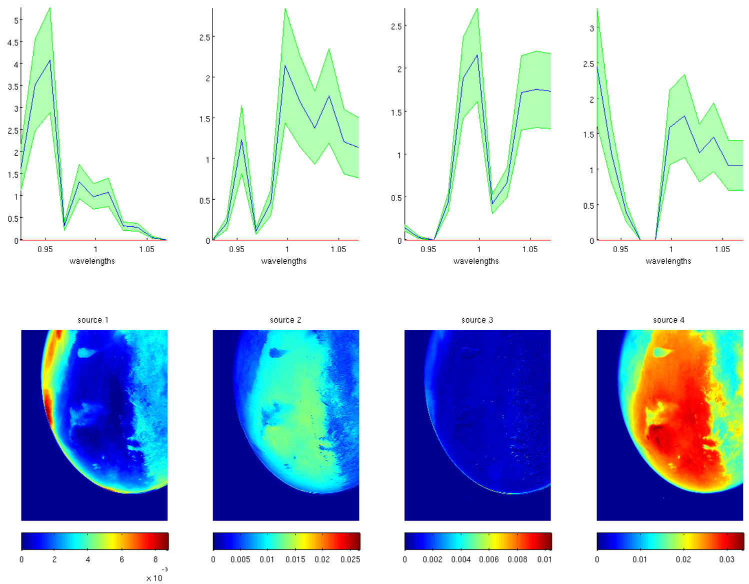
Conclusion

→ The toolbox can be used in an automatic mode (parameters can be changed)

→ Adaptable to any otherdataset

→ Articles has been submitted in WHISPERS

→ Many teams were interested in testing it
- Osiris
- Merid
- Virtis



Results on OSIRIS
4 sources

Conclusion

→ The toolbox can be used in an automatic mode (parameters can be changed)

→ Adaptable to any otherdataset

→ Articles has been submitted in WHISPERS

→ Many teams were interested in testing it
- Osiris
- Merid
- Virtis

Questions ?