Parallel Nonnegative Matrix Factorization Algorithms for Hyperspectral Images

A Masters Thesis

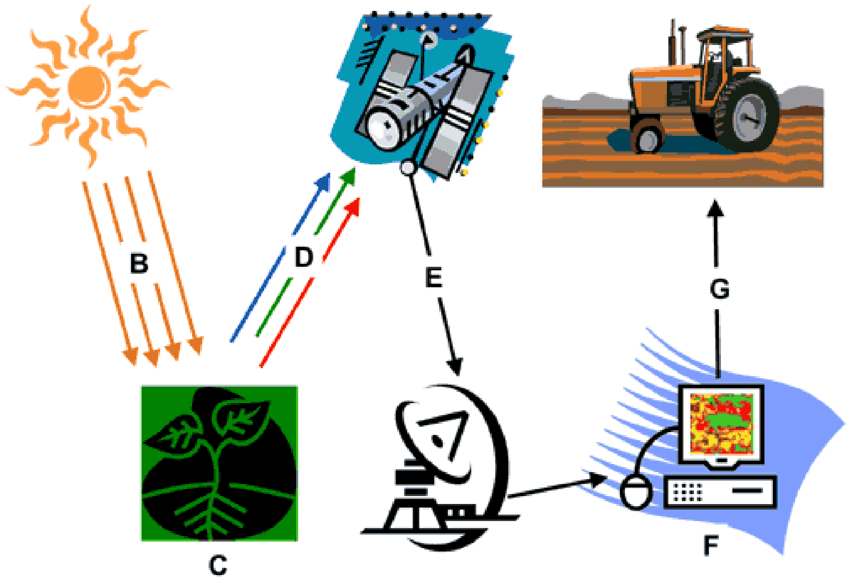
Lukasz Grzegorz Maciak

Montclair State University Department of Computer Science

Objectives

1. **Background**
2. Investigate Novel Feature Extraction methods (NMF)
3. Design, implement and test parallel NMF algorithms
4. Initiate development of Java based hyperspectral imaging toolkit

What is Remote Sensing?



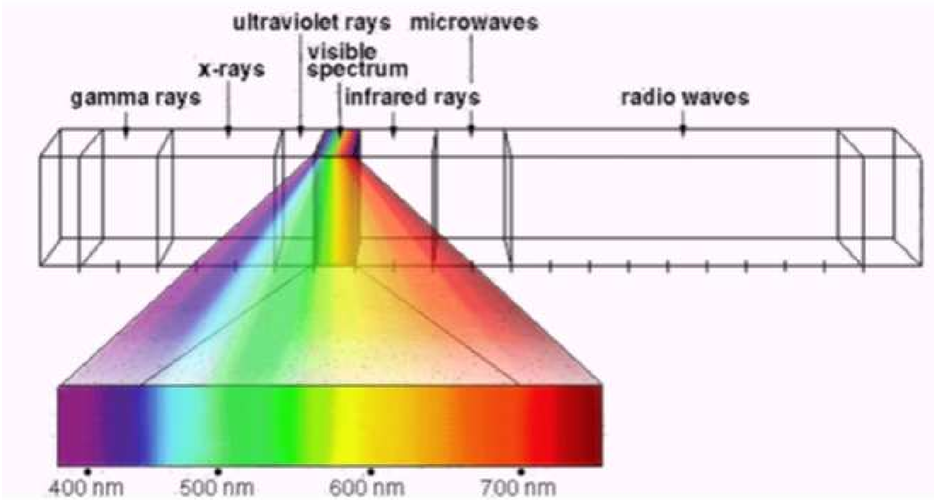
The study of Earth's surface and atmospheric features from a distance.

● direct observation ● reconnaissance ● aerial and/or satellite photography/video ●

Applications in:

● agriculture and forestry ● meteorology ● military surveilance ●

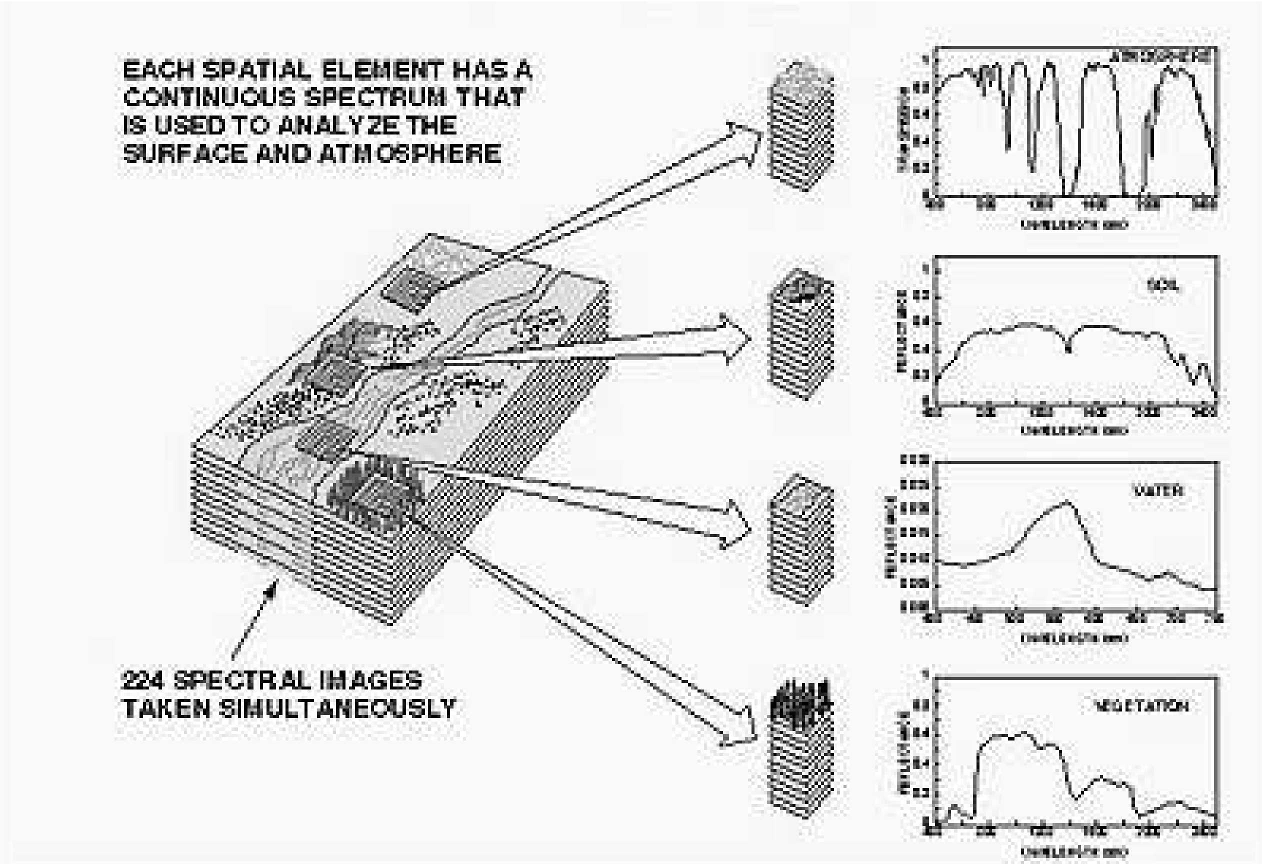
Hyperspectral Images



Visible Light: $0.4\mu m$ to $0.7\mu m$

Hyperspectral Lens: $0.4\mu m$ and $2.4\mu m$

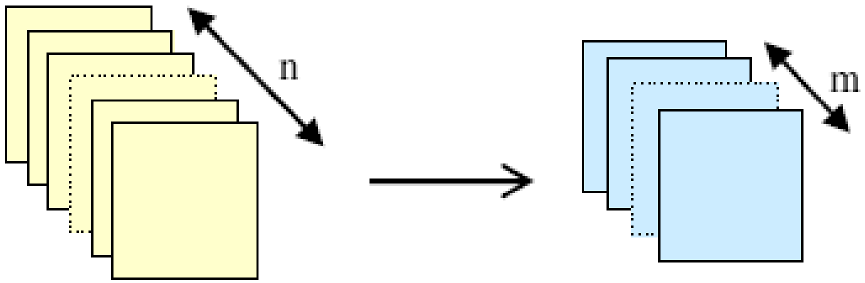
Hyperspectral Images



● spectra ● mixed pixels ●

Feature Extraction

Reduce dimensionality of the data sample without data loss



Source Separation

Spectral Unmixing

Linear Mixing Model

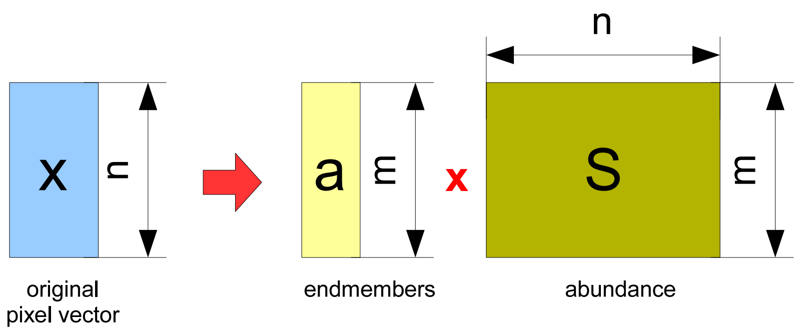
Each pixel p is composed of d endmembers

Each endmember has a natural intensity e

Each endbember contributes fractional amount a to the total pixel intensity

$$p = \begin{bmatrix} e_1 & e_2 & ... & e_d \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_d \end{bmatrix} \quad (1)$$

Linear Mixing Model



$$X = \sum_{i=1}^{m} a_i s_i + w = Sa + w \qquad (2)$$

$$where \qquad a_i \geq 0, i = 1, ..., m \qquad and \qquad \sum_{i=1}^{m} a_i = 1 \qquad (3)$$

Unsupervised Feature Extraction

No training data available

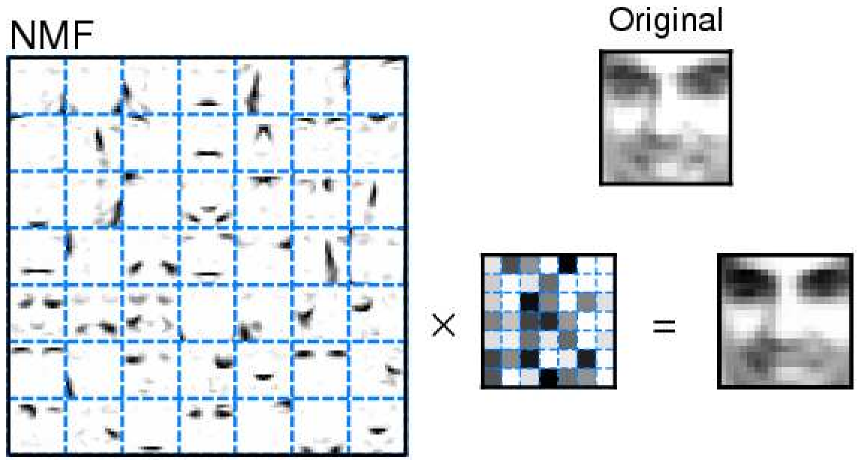
Extraction must be performed algorithmically

Nonnegative Matrix Factorization

Objectives

1. Background
2. **Investigate Novel Feature Extraction methods (NMF)**
3. Design, implement and test parallel NMF algorithms
4. Initiate development of Java based hyperspectral imaging toolkit

Nonnegative Matrix Factorization



Relatively Fast

Straightforward Algorithm

Good Source Separation

The NMF Problem

Given a nonnegative matrices

$$Y \in \mathbb{R}^{mxn}, W \in \mathbb{R}^{mxk}, H \in \mathbb{R}^{kxn}$$

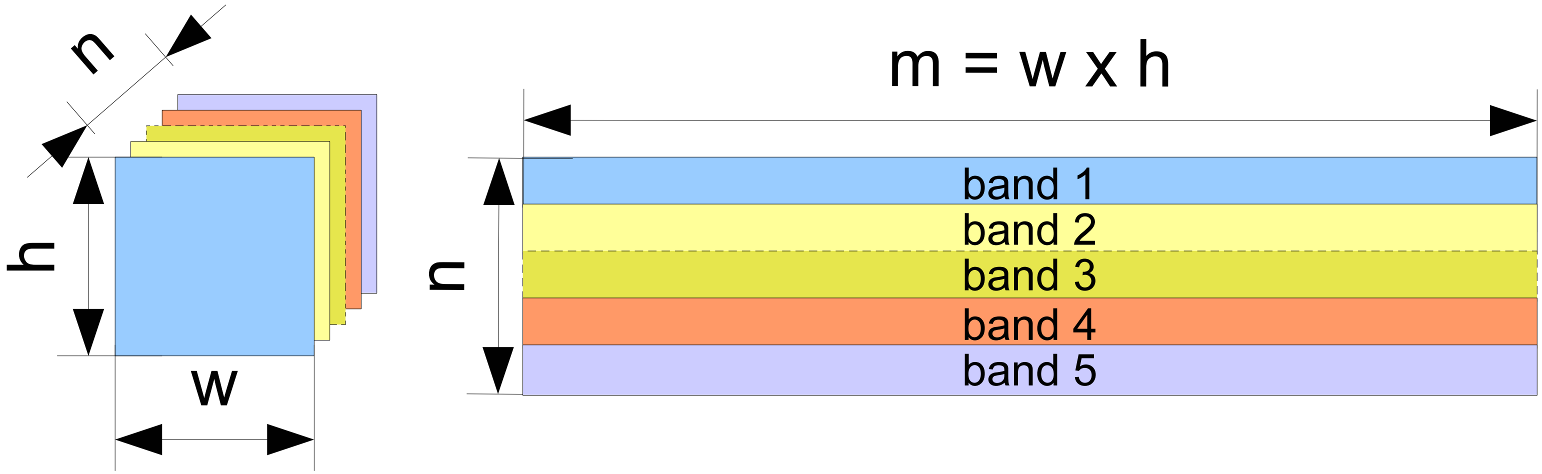
and a positive integer

$$k \leq min\{m, n\}$$

find W and H which minimize the function:

$$f(W, H) := \frac{1}{2} \|Y - WH\|_F^2 \tag{4}$$

Applying NMF to Hyperspectral Data



a) original 3D image cube

b) 2D representation used in NMF

Unrolling the hyperspectral image cube into an array of bandvectors

How does NMF work?

Start with random W and H and repeatedly update them using following rules:

$$W = W \frac{(YH^T)}{(WHH^T) + \epsilon} \quad (5)$$

$$H = H \frac{(W^TY)}{(W^tHWH) + \epsilon} \quad (6)$$

where ϵ is a very small positive quantity ($\epsilon \leq 10^{-9}$)

NMF Algorithm

1. Given

$$Y \in \mathbb{R}^{mxn} \geq 0, k > 0 \quad and \quad k \ll min(m, n) \tag{7}$$

randomly initialize matrices $W \in \mathbb{R}^{mxk}$ and $H \in \mathbb{R}^{kxn}$ with nonnegative values

2. Scale the columns of W to sum up to one

3. Create temporary variables $\bar{W}$ and $\bar{H}$. Sent their contents to be equal to H and W respectively.

4. Repeatedly apply the following steps until convergence criteria are met:

   (a) Update $\bar{W}$ and $\bar{H}$ by using:

$$\bar{H}_{ij} \Leftarrow H_{cj} \frac{(W^T Y)_{cj}}{(W^T W H)_{cj} + \epsilon} \qquad (1 \leq c \leq k) \quad (1 \leq j \leq n) \tag{8}$$

$$\bar{W}_{ic} \Leftarrow W_{ic} \frac{(Y H^T)_{ic}}{(W H H^T)_{ic} + \epsilon} \qquad 1 \leq i \leq m) \quad (1 \leq c \leq k) \tag{9}$$

   (b) Set $W = \bar{W}$ and $H = \bar{H}$

   (c) Scale the columns of W to sum up to one

Projected Gradient NMF (PG-NMF)

Alternatively fix one matrix and update another.

or

find W_{k+1} such that $f(W_{k+1}, H_k) \leq f(W_k, H_k)$

and

find H^{k+1} such that $f(W^{k+1}, H^{k+1}) \leq f(W^{k+1}, H^k)$

PG-NMF

Update H and W using following rules:

$$H_{k+1} = H_k - \alpha W_k^T (W_k H_k - Y) \tag{10}$$

$$W_{k+1} = W_k^T - \alpha H_{k+1} (H_{k+1}^T W_k^T - Y^T) \tag{11}$$

PG-NMF: Finding α

Substitute H or W for X as necessary:

$$f(X_{k+1}) - f(X_k) \leq \alpha \|\nabla f(X_k)^T (X_{k+1} - X_k)\| \tag{12}$$

where $\nabla f(H) = W^T(WH - Y)$ and $\nabla f(W) = H(H^T W^T - Y^T)$

PG-NMF: α Updates

if α_k satisfies 12 then repeatedly increase α

$$\alpha_k \leftarrow \alpha_k / \beta$$

as long as it still satisfies 12.

else repeatedly decrease α

$$\alpha_k \leftarrow \alpha_k * \beta$$

until it satisfies 12.

in our case we used $\beta = 0.1$

PPG-NMF: α Updates rewritten by Chih-Jen Lin

$$(1-\sigma)\langle W_k^T(W_kH_k-Y), H_{k+1}-H_k\rangle + \frac{1}{2}\langle H_{k+1}-H_k, (W_k^TW_k)(H_{k+1}-H_k)\rangle \leq 0 \quad (13)$$

$$(1-\sigma)\langle H_{k+1}(H_{k+1}^TW_k^T-Y^T), W_{k+1}^T-W_k^T\rangle + \frac{1}{2}\langle W_{k+1}-W_k, (H_{k+1}H_{k+1}^T)(W_{k+1}^T-W_k^T)\rangle \leq 0 \quad (14)$$

where $\langle , \rangle$ denotes the sum of component wise products of two matrices

PG-NMF Algorithm

1. Given

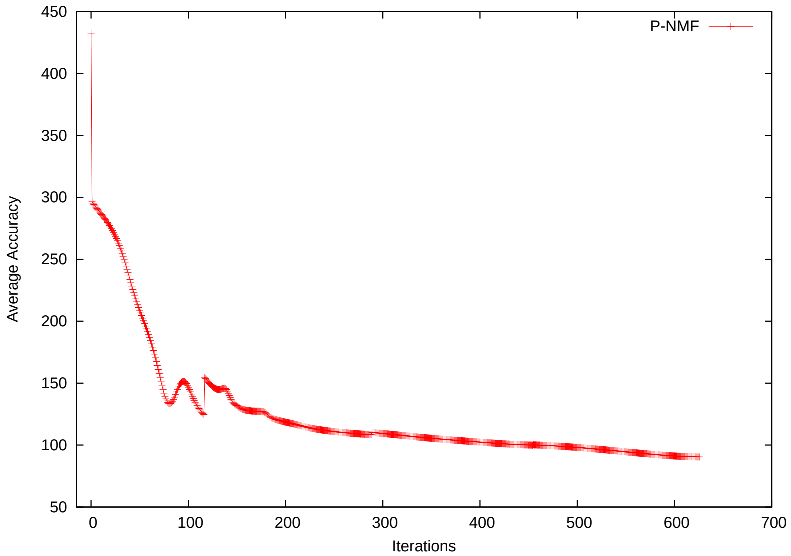
$$Y \in \mathbb{R}^{mxn} \geq 0, k > 0, k \ll min(m, n), \alpha = 1, \beta = 0.1, \sigma = 0.01 \quad (15)$$

   randomly initialize matrices $W \in \mathbb{R}^{mxk}$ and $H \in \mathbb{R}^{kxn}$ with nonnegative values.

2. Find H_{k+1} using Equation 10
3. Evaluate Equation 10 and:
   (a) if Equation 13 is satisfied then:
      i. if at the last iteration Equation 13 was not satisfied set $H_{k+1} \leftarrow \bar{H}_{k+1}$ and goto 4
      ii. else save the value of H_{k+1} in a temporary buffer $\bar{H}_{k+1}$
      iii. save the outcome of Equation 13
      iv. update $\alpha \leftarrow \alpha / \beta$
      v. go back to 2.

   (b) if Equation 13 is not satisfied then:
      i. if at the last iteration Equation 13 was satisfied set $H_{k+1} \leftarrow \bar{H}_{k+1}$ and goto 4
      ii. else save the value of H_{k+1} in a temporary buffer $\bar{H}_{k+1}$
      iii. save the outcome of Equation 13
      iv. update $\alpha \leftarrow \alpha * \beta$
      v. go back to 2.
4. Find W_{k+1} using Equation 11
5. Evaluate Equation 10 and perform steps analogous to 3a and 3b for W.
6. Set $H = H_{k+1}$ and $W = W_{k+1}$
7. Go back to 2 until convergence criteria are met.

Convergence Criteria



1. Change in $f(W, H)$
2. Desired Value of $f(W, H)$
3. Number of Iterations
4. Execution Time

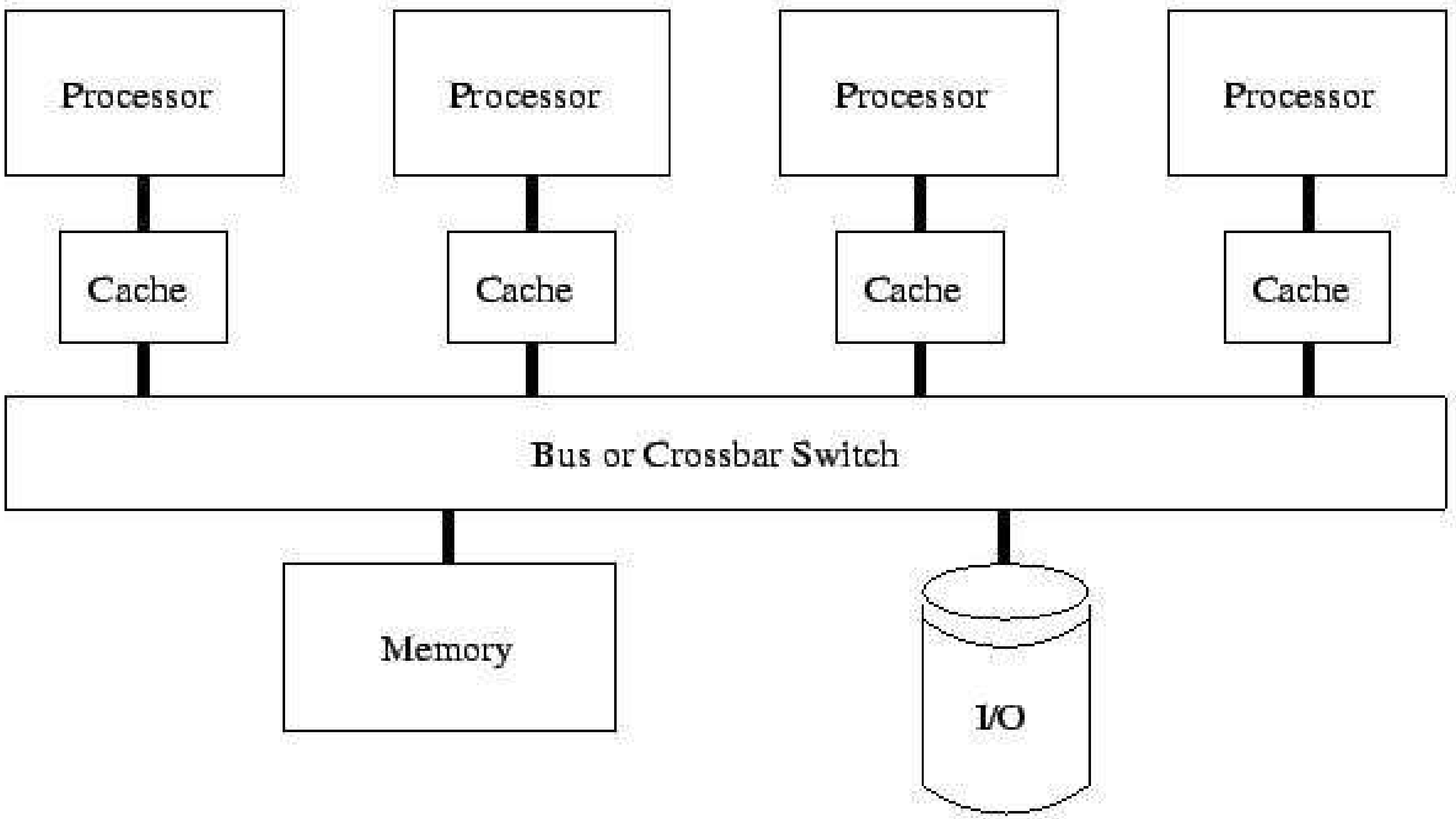
Objectives

1. Background
2. Investigate Novel Feature Extraction methods (NMF)
3. **Design, implement and test parallel NMF algorithms**
4. Initiate development of Java based hyperspectral imaging toolkit

Need For Parallelization

1. Large Image Resolution
2. Many Bands
3. Large Image Size
4. Computational Complexity

Shared Memory: SMP

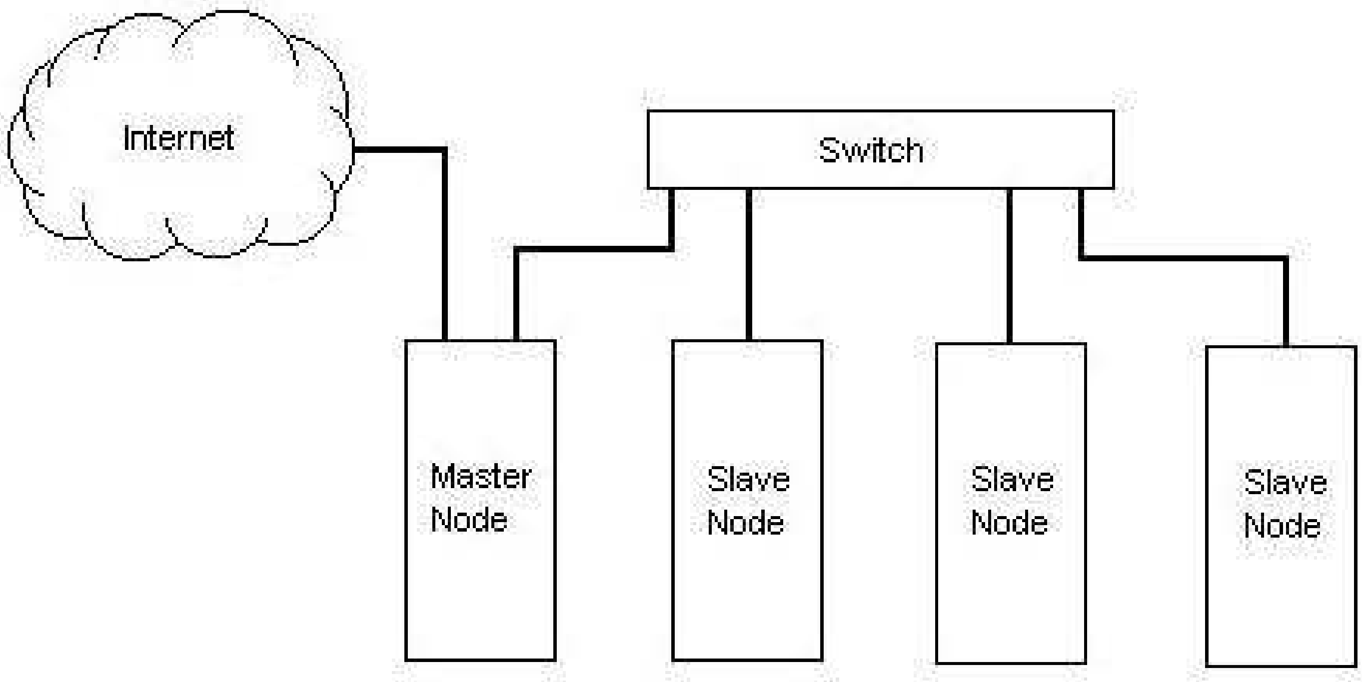


SMP - Symmetric Multiprocessor

SMP Features

1. CPU Scheduling transparent to the user
2. Java Threads leverage SMP architecture
3. No communication overhead
4. Not very scalable

Distributed Memory



Distributed Memory Features

1. Communication between nodes is slow

2. Data must be distributed manually

3. Unlimited scalability

4. Require 3rd party libraries

There also exist hybrid systems - clusters of SMP's

Data Partitioning

1. Spectral Data Partitioning

2. Spatial Data Partitioning

3. Other Approaches

Speedup

$$S_p = \frac{T_p}{T_s} \tag{16}$$

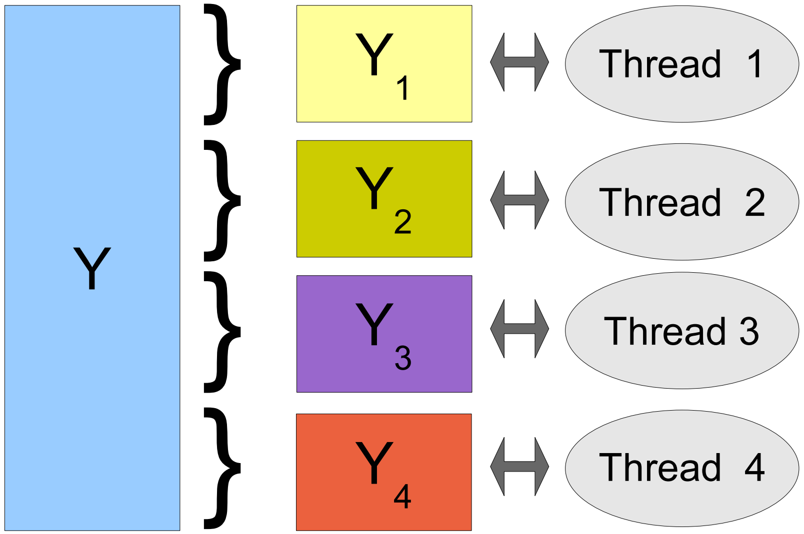
T_s time of sequential execution

T_p time of parallel execution with p processors.

An ideal speedup is linear:

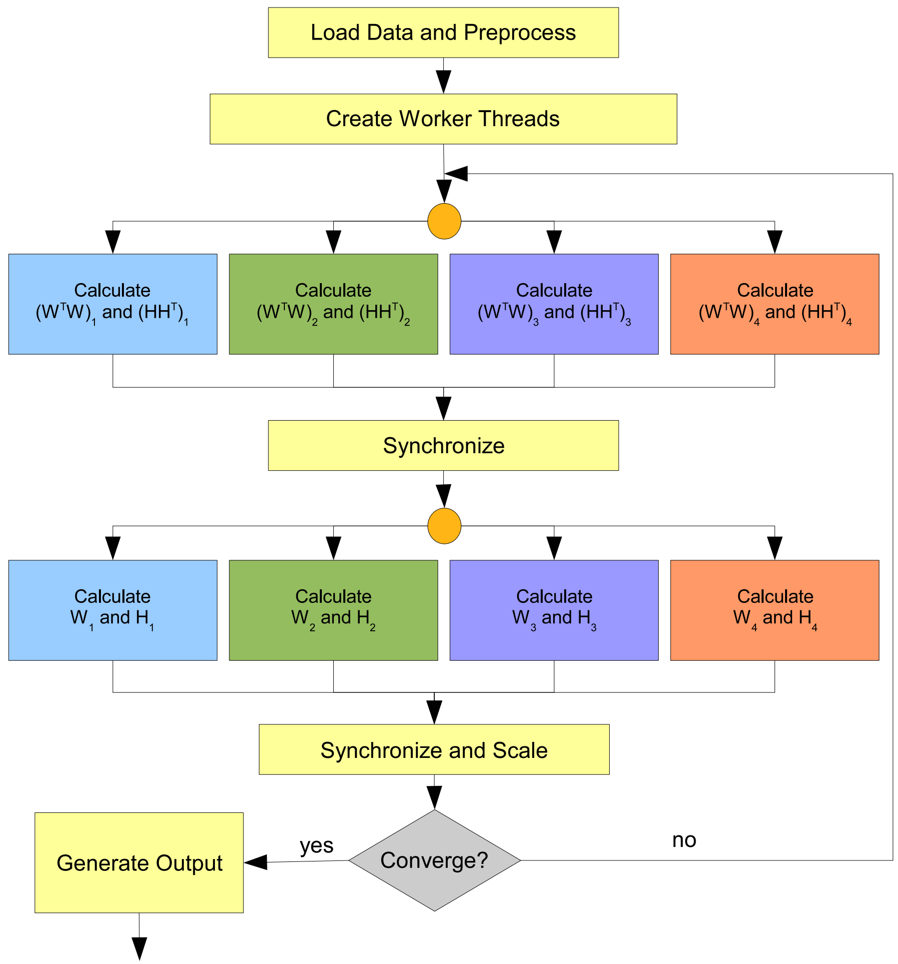
$$S_p = p$$

Parallel NMF (P-NMF)



using spectral data partitioning

P-NMF Parallel Execution



Data Distribution

```
int start, end;

NMFThread[] tmp = new NMFThread[number_of_threads];

for(int i=0; i<number_of_threads; i++)
{
        start = i*(k/number_of_threads);
        end = (i+1)*(k/number_of_threads);

        if(tt == (number_of_threads -1))
        end = k;

        tmp[tt] = new NMFThread(start, end, times);
        tmp[tt].start();
}
```

Parallel Projected Gradient NMF (PPG-NMF)

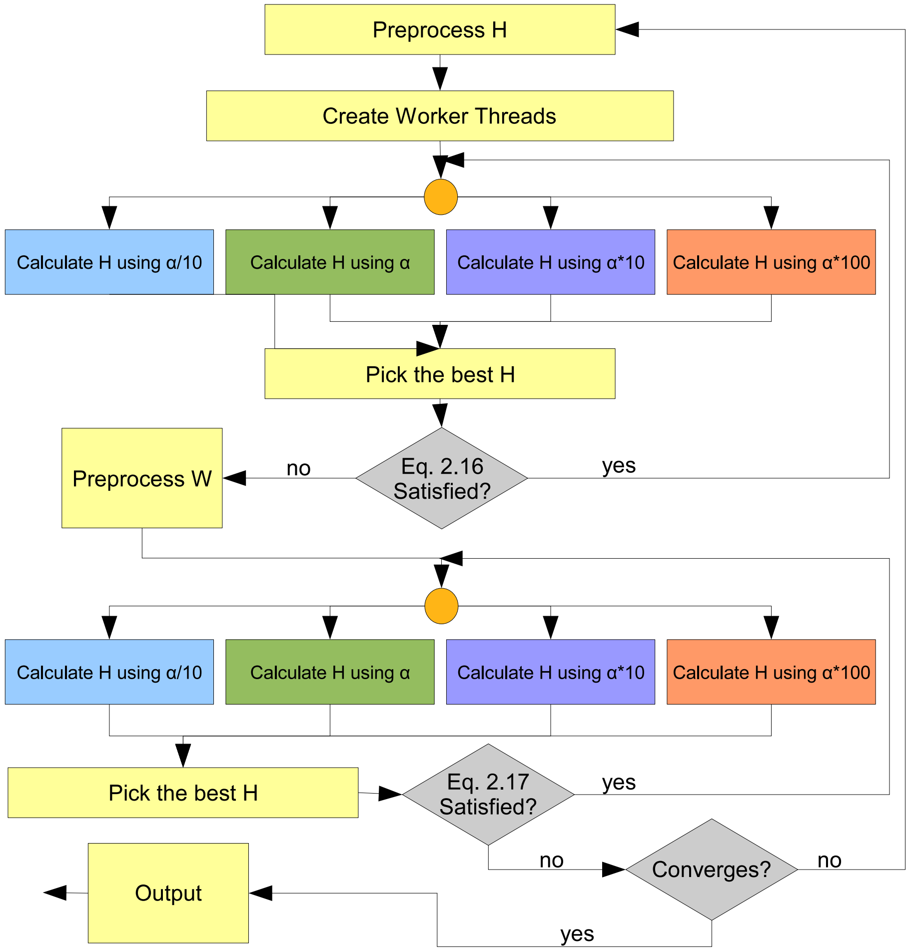
Spatial and Spectral Partitioning Strategies

NOT APPROPRIATE

instead

each CPU evaluates one α value

PPG-NMF Execution



PPG-NMF Distribution

1. If we do not have a previous sum (ie. this is the first run):

   Initialize a single thread with $\alpha = 1$ and evaluate it

2. if we do have a previous sum S then:

   Let i be the number of threads

   such that $i \in \{1, 2, ..., n\}$

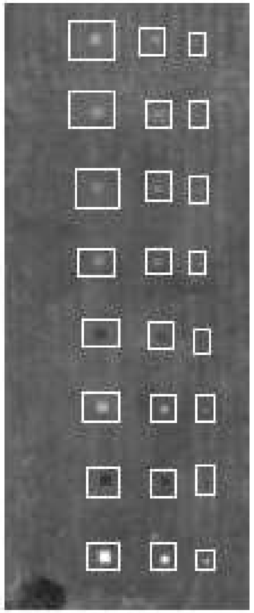
   where n is the total number of threads;

   Let e be an integer such that

   $e = 1$ if $S <= 0$ or $e = -1$ if $S > 0$.

   Initialize all the threads with $\alpha = \beta^{ie}$

Experimental Data



(a)

(b)

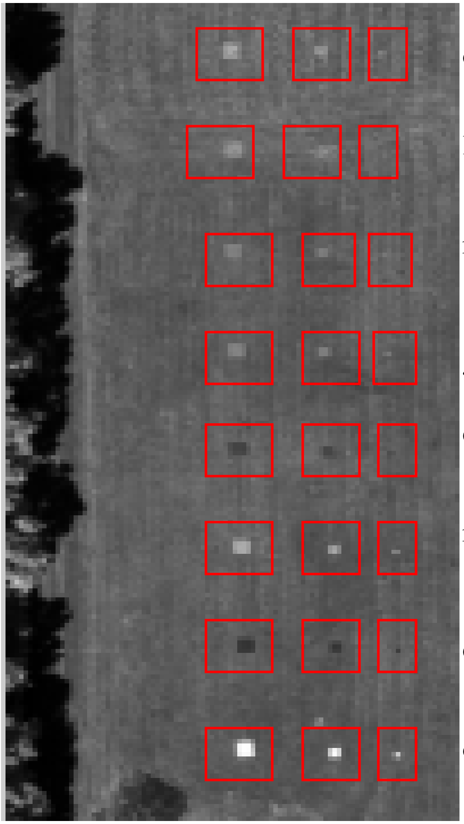
a) Hyperspectral Digital Imagery Collection Experiment (HYDICE)

b) Photo taken using SOC 700 hyperspectral sensor

Experimental Data

1. HYDICE
   (a) 85x185 pixels and 40 bands
2. SOC 700
   (a) 160x160 pixels and 120
   (b) only 40 bands used

HYDICE Data Set



dark olive parachute (nylon)

light olive parachute (nylon)

nomex kevlar (woodland)

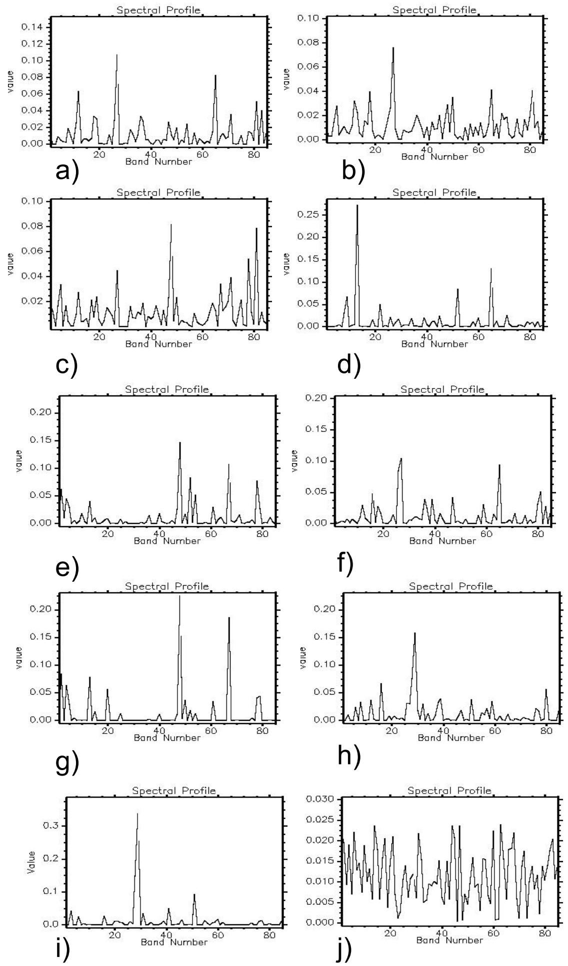
green tenting

cotton (green woodland)

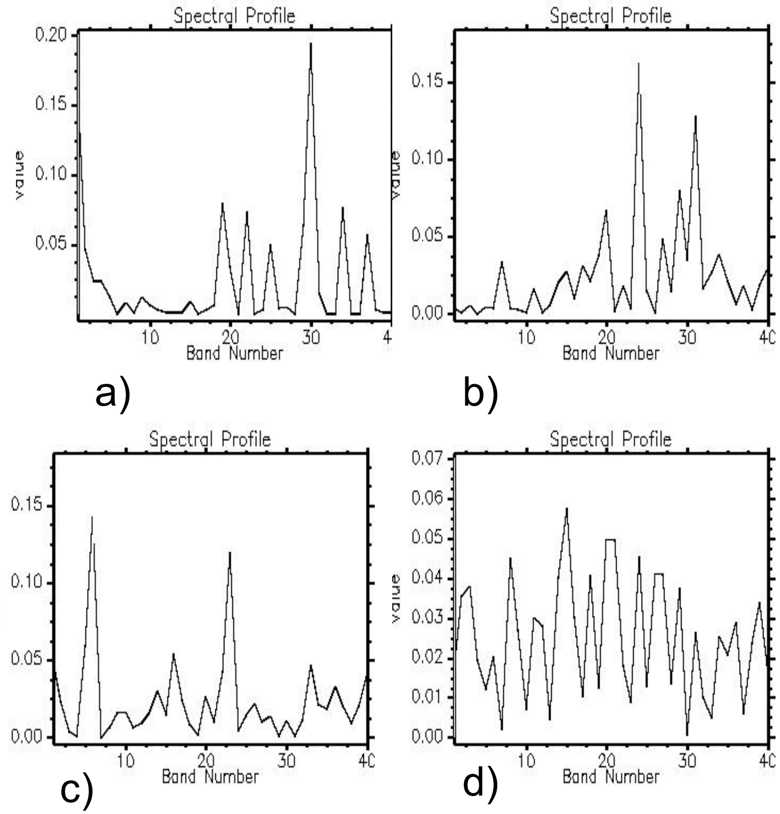
nylon (green woodland)

cotton (green)

desert BDU (nylon)



SOC 700 Data Set



Testing Platform

1. SunFire v880
2. 4 UltraSparc9 750MHz CPU's
3. 8 GB of RAM
4. Solaris 8
5. Publicly Accessible Server

Test Procedure

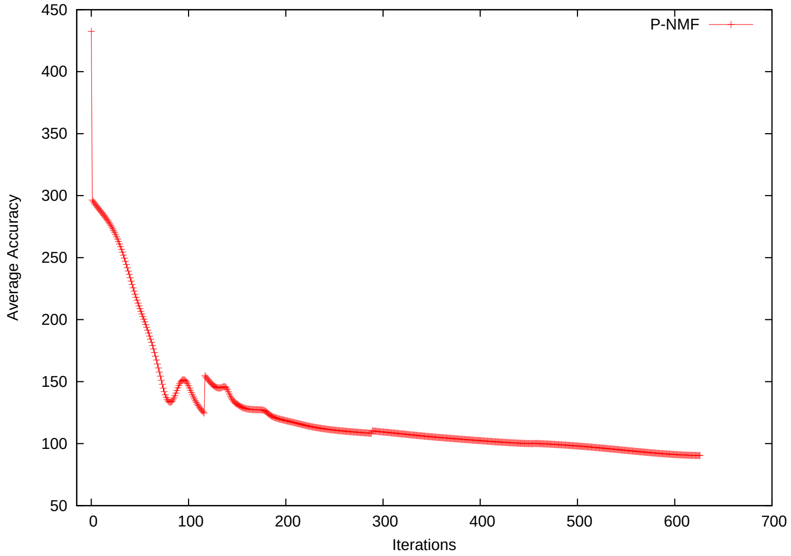
5 P-NMF and 5 PPG-NMF tests per sample

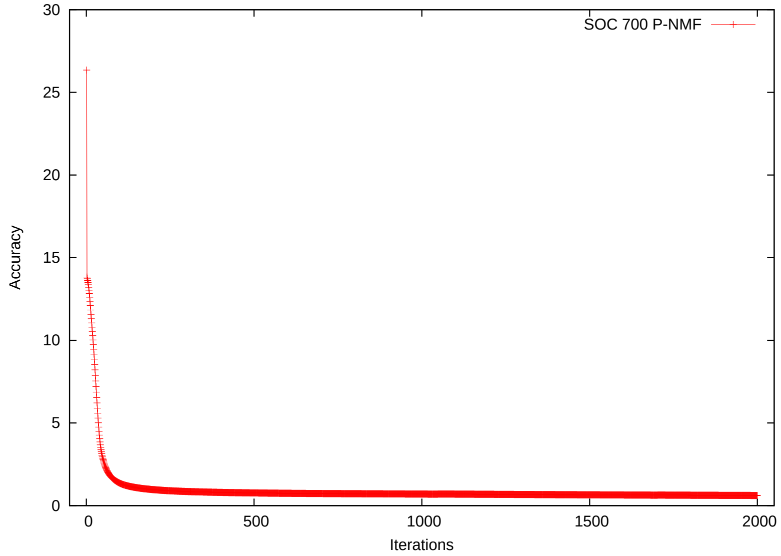
Each test: run 1-8 threads until converges

Convergence Criteria: change in $f(W, H) \leq 0.001$

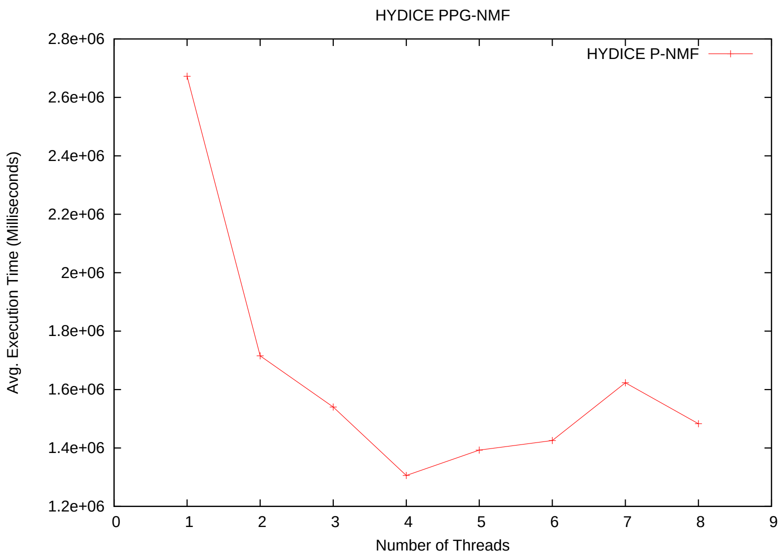
All runs in a test initialized with same seed value

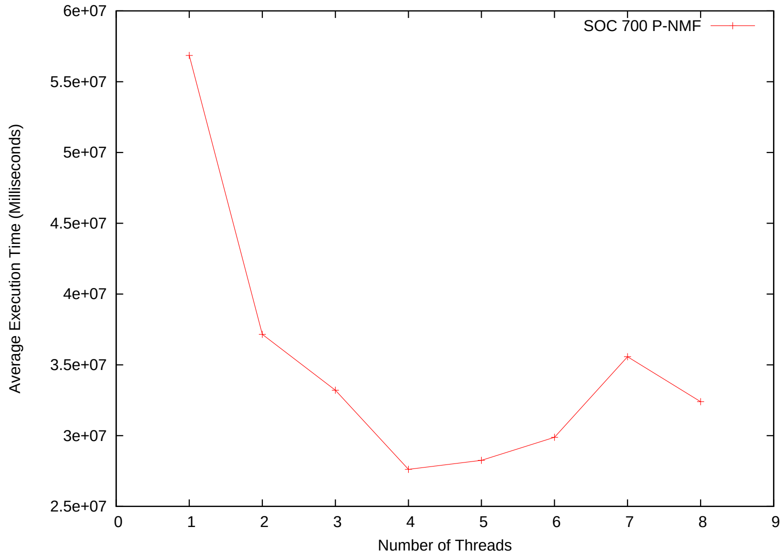
P-NMF Accuracy



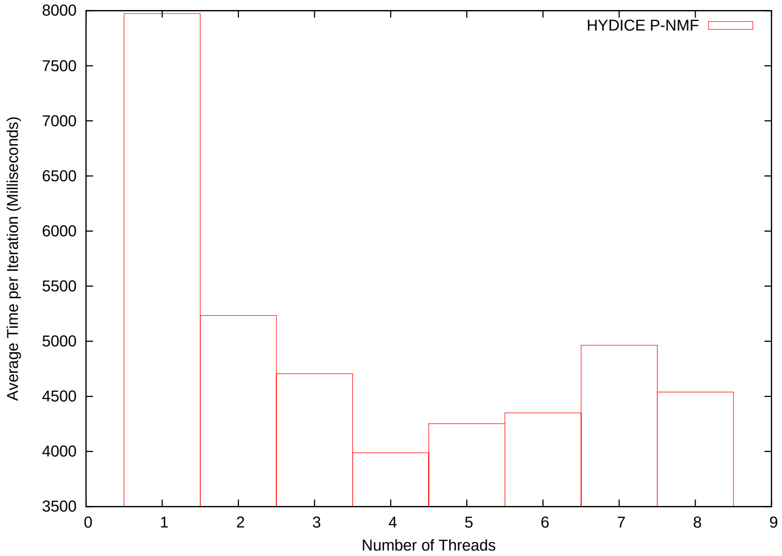


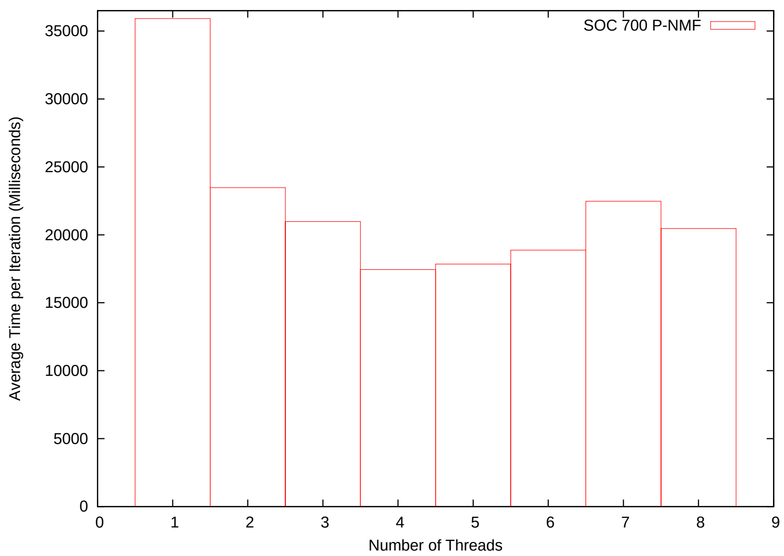
P-NMF Execution Time



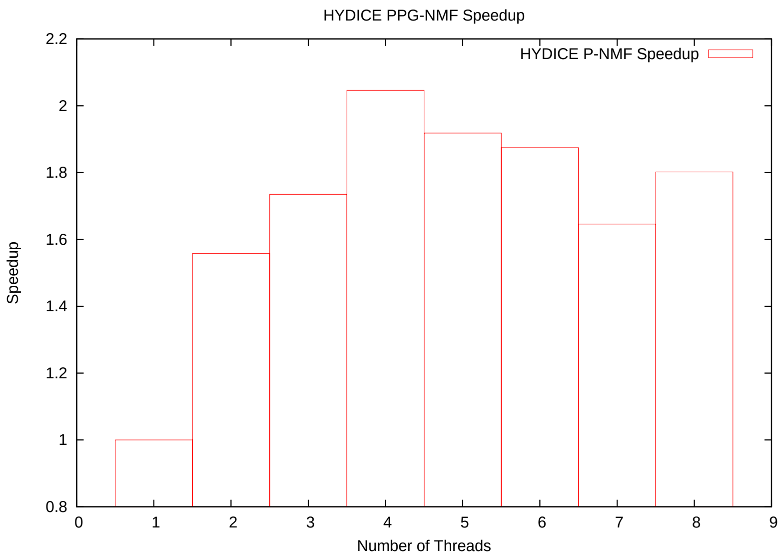


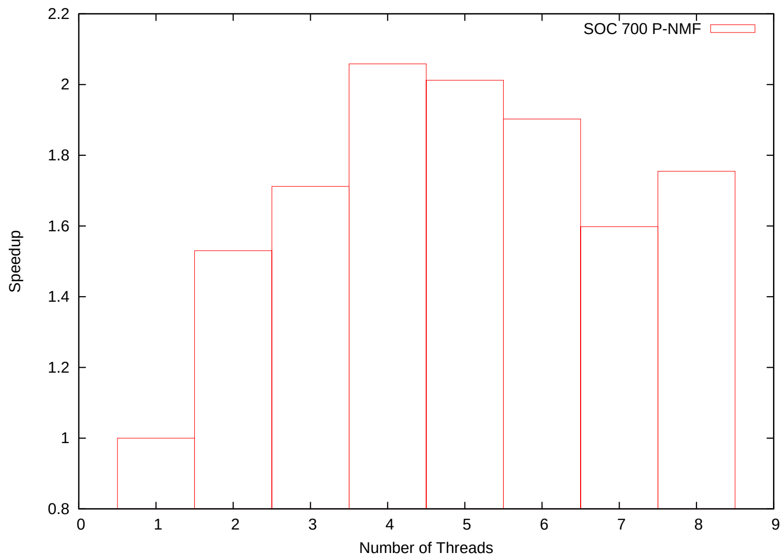
P-NMF Average Time Per Iteration



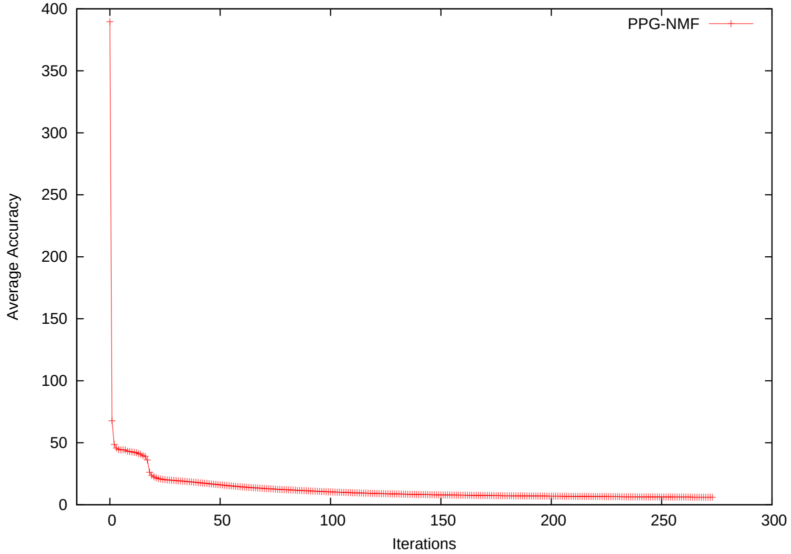


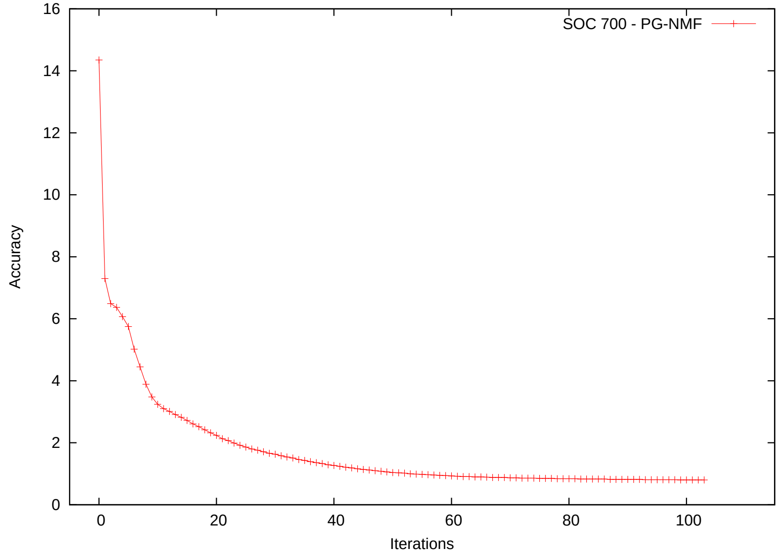
P-NMF Speedup



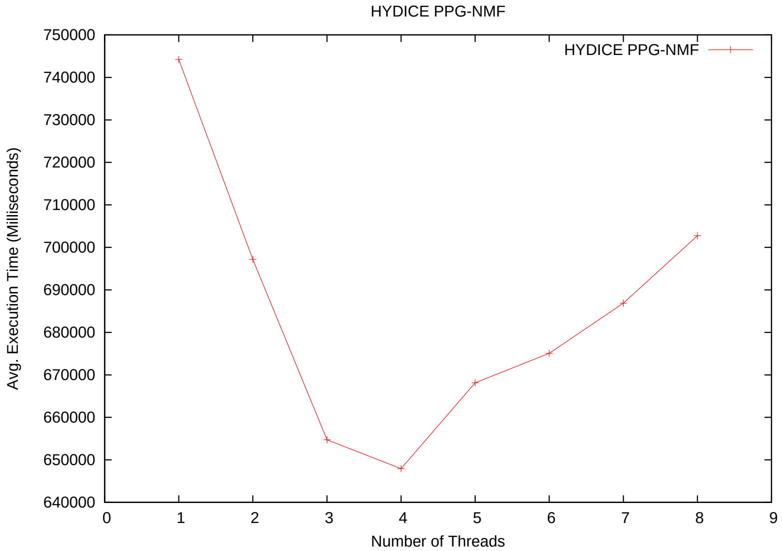


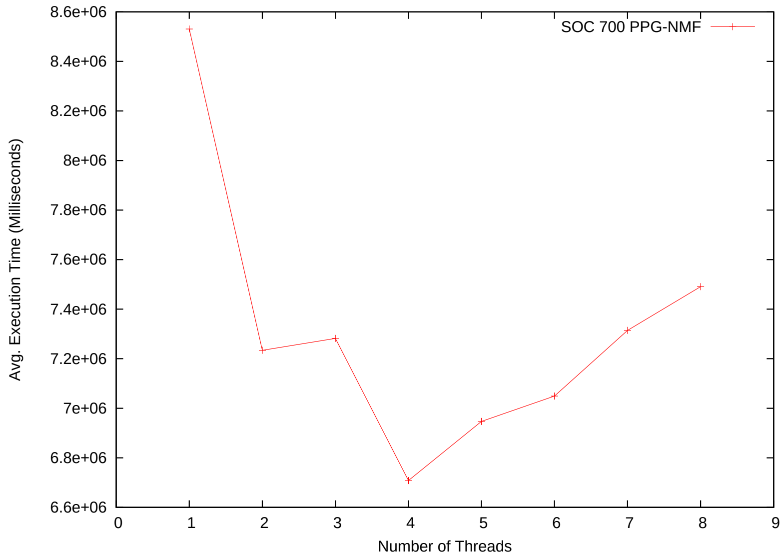
PPG-NMF Accuracy



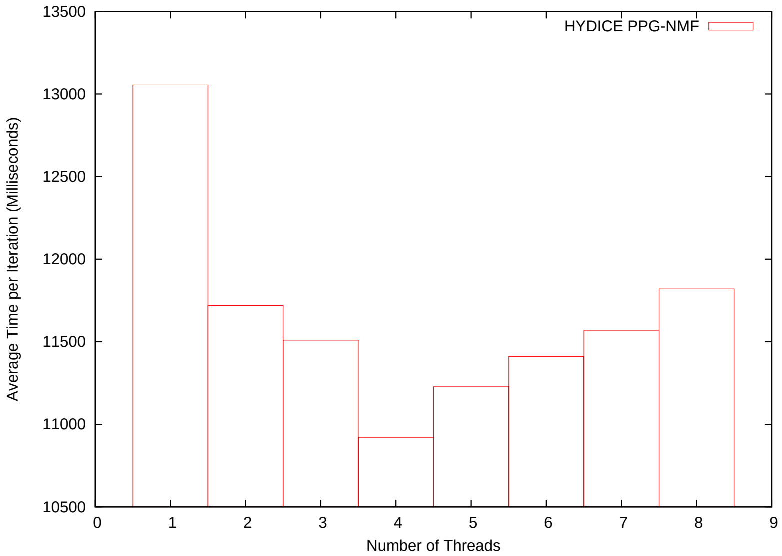


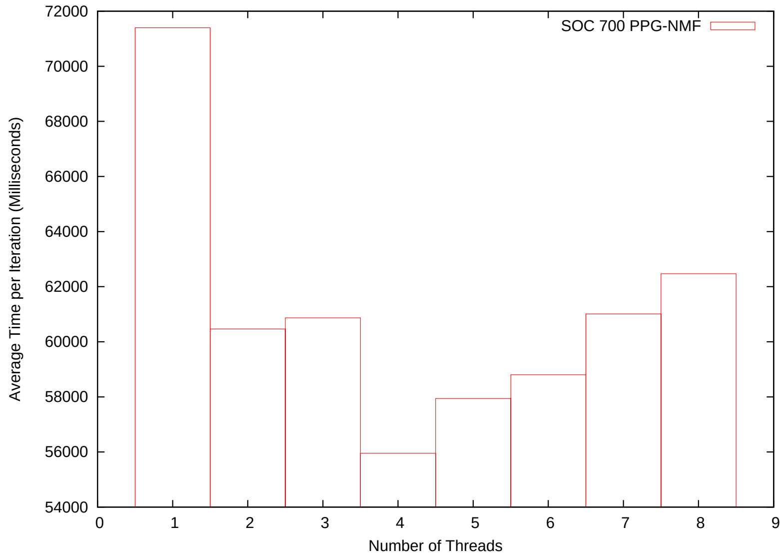
PPG-NMF Execution Time



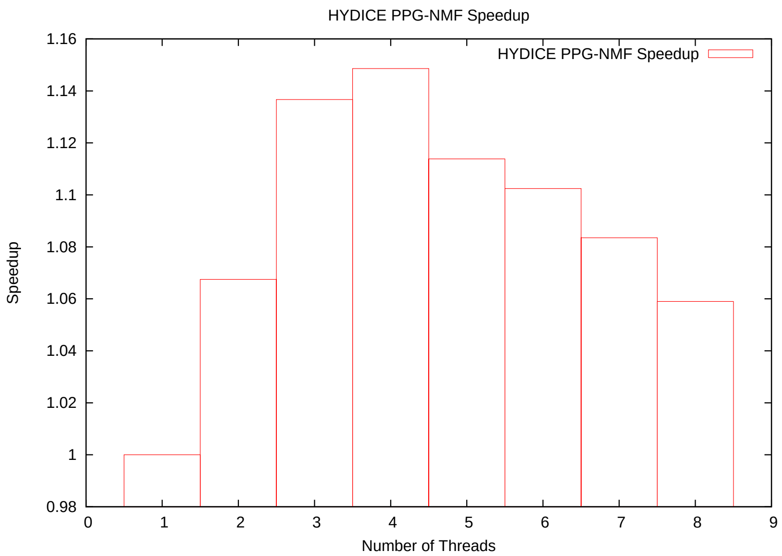


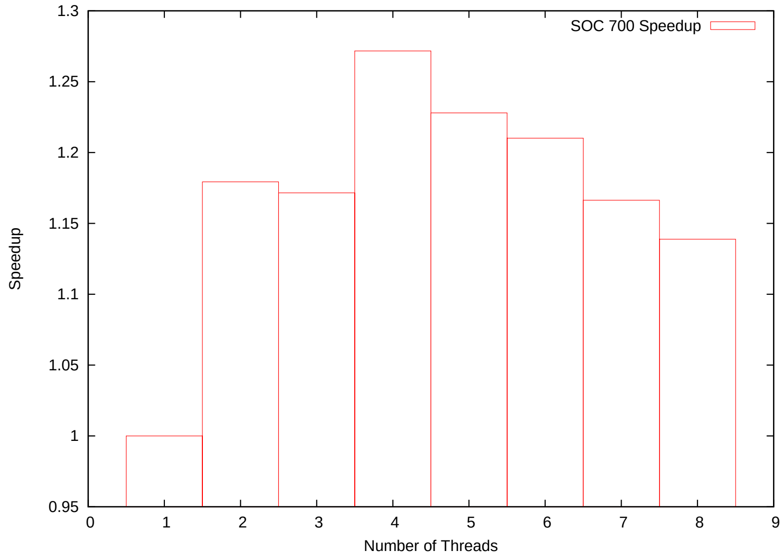
PPG-NMF Average Time Per Iteration



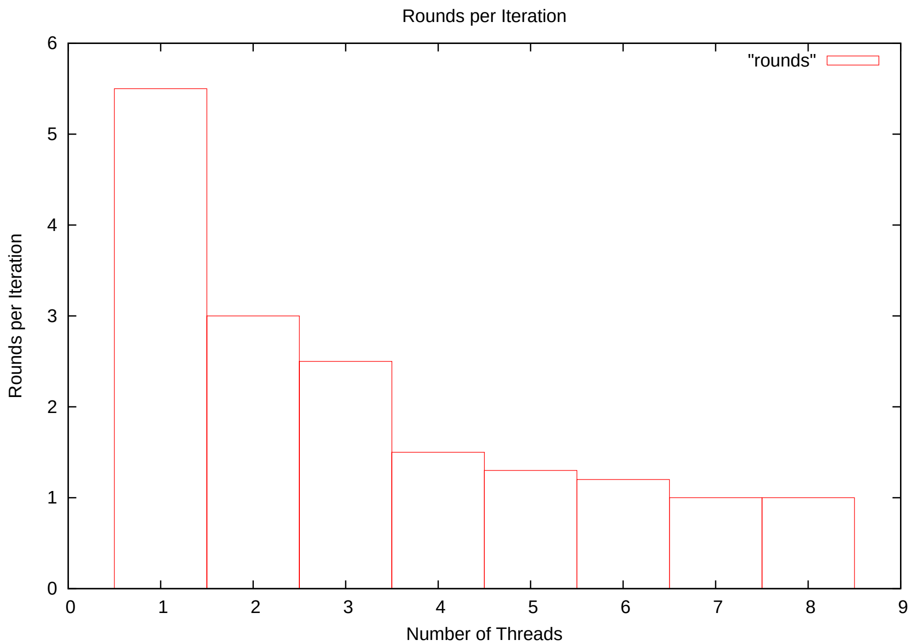


PPG-NMF Speedup





PPG-NMF Rounds per Iteration



for SOC 700 Data

Objectives

1. Background
2. Investigate Novel Feature Extraction methods (NMF)
3. Design, implement and test parallel NMF algorithms
4. **Initiate development of Java based hyperspectral imaging toolkit**

Hyperspectral Java Toolkit

Read and Write Hyperspectral files

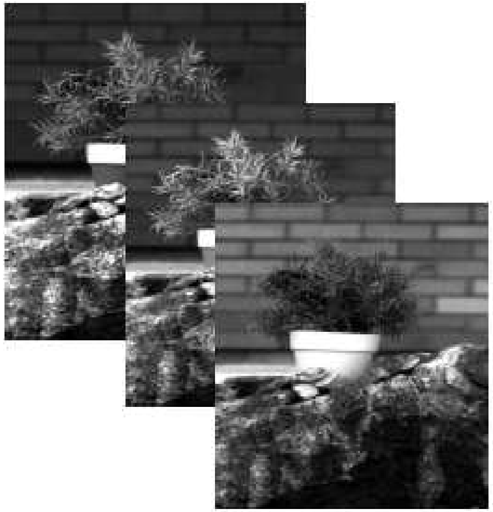
Convert between different data types

Convert between big endian and little endian

Launch various feature extraction algorithms and collect the results

Load and display hyperspectral images on the screen

Visualization



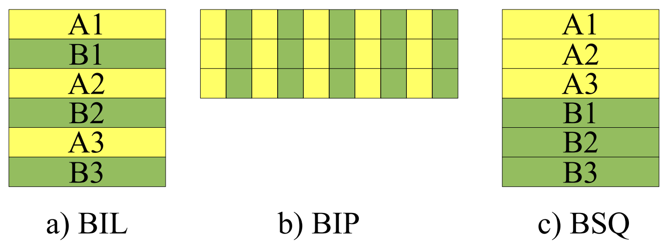
a)

b)

a) each band displayed separately as a grayscale image

b) 3 bands combined to form a single RGB image

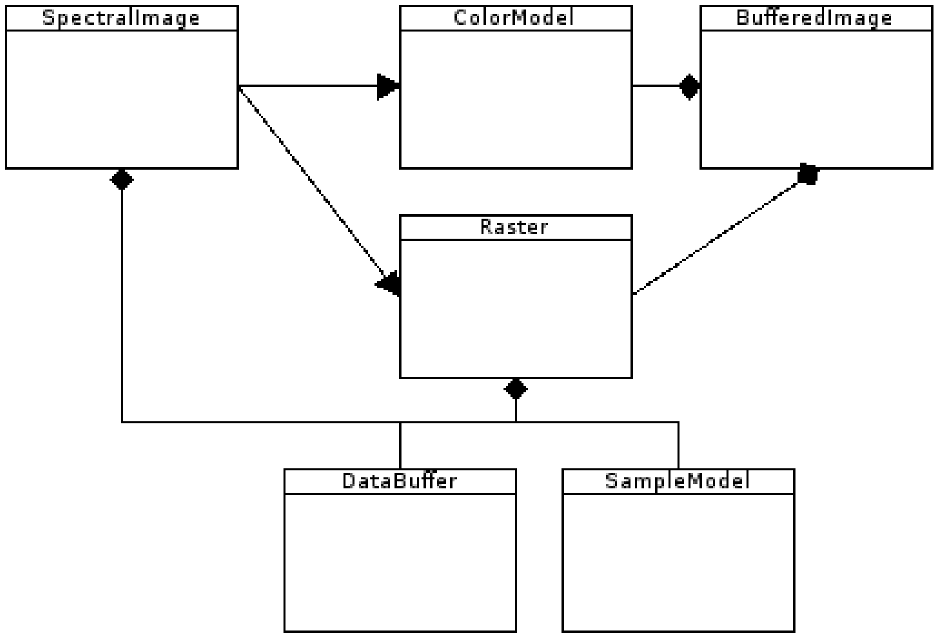
I/O Encoding



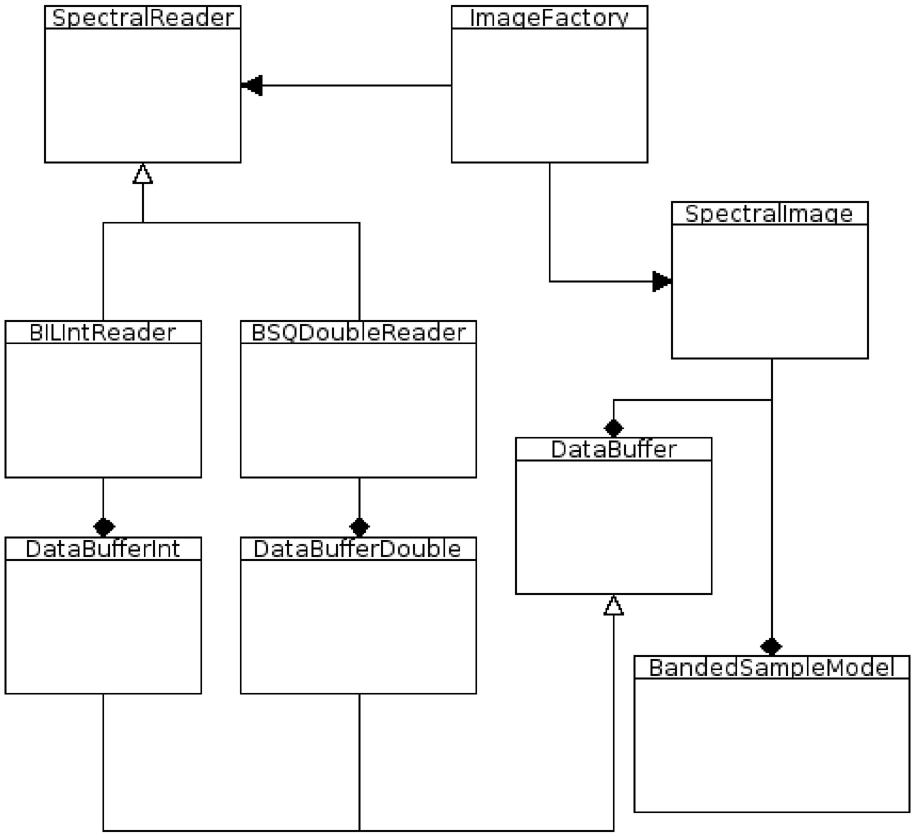
1. byte (signed or unsigned)
2. 32 bit integer (signed or unsigned)
3. 64 bit integer (signed or unsigned)
4. 32 or 64 bit float

Pixels can be in big endian or little endian.

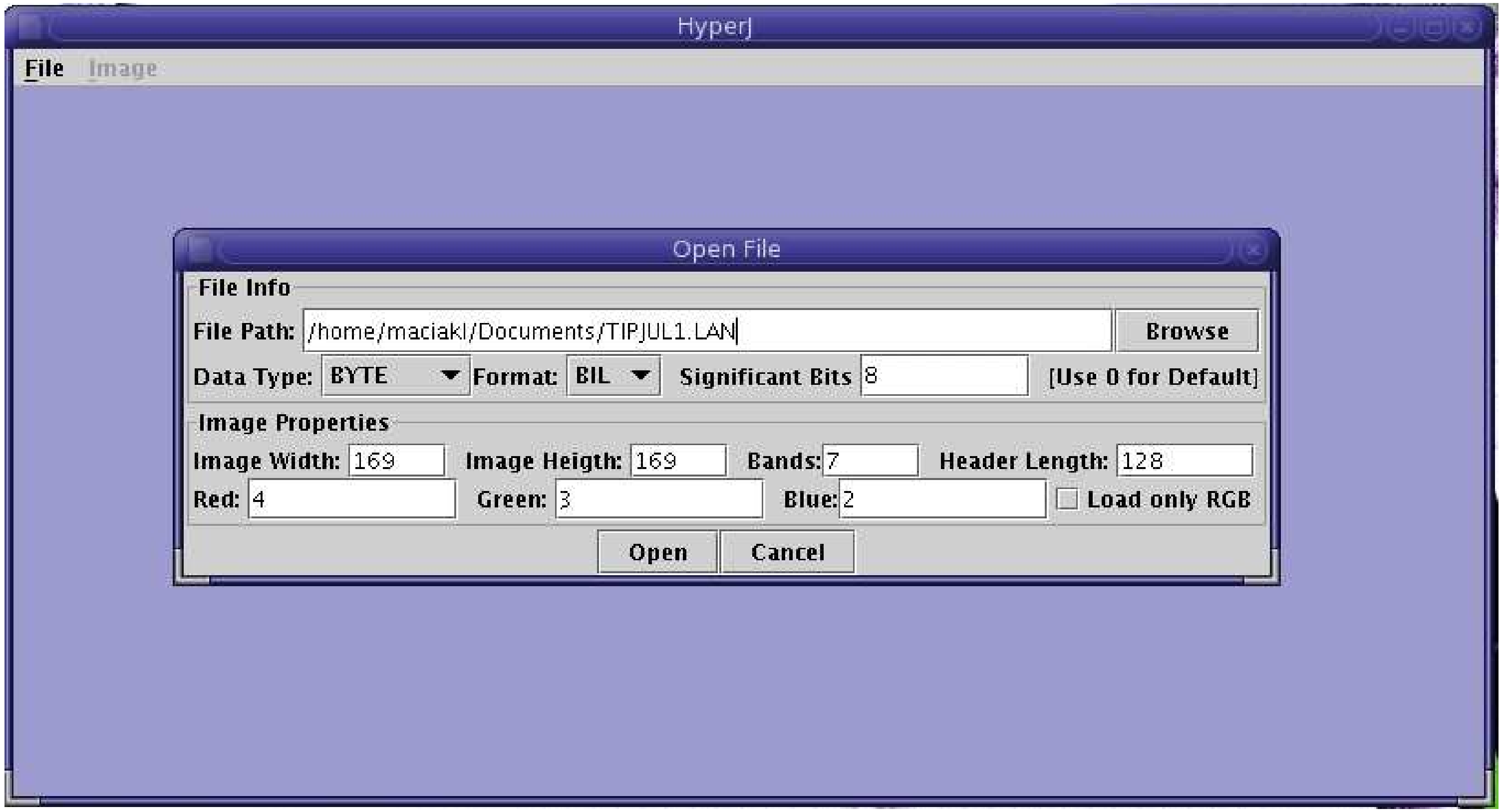
Hyperspectral Java Toolkit: Visualization



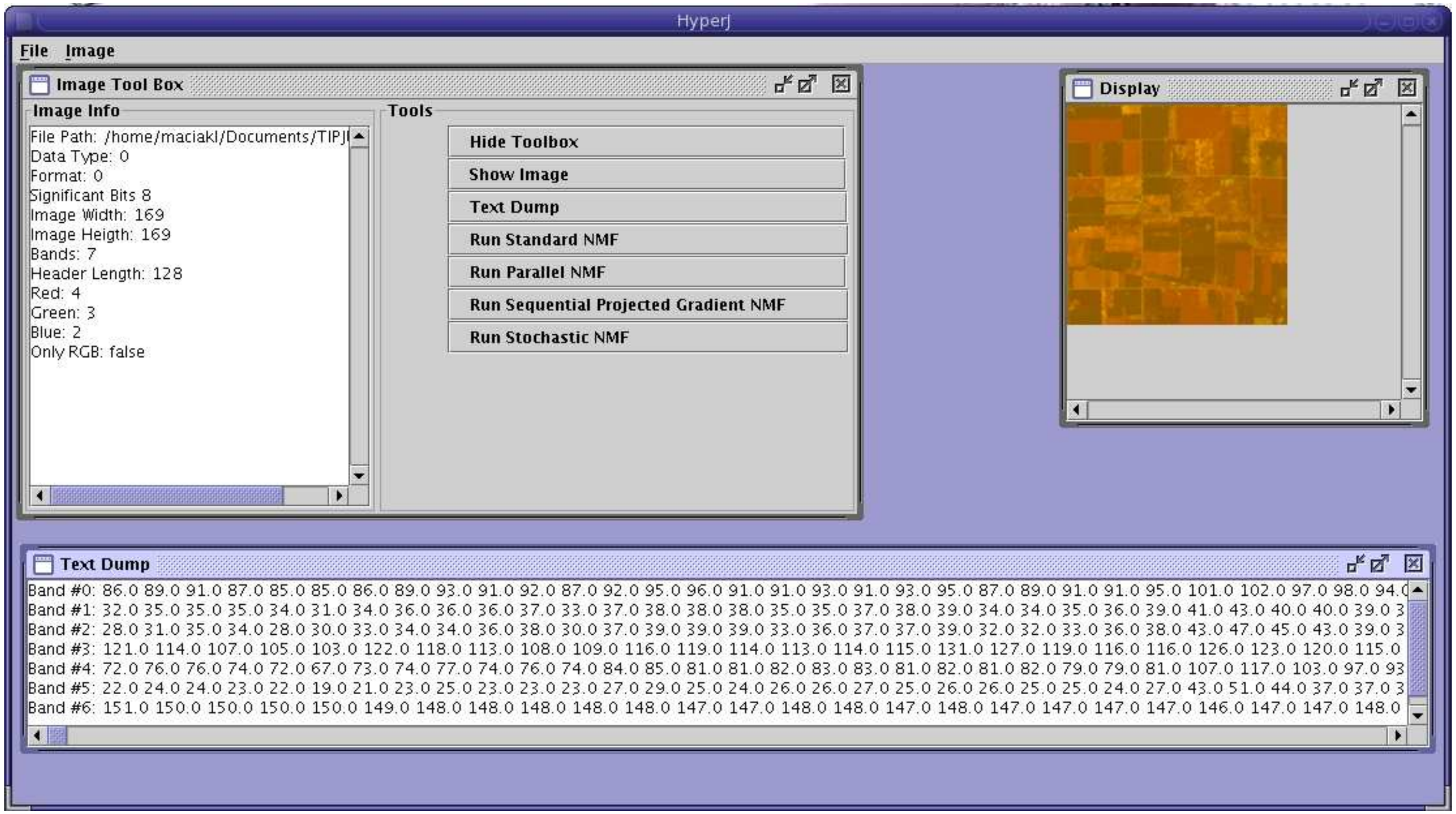
Hyperspectral Java Toolkit: I/O



Hyperspectral Java Toollit: GUI



Hyperspectral Java Toollit: GUI



Future Work

More testing on a dedicated 6-8 CPU SMP

Develop a distributed memory version of P-NMF

Optimize the memory requirements

Plugin support for the toolkit

Better GUI support

Publications

S. A. Robila, L. Maciak, Novel Approaches for Feature Extraction in Hyperspectral Images, Proceedings IEEE LISAT, 2006, 7 pgs. on CD.

S. A. Robila, L. Maciak, Parallel Unmixing Algorithms for Hyperspectral Image Processing, SPIE Intelligent Robots and Computer Vision XXIV, vol. 6384, 2006, 10pgs., in print

S. A. Robila, L. Maciak, Sequential and Parallel Feature Extraction using Nonnegative Matrix Factorization, Proceedings IEEE LISAT 2007, 8 pgs on CD

S. A. Robila, L. Maciak, Parallel Projected Gradient Nonnegative Matrix Factorization for Hyperspectral Images, submitted to IEEE IGARSS 2007