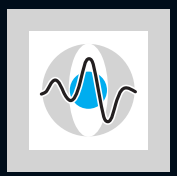
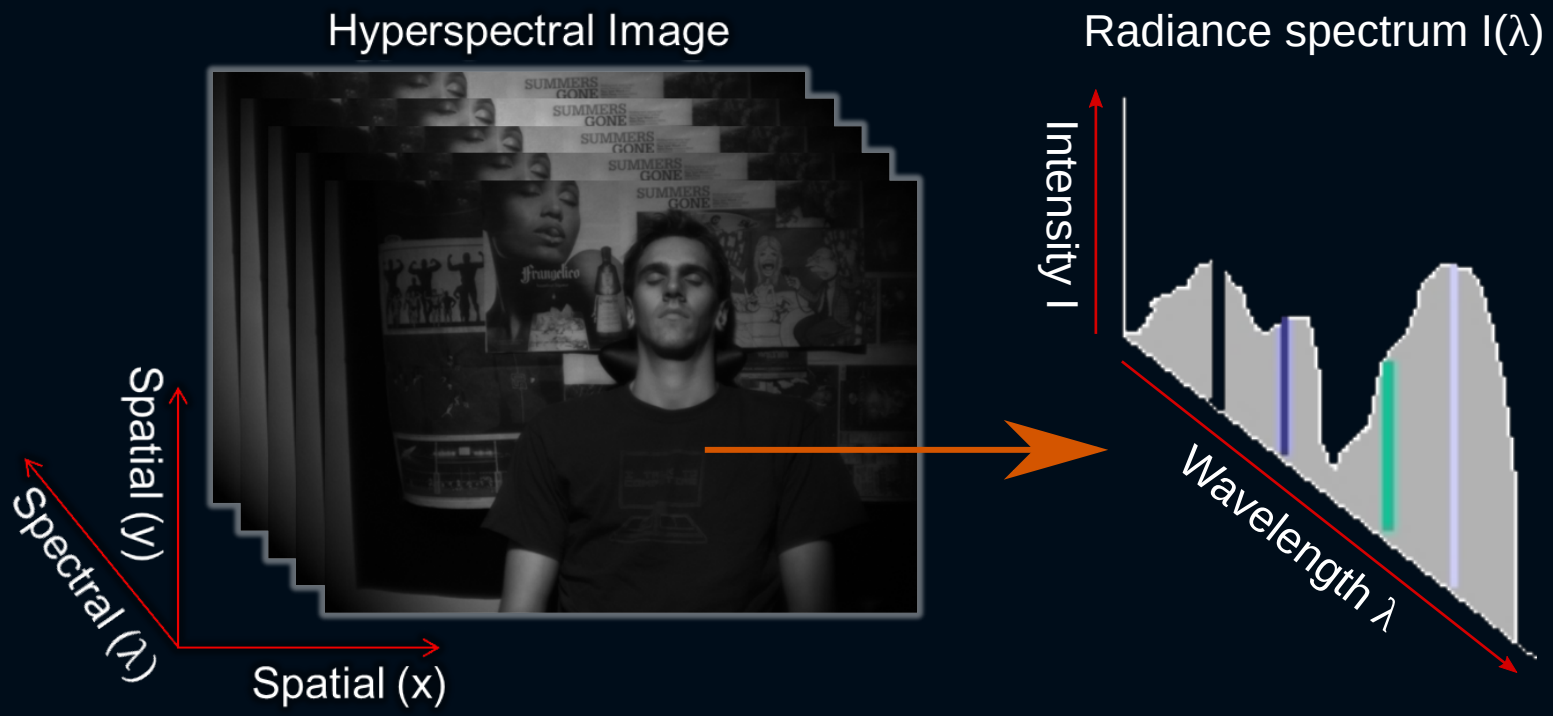
An Unsupervised Material Learning Method for Imaging Spectroscopy

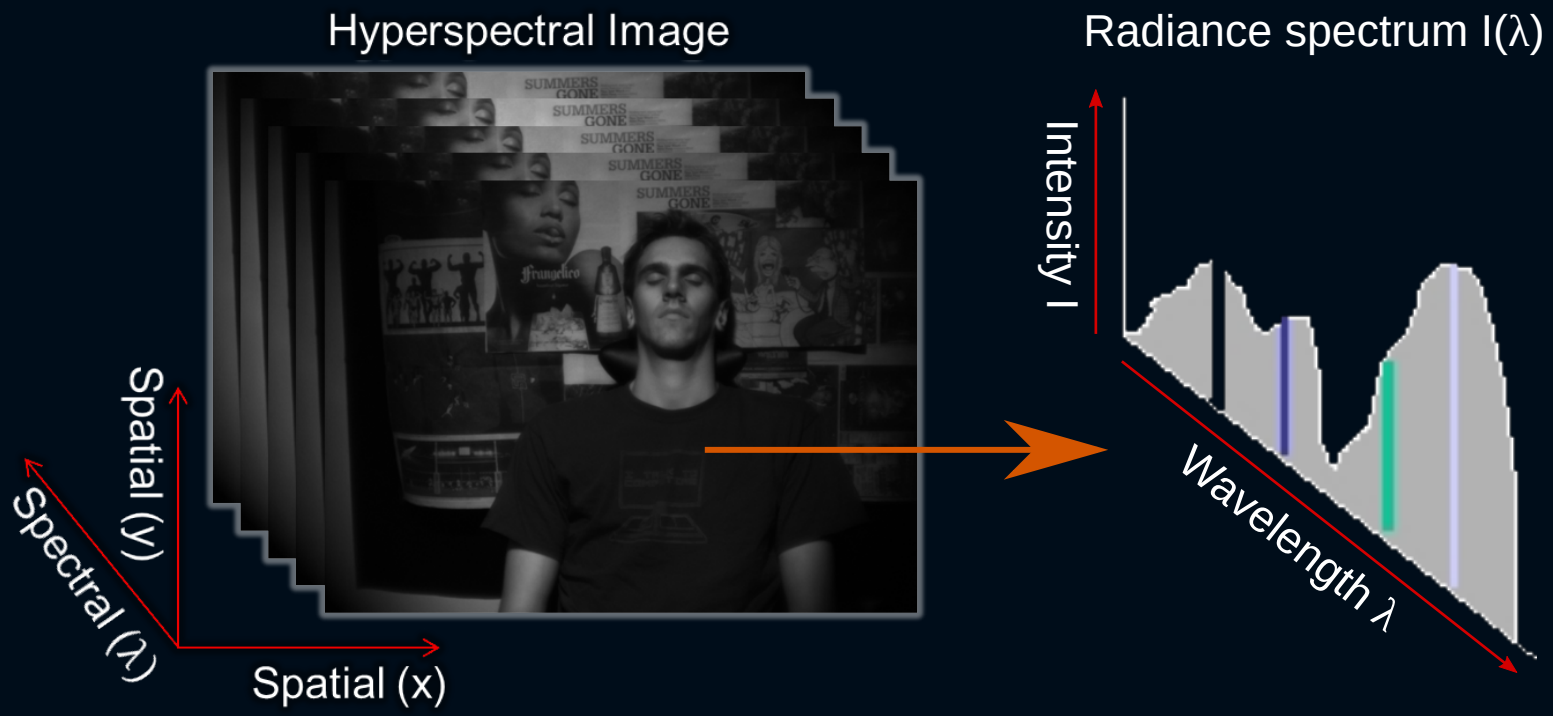
Johannes Jordan, Elli Angelopoulou, Antonio Robles-Kelly
University of Erlangen-Nuremberg; National ICT Australia



Multispectral Imaging / Imaging Spectroscopy



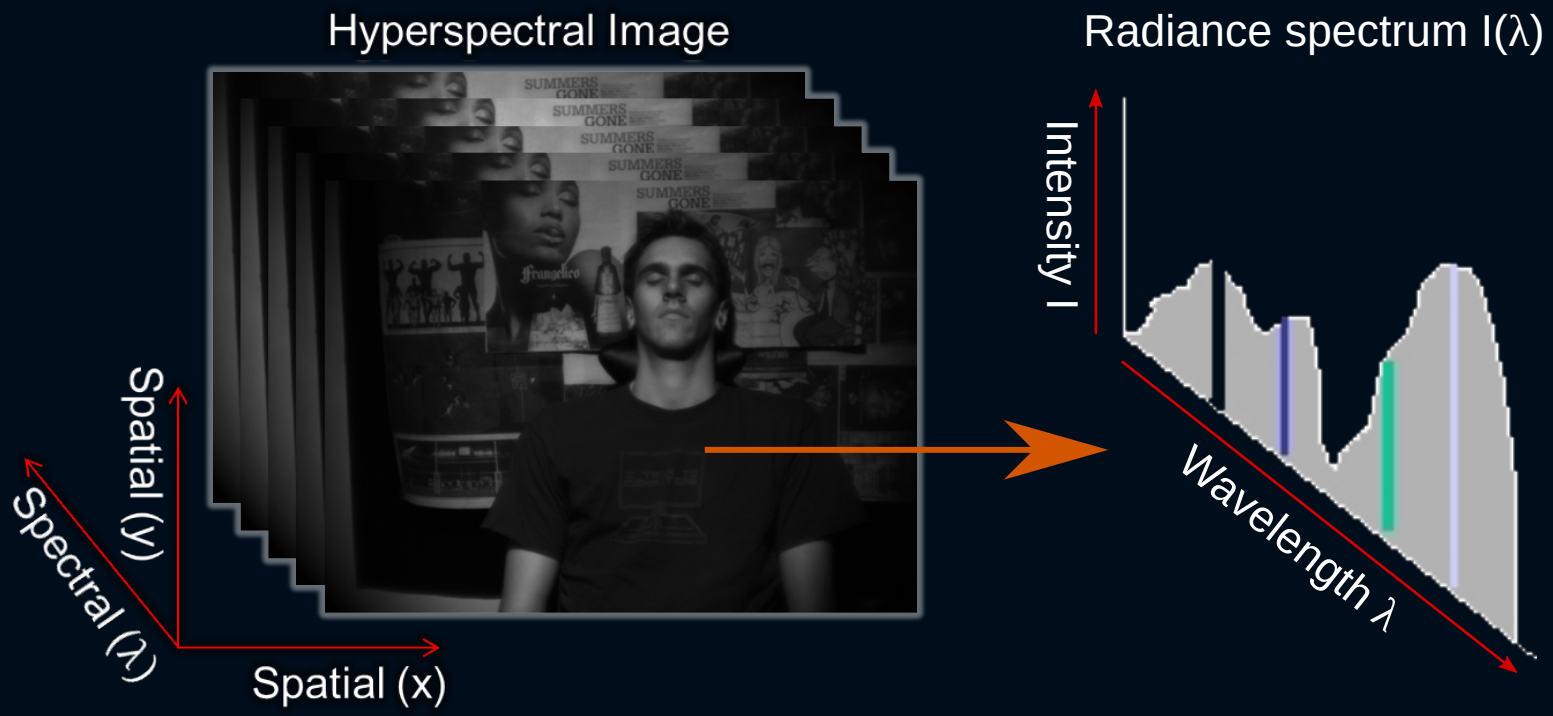
Multispectral Imaging / Imaging Spectroscopy



Benefits:

- feature-rich
- high accuracy possible

Multispectral Imaging / Imaging Spectroscopy



Benefits:

- feature-rich
- high accuracy possible

Challenges:

- high-dimensional
- highly correlated, but noisy

Contribution

Application: Material Learning

- False-colour visualisation of scene materials
- Extraction of material prototypes
    - Common application in remote sensing: Spectral unmixing

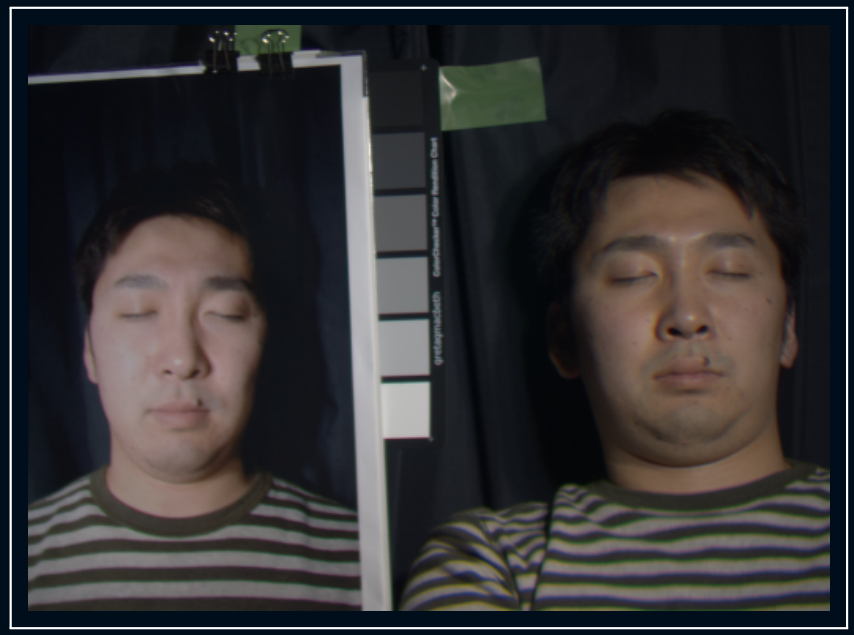
Contribution

Application: Material Learning

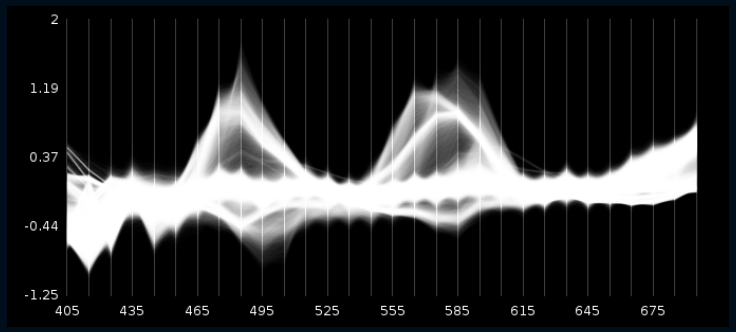
- False-colour visualisation of scene materials
- Extraction of material prototypes
    - Common application in remote sensing: Spectral unmixing

Methodology:

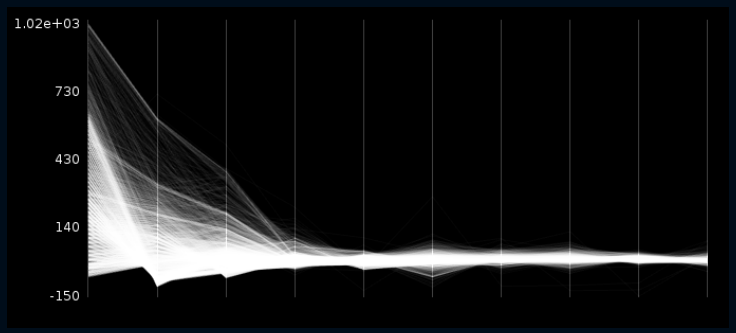
- Unsupervised manifold learning
- Fast clustering using established methods



Manifold Learning vs Dimensionality Reduction

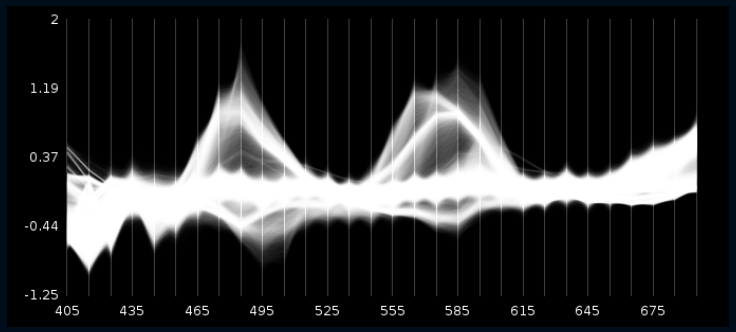


Original data (image spectral gradient)

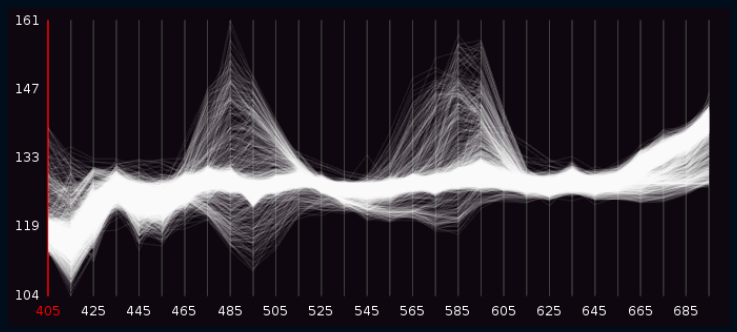


Principal Component Analysis

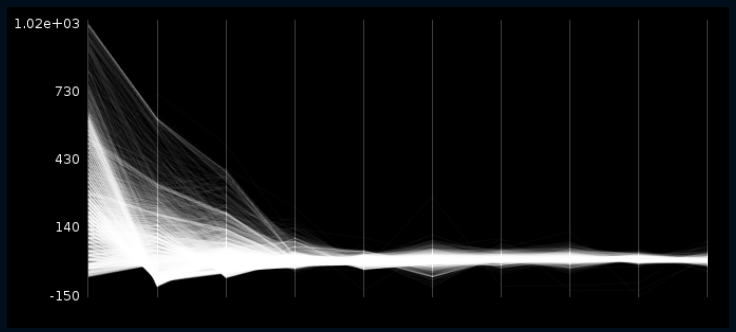
Manifold Learning vs Dimensionality Reduction



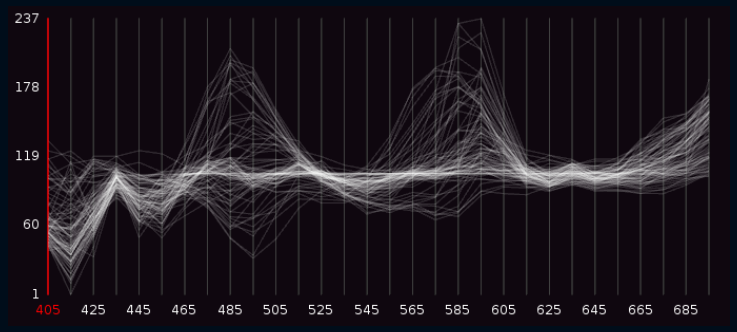
Original data (image spectral gradient)



Sparse representation



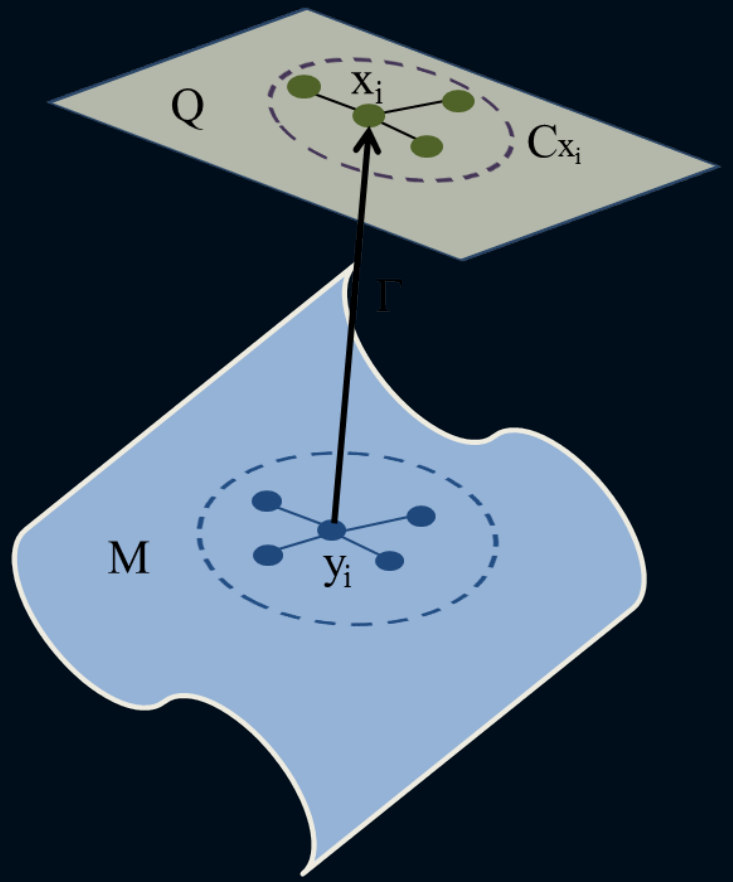
Principal Component Analysis



Material prototypes

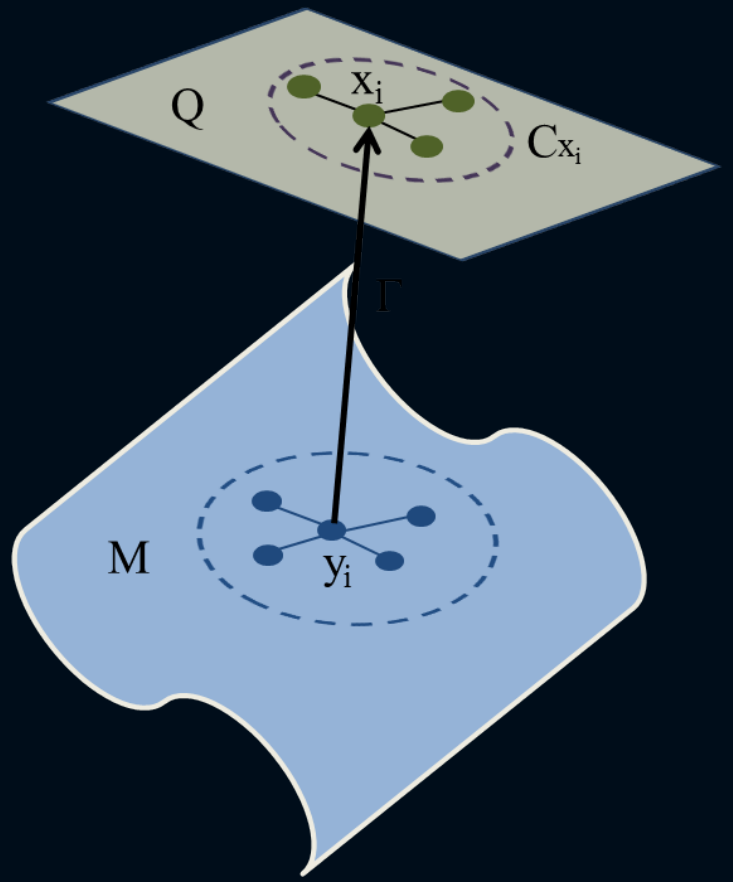
Concept

- graph $G_M = (Y, E_M)$,
  node set Y comprises of $y_i \in \mathbb{R}^m$
- determining E_M is a hard problem
- graph $G_Q = (X, E_Q)$,
  constructed to help derive the manifold topology



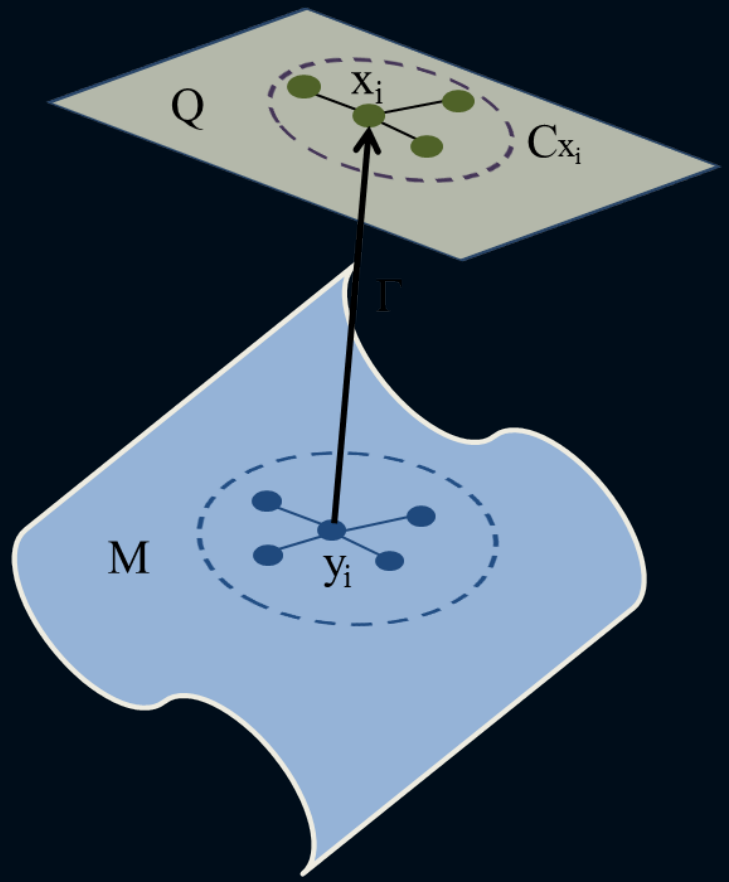
Concept

- graph $G_M = (Y, E_M)$,
  node set Y comprises of $y_i \in \mathbb{R}^m$
- determining E_M is a hard problem
- graph $G_Q = (X, E_Q)$,
  constructed to help derive the manifold topology
- X, Y are linked via
  $x_j = \sum_{y_k \in Y} \mathbb{I}(x_j \sim y_k)\Gamma(y_k)$
- $\Gamma : \mathbb{R}^m \mapsto \mathbb{R}^q$,
  $\mathbb{I}(x_j \sim y_k)$ indicator function for adjacency



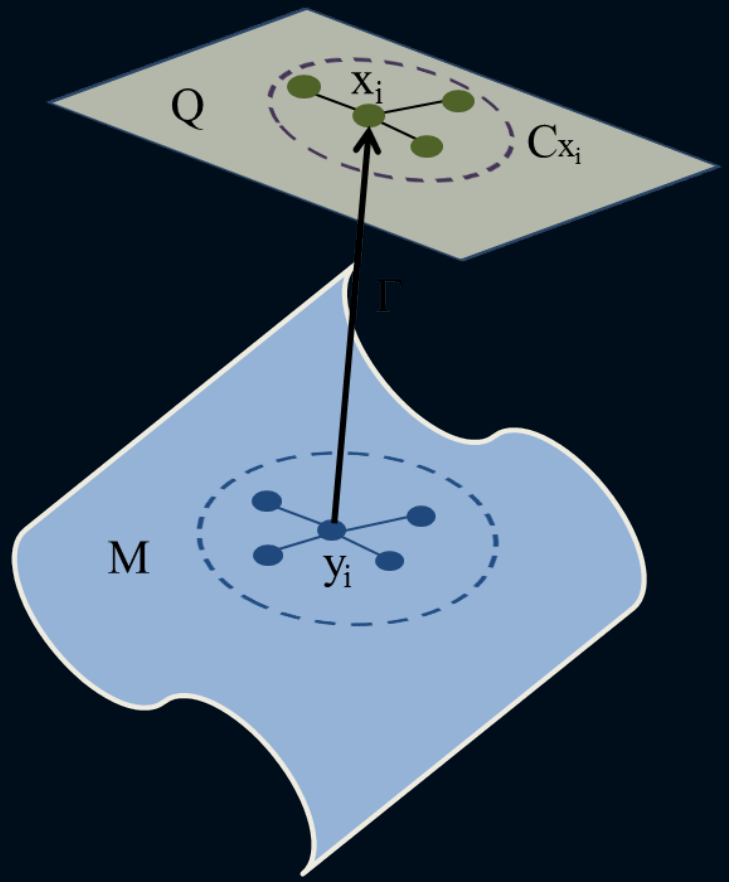
Concept (1)

- G_Q is a **Gibbs Field**
- conditional probability of vertex x_i:
  $P(x_i \mid \mathcal{C}_{x_i}) = \frac{1}{Z_Q} \prod_{x_j \in \mathcal{C}_{x_i}} f_Q(x_j, x_i)$
- $\mathcal{C}_{x_i}$ clique centered at x_i
- Z_Q partition function,
  $f_Q(x_j, x_i)$ potential function, also
  edge weight between x_j and x_i

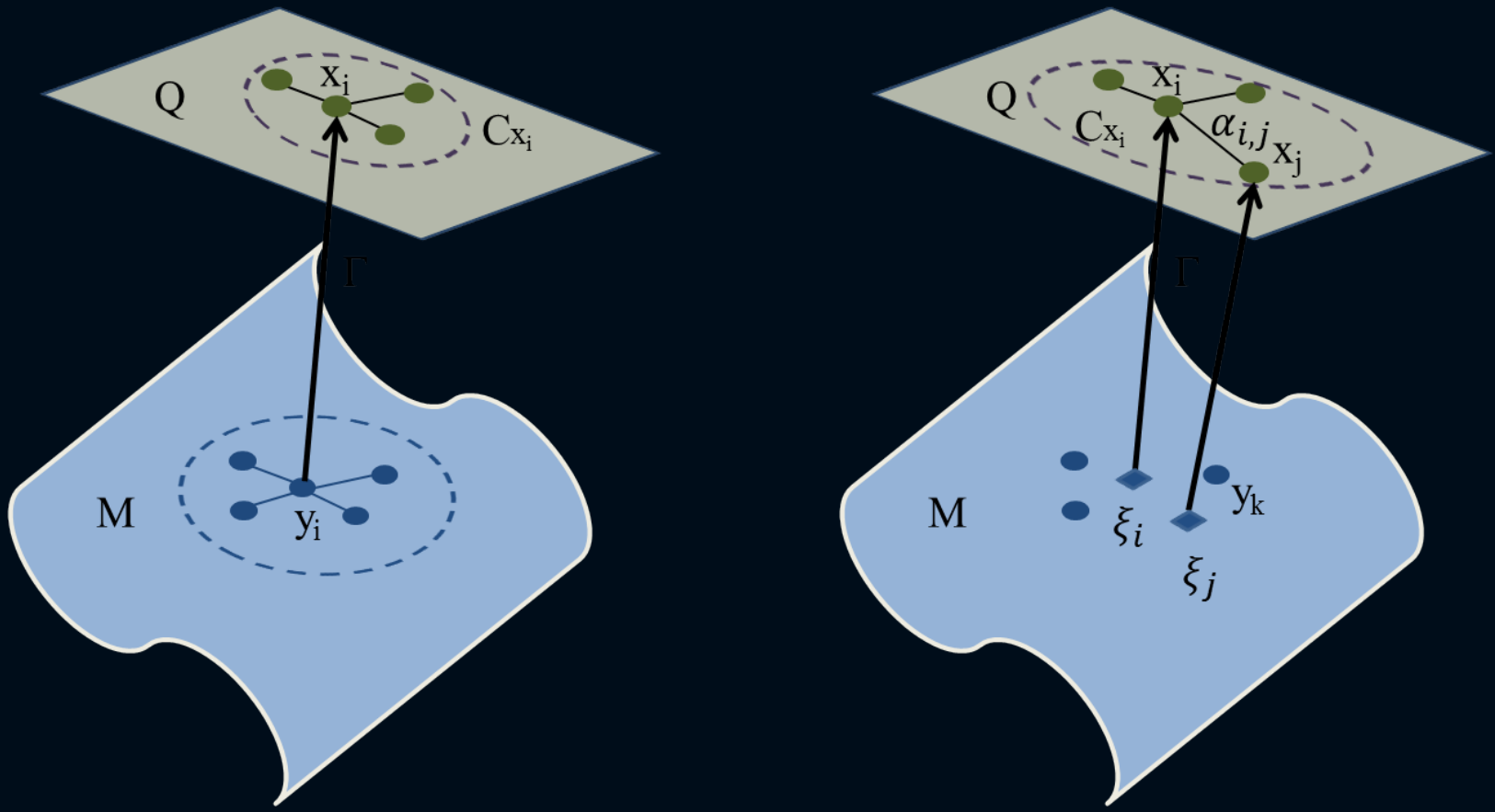


Concept (1)

- G_Q is a **Gibbs Field**
- conditional probability of vertex x_i:
  $P(x_i \mid \mathcal{C}_{x_i}) = \frac{1}{Z_Q} \prod_{x_j \in \mathcal{C}_{x_i}} f_Q(x_j, x_i)$
- $\mathcal{C}_{x_i}$ clique centered at x_i
- Z_Q partition function,
  $f_Q(x_j, x_i)$ potential function, also
  edge weight between x_j and x_i
- maximum-likelihood estimation
  (MLE) on the vertex-set $Y \in G_M$
  based upon the vertices $X \in G_Q$



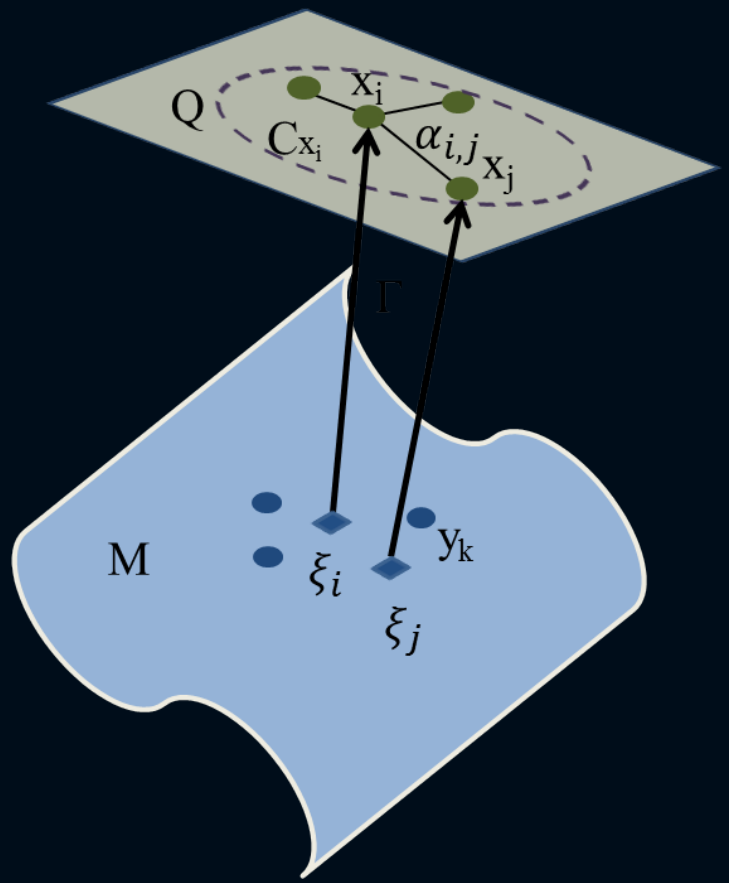
Concept (2)



We introduce the parameter set ξ_j with correspondence to x_j.

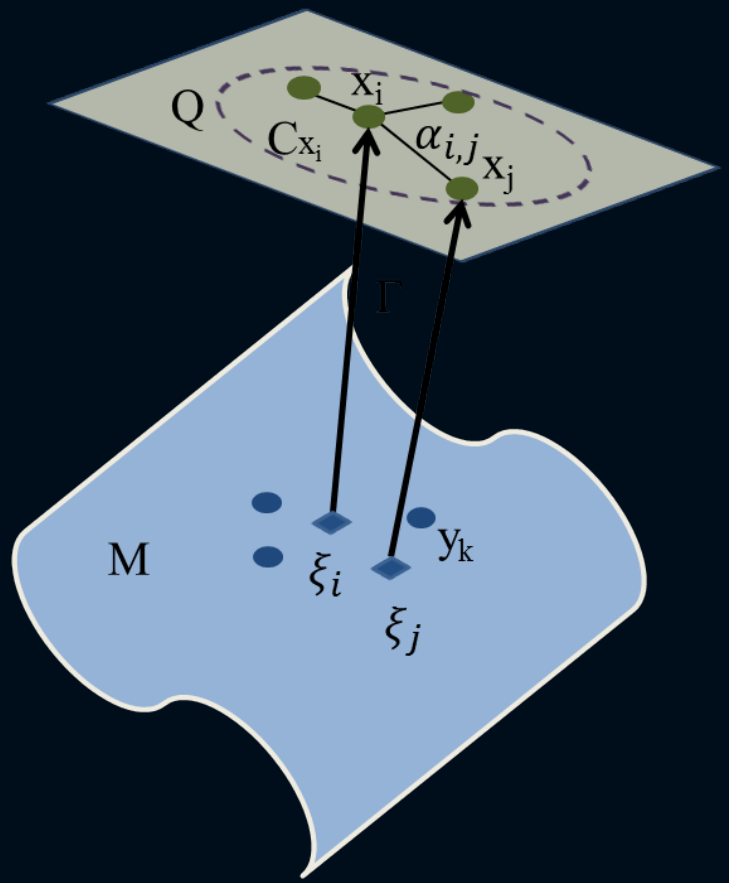
Concept (3)

- $x_j = \sum_{y_k \in Y} \eta_j \mathcal{K}(y_k, \xi_j)$
- prior η_j, kernel function $\mathcal{K}(\cdot)$
- vertex x_j in G_Q corresponds to parameter vector ξ_j



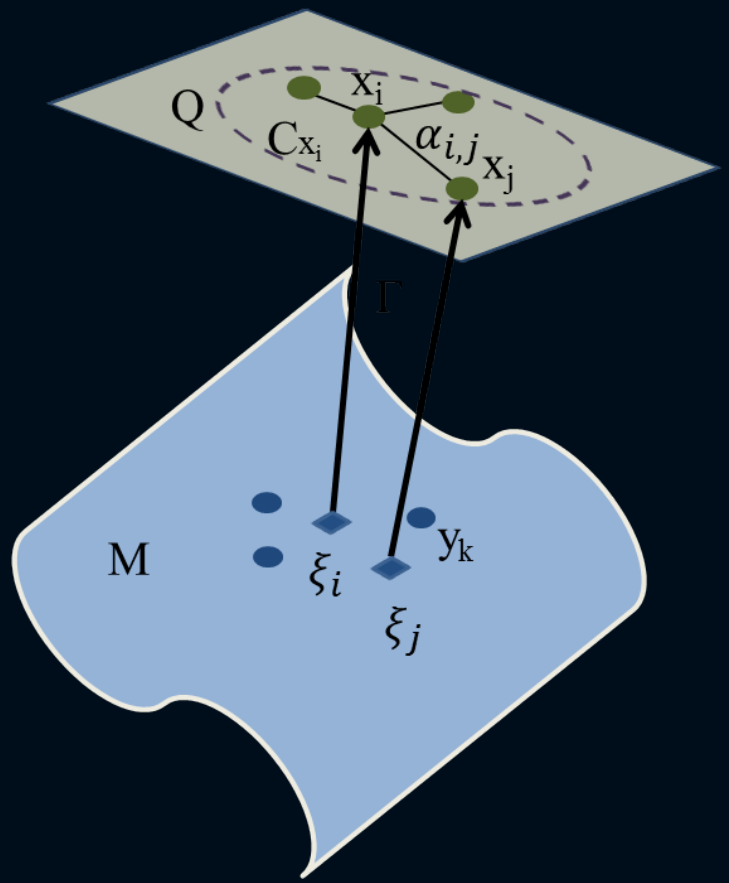
Concept (3)

- $x_j = \sum_{y_k \in Y} \eta_j \mathcal{K}(y_k, \xi_j)$
- prior η_j, kernel function $\mathcal{K}(\cdot)$
- vertex x_j in G_Q corresponds to parameter vector ξ_j
- conditional probability of vertex x_i:
  $P(x_i \mid \mathcal{C}_{x_i}, Y) =$
  $\frac{1}{Z_Q} \prod_{x_j \in \mathcal{C}_{x_i}} f_Q\left(\sum_{y_k \in Y} \mathcal{K}(y_k, \xi_j), x_i \right)$



Concept (3)

- $x_j = \sum_{y_k \in Y} \eta_j \mathcal{K}(y_k, \xi_j)$
- prior η_j, kernel function $\mathcal{K}(\cdot)$
- vertex x_j in G_Q corresponds to parameter vector ξ_j
- conditional probability of vertex x_i:
  $P(x_i \mid \mathcal{C}_{x_i}, Y) =$
  $\frac{1}{Z_Q} \prod_{x_j \in \mathcal{C}_{x_i}} f_Q\left(\sum_{y_k \in Y} \mathcal{K}(y_k, \xi_j), x_i \right)$
- **Choose $f_Q(\cdot)$, $\mathcal{K}(\cdot)$, obtain MLE estimate of parameter ξ_j.**



Parameter Learning

Potential function:

- Gibbs distribution:

$$f_Q(x_j, x_i) = \sum_{y_k \in Y} \alpha_{i,j} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$

Parameter Learning

Potential function:

- Gibbs distribution:

$$f_Q(x_j, x_i) = \sum_{y_k \in Y} \alpha_{i,j} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$

- Kernel function:

$$\mathcal{K}(y_k, \xi_j) = \sum_{y_k \in Y} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$

Parameter Learning

Potential function:

- Gibbs distribution:

$$f_Q(x_j, x_i) = \sum_{y_k \in Y} \alpha_{i,j} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$

- Kernel function:

$$\mathcal{K}(y_k, \xi_j) = \sum_{y_k \in Y} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$

- Rewrite using kernel:

$$f_Q(x_j, x_i) = \alpha_{i,j} \sum_{y_k \in Y} \exp\left\{ -\frac{1}{T} E_{\xi_j}(y_k) \right\} = \eta_i \eta_j \mathcal{K}(y_k, \xi_j)$$

Parameter Learning

Potential function:

- Gibbs distribution:
  $$f_Q(x_j, x_i) = \sum_{y_k \in Y} \alpha_{i,j} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$
- Kernel function:
  $$\mathcal{K}(y_k, \xi_j) = \sum_{y_k \in Y} \exp\left\{ -\frac{1}{T} d_M(y_k, \xi_j)^2 \right\}$$
- Rewrite using kernel:
  $$f_Q(x_j, x_i) = \alpha_{i,j} \sum_{y_k \in Y} \exp\left\{ -\frac{1}{T} E_{\xi_j}(y_k) \right\} = \eta_i \eta_j \mathcal{K}(y_k, \xi_j)$$

Algorithm: Expectation-Maximization on ξ_j, $\alpha_{i,j}$

Relation to Self-Organizing Maps

Consider a sampling process in M,

- $\alpha_{i,j} = \mathbb{I}(x_j \sim \rho) h\left(\frac{1}{T} d_Q(x_i, x_j)^2\right)$
- ρ resembles the *best-matching unit* in a SOM
- $\mathbb{I}(x_j \sim \rho)$ indicates the vertex x_j connected to ξ_j with minimum distance to the input sample

Relation to Self-Organizing Maps

Consider a sampling process in M,

- $\alpha_{i,j} = \mathbb{I}(x_j \sim \rho) h\left(\frac{1}{T} d_Q(x_i, x_j)^2\right)$
- ρ resembles the *best-matching unit* in a SOM
- $\mathbb{I}(x_j \sim \rho)$ indicates the vertex x_j connected to ξ_j with minimum distance to the input sample

Observations:

- Clique size eq. SOM neighborhood size
- Gibbs temperature T eq. SOM learning rate
- More general formulation, but binary neighborhood function

Experiments

Input: publicly available test images

- 1020 x 1340 pixels, 33 bands (400*nm* - 720*nm*)

Experiments

Input: publicly available test images

- 1020 x 1340 pixels, 33 bands (400nm - 720nm)

Algorithmic parameters:

- G_M: 10^3 nodes in 3D, 6-connected lattice
- 50 000 iterations

Experiments

Input: publicly available test images

- 1020 x 1340 pixels, 33 bands ($400nm$ - $720nm$)

Algorithmic parameters:

- G_M: 10^3 nodes in 3D, 6-connected lattice
- 50 000 iterations

Visualization:

$$u = \sum_{\substack{y_k \in Y \\ x_j \in C_\rho}} \mathbb{I}(x_j \sim y_k) \exp\left\{ -\tfrac{1}{T} d_Q(x_j, \rho)^2 \right\} \exp\left\{ -\tfrac{1}{T} (y_k - \xi_\rho)^2 \right\}$$

Experiments

Input: publicly available test images

- 1020 x 1340 pixels, 33 bands ($400nm$ - $720nm$)

Algorithmic parameters:

- G_M: 10^3 nodes in 3D, 6-connected lattice
- 50 000 iterations

Visualization:

$$u = \sum_{\substack{y_k \in Y \\ x_j \in \mathcal{C}_\rho}} \mathbb{I}(x_j \sim y_k) \exp\left\{ -\tfrac{1}{T} d_Q(x_j, \rho)^2 \right\} \exp\left\{ -\tfrac{1}{T} (y_k - \xi_\rho)^2 \right\}$$

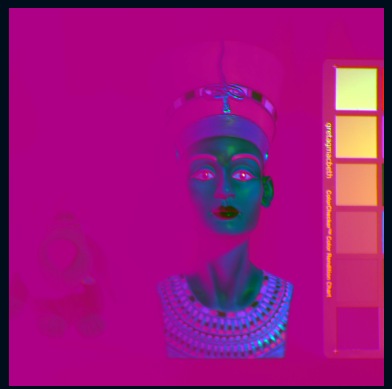
Material clustering:

- cluster on parameters ξ_j
- fast-adaptive mean shift algorithm
- hill-climbing to find material prototypes

Results: Learning process

Input

Pseudocolor



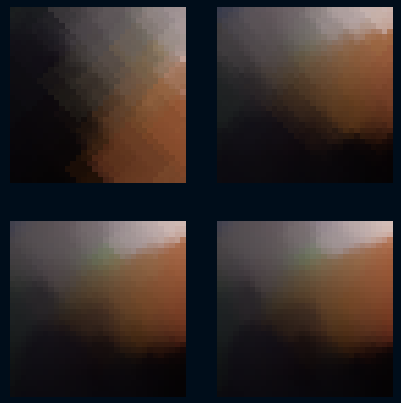
PCA

SOM-Euclidean

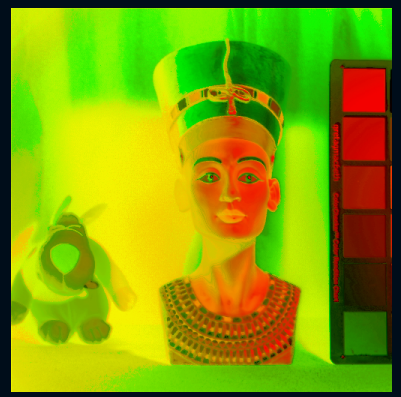
Parameter map

Visualisation

SOM-Constant

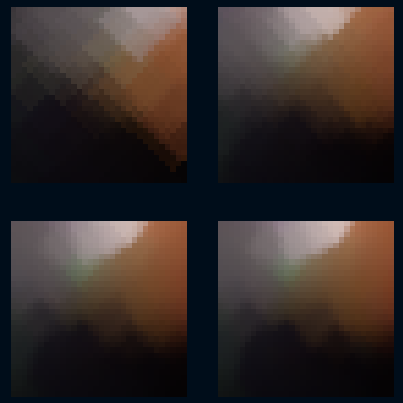


Parameter map



Visualisation

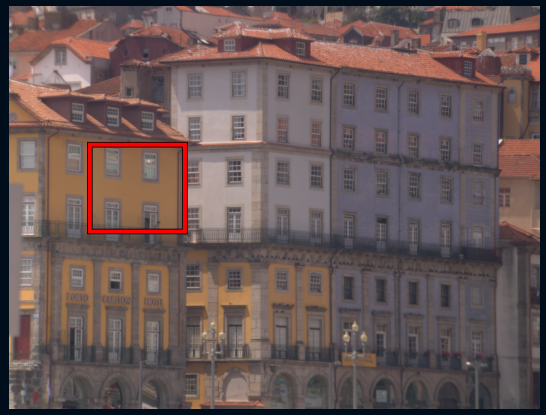
Our method

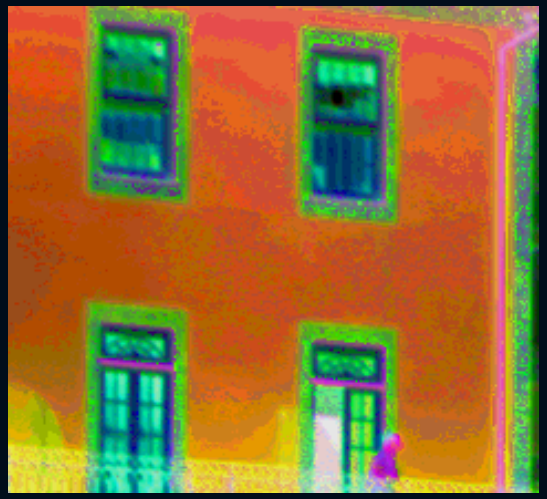


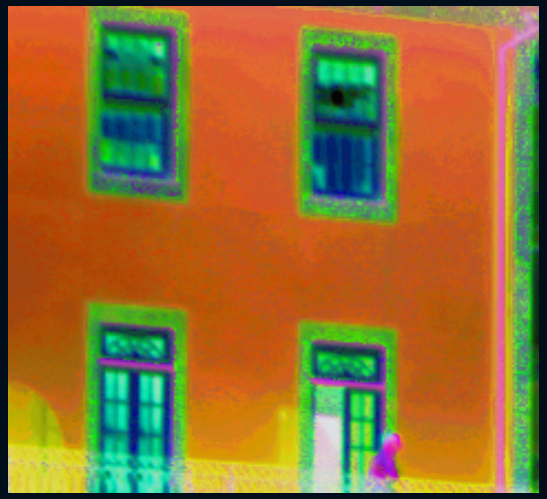
Parameter map

Visualisation

Results: Visualisation





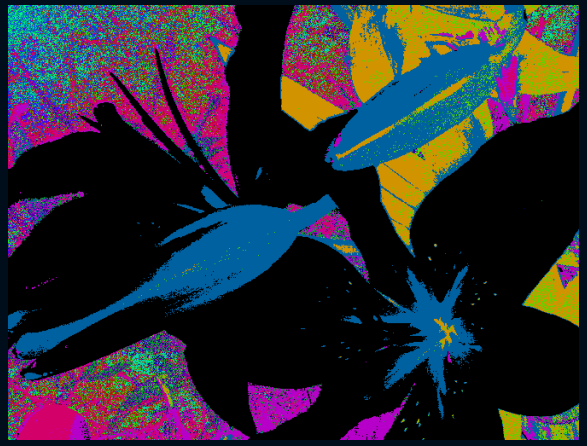


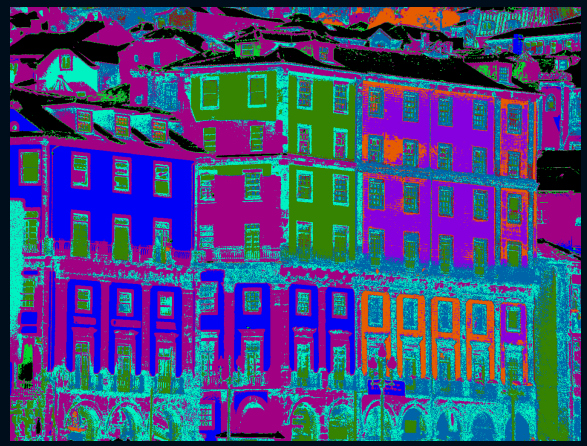
Input in pseudocolour

Visualisation using only the best matching unit

Visualisation recovered using clique information

Results: Material clustering



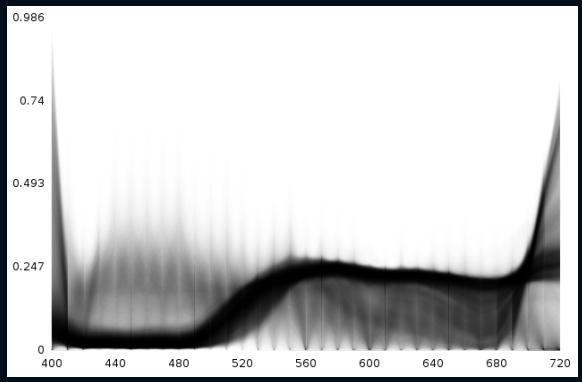


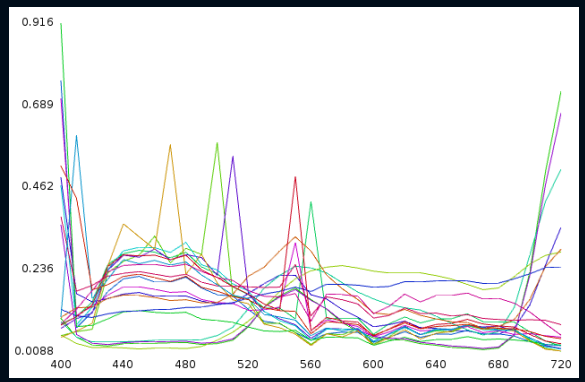
Input in pseudocolour

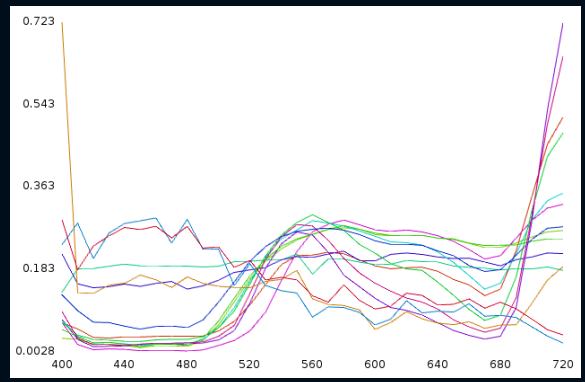
Huynh and Robles-Kelly

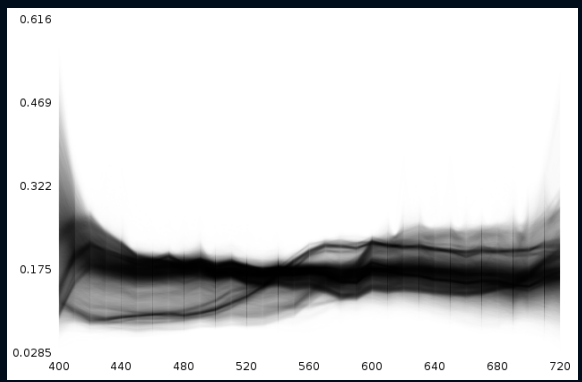
Our method

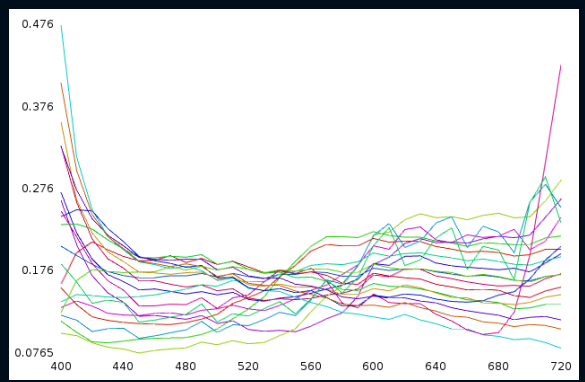
Results: Material clustering (2)











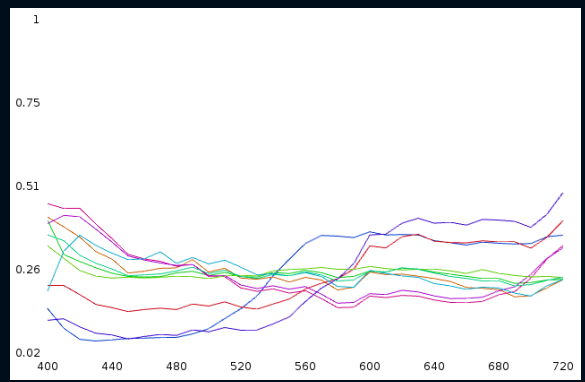
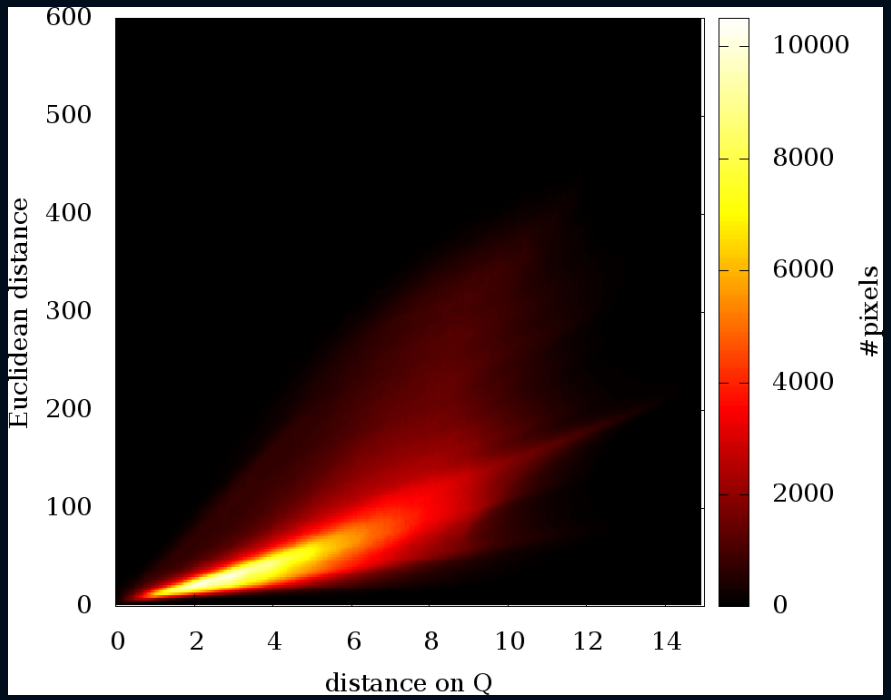
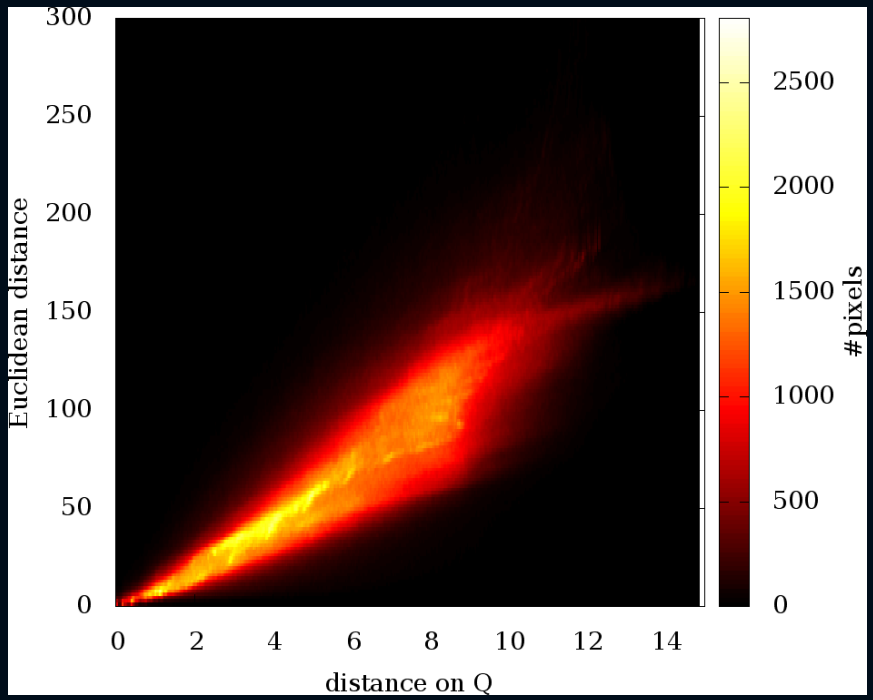


Image spectra | Huynh and Robles-Kelly | Our method

Results: Dimensionality Reduction

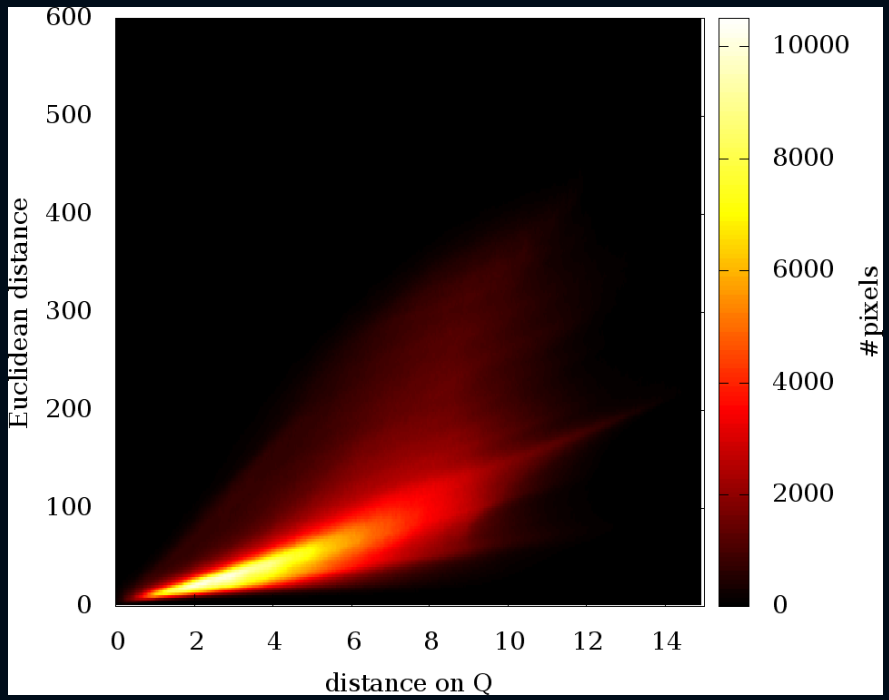


`Cbrufefields` (Foster dataset)

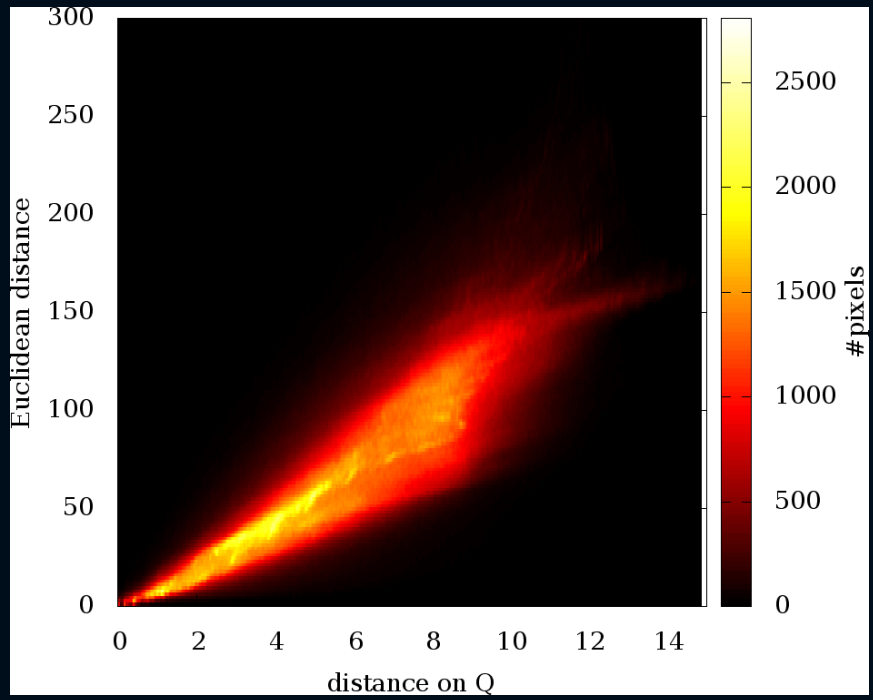


`D.C. Mall` (Remote sensing)

Results: Dimensionality Reduction



`Cbrufefields` (Foster dataset)



`D.C. Mall` (Remote sensing)

Timing Results

Training: **3 s**, mapping: **12 s**, clustering: **3 s**, total **17.7 s**

Conclusions

- We propose a new unsupervised statistical learning method

Conclusions

- We propose a new unsupervised statistical learning method
- We model the problem of manifold learning with two linked graphs
  1. G_M represents data samples, supports parameters ξ_j
  2. G_Q is a constructed Gibbs field, enables ξ_j inference

Conclusions

- We propose a new unsupervised statistical learning method
- We model the problem of manifold learning with two linked graphs
  1. G_M represents data samples, supports parameters ξ_j
  2. G_Q is a constructed Gibbs field, enables ξ_j inference
- Our method is shown to be effective for:
  1. material prototype recovery
  2. scene material segmentation
  3. visualisation and dimensionality reduction

Conclusions

- We propose a new unsupervised statistical learning method
- We model the problem of manifold learning with two linked graphs
  1. G_M represents data samples, supports parameters ξ_j
  2. G_Q is a constructed Gibbs field, enables ξ_j inference
- Our method is shown to be effective for:
  1. material prototype recovery
  2. scene material segmentation
  3. visualisation and dimensionality reduction
- Our method is computationally efficient in terms of computing power and memory use
- We improve over state-of-the-art in speed and performance