Image Processing

Graph-based image processing (Part 1)

Minh-Tan Pham

IRISA-UBS, Vannes, France

minh-tan.pham@univ-ubs.fr

Table of contents

Lecture overview

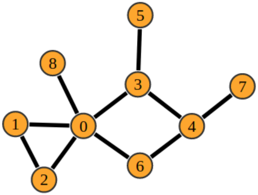
The objective of this lecture & lab (2×3 hours) is to introduce some basic tools of graph-based image analysis and processing. The lecture includes 2 parts:

- Part 1 (4h of lecture & lab):
  - General introduction of graph
  - How to construct a graph from an image?
  - Application 1: Superpixel and Graph-cut for segmentation
  - Application 2: Keypoint local graph for change detection
- Part 2 (2h of lecture & demo):
  - The spectral domain of graph
  - Application 3: Non-local graph for image denoising
  - Other applications

Table of contents

A graph $G = \{V, E\}$ consists of

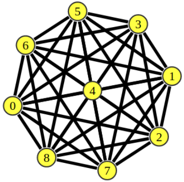
- a set of vertices (nodes) $V = \{v_i; i = 1, \ldots, |V|\}$
- a set of edges E



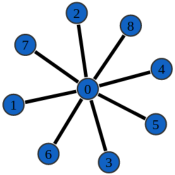
Definitions

- order of G: number of nodes $N = |V|$
- size of graph G: number of edges E
- degree of a node $d(v_i)$: number of edges connected to v_i
- partial graph $G' = \{V, E'\}$ where $E' \subseteq E$
- sub-graph $G' = \{V', E'\}$ where $V' \subseteq V$ and $E' \subseteq E$

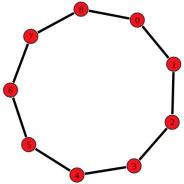
Some special graph models [1]:



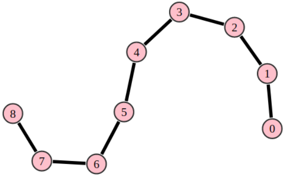
Complete graph



Star graph



Circle (ring) graph



Line graph

[1] Image source: `https://www.vincent-gripon.com/?l=en&p1=330&p2=0&`

A weighted graph $\mathcal{G} = \{V, E, w\}$ consists edges that involve a weighting function (e.g. a measure of similarity between vertices).

- $w(i,j)$:weight of an edge linking v_i and v_j
- adjacency matrix $\mathcal{W}$ (i.e. similarity matrix, matrix of weights):

$$\mathcal{W}_{ij} = \begin{cases} w\,(i,j) & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

  size of W is $N \times N$
- for undirected graphs, W is symmetric

An example of a weighted graph $\mathcal{G}$ with:

- $N = 4$ vertices, $V = \{v_1, v_2, v_3, v_4\}$
- $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_3, v_4)\}$.

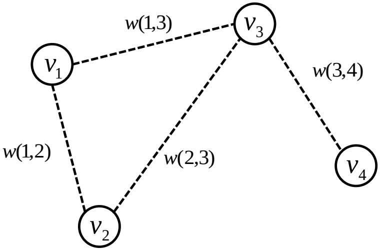


Figure: Example of a weighted, undirected and unlooped graph.

Some basic matrices in graph theory are defined:

- The degree matrix $\mathcal{D}$ which is diagonal:

$$\begin{cases} \mathcal{D}_{ii} = \sum_j \mathcal{W}_{ij}, \\ \mathcal{D}_{ij} = 0, \forall i \neq j. \end{cases} \tag{2}$$

- The Laplacian matrix:

$$\mathcal{L} = \mathcal{D} - \mathcal{W}. \tag{3}$$

- The normalized Laplacian matrix:

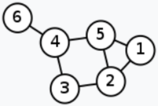
$$\mathcal{L}_{\text{nor}} = \mathcal{D}^{-\frac{1}{2}} \mathcal{L} \mathcal{D}^{-\frac{1}{2}} = \mathcal{I} - \mathcal{D}^{-\frac{1}{2}} \mathcal{W} \mathcal{D}^{-\frac{1}{2}}, \tag{4}$$

  where $\mathcal{I}$ is the $N \times N$ identity matrix.

- The random walk matrix:

$$\mathcal{P} = \mathcal{D}^{-1} \mathcal{W}. \tag{5}$$

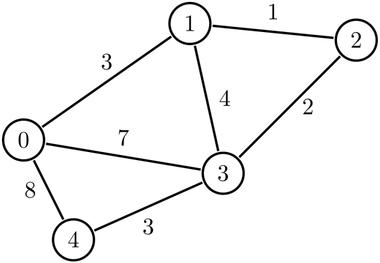
Let's take a look on the following graph[2]

Labelled graph	Degree matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Adjacency matrix	Laplacian matrix
$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

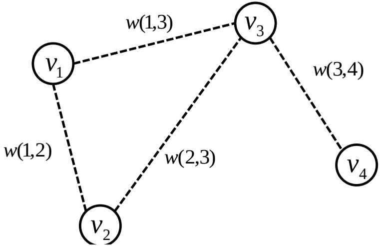
[2]Source: `https://en.wikipedia.org/wiki/Laplacian_matrix`

Now compute those matrices on the following weighted graph!



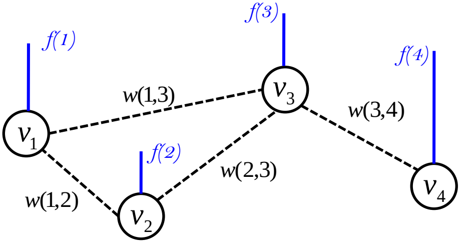
Signal on graph

So a common graph have nodes, edges and weights between edges, but not all!
The nodes also involves some features (or attributes).



Signal on graph

Let $f = \{f(n) \in \mathbb{R}; n = 1, \ldots, N\}$ be the signal or function on graph vertices called graph signal. Each node v_i encapsulates a signal sample $f(i)$ which is a feature.



A signal f on graph $\mathcal{G}$. Each vertex v_i encapsulates a signal sample $f(i)$.

Some operations with graph signals

Given a signal on graph $f \in \mathbb{R}^N$, the following equations are obtained:

$$(\mathcal{W}f)(n) = \sum_{v_m \sim v_n} w(n,m) f(m), \tag{6}$$

$$(\mathcal{P}f)(n) = \frac{1}{\sum_{v_m \sim v_n} w(n,m)} \sum_{v_m \sim v_n} w(n,m) f(m), \tag{7}$$

$$(\mathcal{L}f)(n) = \sum_{v_m \sim v_n} w(n,m) \left[f(n) - f(m) \right], \tag{8}$$

$$f^T \mathcal{L} f = \sum_{v_m \sim v_n} w(n,m) \left[f(n) - f(m) \right]^2, \tag{9}$$

where the notation $v_m \sim v_n$ means that two vertices v_m and v_n are connected (i.e. $(v_m, v_n) \in E$).

Some operations with graph signals

Remarks:

- $(\mathcal{W}f)(n)$ and $(\mathcal{P}f)(n)$ perform the concentration of information from neighboring vertices to the understudied vertex v_n
  $\Longrightarrow$ we can consider $\mathcal{W}f$ as a local filtering operator (low-pass) performed in graph vertex domain and $\mathcal{P}f$ as a normalized version of this filter.
- $\mathcal{L}f$ can be considered as a difference operator.
- $f^T\mathcal{L}f$ represents the total variation of the signal f on graph.

But yes!!! We are going to far away!!! BACK TO GRAPH FOR IMAGES.

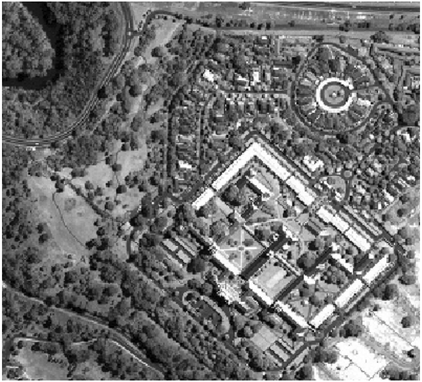
Graph basis

Brief summary:

- Graph has node, edge, edge weight and node features (modeled by a signal)
- Graph is modeled by different matrices: adjacency $\mathcal{W}$, laplacian $\mathcal{L}$, degree $\mathcal{D}$, random walk $\mathcal{P}$
- and also not feature matrix $\mathcal{F}$ *(not introduced yet but you know that it exists)!!!!*
- Given a weighted graph with signal, we could play with:
  - spatial domain (nodes, edges)
  - spatial + feature (signal)
  - and yes!!! spectral domain (see later in this lecture)

Table of contents

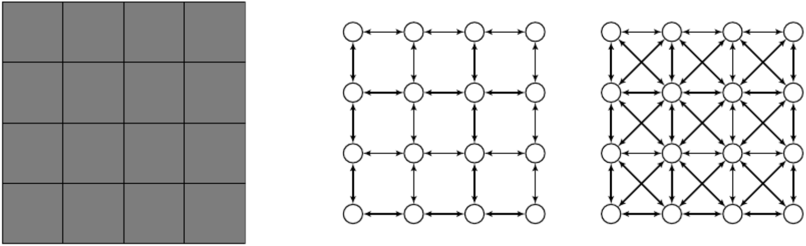
Given an image, how could we model it with a graph??? [3]



[3] Image source: PhD of M-T Pham

Regular graph with pixel adjacency

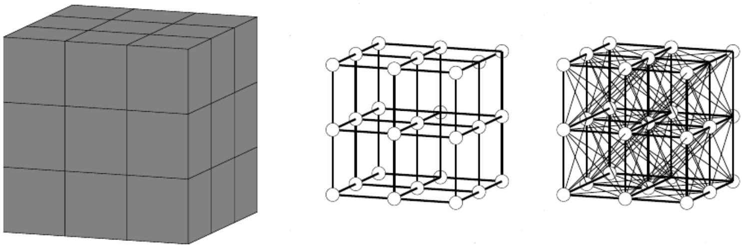
- *Pixel adjacency graph*
- Each pixel is a node
- 4-connected or 8-connected edges[4]



[4] Image source: `http://www.cb.uu.se/~filip/ImageProcessingUsingGraphs/`

Regular graph with pixel adjacency

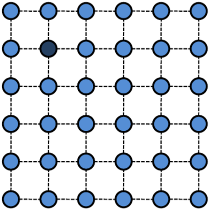
- *Pixel adjacency graph*
- Each pixel is a node
- Extension on 3D images [5]

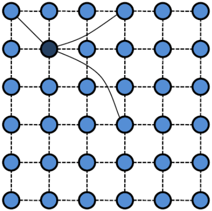


[5] Image source: `http://www.cb.uu.se/~filip/ImageProcessingUsingGraphs/`

Regular graph with non-local edges

- Each pixel is a node
- 4-connected for locally regular links
- some non-neighborhood nodes could be linked[6]

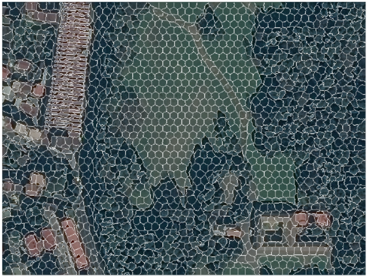




[6] Image source: PhD of M-T Pham

Irregular graph with region adjacency

- *Region adjacency graph (RAG)*
- Over-segment image into super-pixels
- Each super-pixel = node
- Edge linking neighboring super-pixels[7]



Superpixel segmentation

[7] Image source: PhD of M-T Pham

What are super-pixels???? (just in case!!!)

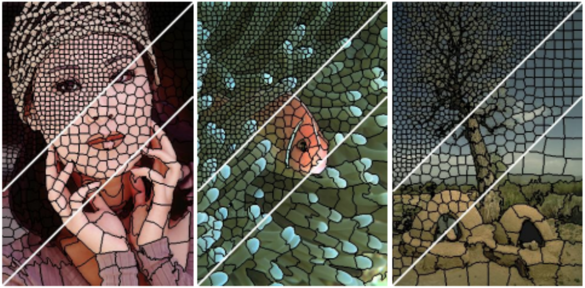


Image segmented into superpixels of (approximate) size 64, 256, and 1024 pixels. The superpixels are compact, uniform in size, and adhere well to region boundaries[8]

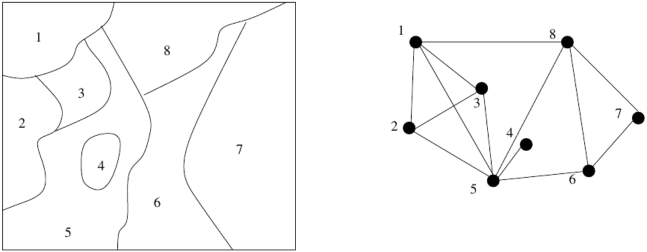
Let's understand the SLIC generation here [9]

[8] Image source: SLIC superpixels compared to state-of-the-art superpixel methods, IEEE PAMI 2012

[9] https://darshita1405.medium.com/superpixels-and-slic-6b2d8a6e4f08

Irregular graph with region adjacency

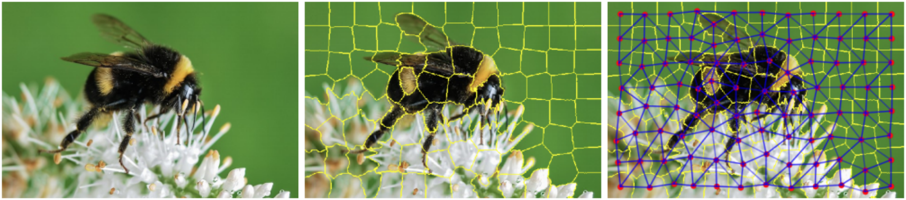
- *Region adjacency graph (RAG)*
- Over-segment image into super-pixels
- Each super-pixel = node
- Edge linking neighboring super-pixels [10]



RAG (Region Adjacency Graph)

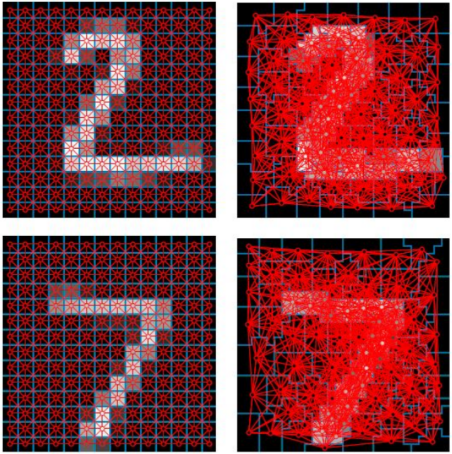
[10] Image source: `https://perso.telecom-paristech.fr/tupin/cours/AIC/graphes.pdf`

Examples of RAGs with super-pixels [11]



[11] Image source: Superpixel Image Classification with Graph Attention Networks, https://arxiv.org/pdf/2002.05544.pdf

Examples of Regular vs Irregular Graphs [12]



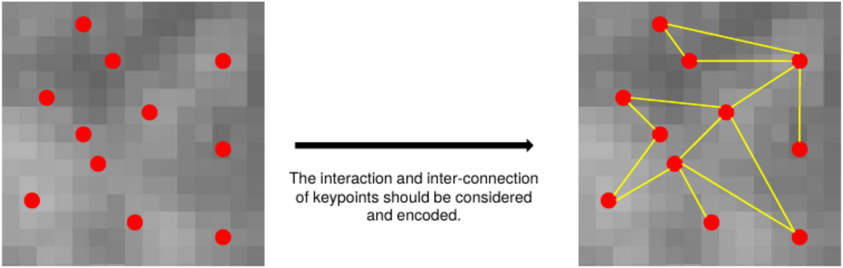
Regular grid Superpixels

[12] Image source: Master Thesis of Anthony Frion (IRISA, 2021)

Graph for images Irregular graph

Irregular graph with keypoint adjacency

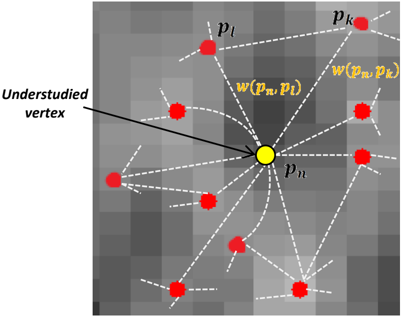
- Keypoint detection/extraction from image
- Each keypoint = node
- Edge linking each keypoint to *K* closest neighbors [13]



[13] Image source: PhD of M-T Pham

Irregular graph with keypoint adjacency

- Keypoint detection/extraction from image
- Each keypoint = node
- Edge linking each keypoint to K closest neighbors [14]

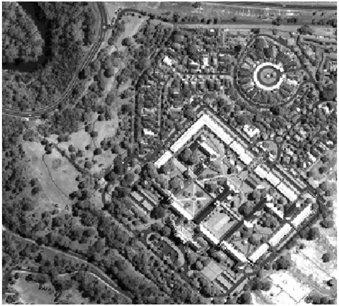


[14] Image source: PhD of M-T Pham

Graph for images

Brief summary: To model an image using graph

- Node = pixel, region (super-pixel), keypoint
- Edge = local, non-local links
- And...node features = color/spectral information (red, green, blue, infrared, etc.), spatial/texture (shape, size, heterogeneity, etc.)



Exercise (for reminder)

Giving the following super-pixel map, provide the corresponding adjacency matrix. Then calculate the degree matrix as well as the Laplician matrix.

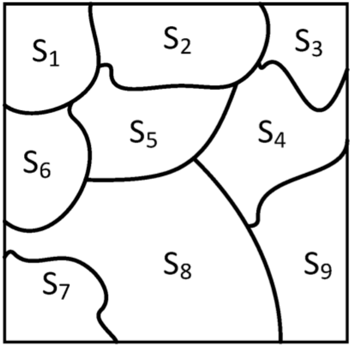


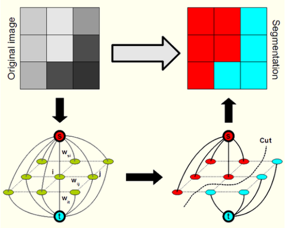
Table of contents

Graph-cut

Graph-cut is one of the popular tools for image segmentation. [15]

- Model image as a graph (pixel adjacency or region adjacency)
- Cut = a partition of graph vertices into two disjoint subsets
- Goal: minimize the cost of cut

(more details will be given in a dedicated lecture of Segmentation)

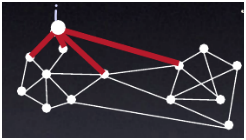


[15] Image source: https://github.com/AmarJ/GraphCut

Graph-cut Definitions[16]

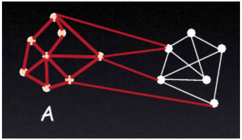
Node degree (reminder)

$d(i) = \sum_{j \sim i} w(i,j)$



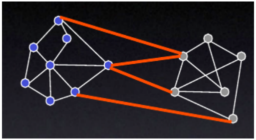
Volume of set

$vol(A) = \sum_{i \in A} d(i), A \subseteq V$



Cost of a cut

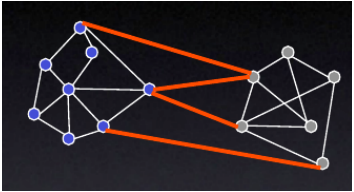
$cut(A,B) = \sum_{i \in A, j \in B} w(i,j)$



[16] Image source: http://lagis-vi.univ-lille1.fr/ lm/classpec/

Criteria: By minimizing the cost of cut, we can optimally bi-partition the graph into 2 subsets (corresponding to 2 image segments)[17].

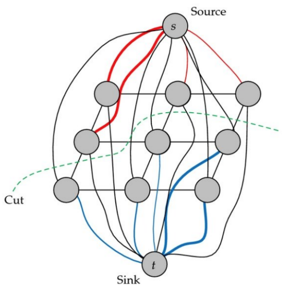
$$\min cut(A,B) = \min \sum_{i \in A, j \in B} w(i,j)$$



[17] Image source: http://lagis-vi.univ-lille1.fr/ lm/classpec/

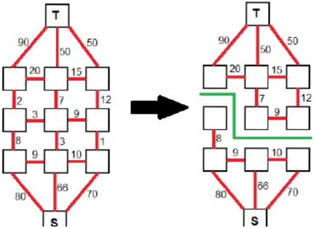
Algorithm Min-cut Max-flow (Ford & Flkerson, 1956)

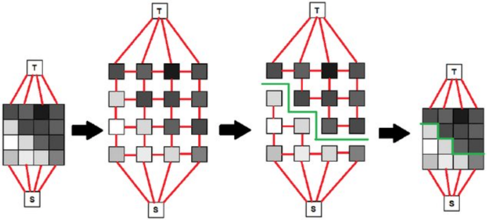
- Given a source (s) and a sink node (t)
- Consider water flow from s to t, supposing the edge weight as the capacity of pipes
- Min-cuts are given by the saturated edges for maximum flows [18].



[18] Image source: Chen et al., Markov Models for Image Labeling, Mathematical Problems in Engineering, 2012

Algorithm Min-cut Max-flow (Ford & Flkerson, 1956)

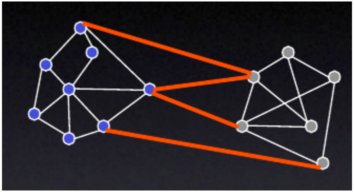




Problem of Min cut minCut often performs cutting isolated nodes in the graph due to the small values achieved by partitioning such nodes!!!

Normalized cut = balanced cut computes the cut cost as a fraction o the total edge connections to all graph nodes[19].

$$Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$



[19] Image source: http://lagis-vi.univ-lille1.fr/ lm/classpec/

10 minute reading: Interactive graph-cut segmentation
`https://www.datasciencecentral.com/profiles/blogs/interactive-image-segmentation-with-graph-cut-in-python`

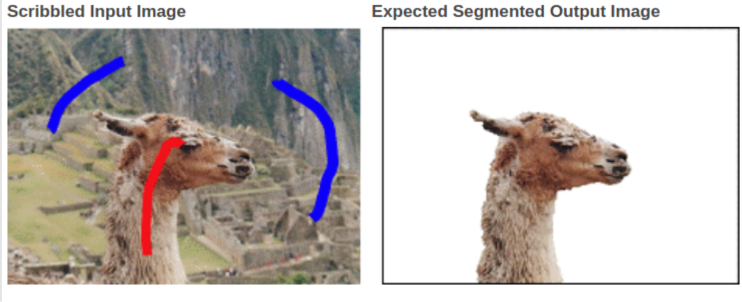
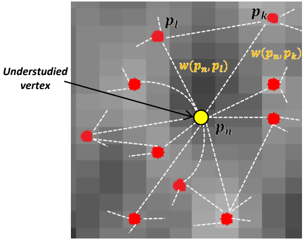


Table of contents

Keypoint graph for change detection[20]

Graph-based approach:

- Relevant to encode keypoint interaction
- Vertex domain $\Rightarrow$ allow to study the concentration/diffusion of information
- Graph construction: each vertex connects to its K-closest neighbors



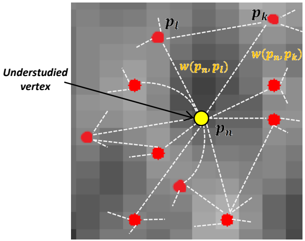
$$(\mathcal{W}f)(p_n) = \sum_{p_k \sim p_n} w(p_n, p_k) f(p_k)$$

[20] Image source: PhD of M-T Pham

Keypoint graph for change detection[20]

Graph-based approach:

- Relevant to encode keypoint interaction
- Vertex domain ⇒ allow to study the concentration/diffusion of information
- Graph construction: each vertex connects to its K-closest neighbors



$$(\mathcal{W}f)(p_n) = \sum_{p_k \sim p_n} w(p_n, p_k) f(p_k)$$

For each understudied vertex, the local graph topology:

- encode radiometric information and geometric structure of its local environment
- allow to study the coherence/compatibility of information

[20] Image source: PhD of M-T Pham

Keypoint graph for change detection[21]

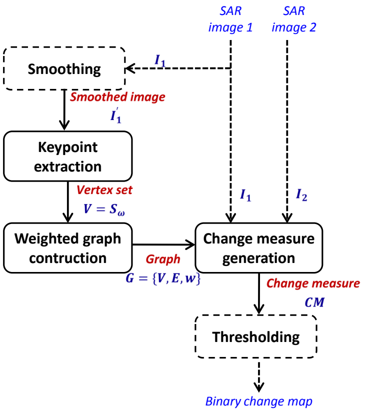
Proposed framework

- Construct a graph $\mathcal{G}$ to connect keypoints from I_1
  - vertex set = keypoin set
  - each vertex connects to its K closest neighbors
  - weight computation:

$$w(p_n, p_k) = e^{-\gamma\left[\mathrm{dist}_{\mathrm{LR}}(p_n, p_k)\right]}$$

$$\mathrm{dist}_{\mathrm{LR}}(p_n, p_k) = \left|\log \frac{\mu_1(p_n)}{\mu_1(p_k)}\right|$$

$\mu_1(p_n)$ (resp. $\mu_1(p_k)$): the mean intensity of small pixel patch



[21] Image source: PhD of M-T Pham

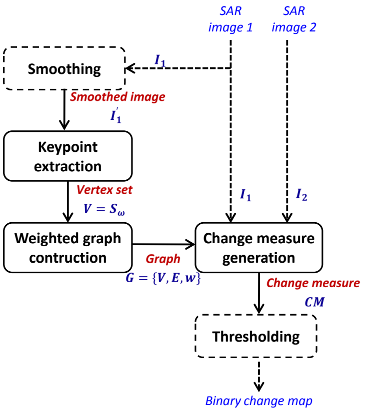
Keypoint graph for change detection[21]

Proposed framework

Change measure

$\Updownarrow$

coherence/compatibility of radiometric information from I_1 and I_2 captured by structure of $\mathcal{G}$.



[21] Image source: PhD of M-T Pham

Keypoint graph for change detection[21]

Proposed framework

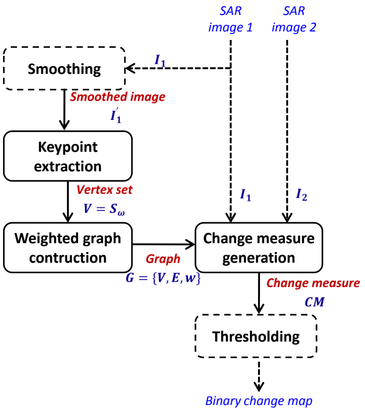
- 2 signals on graph:

$$f_1 = [\log \mu_1(p_1), \log \mu_1(p_2), \ldots, \log \mu_1(p_N)]^T$$
$$f_2 = [\log \mu_2(p_1), \log \mu_2(p_2), \ldots, \log \mu_2(p_N)]^T$$

(2 images I_1 and I_2 diffuse their information on $\mathcal{G}$)

- Change measure at each keypoint:

$$CM(p_n) = \left|\left|(\mathcal{W}f_1)(p_n) - (\mathcal{W}f_2)(p_n)\right|\right|_1$$
$$= \sum_{p_k \sim p_n} w(p_n, p_k) \left| \log \frac{\mu_1(p_k)}{\mu_2(p_k)} \right|$$



[21] Image source: PhD of M-T Pham

Keypoint graph for change detection[21]

Proposed framework

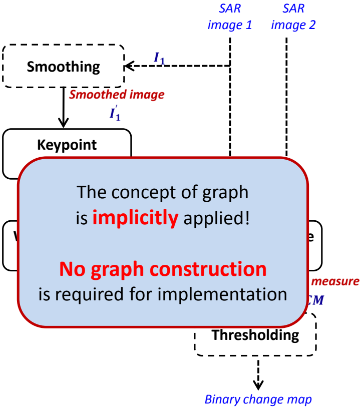
- 2 signals on graph:

$$f_1 = [\log \mu_1(p_1), \log \mu_1(p_2), \ldots, \log \mu_1(p_N)]^T$$
$$f_2 = [\log \mu_2(p_1), \log \mu_2(p_2), \ldots, \log \mu_2(p_N)]^T$$

(2 images I_1 and I_2 diffuse their information on $\mathcal{G}$)

- Change measure at each keypoint:

$$CM(p_n) = \left|\left|(\mathcal{W}f_1)(p_n) - (\mathcal{W}f_2)(p_n)\right|\right|_1$$
$$= \sum_{p_k \sim p_n} w(p_n, p_k) \left| \log \frac{\mu_1(p_k)}{\mu_2(p_k)} \right|$$



The concept of graph is **implicitly** applied!

No graph construction is required for implementation

[21] Image source: PhD of M-T Pham

Keypoint graph for change detection

10 minute reading: Pham, Minh-Tan et al. **Change detection between SAR images using a pointwise approach and graph theory.** IEEE Transactions on Geoscience and Remote Sensing 54.4 (2015): 2020-2032.

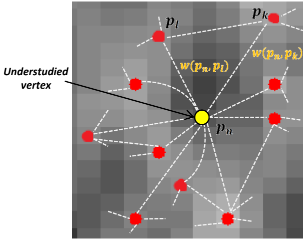


Table of contents

Assignment (pair work)

Graph-based methods (non deep learning) applied to remote sensing

- Search and read, discuss and summarize two papers using graph-based methods applied to remote sensing data (optical, SAR, hyperspectral, Lidar)
- Provide a 2- or 3-page short report including, for each paper:
  - summary of the proposed methodology and the obtained results
  - which techniques you have learned/known from the Computer Vision course
  - pros/cons of the proposed approaches

E.g. Hierarchical graph-based segmentation for extracting road networks from high-resolution satellite images

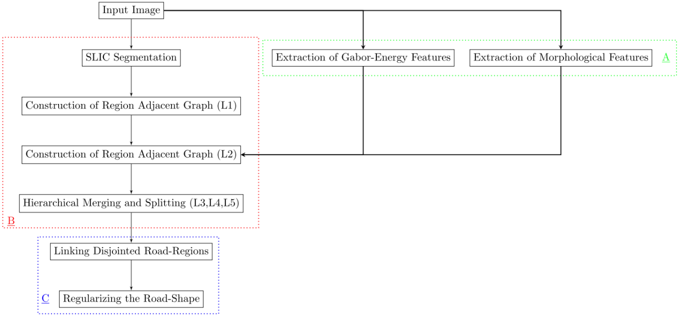


Fig. 1. Flowchart of the proposed method. It consists of (A) Pre-processing, (B) Graph-based segmentation, and (C) Post-processing.

Lab

Graph construction and graph-cut segmentation with skimage library on Colab

- Download the notebook file and images from Moodle
- Exo 1: Regular graph construction (pixel adjacency graph)
- Exo 2: Irregular graph construction from super-pixels
- Exo 3: Graph-cut for segmentation

References I

Books & articles:

1. Lézoray, Olivier, and Leo Grady, eds. **Image processing and analysis with graphs: theory and practice.** CRC Press, 2012.
2. Achanta, Radhakrishna, et al. **SLIC superpixels compared to state-of-the-art superpixel methods.** IEEE PAMI 34.11 (2012): 2274-2282.
3. Pham, Minh-Tan, et al. **Change detection between SAR images using a pointwise approach and graph theory.** IEEE TGRS 54.4 (2015): 2020-2032.
4. Chen et al., **Markov Models for Image Labeling**, Mathematical Problems in Engineering, 2012

Courses:

1. Image Processing using Graphs - Centre for Image Analysis. `http://www.cb.uu.se/~filip/ImageProcessingUsingGraphs/`
2. Tutorial Graph Based Image Segmentation. `http://lagis-vi.univ-lille1.fr/lm/classpec/`
3. Graphs for image processing, analysis and pattern recognition `https://perso.telecom-paristech.fr/tupin/cours/AIC/graphes.pdf`