# Embedding and Spectrum of Graphs

Ali Shokoufandeh,

Department of Computer Science, Drexel University

# Overview

Approximation Algorithms

Geometry of Graphs and Graphs Encoding the Geometry

Spectral Graph Theory

# Algorithmic Graph Theory:

- **Objective:** Designing efficient combinatorial methods for solving decision or optimization problems.
  - Runs in polynomial number of steps in terms of size of the graph; $\boldsymbol{n}=|\boldsymbol{V}(\mathbf{G})|$ and $\boldsymbol{m}=|\boldsymbol{E}(\boldsymbol{G})|$.
  - Optimality of solution.

- **Bad news:** most of the combinatorial optimization problems involving graphs are computationally intractable:
  - traveling salesman problem, maximum cut problem, independent set problem, maximum clique problem, minimum vertex cover problem, maximum independent set problem, multidimensional matching problem,…

# Algorithmic Graph Theory:

- **Dealing with the intractability:**
  - Bounded approximation algorithms
  - Suboptimal heuristics.

# Algorithmic Graph Theory:

**Bounded approximation algorithms**

- Example: Vertex cover problem:
  - **A *vertex cover* of an undirected graph $G=(V,E)$ is a subset $V'$ of $V$ such that if $(u,v)$ is an edge in $E$, then $u$ or $v$ (or both) belong to $V'$.**

# Algorithmic Graph Theory:

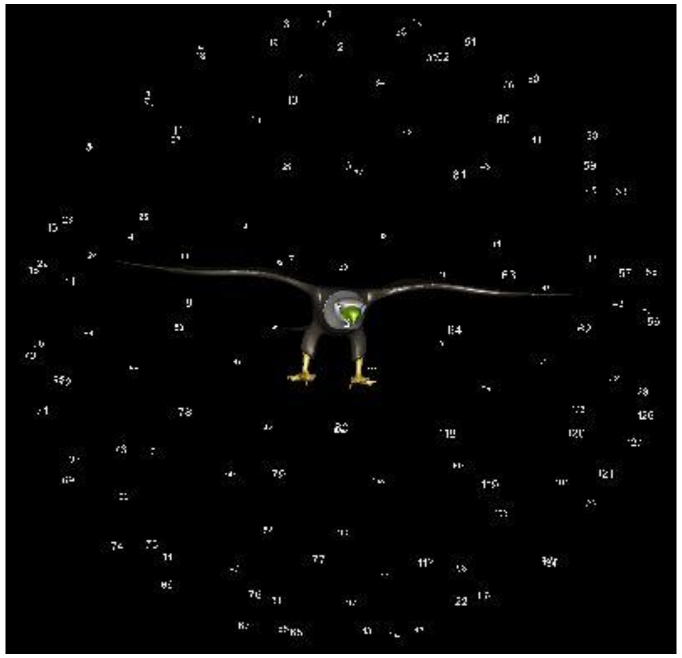
**Bounded approximation algorithms**

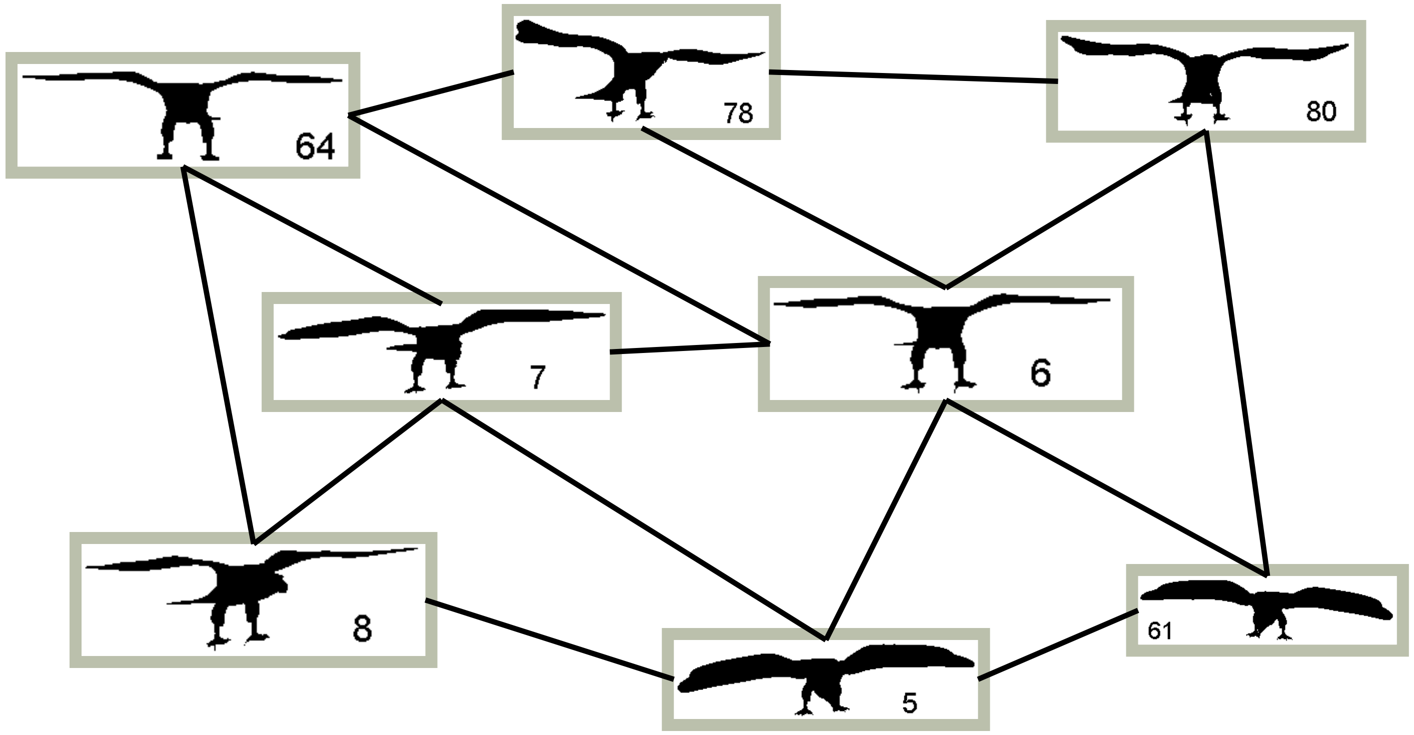
- Example: Vertex cover problem:
  - A ***vertex cover*** of an undirected graph ***G***=(***V,E***) is a subset ***V'*** of ***V*** such that if (***u,v***) is an edge in ***E***, then ***u*** or ***v*** (or both) belong to ***V'***.
  - **The *vertex cover problem* is to find a vertex cover of minimum size in a given undirected graph.**

# Algorithmic Graph Theory:

**Bounded approximation algorithms**

- Example: Vertex cover problem:
  - A ***vertex cover*** of an undirected graph $G=(V,E)$ is a subset $V'$ of $V$ such that if $(u,v)$ is an edge in $E$, then $u$ or $v$ (or both) belong to $V'$.
  - The ***vertex cover problem*** is to find a vertex cover of **minimum** size in a given undirected graph.

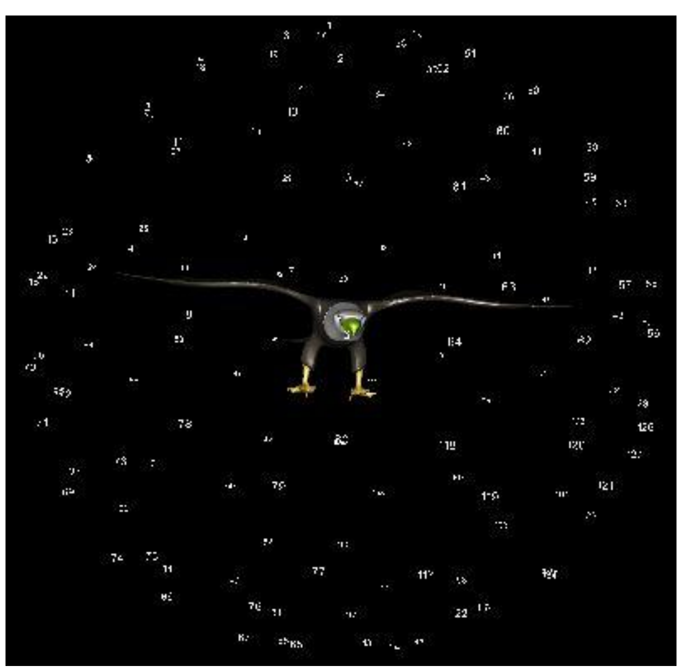


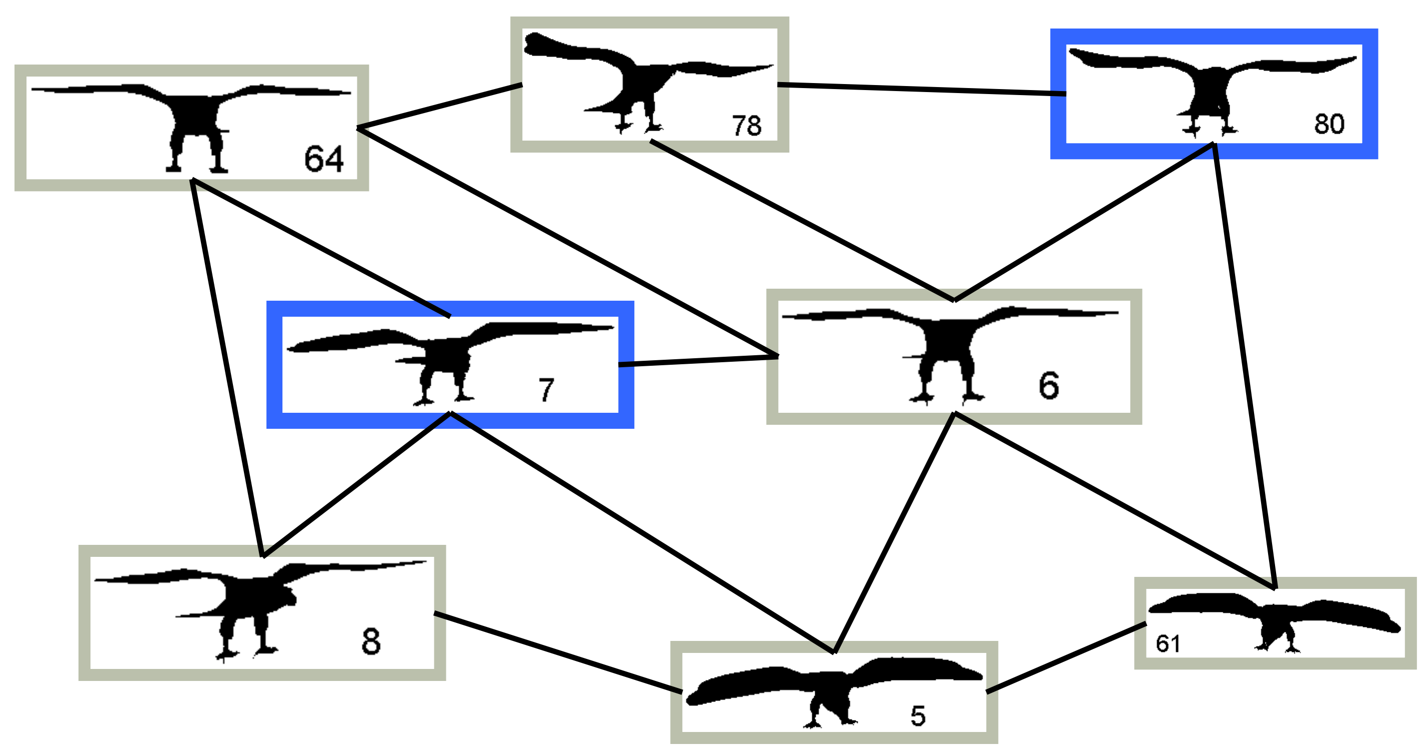


# Algorithmic Graph Theory:

**Bounded approximation algorithms**

- Example: Vertex cover problem:
  - A ***vertex cover*** of an undirected graph ***G***=(***V,E***) is a subset ***V'*** of ***V*** such that if (***u,v***) is an edge in ***E***, then ***u*** or ***v*** (or both) belong to ***V'***.
  - The ***vertex cover problem*** is to find a vertex cover of **minimum** size in a given undirected graph.





# Algorithmic Graph Theory:

## Bounded approximation algorithms

- Example: Vertex cover problem:
  - A ***vertex cover*** of an undirected graph ***G***=(***V,E***) is a subset ***V'*** of ***V*** such that if (***u,v***) is an edge in ***E***, then ***u*** or ***v*** (or both) belong to ***V'***.
  - The size of a vertex cover is the number of vertices in it.
  - The ***vertex cover problem*** is to find a vertex cover of **minimum** size in a given undirected graph.
  - **We call such a vertex cover an *optimal vertex cover.***
  - **The vertex cover problem was shown to be NP-complete.**

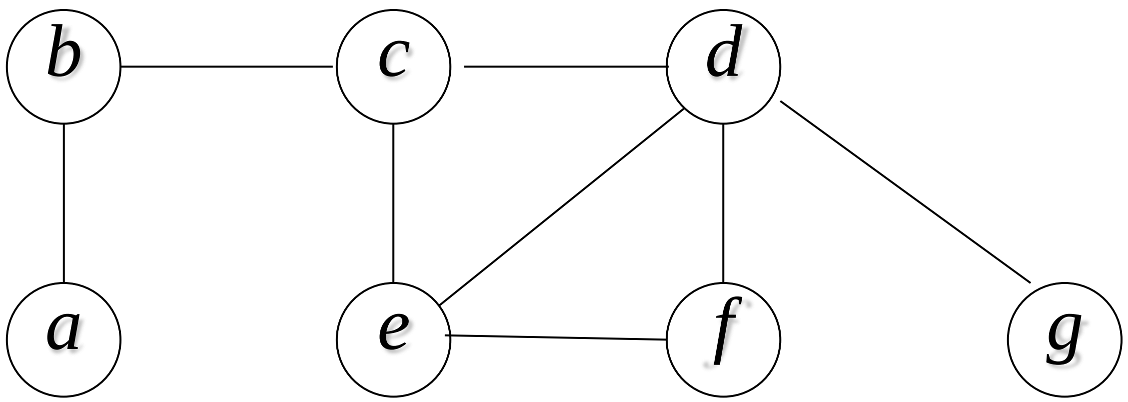
# Algorithmic Graph Theory:

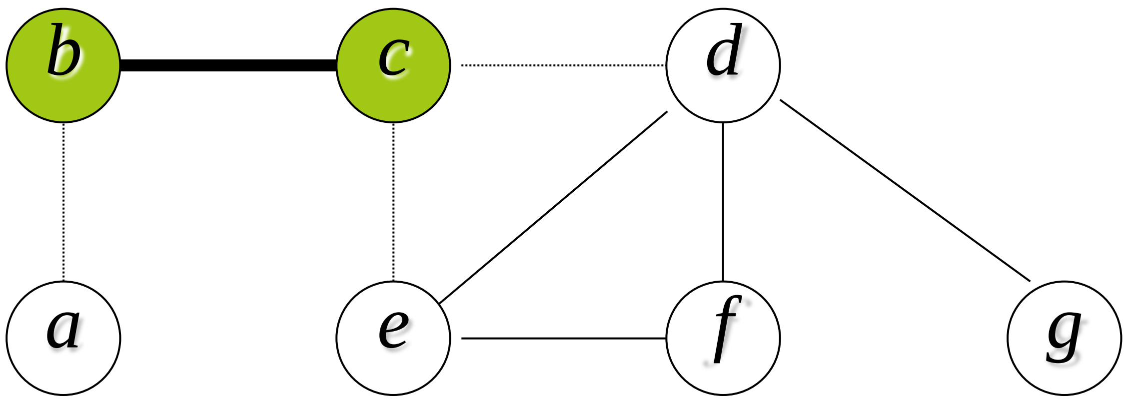
**Vertex cover problem:**

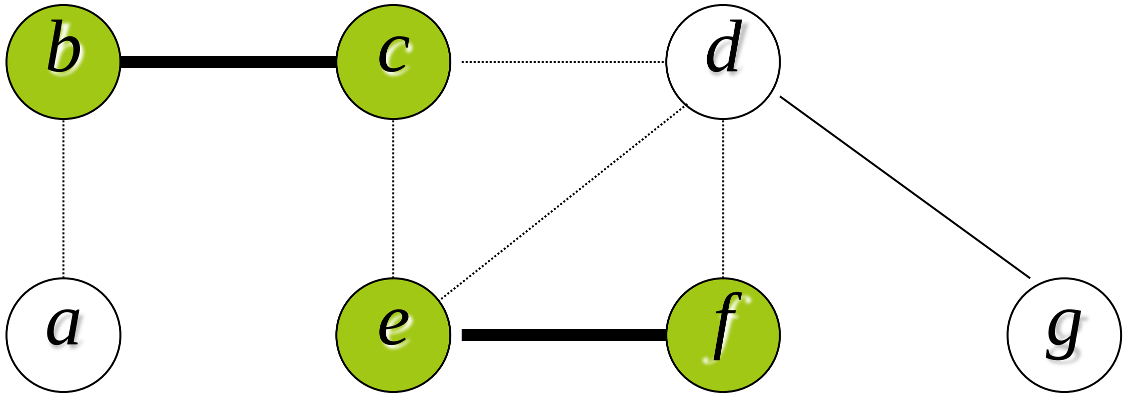
- The following approximation algorithm takes as input an undirected graph $G$ and returns a vertex cover whose size is guaranteed no more than twice the size of optimal vertex cover:

1. $C \leftarrow \emptyset$
2. $E' \leftarrow E[G]$
3. While $E' \neq \emptyset$ do
4. Let $(u, v)$ be an arbitrary edge in $E'$
5. $C \leftarrow C \cup \{u, v\}$
6. Remove from $E'$ every edge incident on either $u$ or $v$
7. Return $C$

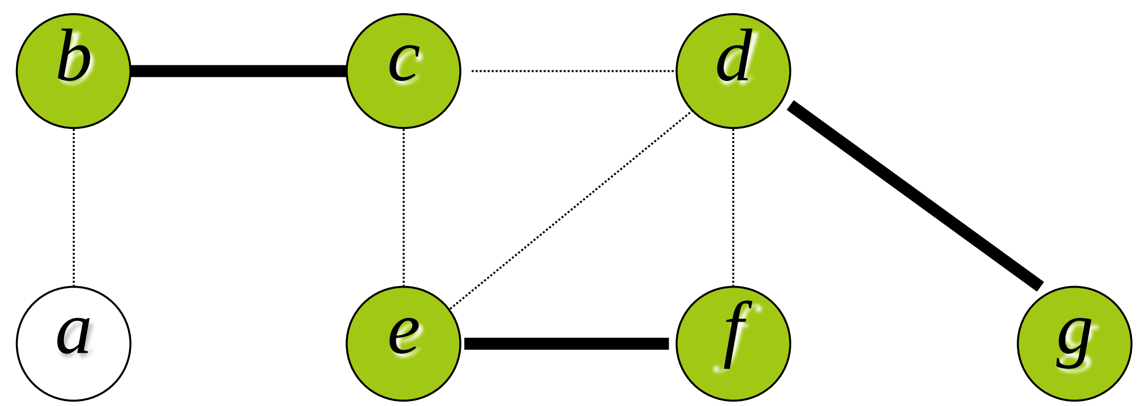
# Algorithmic Graph Theory:

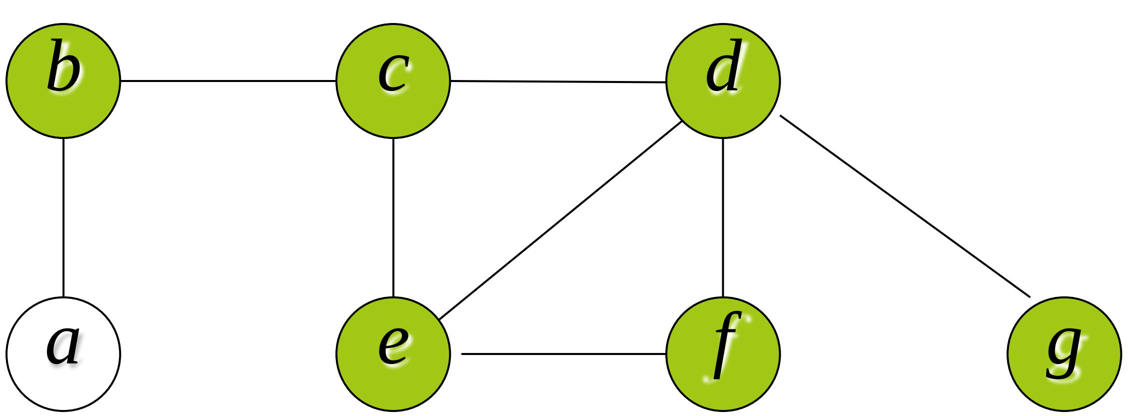


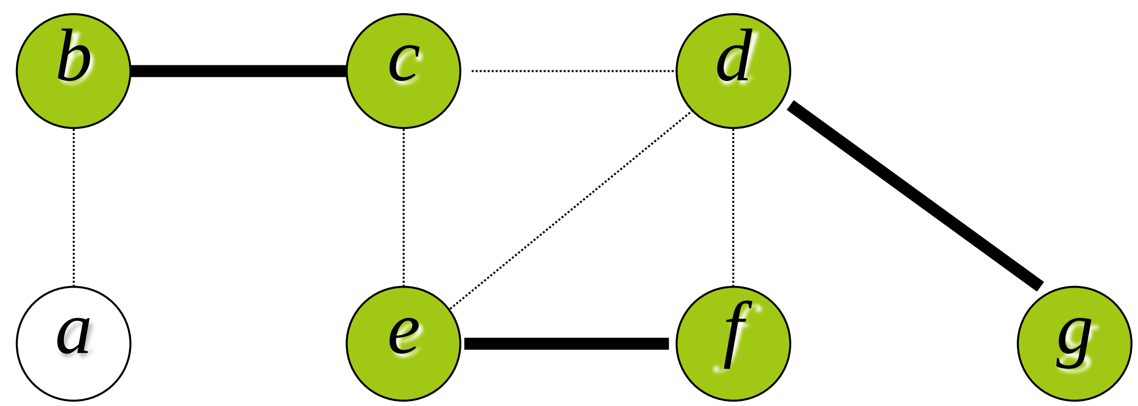



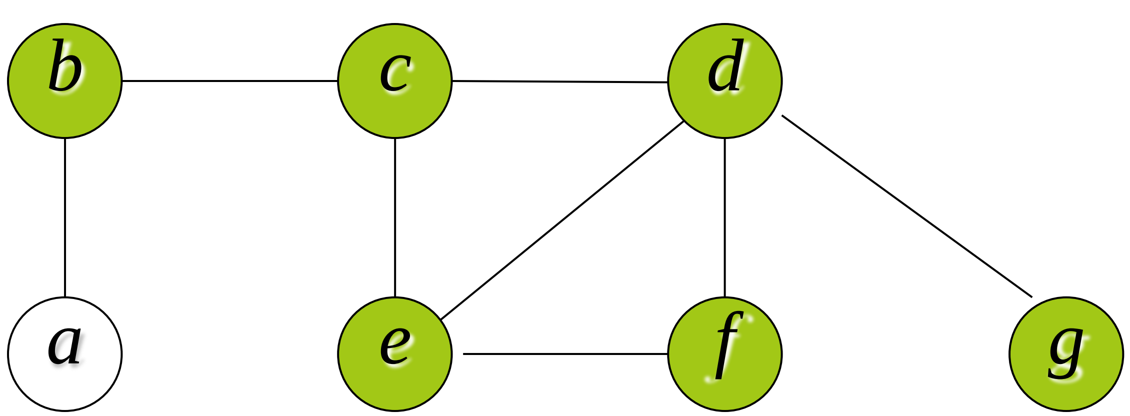
# The Vertex Cover Problem

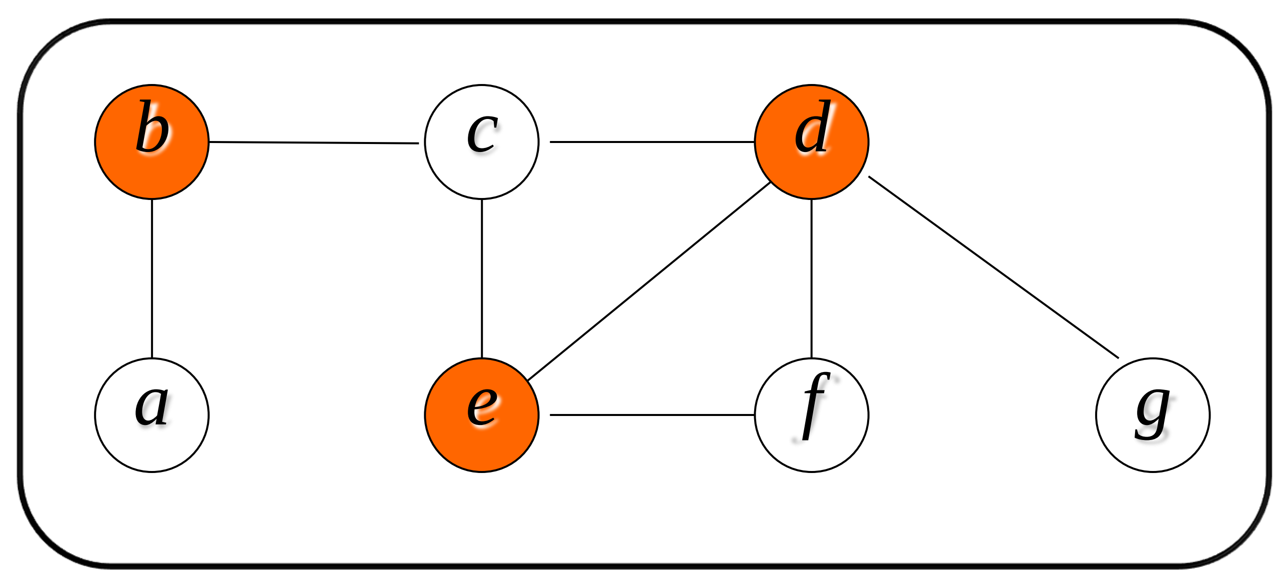


# The Vertex Cover Problem



# Algorithmic Graph Theory:

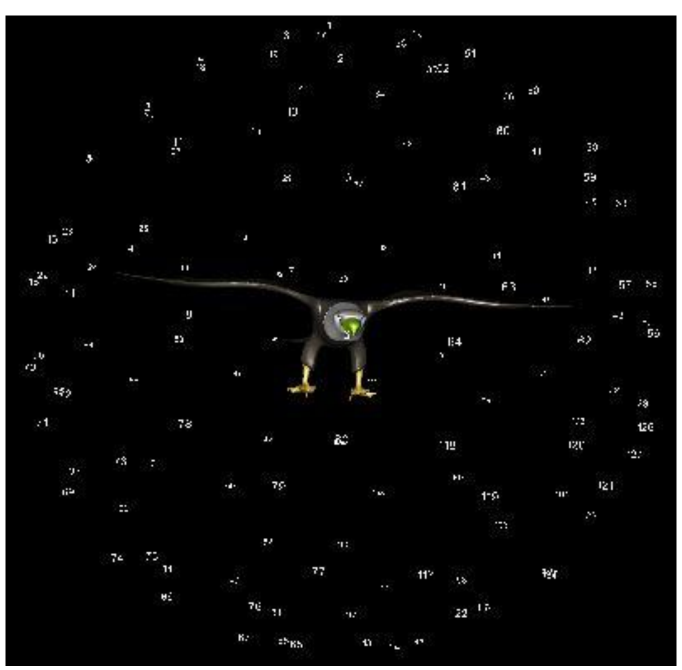
**Theorem**: Approximate vertex cover has a ratio bound of 2.

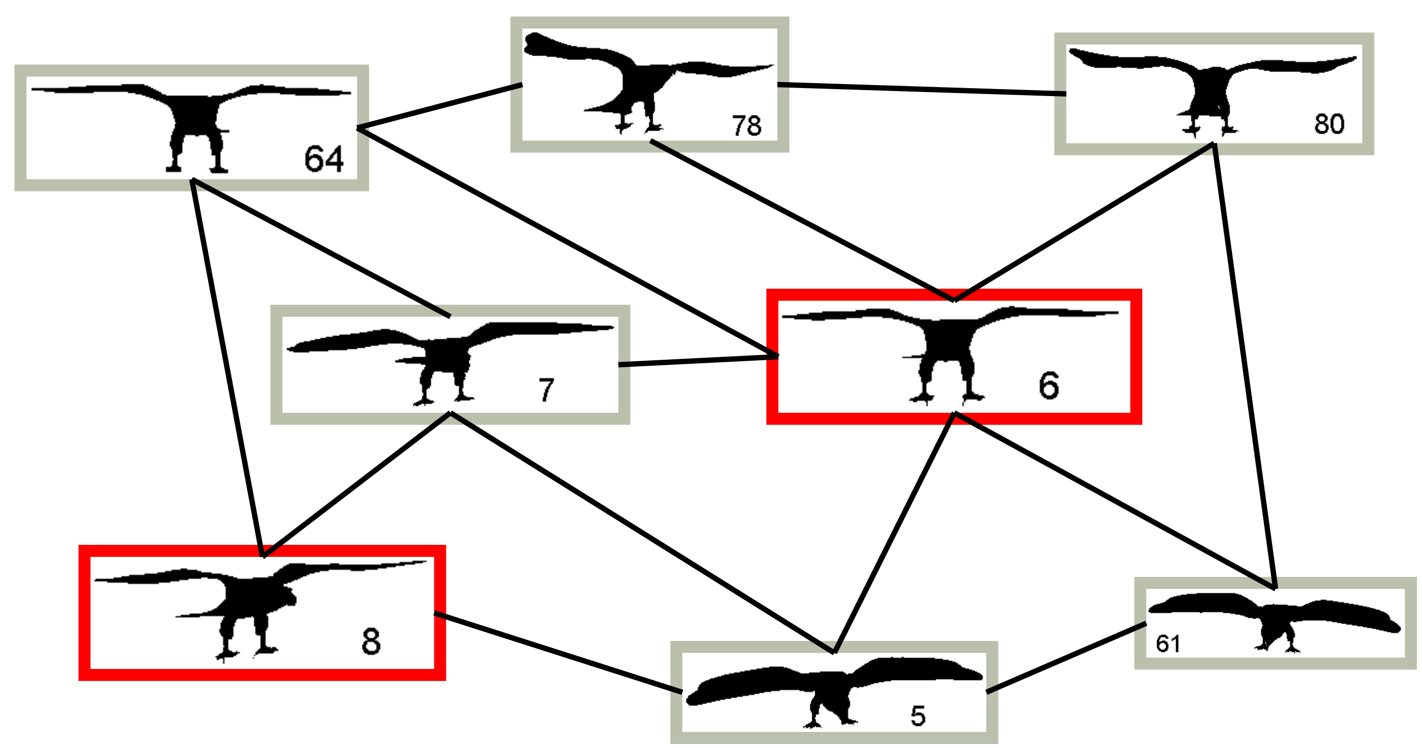
- ***Proof***:
  - It is easy to see that ***C*** is a vertex cover.
  - To show that the size of ***C*** is twice the size of optimal vertex cover.
  - Let ***A*** be the set of edges picked in line 4 of algorithm.
  - No two edges in ***A*** share an endpoint, therefore each new edge adds two new vertices to ***C***, so |***C***|=**2**|***A***|.
  - Any vertex cover should cover the edges in ***A***, which means at least one of the end points of each edge in *A* belongs to ***C**.**
  - So, |***A***|<=|***C****|, which will imply the desired bound.

# Algorithmic Graph Theory:

**Bounded approximation algorithms**

- Example: Vertex cover problem:
  - A ***vertex cover*** of an undirected graph ***G***=(***V,E***) is a subset ***V'*** of ***V*** such that if (***u,v***) is an edge in ***E***, then ***u*** or ***v*** (or both) belong to ***V'***.
  - The ***vertex cover problem*** is to find a vertex cover of **minimum** size in a given undirected graph.





# Overview

**Geometry of Graphs and Graphs Encoding the Geometry**

Spectral Graph Theory

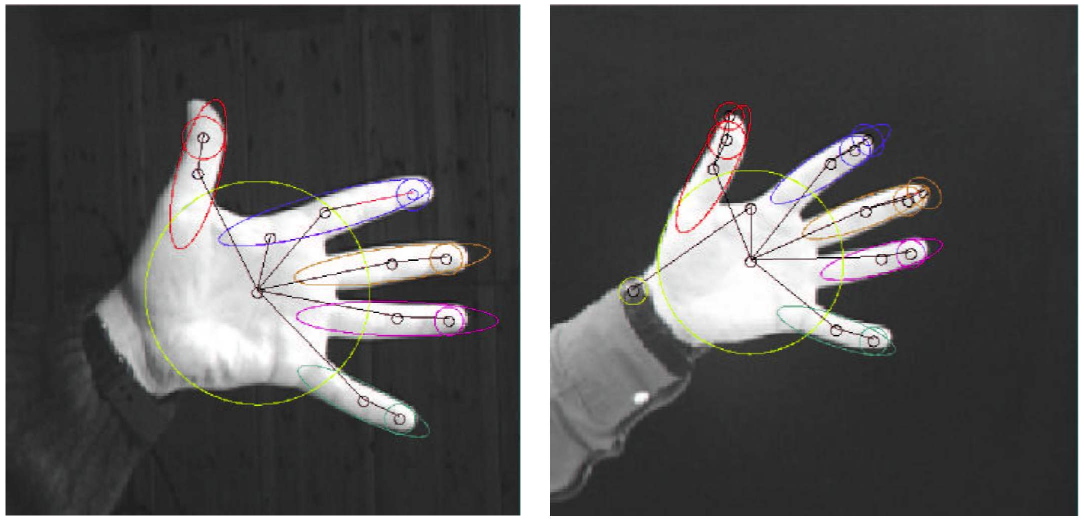
# Motivation:

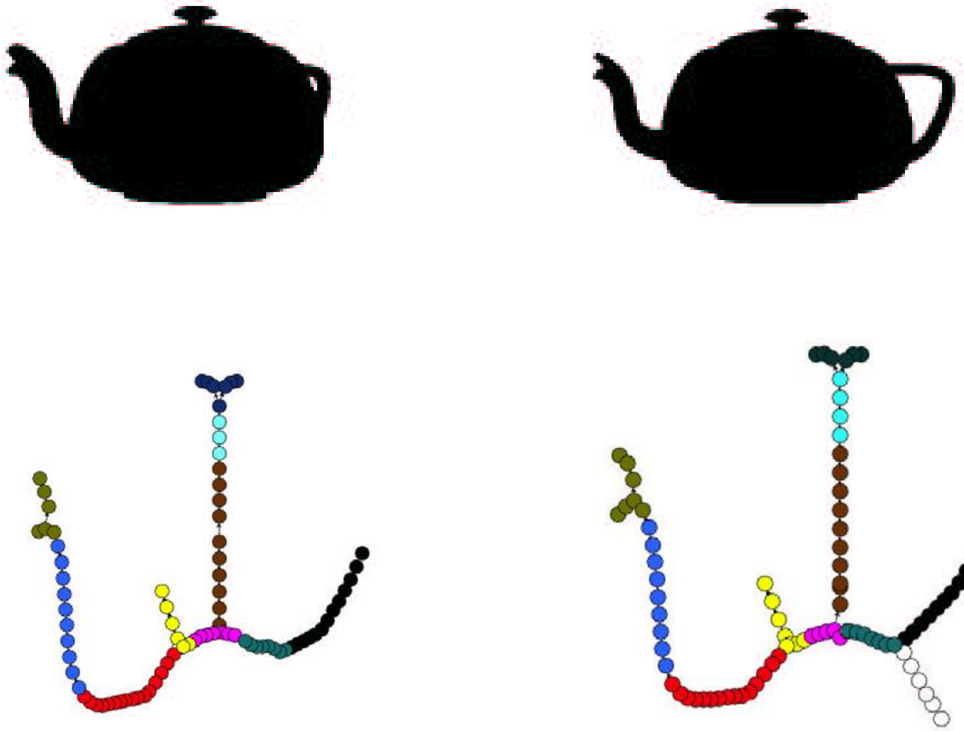
- In some scenarios geometrical problem in a finite metric space is easier to solve (approximate) than the corresponding combinatorial or optimization problem.
- Example: Many-to-many graph matching.

# Motivation:

- In some scenarios geometrical problem in a finite metric space is easier to solve (approximate) than the corresponding combinatorial or optimization problem.

- Example: Many-to-many graph matching.

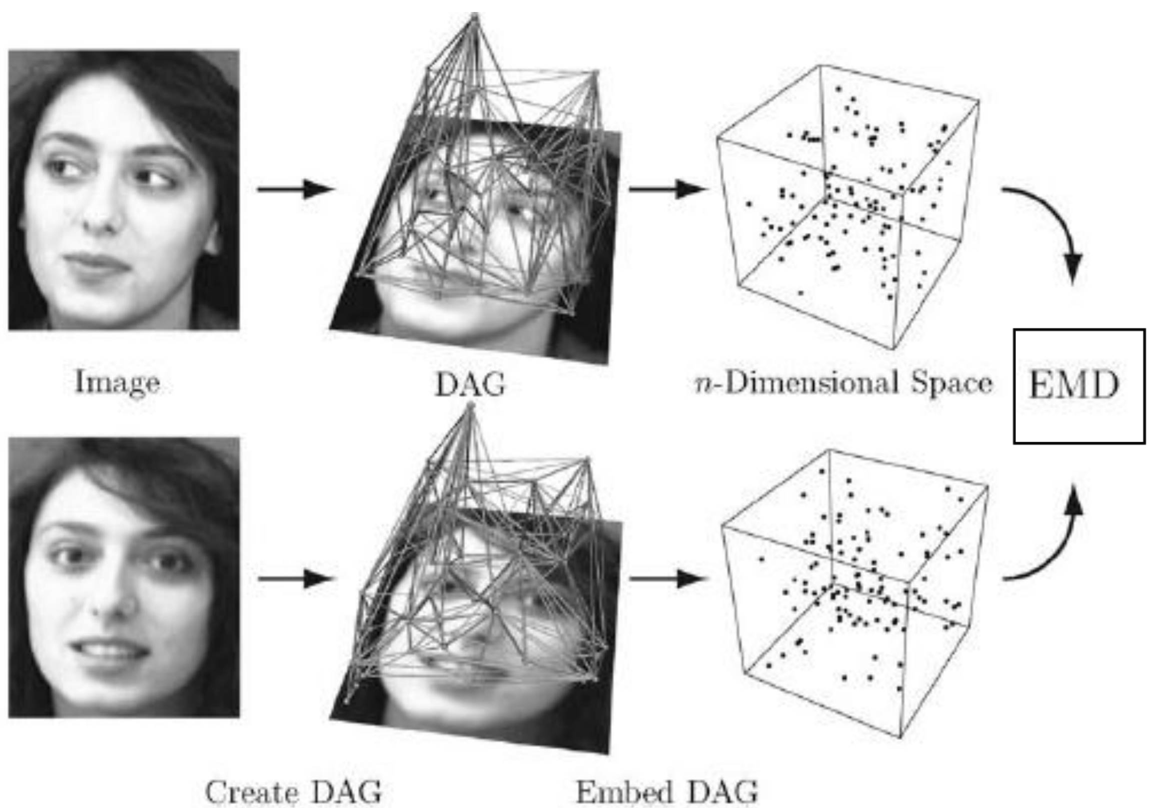




# Motivation:

- In some scenarios geometrical problem in a finite metric space is easier to solve (approximate) than the corresponding combinatorial or optimization problem.

- Example: Many-to-many graph matching.



# Some Formalities:

**(semi) metric**$(M, \rho)$: $M$ a (finite) set of points, $\rho$ a distance function satisfying for all $x, y, z$ in $M$:

- $\rho(x,x)=0$,
- $\rho(x,y)=\rho(y,x)$,
- $\rho(x,z) \leq \rho(x,y)+\rho(y,z)$.

**Embedding:** a mapping $f:(M, \rho) \rightarrow (H,v)$ of a metric space **M** into a host metric space $H$, that (possibly) preserves the geometry (distances) of $M$.

**Distortion of embedding** $f$: the least $K \geq 1$ for which exists $C > 0$ such that for all $x, y$ in $M$:

$$C \times \rho(x,y) \leq v(x,y) \leq K \times C \times \rho(x,y)$$

# Non-embedability:

- **Given:** $\boldsymbol{\rho}$ the (Shortest Path) metric of the graph $\boldsymbol{C_4}$, a cycle on four nodes.
- **Question:** Is there an isometric embedding of $\boldsymbol{C_4}$ in Euclidean space?

# Non-embedability:

- **Given:** $\boldsymbol{\rho}$ the (Shortest Path) metric of the graph $\boldsymbol{C_4}$, a cycle on four nodes.
- **Question:** Is there an isometric embedding of $\boldsymbol{C_4}$ in Euclidean space?
- **No:**
  - Denote the vertices on the $\boldsymbol{C_4}$ by $\boldsymbol{a_1}, \ldots, \boldsymbol{a_4}$.
  - Suppose an ***isometric*** embedding exists.
  - Note that $\rho(a_1, a_3) = \rho(a_1, a_2) + \rho(a_2, a_3)$, hence the triangle inequality holds with equality, which means (for Euclidean spaces) that $f(a_2)$ is in the middle of the segment $[f(a_1), f(a_3)]$.

# Non-embedability:

- **Given:** $\rho$ the (Shortest Path) metric of the graph $C_4$, a cycle on four nodes.
- **Question:** Is there an isometric embedding of $C_4$ in Euclidean space?
- **No:**
  - Denote the vertices on the $C_4$ by $a_1, \ldots, a_4$.
  - Suppose an ***isometric*** embedding exists.
  - Note that $\rho(a_1, a_3) = \rho(a_1, a_2) + \rho(a_2, a_3)$, hence the triangle inequality holds with equality, which means (for Euclidean spaces) that $f(a_2)$ is in the middle of the segment $[f(a_1), f(a_3)]$.
  - Analogously, $f(a_4)$ is in the middle of the segment $[f(a_1), f(a_3)]$.
  - Hence $f(a_2) = f(a_4)$. ➔⟵

# Non-embedability:

- **Given:** $\boldsymbol{\rho}$ the (Shortest Path) metric of the graph $\boldsymbol{C_4}$, a cycle on four nodes.
- **Question:** Is there an isometric embedding of $\boldsymbol{C_4}$ in Euclidean space?
  - **No.**
- Embedding of $\boldsymbol{C_4}$ as a square in the plain is the best embedding in Hilbert space, (distortion= $\sqrt{\mathbf{2}}$).

# Example Application:

**Sparsest Cut and Flux Minimization Problem:**

- A cut in graph ***G*** = (***V,E***) is a partition of ***V*** into two nonempty subsets ***A*** and ***B***=***V***-***A***.
- The density or flux of the cut (***A***,***B***) is

$$\mathrm{Y}(A,B) = \frac{e(A,B)}{|A| \times |B|}$$

where ***e***(***A***,***B***) is the number (or the weight) of edges crossing the cut.

- The sparsity of an (***A,B***)-cut will be defined as

$$a(A,B) = \frac{e(A,B)}{\min\left(|A|,|B|\right)}$$

# Example Application:

**Sparsest Cut and Flux Minimization Problem:**

- It is not hard to see that

$$\frac{\alpha(A,B)}{|V|} \le \Psi(A,B) \le \frac{2 \cdot \alpha(A,B)}{|V|}$$

# Example Application:

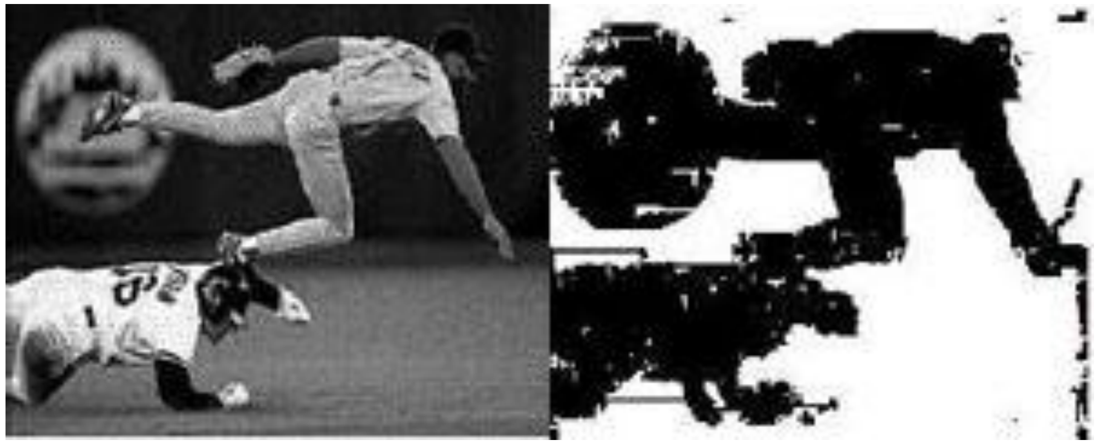
**Sparsest Cut Problem:**

- In sparsest cut problem we look for a cut of the smallest possible density.
- This problem is known to be **NP-**hard.
- As optimization problems this are **minimization** problems and intractable.

# Example Application:

**Sparsest Cut Problem:**

- In sparsest cut problem we look for a cut of the smallest possible density.
- This problem is known to be **NP-**hard.
- As optimization problems this are **minimization** problems and intractable.



Shi and Malik, 1999

# Example Application:

Flux Minimization Problem:

- The flux problem can be formulated as embedding:

  Find a mapping ***ϕ***: ***V*** ➔{0,1} that minimizes:

$$\frac{\sum_{(u,v)\in E} |\phi(u) - \phi(v)|}{\sum_{(u,v)\in V^2} |\phi(u) - \phi(v)|}$$

# Example Application:

**Flux minimization problem:**

$$\min_f \frac{\sum_{(u,v)\in E} |f(u) - f(v)|}{\sum_{(u,v)\in V^2} |f(u) - f(v)|}$$

- Simple modification of the flux formulation:
  - letting $d_{u,v} = |\phi(u) - \phi(v)|$,
  - Setting denominator $\sum_{(u,v)\in V^2} |f(u) - f(v)| \geq 1$
  - Enforcing triangle inequality $d_{u,v} \leq d_{u,w} + d_{w,v}$

# Example Application:

**Flux minimization problem:**

- Simple modification of the flux formulation:
  - letting $\boldsymbol{d_{u,v}} = |\boldsymbol{\phi(u)} - \boldsymbol{\phi(v)}|$,
  - Setting denominator $\sum_{(u,v)\in V^2} |\phi(u) - \phi(v)| \geq 1$
  - Enforcing triangle inequality $\boldsymbol{d_{u,v}} \leq \boldsymbol{d_{u,w}} + \boldsymbol{d_{w,v}}$
  - Relax the $d_{u,v} \in \{0,1\}$ and solve:

$$\min \sum_{(u,v)\in E} d_{u,v}$$

$$\text{s.t.} \begin{cases} \sum_{(u,v)\in V^2} d_{u,v} \geq 1 \\ d_{u,v} \leq d_{u,w} + d_{w,v} \\ 0 \leq d_{u,v} \leq 1 \end{cases}$$

# Example Application:

**Now what?**

- The solution of LP gives us a metric (***V***,***d***).
- We can use Bourgain's theorem:

*For any metric space* ($V$,$d$) *with* $|V|=n$ *there is an embedding into* $R^{(\log n)^2}$ *under* $L_1$ *with* $O(\log n)$ *distortion.* ***And we can construct this embedding in poly-time*** *using a* <u>*randomized algorithm*</u>.

# Example Application:

**Now what?**

- The solution of LP gives us a metric ($\boldsymbol{V},\boldsymbol{d}$).

- We can use Bourgain's theorem:

  *For any metric space (V,d) with |V|=n there is an embedding into $R^{(\log n)^2}$ under $L_1$ with $O(\log n)$ distortion.* ***And we can construct this embedding in poly-time*** *using a randomized algorithm.*

- Suppose $\omega: V \rightarrow R^{(\log n)^2}$ is such an embedding, we have

$$\boldsymbol{d_{u,v}} \leq |\boldsymbol{\omega}(\boldsymbol{u}) - \boldsymbol{\omega}(\boldsymbol{v})| \leq \boldsymbol{d_{u,v}} \times \log^2 \boldsymbol{n}$$

# Example Application:

**Now what?**

- Form the cut $\boldsymbol{S_{i,j}} = (\boldsymbol{A_{i,j}}, \boldsymbol{B_{i,j}})$ , for $\boldsymbol{j}$ in $\{1, \ldots, \boldsymbol{n}\text{-}1\}$ as follows:
  - Fix a coordinate $\boldsymbol{i}$ in $\{1, \ldots , \log^2 n\}$.
  - Order the vector with respect to their $\boldsymbol{i}$-th coordinate $\boldsymbol{\omega_i}(\boldsymbol{u})$
  - Take the first $\boldsymbol{j}$ points as $\boldsymbol{A_{i,j}}$
  - Take the other $\boldsymbol{n\text{-}j}$ points as $\boldsymbol{B_{i,j}}$

# Example Application:

**Now what?**

- Form the cut $\boldsymbol{S_{i,j}} = (\boldsymbol{A_{i,j}}, \boldsymbol{B_{i,j}})$ , for $\boldsymbol{j}$ in {1, …, $\boldsymbol{n}$-1} as follows:
  - Fix a coordinate $\boldsymbol{i}$ in {1, … , $\log^2 n$}.
  - Order the vector with respect to their $\boldsymbol{i}$-th coordinate $\boldsymbol{\omega_i}(\boldsymbol{u})$
  - Take the first $\boldsymbol{j}$ points as $\boldsymbol{A_{i,j}}$
  - Take the other $\boldsymbol{n-j}$ points as $\boldsymbol{B_{i,j}}$
- This will result in $\boldsymbol{n}\times\log^2 \boldsymbol{n}$ cuts of the form $\boldsymbol{S_{i,j}}$.
- Choose the one the give the minimum flux value.
- **Theorem:** The procedure described above generates a cut within a factor of $O(\log \boldsymbol{n})$ to the optimal in poly-time.

# Overview

Geometry of Graphs and Graphs Encoding the Geometry

**Spectral Graph Theory**

# Introduction:

- Spectral graph theory is a branch of Algebraic graph Theory (the study of matrices associated with a graph).

- Spectral graph theory deals with studying spectral operators associated with a graph:
  - For an $\boldsymbol{n \times n}$ matrix $\boldsymbol{A}$ having a basis of right-eigenvalues $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{\boldsymbol{n}}$ means:

$$Av_i = \lambda_i v_i$$

  - Assuming $\boldsymbol{x} = \boldsymbol{c}_{\mathbf{1}}\boldsymbol{v}_1+\ldots+\boldsymbol{c}_{\boldsymbol{n}}\boldsymbol{v}_{\boldsymbol{n}}$ , as an operator, the behavior of $\boldsymbol{A}$ on vector $\boldsymbol{x}$ can be expressed as

$$A^k x = \sum_i c_i A^k v_i = \sum_i c_i \lambda^k v_i$$

# Notations:

- Adjacency operator:

$$A_G(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E(G) \\ 0 & \text{Otherwise} \end{cases}$$

- Observer that for a vector ***x***:

$$(A_G x)(u) = \sum_{b:(u,v)\in E} x(v)$$

- Define ***d***(***v***)=|{***u***| (***u***,***v***) in ***E***(***G***)}| then degree matrix

$$D_G(u,v) = \begin{cases} d(u) & \text{if } (u,v) \in E(G) \\ 0 & \text{Otherwise} \end{cases}$$

# Notations:

- Using Degree matrix

$$D_G(u,v) = \begin{cases} d(u) & \text{if } (u,v) \in E(G) \\ 0 & \text{Otherwise} \end{cases}$$

- Diffusion matrix operator:

$$W_G = A_G D_G^{-1}$$

- The action of this operator on a vector ***x***:

$$(W_G x)(u) = \sum_{v:(u,v) \in E} x(v) / d(u)$$

# Quadratic forms:

- Laplacian forms:

$$x^T L_G x = \sum_{(u,v)\in E} L_G(u,v)\cdot(x(u) - x(v))^2$$

- **Motivation:**
  - measures the smoothness of walk denoted by function ***x*** (its value is small if ***x*** does not change dramatically along each edge).
  - As a matrix operator:

$$L_G = D_G - A_G$$

  - Normalized version

$$N_G = D^{-1/2} L_G D^{-1/2} = I - D^{-1/2} A_G D^{-1/2}$$

# Courant-Fisher Theorem:

- The Rayleigh quotient of a nonzero vector ***x*** with resect to symmetric matrix ***A***:

$$\frac{x^T A x}{x^T x}$$

- **Theorem:** Let ***A*** be a symmetric matrix with spectrum $\boldsymbol{\alpha_1} \geq \ldots \geq \boldsymbol{\alpha_n}$. Then

$$a_k = \max_{\substack{S \subseteq R^n \\ \dim(S)=k}} \min_{\substack{x \in S \\ x \neq 0}} \frac{x^T A x}{x^T x} = \min_{\substack{T \subseteq R^n \\ \dim(T)=n-k+1}} \max_{\substack{x \in S \\ x \neq 0}} \frac{x^T A x}{x^T x}$$

# Low-rank Approximation:

- Eigenvalues and eigenvectors provide low-rank approximation of a matrix.
- Recall, for matrix A with spectrum $\boldsymbol{\alpha_1} \geq \ldots \geq \boldsymbol{\alpha_n}$:

$$A = \sum_i \alpha_i v_i v_i^T$$

- Consequence of Courant-Fischer:
  - For every ***k***, the best approximation of ***A*** by a rank ***k*** matrix can be obtained by

$$\hat{A} = \sum_{i=1}^{k} \alpha_i v_i v_i^T$$

  - i.e

$$\hat{A} = \arg\min_{rank(B)=k} \|A - B\|_F$$

# Notes:

- The all-ones vector is an eigenvector of $L_G$.
- Let $\alpha_1 \geq \ldots \geq \alpha_n$ denote the spectrum of $A_G$, then:

$$\bar{d}(G) \leq \alpha_1 \leq \Delta(G).$$

- The all-ones is an eigenvector of $A_G$ only if $G$ is a regular graph.
- Multiplicity of **0** eigenvalue of $L_G$ is the number of connected components of $G$.
- Let $\lambda_1 \geq \ldots \geq \lambda_n$ denote the spectrum of $L_G$, then:

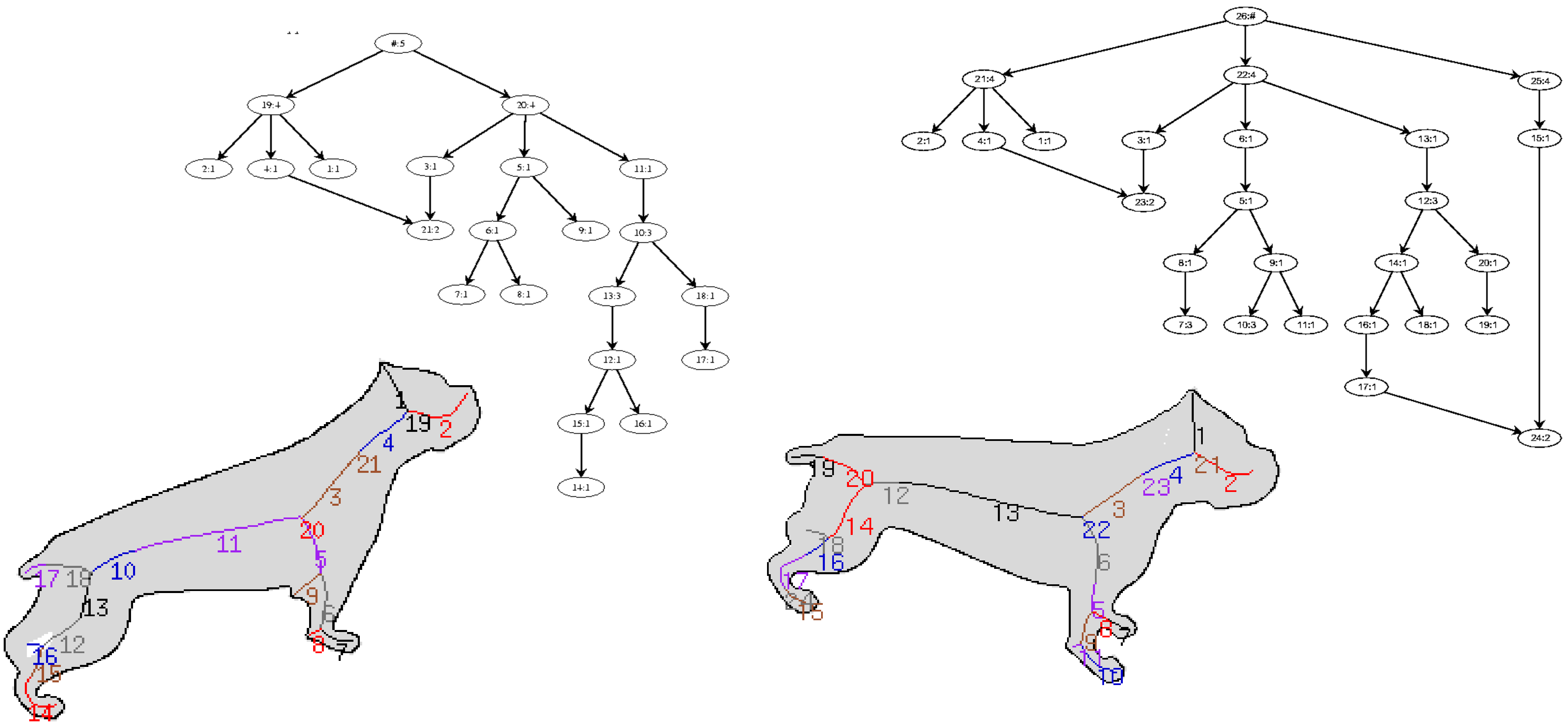
$$\lambda_1 \leq 2 \times \Delta(G).$$

- If $\alpha_1 = -\alpha_n$ only if $G$ is a bipartite graph.

# Matching Spectral Abstractions of Graph Structures

- Image features and their relations can be conveniently represented by labeled graphs.
- When features are multi-scale, or when part/whole relations exist between features, resulting graphs can be represented as directed acyclic graphs.
- Object recognition can therefore be formulated as hierarchical graph matching.
- Using spectral graph theory, we embed discrete graphs into low-dimensional continuous spaces.

# Matching Spectral Abstractions of Graph Structures

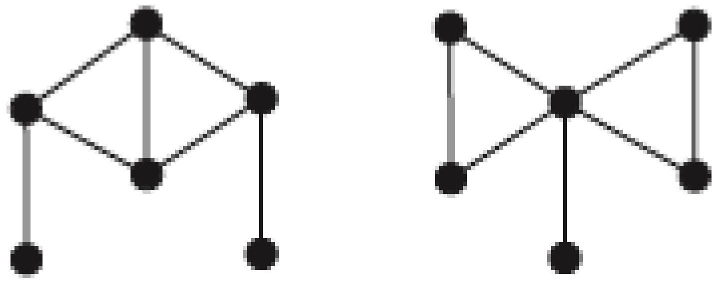


# The Eigenspace and Isomorphism

- If two graphs have different spectra (equivalently, different characteristic polynomials) of the adjacency matrix, then they are not isomorphic

- However, non-isomorphic graphs can be co-spectral!

- But, are they unique? No, but co-spectral graphs are not that common.



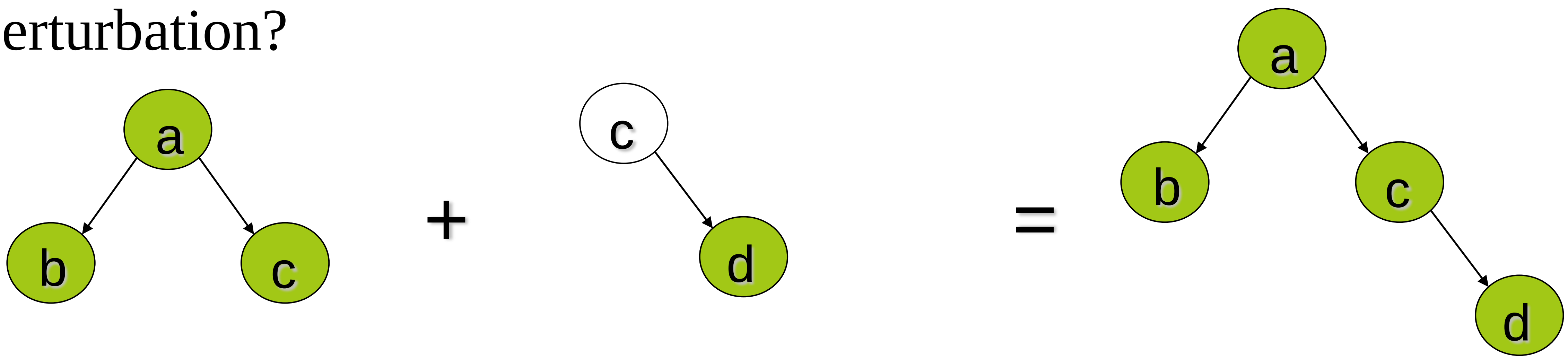
$$p(x) = x^6 - 7x^4 - 4x^3 + 7x^2 + 4x - 1$$

# The Eigenspace and Isomorphism

- Clearly, isomorphic graphs must have the same adjacency and Laplacian spectrum (i.e., Laplacian characteristic polynomial)
- **Bad news**: non-isomorphic graphs can be adjacency or Laplacian cospectral
- [Schwenk 73], [McKay 77] For almost all trees *T* there is a non-isomorphic tree *T'* that has both the same adjacency spectrum and the same Lapalcian spectrum
- **Idea**:
  - Use the spectrum of all subgraphs associated with a graph for its characterization.

# Perturbation

- How robust is the spectrum under noise and minor structural perturbation?




G (original) E (noise) H (perturbed)

| | a | b | c | |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 0 |
| b | -1 | 0 | 0 | 0 |
| c | -1 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |

$\Psi(A_G)$

+

| | a | b | c | d |
|---|---|---|---|---|
| a | 0 | 0 | 0 | 0 |
| b | 0 | 0 | 0 | 0 |
| c | 0 | 0 | 0 | 1 |
| d | 0 | 0 | -1 | 0 |

$A_E$

=

| | a | b | c | |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 0 |
| b | -1 | 0 | 0 | 0 |
| c | -1 | 0 | 0 | 1 |
| | 0 | 0 | -1 | 0 |

$A_H$

# Perturbation:

- Let ***S*** denote a subset of vertices ***V***(***G***), ***A***(***X***), the induced sub-matrix corresponding to set ***X***, and ***A***(***X***,***Y***) the adjacency matrix between sets ***X*** and ***Y***.

- We have

$$A(G) = \begin{bmatrix} A(S) & 0 \\ 0 & A(V-S) \end{bmatrix} + \begin{bmatrix} 0 & A(S, V-s) \\ A(V-S, S) & 0 \end{bmatrix}$$

- How the eigenvalues of ***A*** are related to those of the other matrices?

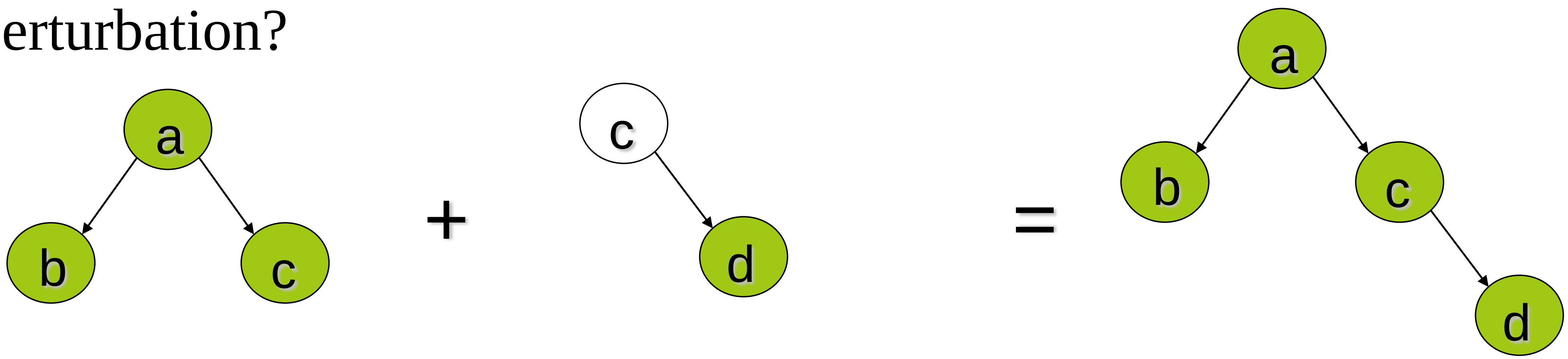
# Perturbation:

- Let $X$ and $Y$ denote two symmetric matrices with eigenvalues $\alpha_1 \geq \ldots \geq \alpha_n$ and $\beta_1 \geq \ldots \geq \beta_n$, respectively, and let $M = X - Y$.
- **Weyl's theorem**:
  - $M$ is symmetric.
  - $|\alpha_i - \beta_i| \leq \|M\|$ for all $i=1,\ldots,n$, where $\|M\|$ is the largest eigenvalue of $M$.
- More generally:
  - Let $v_1,\ldots, v_n$ be an orthonormal basis of eigenvectors of $A$ corresponding to $\alpha_1,\ldots,\alpha_n$ and let $u_1,\ldots,u_n$ be an orthonormal basis of eigenvectors of **B** corresponding to $\beta_1,\ldots,\beta_n$. Let $\theta_i$ be the angle between $v_i$ and $w_i$. Then,

$$\frac{1}{2}\sin 2\theta_i \leq \frac{\|M\|}{\min_{j \neq i}|\alpha_i - \alpha_j|}$$

# Perturbation

- How robust is the spectrum under noise and minor structural perturbation?




G (original) E (noise) H (perturbed)

| | a | b | c | |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 0 |
| b | -1 | 0 | 0 | 0 |
| c | -1 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |

$\Psi(A_G)$

+

| | a | b | c | d |
|---|---|---|---|---|
| a | 0 | 0 | 0 | 0 |
| b | 0 | 0 | 0 | 0 |
| c | 0 | 0 | 0 | 1 |
| d | 0 | 0 | -1 | 0 |

$A_E$

=

| | a | b | c | |
|---|---|---|---|---|
| a | 0 | 1 | 1 | 0 |
| b | -1 | 0 | 0 | 0 |
| c | -1 | 0 | 0 | 1 |
| | 0 | 0 | -1 | 0 |

$A_H$

# Perturbation

- [Wilkinson] If $A$ and $A + E$ are $n \times n$ symmetric matrices, then for all $k$ in $\{1,\cdots,\boldsymbol{n}\}$, and eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$:

$$\lambda_k(A) + \lambda_k(E) \leq \lambda_k(A+E) \leq \lambda_k(A) + \lambda_1(E).$$

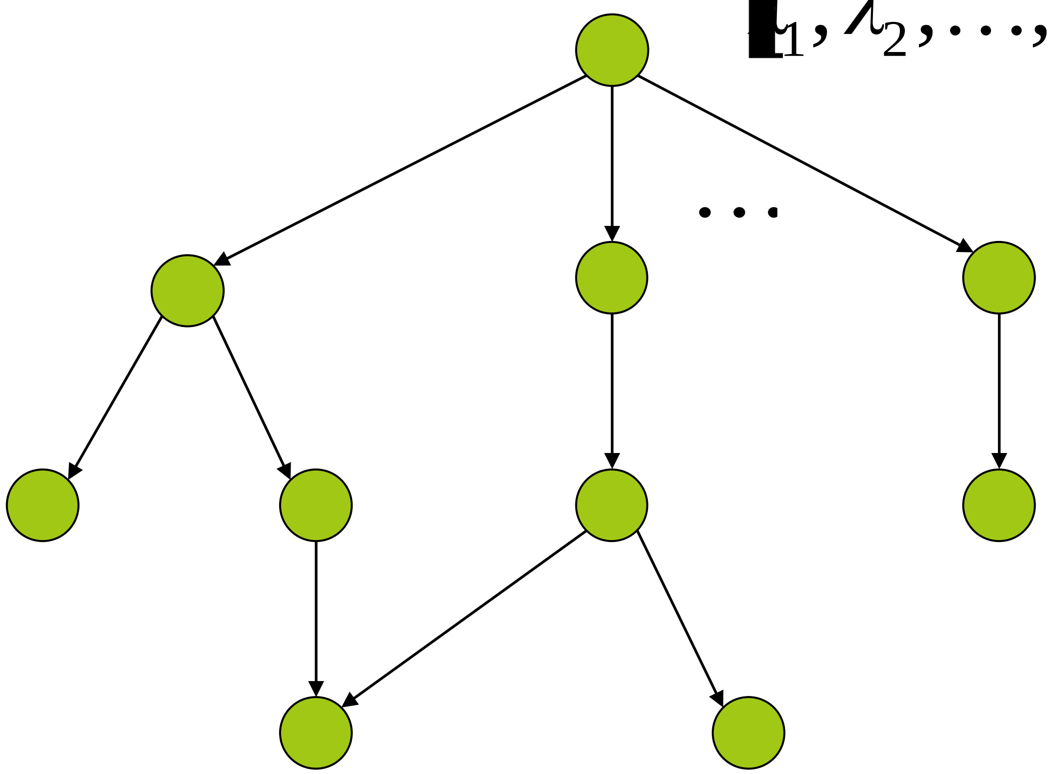
- This is also know as Courant's interlacing theorem
- [Marcini et al.] For $H$ (perturbed graph) and $G$ (original graph), the above theorem yields (after manipulation):

$$\left|\lambda_k(A_H) - \lambda_k(\Psi(A_G))\right| \leq \left|\lambda_1(A_E)\right|$$

- They also extended this result to directed acyclic graphs.

# The Eigenvalues are Stable Now What?

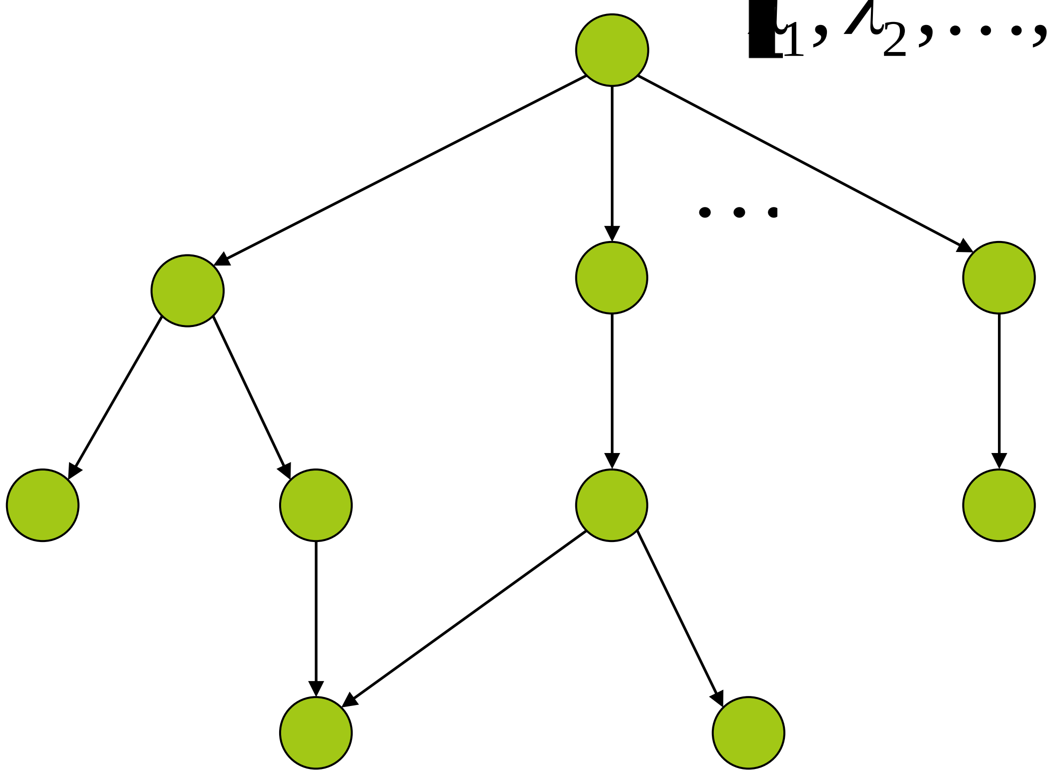
$$[\lambda_1, \lambda_2, \ldots, \lambda_n]$$



We *could* compute the graph's eigenvalues, sort them, and let them become the components of a vector assigned to the graph.

# The Eigenvalues are Stable Now What?

$$[\lambda_1, \lambda_2, \ldots, \lambda_n]$$

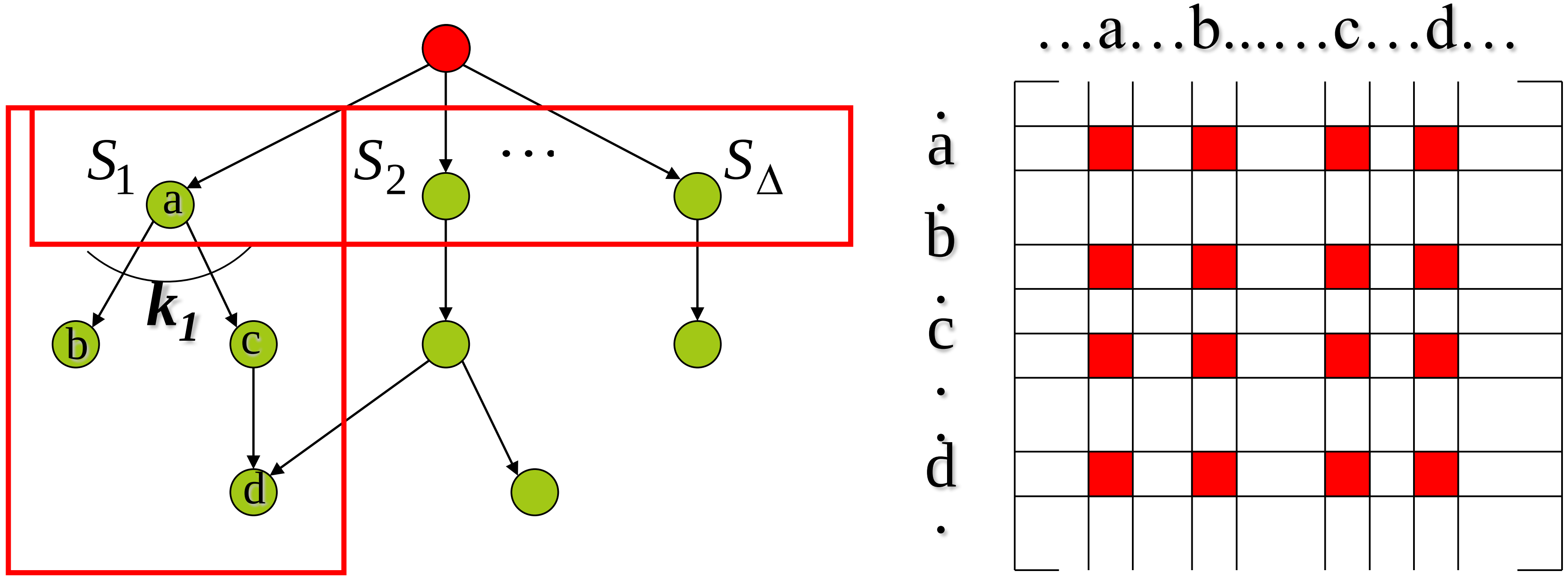


We *could* compute the graph's eigenvalues, sort them, and let them become the components of a vector assigned to the graph.

**But:**

1. Dimensionality grows with size of graph.
2. Eigenvalues are global! Therefore, can't accommodate occlusion or clutter.

# Forming a Structural Signature

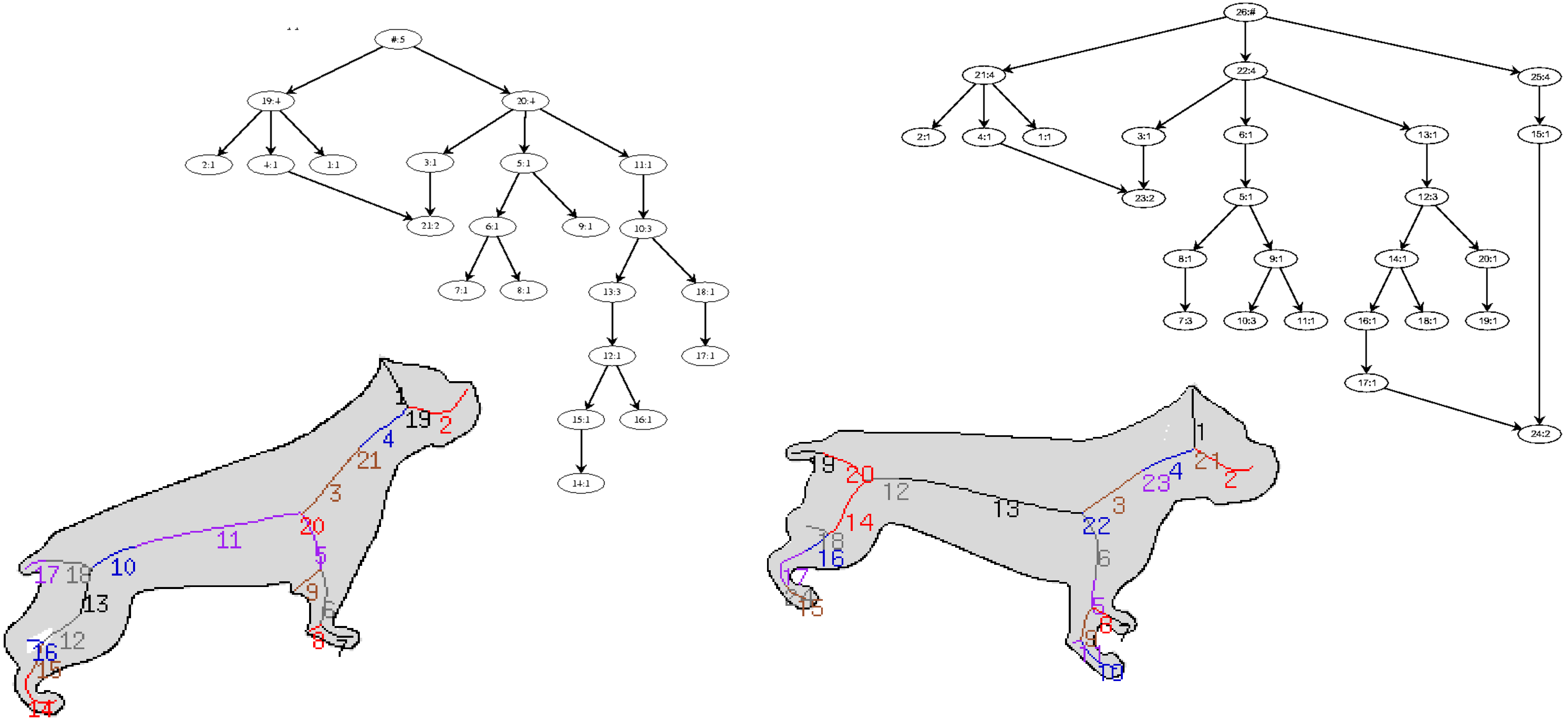



$$V = [S_1, S_2, S_3, \ldots, S_\Delta], \quad S_1 \geq S_2 \geq S_3 \geq \ldots S_\Delta \qquad S_i = |\lambda_1| + |\lambda_2| + \ldots |\lambda_{k_i}|$$

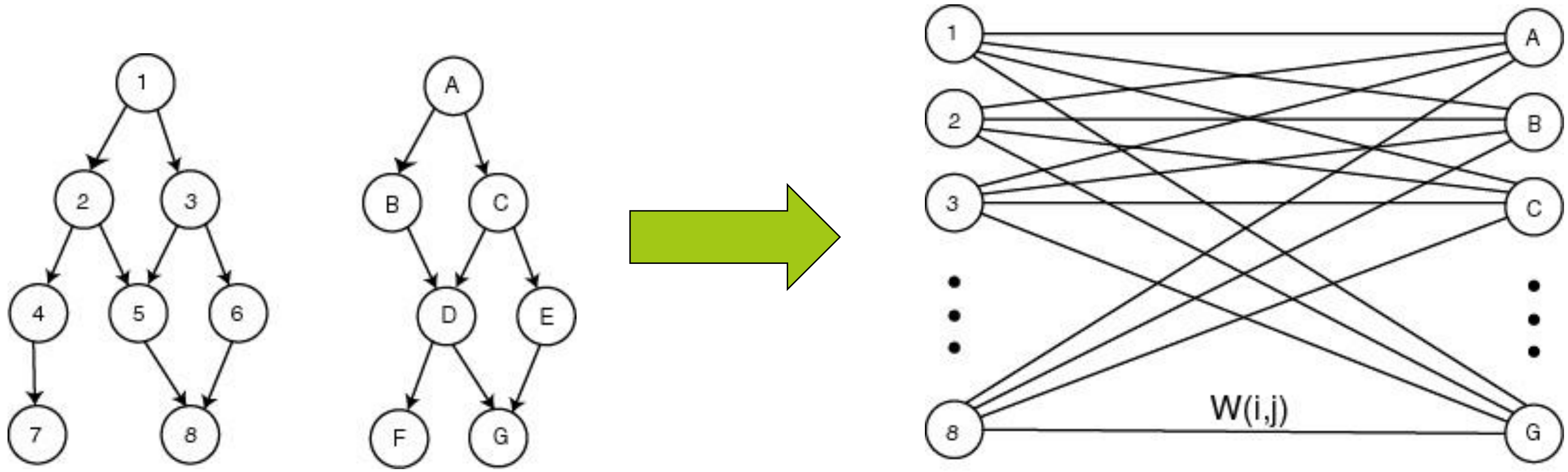
## Why Sum the *k* largest Eigenvalues?

1. Summing reduces dimensionality.
2. Largest eigenvalues most informative.
3. Sums are “normalized” according to richness ($k_i$) of branching structure.

# Matching Spectral Abstractions of Graph Structure



# Matching Problem:

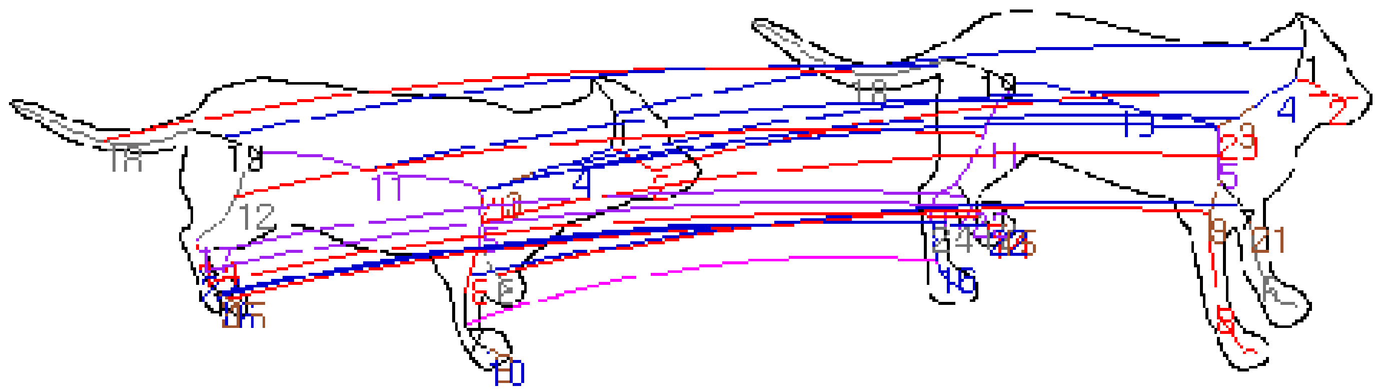


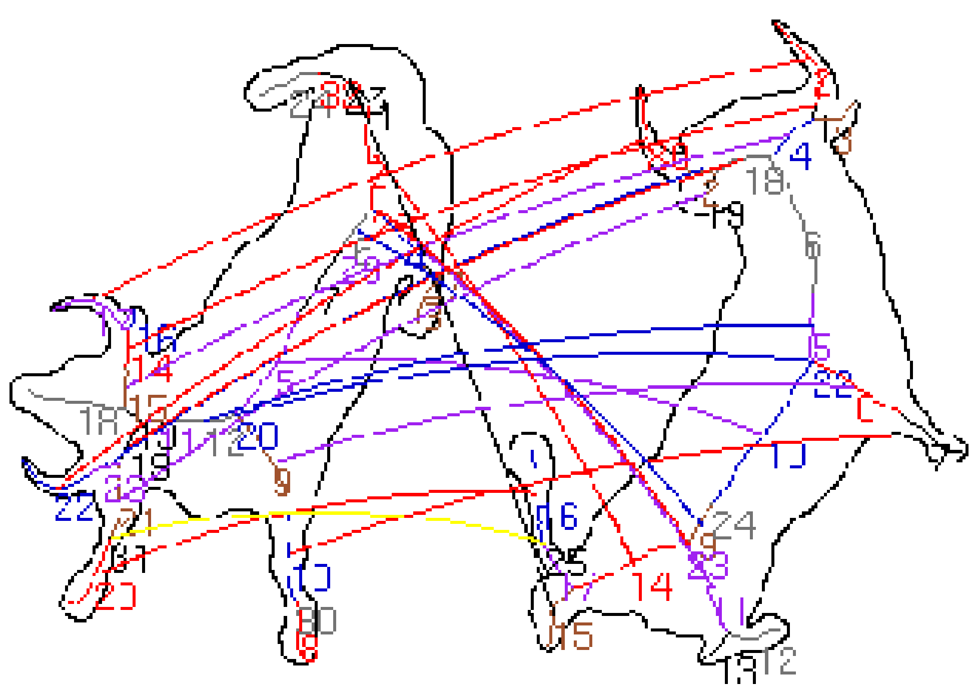

Matching: Consider a bipartite graph matching formulation, in which the edges in the query and model graphs are discarded.

Hierarchical structure is seemingly lost, but can be encoded in the edge weights:

$$W(i,j) = e^{-\left(\alpha_1 d_{struct}(i,j) + \alpha_2 d_{geom}(i,j)\right)}$$

# Sample Matches



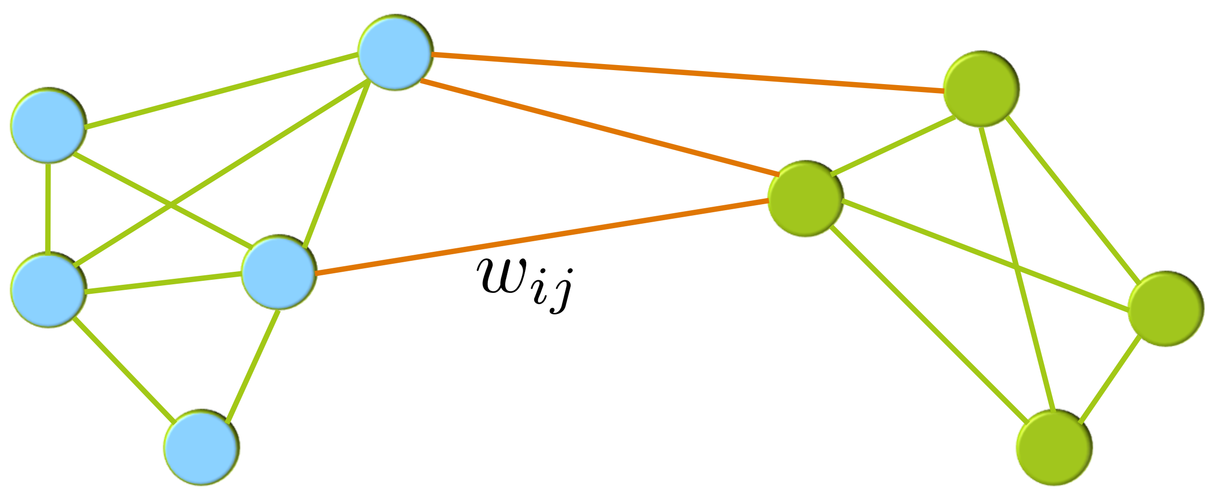


# Connectivity:

- Is there a relationship between eigenvalue distribution and structure of a graph?
- Not hard to show that $\boldsymbol{\lambda}_2(\boldsymbol{G})>0$ iff $\boldsymbol{G}$ is connected.
- **Fiedler eigenvalue problem**: Better connected graphs have higher second eigenvalues!
- There is an eigen-embedding algorithm due to Fiedler (extended by Holst):
  - Compute the eigenvector $\boldsymbol{x}_2$ corresponding to $\boldsymbol{\lambda}_2(\boldsymbol{G})$
  - Form a cut by $\boldsymbol{S}_{t(\leq 0)} = \{\boldsymbol{u} | \boldsymbol{x}_2(\boldsymbol{u}) > \boldsymbol{t}\}$ (and $\boldsymbol{V} \setminus \boldsymbol{S}_t$)
  - Fiedler showed the set $\boldsymbol{S}_t$ forms a (strongly) connected subgraph.

# Cuts and Clustering:

- Recall a cut in a graph is a partition of the vertices to two sets ***S***, ***V-S*.**
- For a weighted graph a weight can be associated with the cut:




$$\partial(S) = \mathrm{cut}(S, V - S) = \sum_{i \in S} \sum_{j \in V - S} w_{ij}$$

# Connectivity and Graph Cut:

- Recall the tradeoff function for sparsest cut or min flux cut (ratio of cut) is:

$$R(S) = \frac{|\partial S|}{|S|\ |V - S|}.$$

- $\boldsymbol{R}(\boldsymbol{S})$ is at least $\boldsymbol{\lambda}_2(\boldsymbol{G})/\boldsymbol{n}$ and eigenvector $\boldsymbol{v}_2$ corresponding to second eigenvalue is related to indicator vector for a set $\boldsymbol{S}$ that minimizes $\boldsymbol{R}(\boldsymbol{S})$:

# Connectivity and Graph Cut:

- Recall the tradeoff function for sparsest cut or min flux cut (ratio of cut) is:

$$R(S) = \frac{|\partial S|}{|S| \; |V - S|}.$$

- $\boldsymbol{R}(\boldsymbol{S})$ is at least $\lambda_2(\boldsymbol{G})/\boldsymbol{n}$ and eigenvector $\boldsymbol{v_2}$ corresponding to second eigenvalue is related to indicator vector for a set $\boldsymbol{S}$ that minimizes $\boldsymbol{R}(\boldsymbol{S})$:
  - Let $\boldsymbol{x_S}$ be the characteristic vector for $\boldsymbol{S}$.
  - We know $x_S^T L_G x_S = |\partial(S)|.$
  - And $\sum_{u<v} \left(x_S(u) - x_S(v)\right)^2 = |S||V - S|.$
  - So $R(S) = \dfrac{x_S^T L_G x_S}{\sum_{u<v} \left(x_S(u) - x_S(v)\right)^2}$

# Connectivity and Partitioning:

- Recall the tradeoff function for sparse or min flux cut (ratio of cut) is:

$$R(S) = \frac{|\partial S|}{|S|\ \ |V - S|}.$$

- ***R***(***S***) is at least $\boldsymbol{\lambda_2(G)/n}$ and eigenvector $\boldsymbol{v_2}$ corresponding to second eigenvalue is related to indicator vector for a set ***S*** that minimizes ***R***(***S***):
  - Let $\boldsymbol{x_S}$ be the characteristic vector for ***S***.
  - We know $x_S^T L_G x_S = \left|\partial(S)\right|,$
  - And $\sum_{u<v} \left(x_S(u) - x_S(v)\right)^2 = |S||V - S|.$
  - So $R(S) = \dfrac{x_S^T L_G x_S}{\sum_{u<v} \left(x_S(u) - x_S(v)\right)^2}$

Fideler's eigenvalue problem

$$\lambda_2(G) = n \times \min_{x \neq 0} \frac{x_S^T L_G x_S}{\sum_{u<v} \left(x_S(u) - x_S(v)\right)^2}$$

# Connectivity and Partitioning:

- Restricting the entries of vector ***x*** being a 0-1will result in the cut that minimizes ***R***(***S***) and is the desirable min cut [**Hagen and Kahng**].

- The weighted variation of the ***R***(***S***) can be stated as

$$\Phi(S) = \frac{w(\partial(S))}{d(S)\, d(V - S)}$$

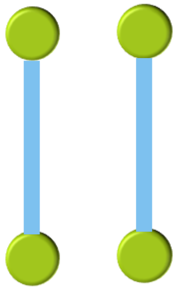
- Which is proportional to normalized cut measure (Lawler and Sokal)

$$\frac{w(\partial(S))}{d(S)} + \frac{w(\partial(V - S))}{d(V - S)}$$

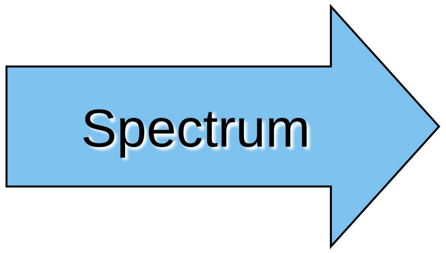
We will see that this is the objective function used by Shi and Malik for their segmentation algorithm.

# Spectral Clustering

- Methods that use the spectrum of the affinity matrix to cluster are known as ***spectral clustering***.
- Normalized cuts, Average cuts, Average association make use of the eigenvectors of the affinity matrix.



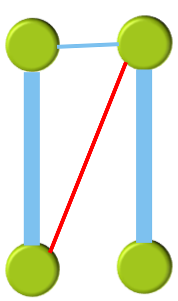
| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |



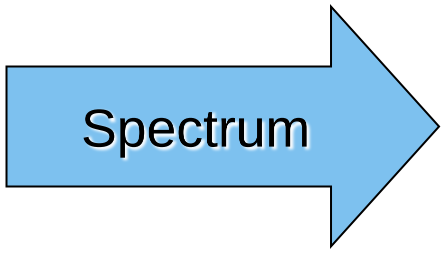

| $\lambda_1 = 2$ | $\lambda_2 = 2$ | $\lambda_3 = 0$ | $\lambda_4 = 0$ |
|---|---|---|---|
| .71 | 0 | | |
| .71 | 0 | | |
| 0 | .71 | | |
| 0 | .71 | | |

# Spectral Clustering

- Methods that use the spectrum of the affinity matrix to cluster are known as ***spectral clustering***.
- Normalized cuts, Average cuts, Average association make use of the eigenvectors of the affinity matrix.



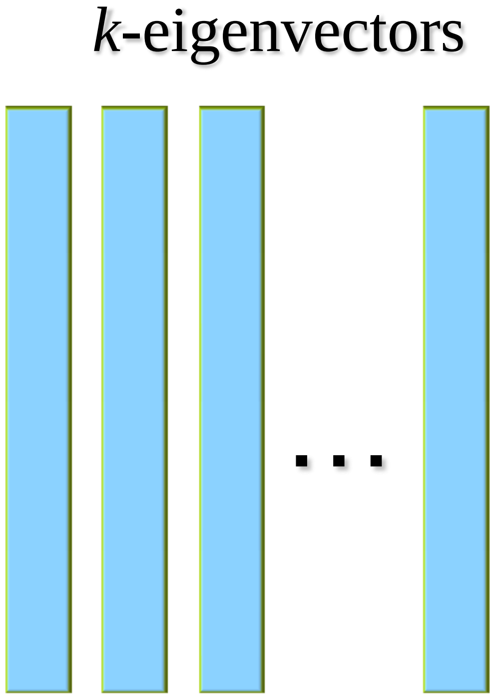
| | | | |
|---|---|---|---|
| 1 | 1 | .2 | 0 |
| 1 | 1 | 0 | -.2 |
| .2 | 0 | 1 | 1 |
| 0 | -.2 | 1 | 1 |




| $\lambda_1$= 2.02 | $\lambda_2$= 2.02 | $\lambda_3$= -0.02 | $\lambda_4$= -0.02 |
|---|---|---|---|
| .71 | 0 | | |
| .69 | -.14 | | |
| .14 | .69 | | |
| 0 | .71 | | |

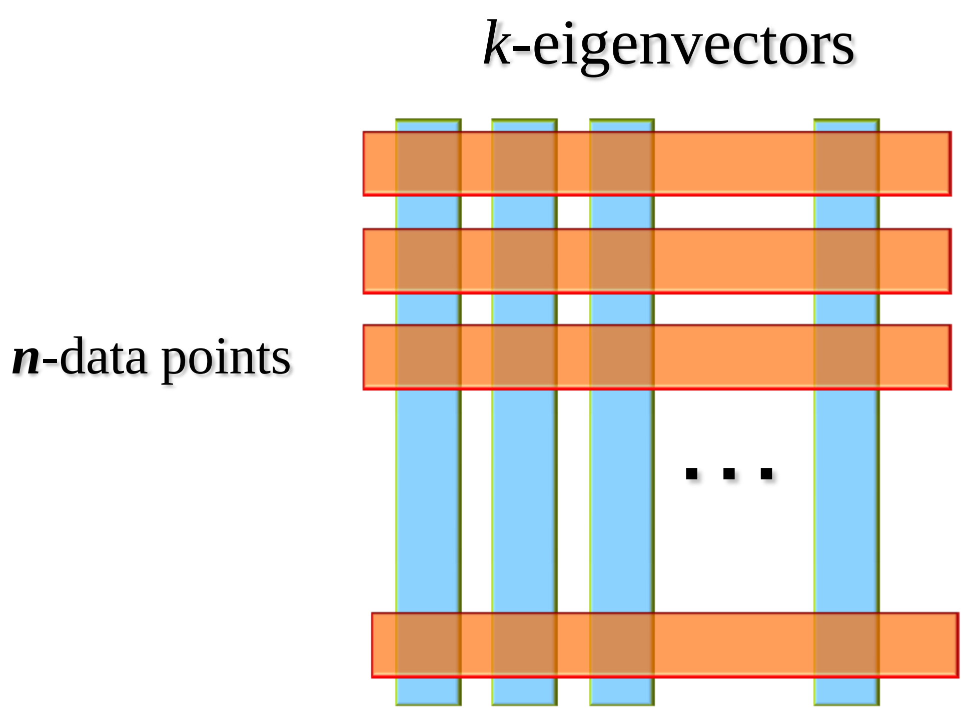
# Spectral Clustering

- We can use ***k*** eigenvectors for embedding of vertices into vector space.



# Spectral Clustering

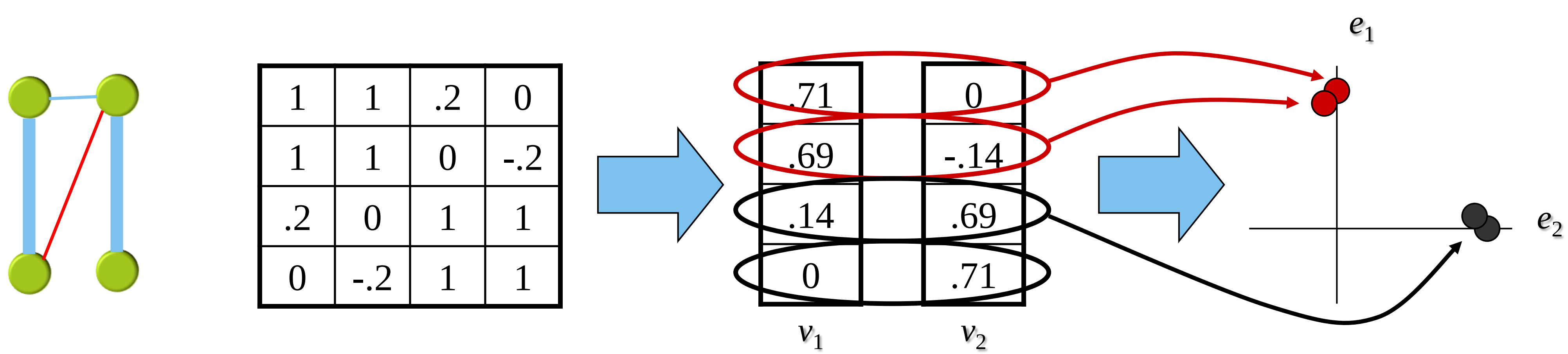
- We can use ***k*** eigenvectors for embedding of vertices into vector space.




- Each Row represents a data point in the eigenvector space.

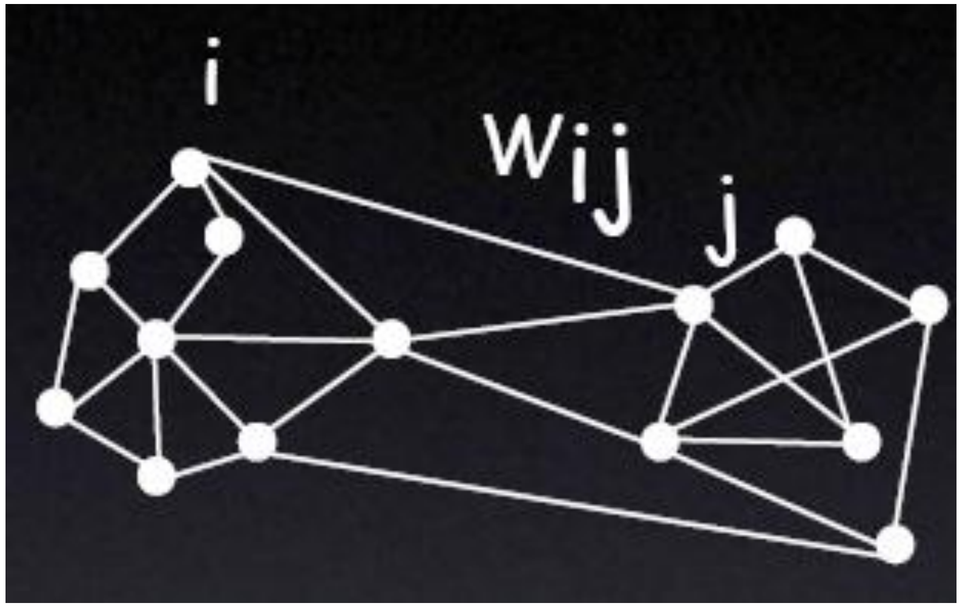
# Spectral Clustering

- We can use ***k*** eigenvectors for embedding of vertices into vector space.




- Each Row represents a data point in the eigenvector space.

# Graph-based Image Segmentation





***G***=(***V***,***E***)

***V***: graph nodes → Pixels

***E***: edges connection nodes → Pixel similarity

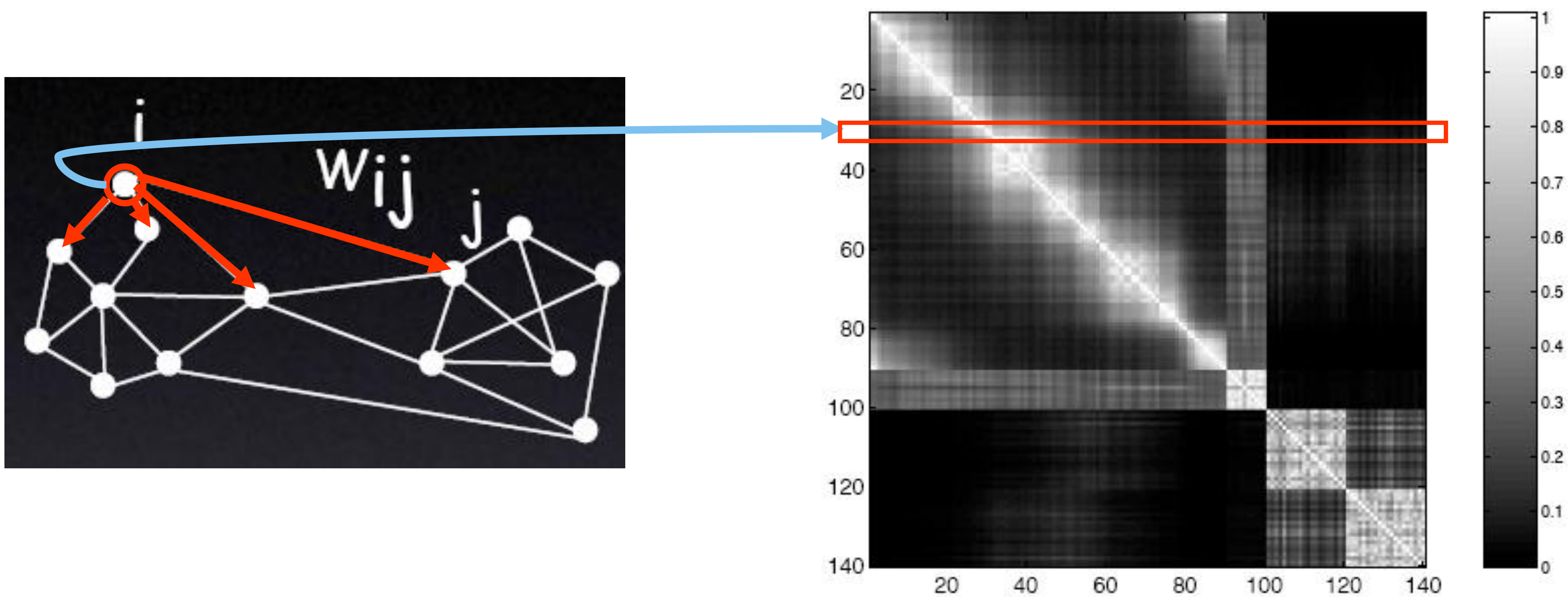
Slides from Jianbo Shi

# Cuts and segmentation

- Similarity matrix:

$$W = \left[ w_{i,j} \right]$$

$$w_{i,j} = e^{\frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$

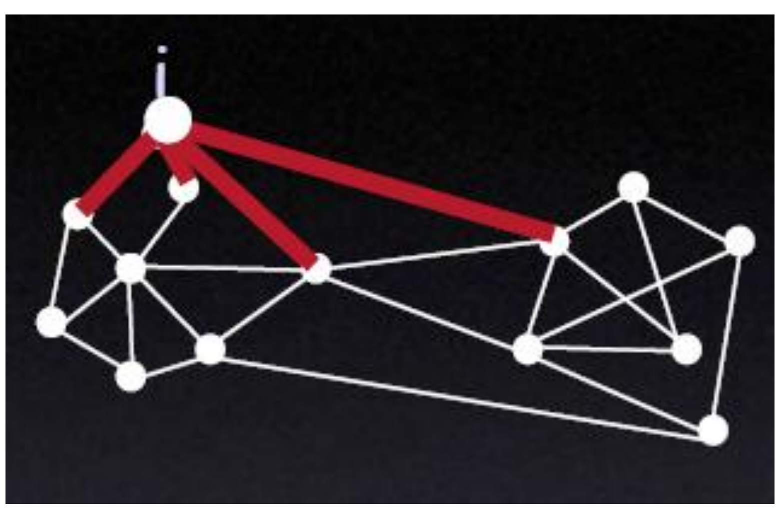


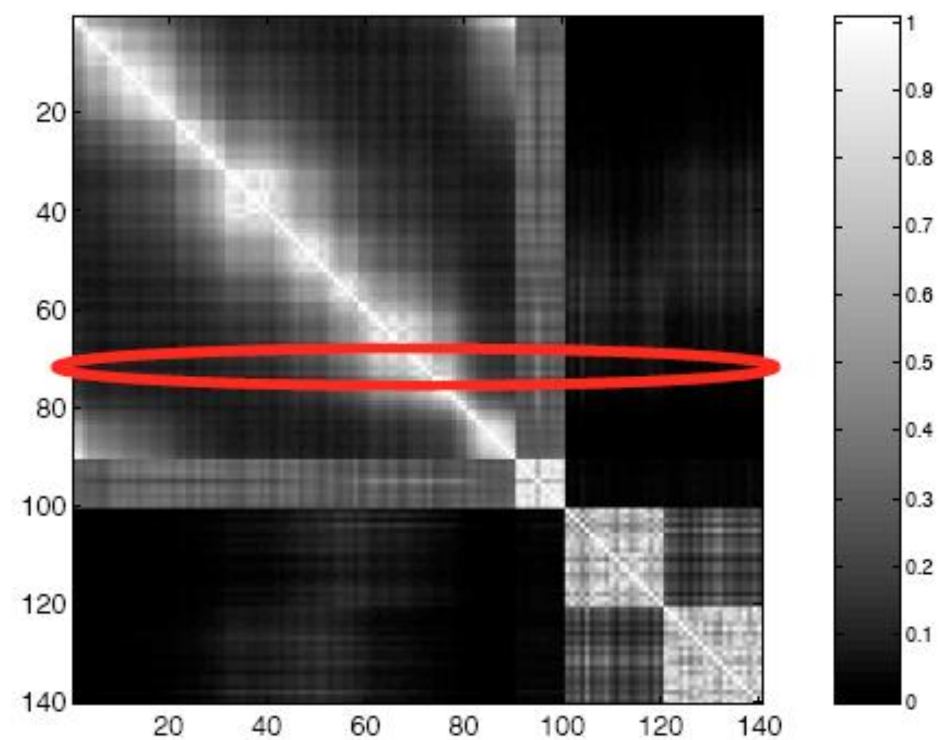

Slides from Jianbo Shi

# Graph terminology

- Degree of node:

$$d_i = \sum_j w_{i,j}$$



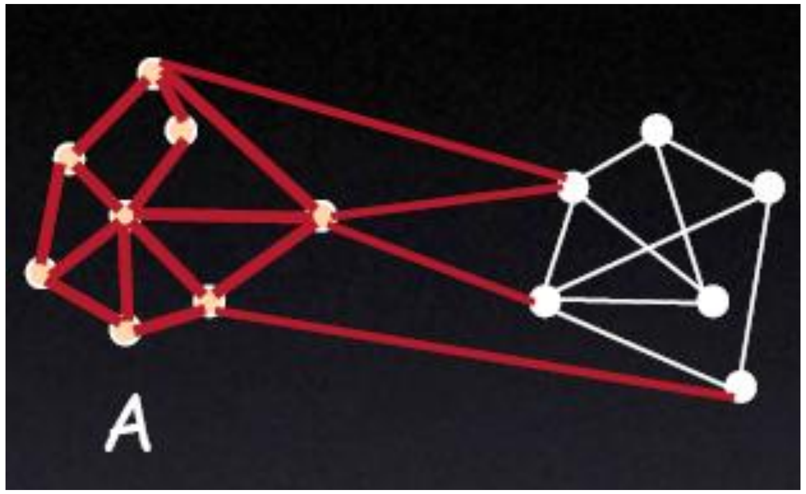



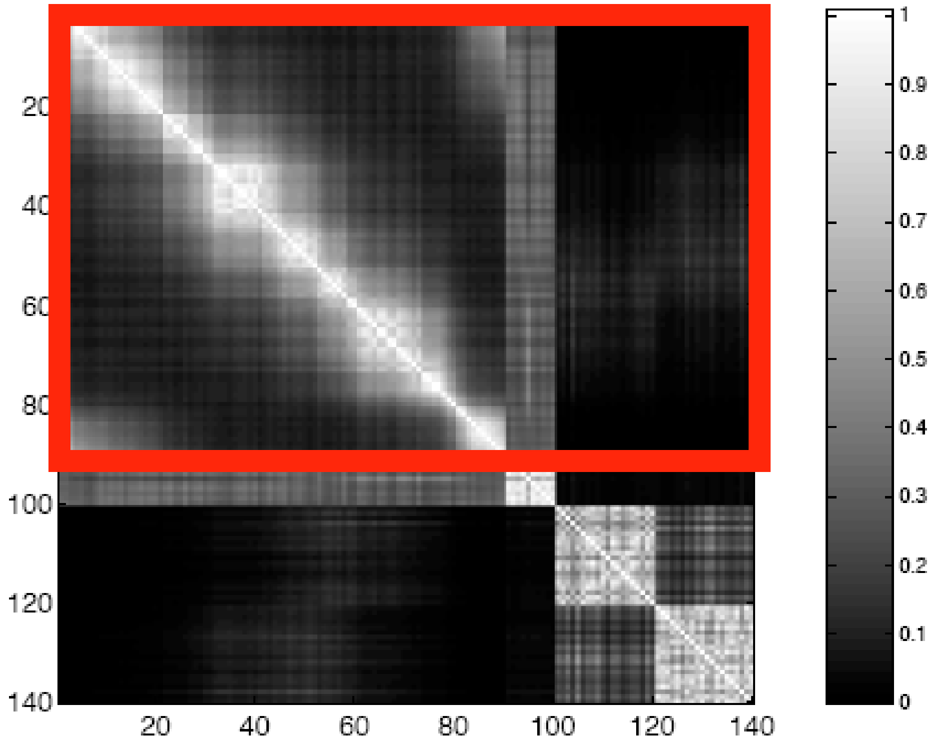

Slides from Jianbo Shi

# Graph terminology

- Volume of set:

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




Slides from Jianbo Shi

# Similarity functions

Intensity

$$W(i,j) = e^{\frac{-\left\| I_{(i)} - I_{(j)} \right\|_2^2}{\sigma_I^2}}$$

Distance

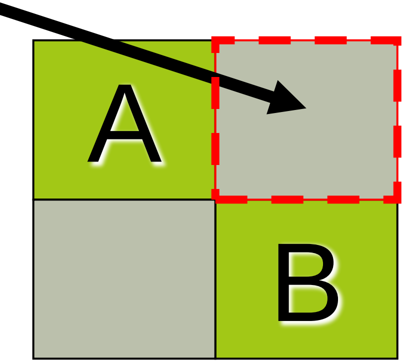
$$W(i,j) = e^{\frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$

Texture

$$W(i,j) = e^{\frac{-\left\| c_{(i)} - c_{(j)} \right\|_2^2}{\sigma_c^2}}$$

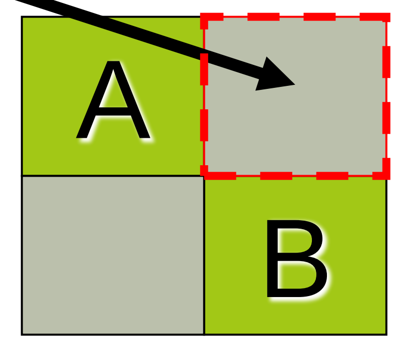
# Minimum cut

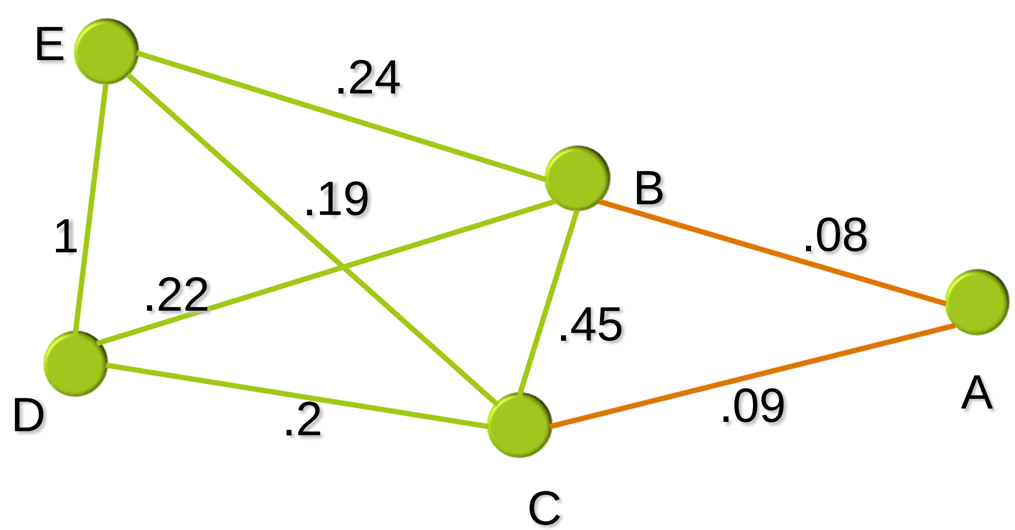
$$\min cut(A,B) = \min_{A,B} \sum_{u \in A, v \in B} w(u,v)$$



# Minimum cut

$$\min cut(A,B) = \min_{A,B} \sum_{u \in A, v \in B} w(u,v)$$



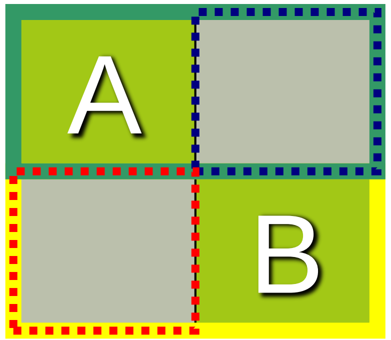




$$Cut(BCDE, A) = 0.17$$

# Normalized Cut

- Define normalized cut: “a fraction of the total edge connections to all the nodes in the graph”:

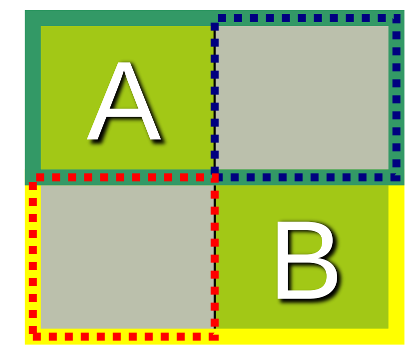
$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

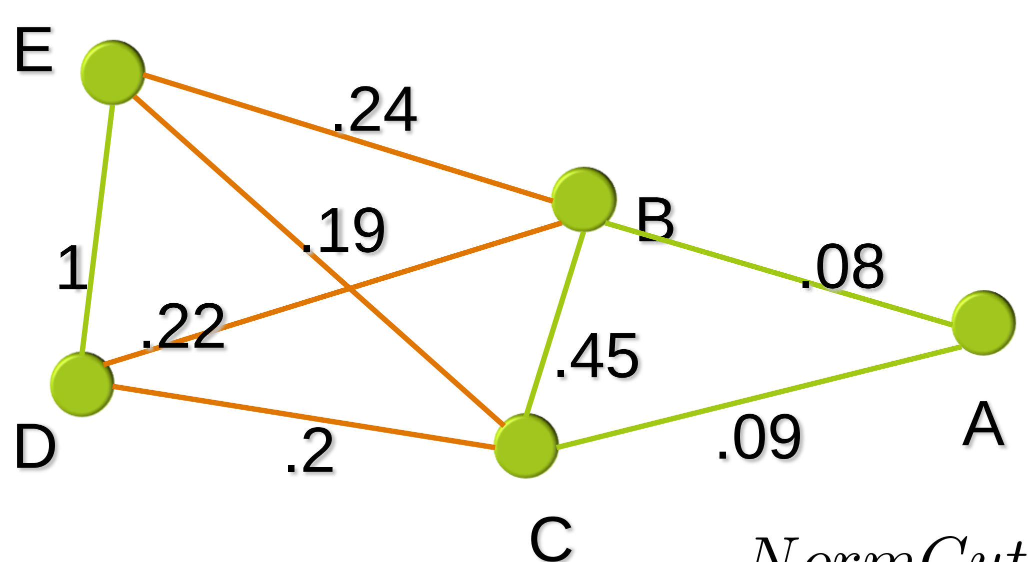


# Normalized Cut

- Define normalized cut: “a fraction of the total edge connections to all the nodes in the graph”:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$




$$NormCut(BCDE, A) = 1.067$$

$$NormCut(ABC, DE) = 1.038$$

# Finding the cut:

- Minimal (bi-partition) normalized cut.

$$\min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)} = \min \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

# Finding the cut:

- Minimal (bi-partition) normalized cut.

$$\min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)} = \min \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

- This can be restated in matrix form as

  $$NCut(A, B) = \frac{y^T(D - W)y}{y^T D y}$$

  - ***D*** is the diagonal (weighted) degree matrix
  - ***W*** is the weighetd adjacency matrix
  - ***D-W*** is the Laplacian matrix

# Finding the cut:

- Minimal (bi-partition) normalized cut.

$$\min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)} = \min \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

- This can be restated in matrix form as

$$NCut(A, B) = \frac{y^T (D - W) y}{y^T D y}$$

- As an optimization problem:

$$\min_y y^T (D - W) y \text{ subject to } y^T D y = 1$$

# Finding the cut:

- Minimal (bi-partition) normalized cut.

$$\min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)} = \min \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

- This can be restated in matrix form as

$$NCut(A, B) = \frac{y^T (D - W) y}{y^T D y}$$

- As an optimization problem:

$$\min_y y^T (D - W) y \text{ subject to } y^T D y = 1$$

- Which is a generalized eigenvalue problem:

$$(D - W) y = \lambda D y$$

# Recall

- ***L* = *D*-*W*** Positive semi-definite $x^T L x \geq 0$
- The first eigenvalue is 0, eigenvector is $\vec{1}$
- The second eigenvalue contains the solution

$$\lambda_2 = \frac{Cut(A,B)}{|A|} + \frac{Cut(A,B)}{|B|}$$

- The corresponding eigenvector contains the cluster indicator for each data point

# Random walks:

- Recall $W_G$ denotes the normalized Laplacian of ***G***.

- Let $\omega_1 \geq \ldots \geq \omega_n$ the spectrum of $W_G$; where $\omega_1$ is equal to 1 and has multiplicity 1. Let ***d*** denote eigenvector corresponding to $\omega_1$. We can define a probability distribution vector $\pi$ for graph ***G*** as follows:

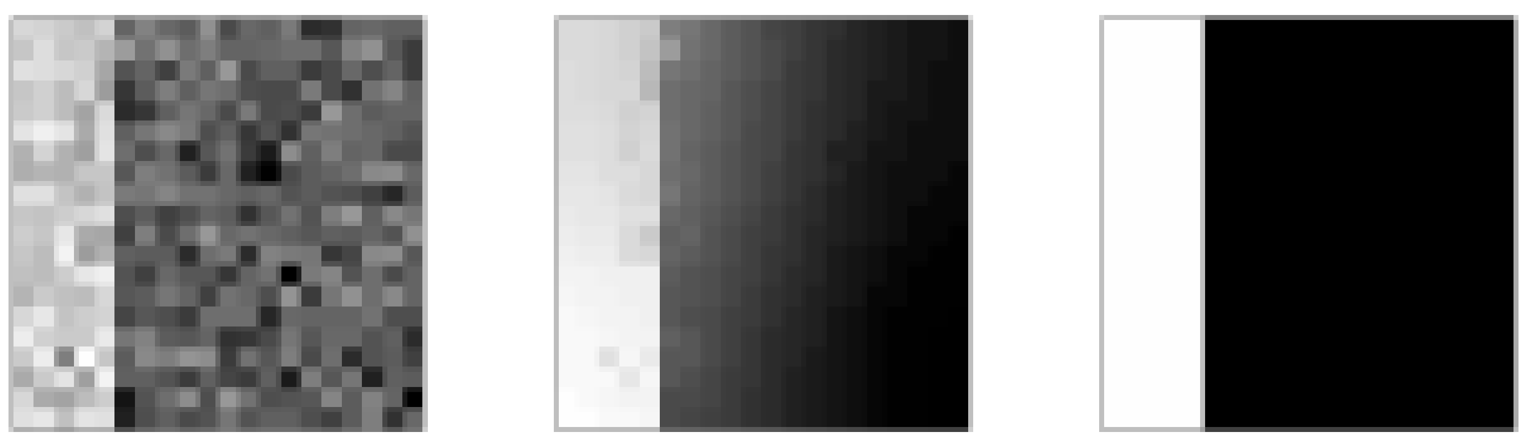
$$\pi(G) = \frac{1}{\sum_u d(u)} \times d$$

  - If $\omega_n \neq -1$, then the distribution of every walk will converge to $\pi$.
  - The rate of converge is a function of $|\omega_1 - \max(|\omega_2|, |\omega_n|)|$.
  - Specifically, let $x_t(v)$ denote the state of the system after ***t*** steps for a walk starting at ***u***:

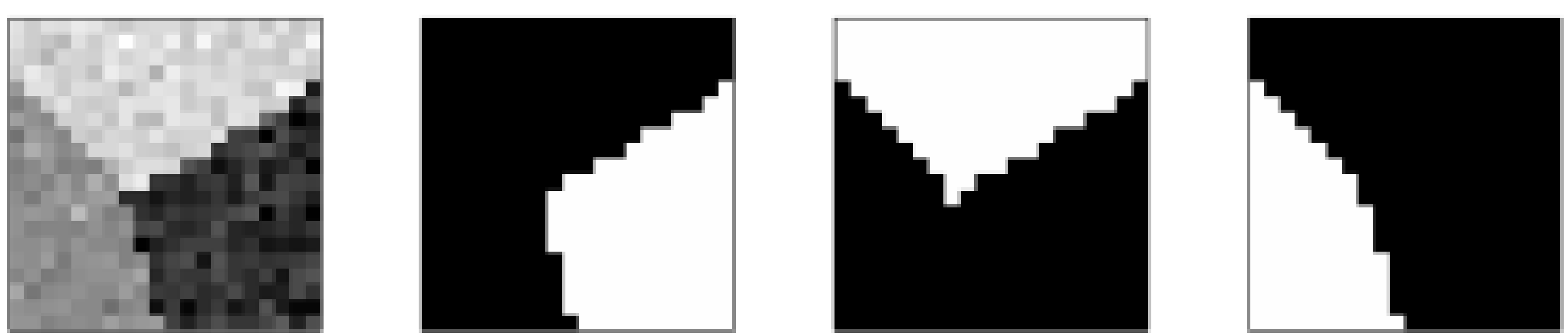
$$\left|p_t(b) - \pi(b)\right| \leq \sqrt{\frac{d(v)}{d(u)}}\left(1 - \max\left(|\omega_2|, |\omega_n|\right)\right)$$

# Experiments

- Synthetic images:



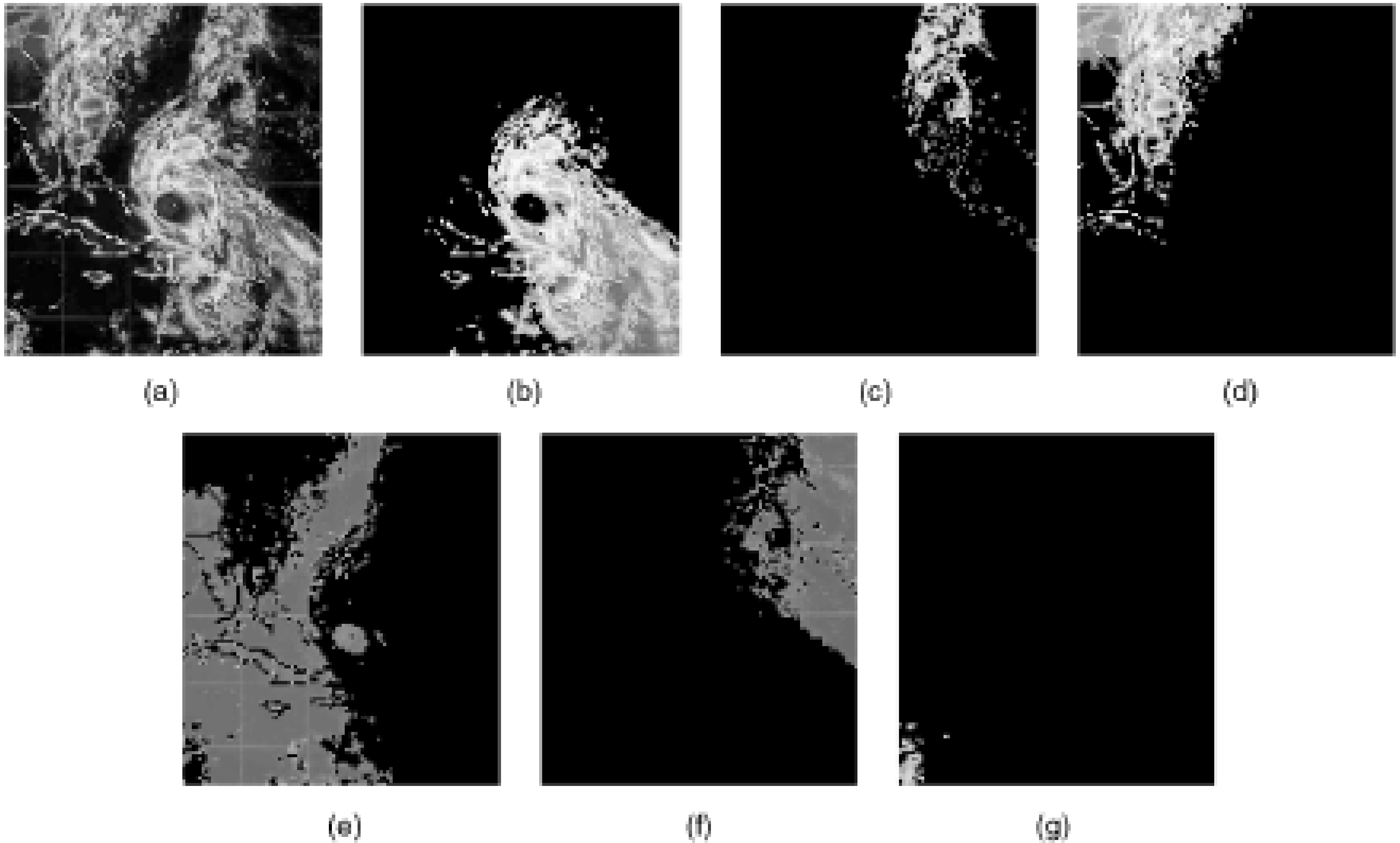
(a) (b) (c)



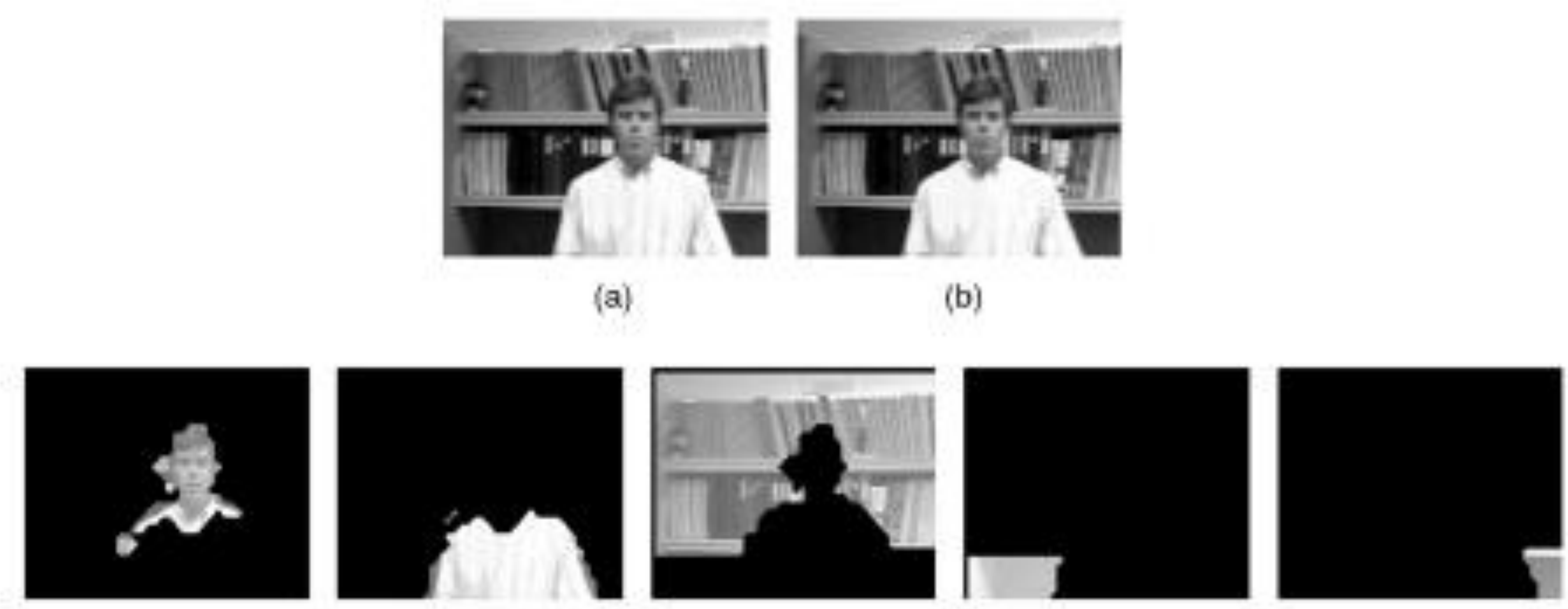
(a) (b) (c) (d)

# Experiments

- Weather radar:



# Experiments



(a) (b)

(c) (d) (e) (f) (g)

# Coloring:

- Valid coloring:
  - Given a graph ***G***, assign a color to every vertex of G so that the endpoints of each edge receive distinct colors.
- As an optimization the objective is to use minimum number of colors.
- The chromatic number $\boldsymbol{\chi}(\boldsymbol{G})$ is the least ***k*** for which ***G*** has a valid ***k***-coloring.
- **[Wilf]** Let $\boldsymbol{\alpha_1} \geq \ldots \geq \boldsymbol{\alpha_n}$ denote the spectrum of graph then

$$c\left(G\right) \pounds 1 + a_1$$

- [**Hoffman**] If ***G*** is a graph with at least one edge, then

$$c\left(G\right) {}^{3}\ 1 + \frac{a_1}{- a_n}$$

# Independent Sets:

- An independent set of vertices of graph ***G***, is a subset of vertices ***S*** such that no edge has both its end points in ***S***.

- As an optimization the objective is to find a maximum size independent set, denoted by ***ρ***(***G***).

- Note that the vertices of any color class of a grapg G form an independent set:

$$\rho(G) \geq \frac{n}{\chi(G)}$$

- [**Hoffman**] If ***G*** is a degree ***d*** regular graph, then

$$\rho(G) \leq n \times \frac{-\alpha_n}{d - \alpha_n}$$

# References


- Doyle, P. G. and Snell, J. L., Random Walks and Electric Networks, Vol. 22 of Carus Mathematical Monographs, Mathematical Association of America, 1984.
- Chung, F. R. K., Spectral Graph Theory, American Mathematical Society, 1997.
- van der Holst, H., Lov`asz, L., and Schrijver, A., "The Colin de Verdi`ere Graph Parameter," Bolyai Soc. Math. Stud., Vol. 7, 1999, pp. 29–85.
- Cvetkovi´c, D. M., Doob,M., and Sachs, H., Spectra of Graphs, Academic Press, 1978.
- Fiedler., M., "Algebraic connectivity of graphs," Czechoslovak Mathematical Journal , Vol. 23, No. 98, 1973, pp. 298–305.
- Fiedler, M., "A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory," Czechoslovak Mathematical Journal , Vol. 25, No. 100, 1975, pp. 618–633.

# References


- Lov´asz, L., “Random walks on graphs: a survey,” Combinatorics, Paul Erd¨os is Eighty, Vol. 2 , edited by T. S. D. Miklos, V. T. Sos, Janos B´olyai Mathematical Society, Budapest, 1996, pp. 353–398.
- Wilf, H. S., “The Eigenvalues of a Graph and its Chromatic Number,” J. London math. Soc., Vol. 42, 1967, pp. 330–332.
- Hoffman, A. J., “On eigenvalues and colorings of graphs,” Graph Theory and its Applications, Academic Press, New York, 1970, pp. 79–92.
- Dodziuk, J., “Difference Equations, Isoperimetric Inequality and Transience of Certain Random Walks,” Transactions of the American Mathematical Society, Vol. 284, No. 2, 1984, pp. 787–794.
- Pothen, A., Simon, H. D., and Liou, K.-P., “Partitioning Sparse Matrices with Eigenvectors of Graphs,” SIAM Journal on Matrix Analysis and Applications, Vol. 11, No. 3, 1990, pp. 430–452.