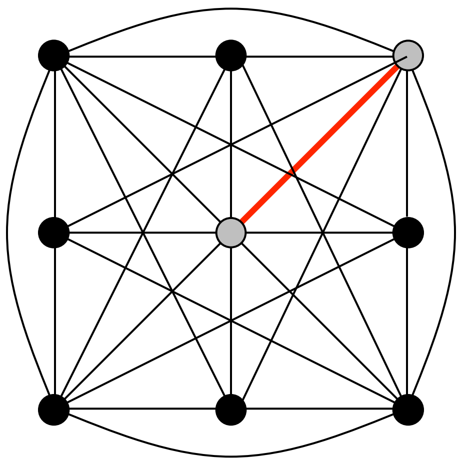
Graph-based Methods

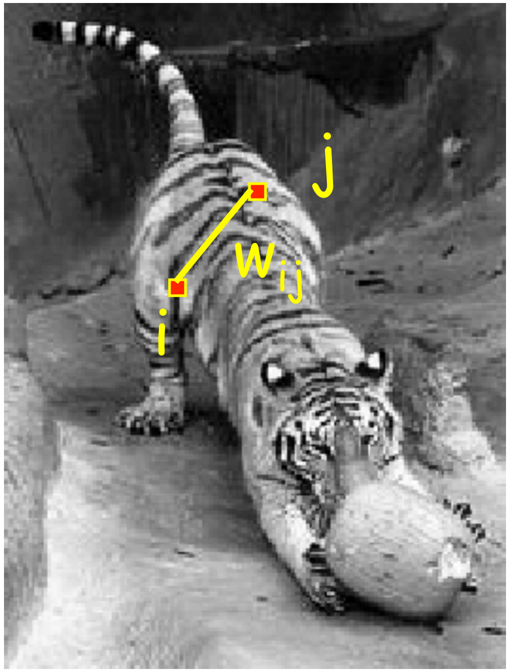
Marcello Pelillo
University of Venice, Italy

Image and Video Understanding

a.y. 2018/19

Images as graphs



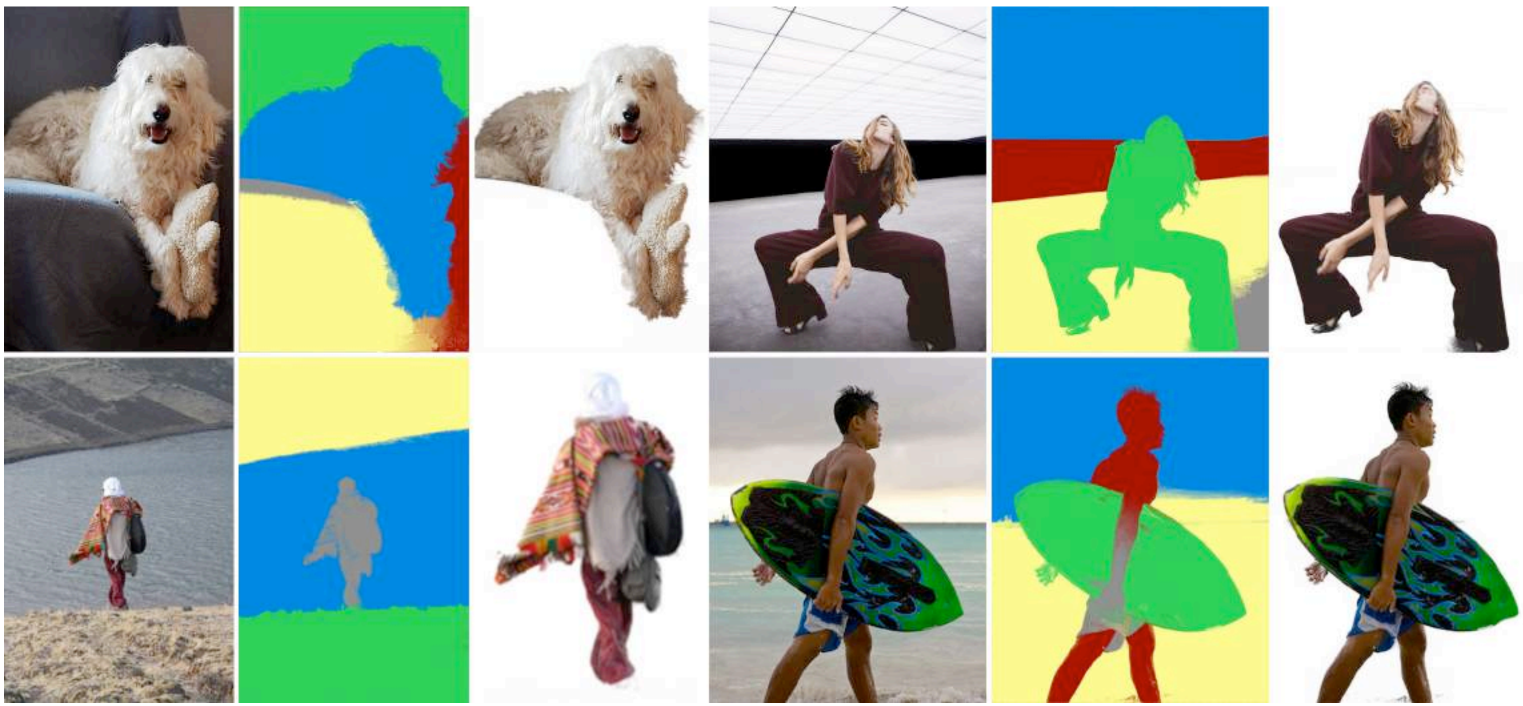


- Node for every pixel
- Edge between every pair of pixels (or every pair of "sufficiently close" pixels)
- Each edge is weighted by the *affinity* or similarity of the two nodes

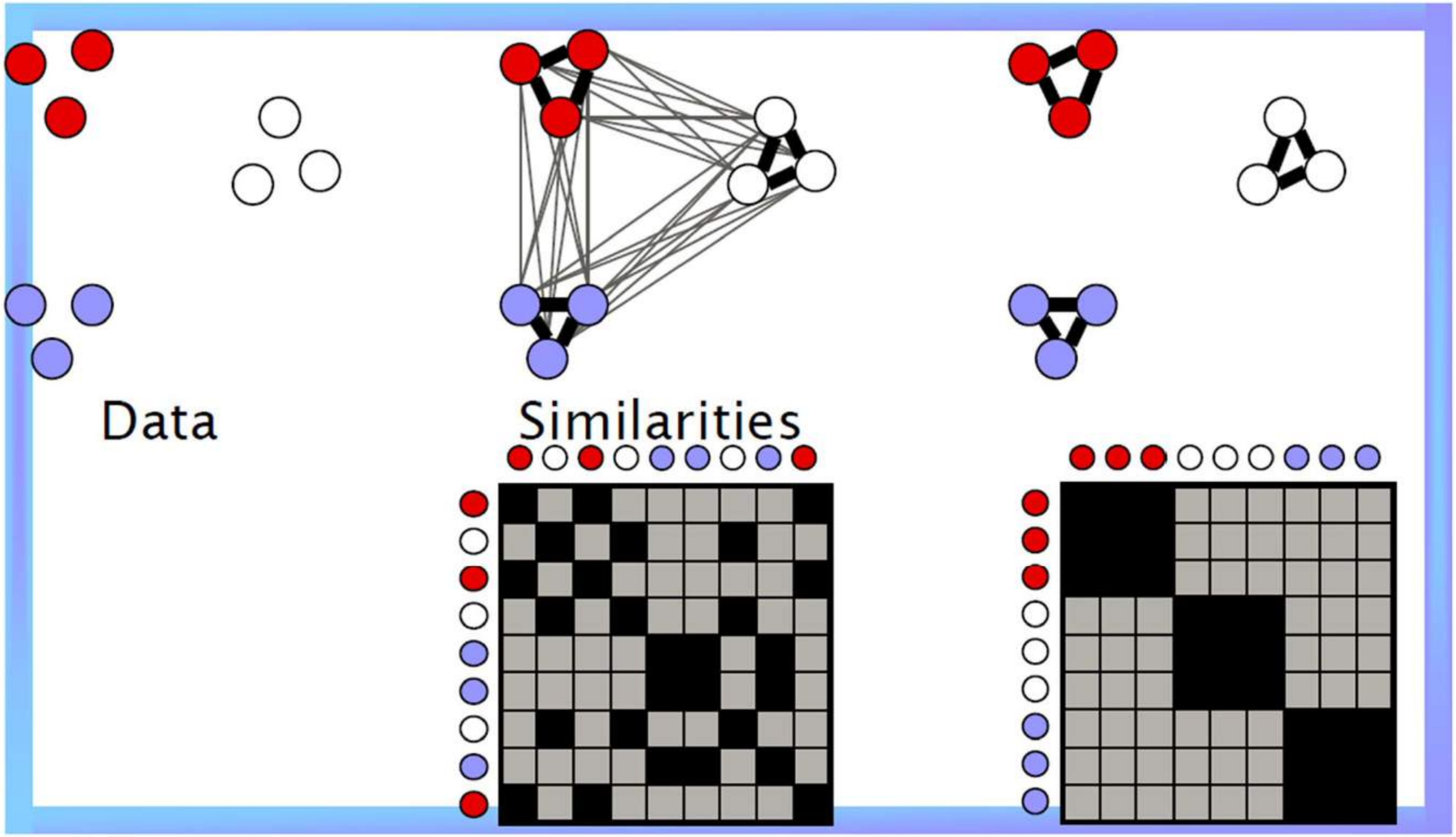
Source: S. Seitz

Graph-theoretic segmentation

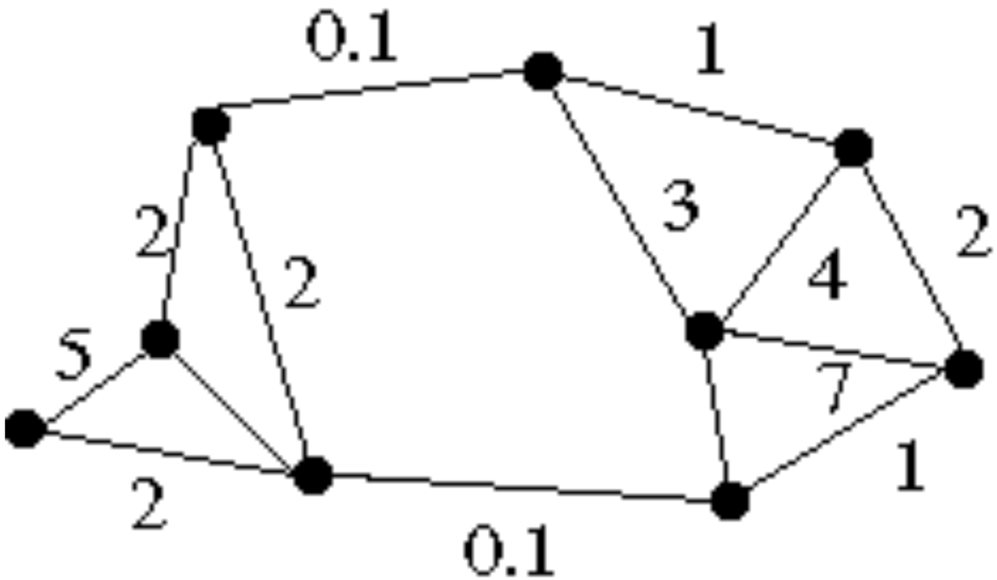
- Represent tokens using a weighted graph.
  - affinity matrix
- Cut up this graph to get subgraphs with strong interior links

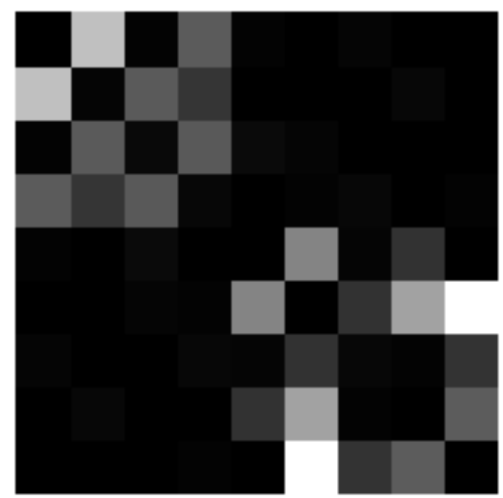


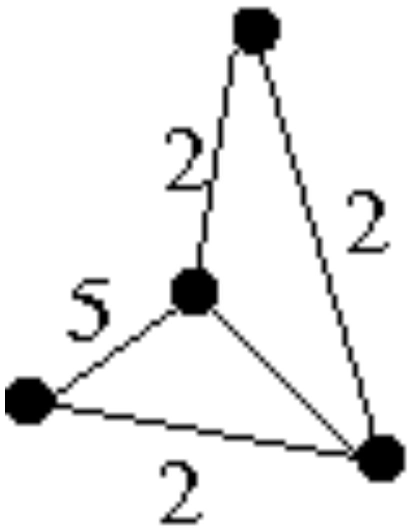
Graphs and matrices

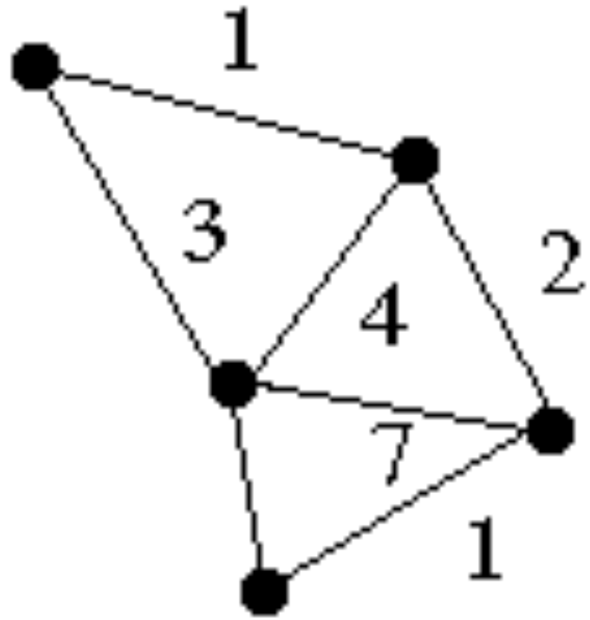


Source: D. Sontag


0.1
1
2
2
3
2
4
5
7
1
2
0.1




2
2
5
2


1
3
2
4
7
1

Measuring affinity

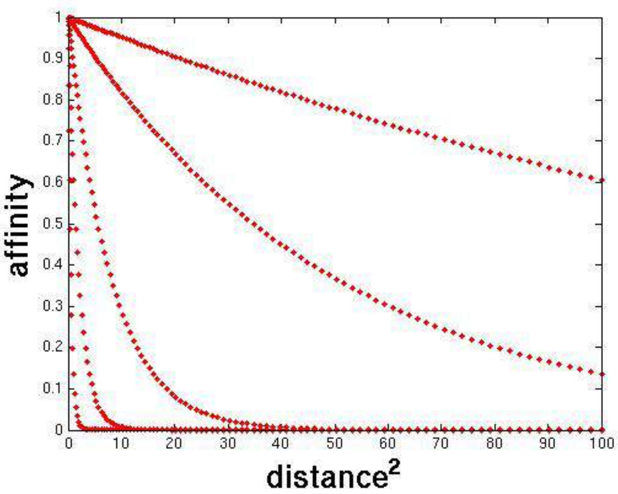
- Suppose we represent each pixel by a feature vector **x**, and define a distance function appropriate for this feature representation

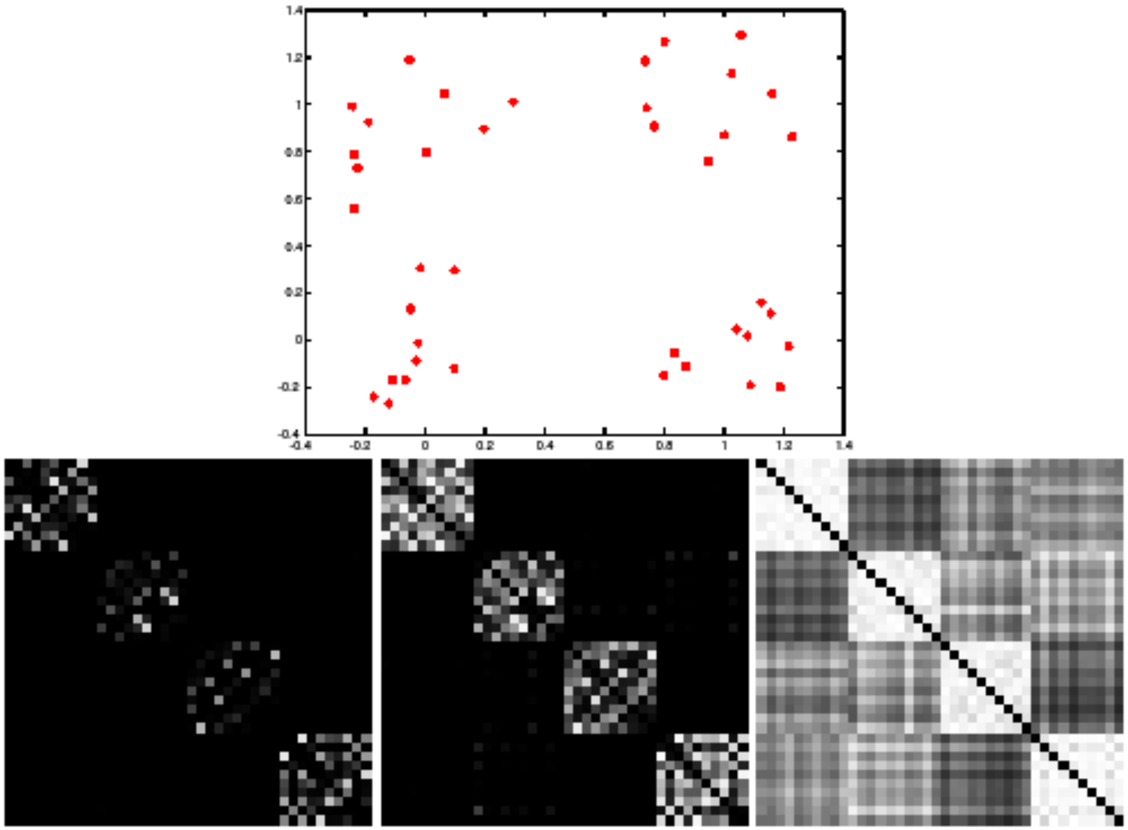
- Then we can convert the distance between two feature vectors into an affinity with the help of a Gaussian kernel:

$$\exp\left(-\frac{1}{2\sigma^2}\operatorname{dist}(\mathbf{x}_i, \mathbf{x}_j)^2\right)$$

Scale affects affinity

- Small σ: group only nearby points
- Large σ: group far-away points





Eigenvector-based clustering

Let us represent a cluster using a vector **x** whose *k*-th entry captures the participation of node *k* in that cluster. If a node does not participate in a cluster, the corresponding entry is zero.

We also impose the restriction that $\mathbf{x}^T\mathbf{x} = 1$

We want to maximize:

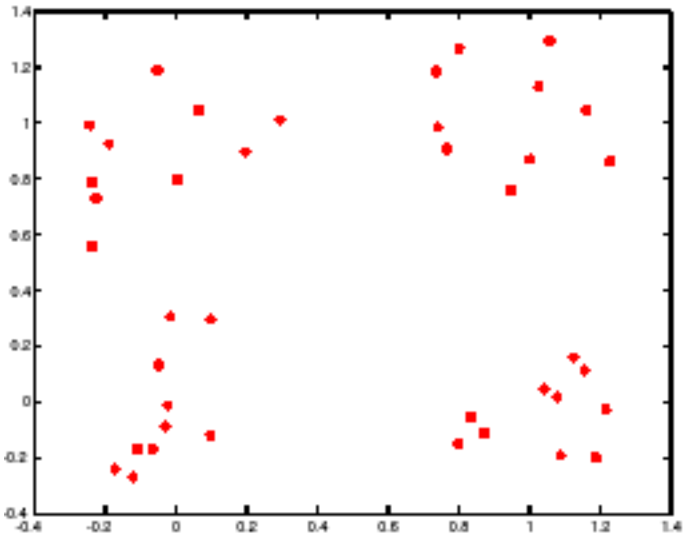
$$\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j = \mathbf{x}^T \mathbf{A} \mathbf{x}$$

which is a measure for the cluster's cohesiveness.

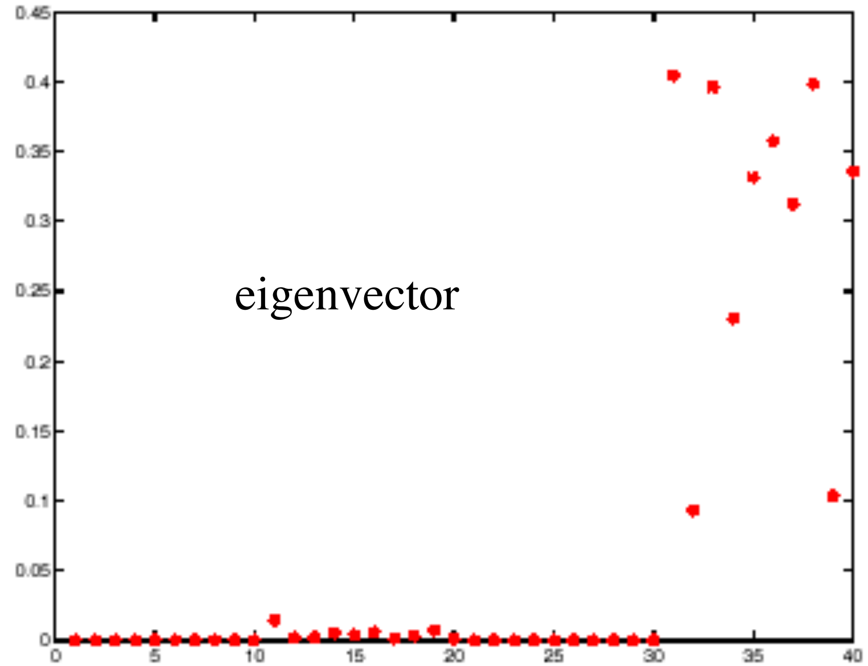
This is an **eigenvalue problem**!

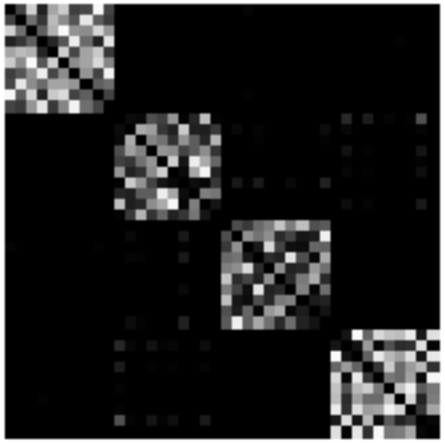
Choose the eigenvector of A with largest eigenvalue

Example eigenvector



points





matrix

More than two segments

- Two options
  - Recursively split each side to get a tree, continuing till the eigenvalues are too small
  - Use the other eigenvectors

Segmentation by eigenvectors: Algorithm

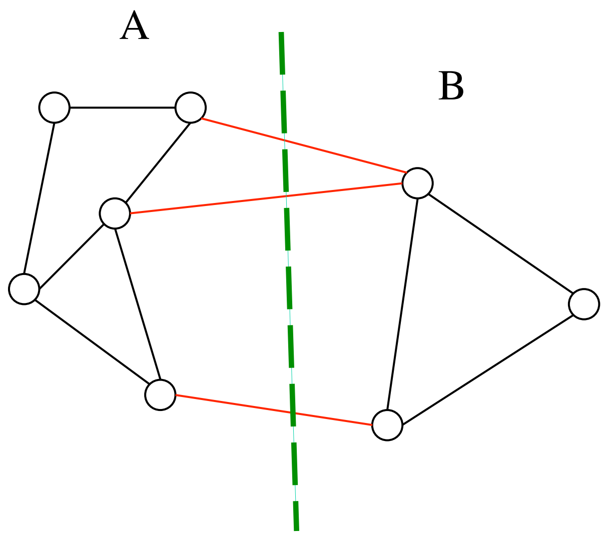
1. Construct (or take as input) the affinity matrix A
2. Compute the eigenvalues and eigenvectors of A
3. Repeat
4. Take the eigenvector corresponding to the largest unprocessed eigenvalue
5. Zero all components corresponding to elements that have already been clustered
6. Threshold the remaining components to determine which elements belong to this cluster
7. If all elements have been accounted for, there are sufficient clusters
8. Until there are sufficient clusters

Segmentation as graph partitioning

Let $G=(V, E, w)$ a weighted graph.

Given a "cut" (A, B), with $B = V \setminus A$, define:

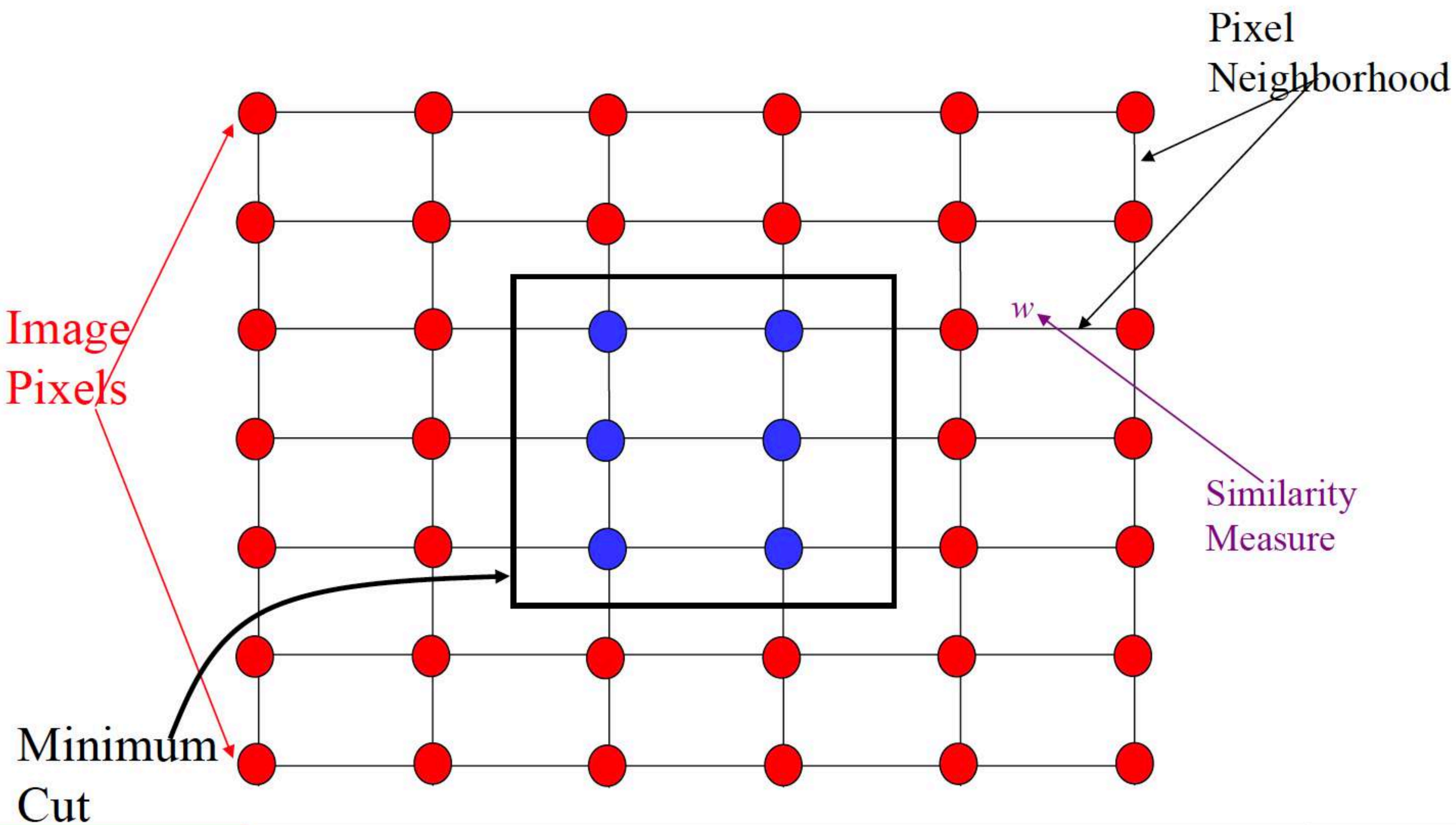
$$cut(A,B) = \sum_{i \in A} \sum_{j \in B} w(i,j)$$



Minimum Cut Problem

Among all possible cuts (A, B),
find the one which minimizes $cut(A, B)$

Segmentation as graph partitioning



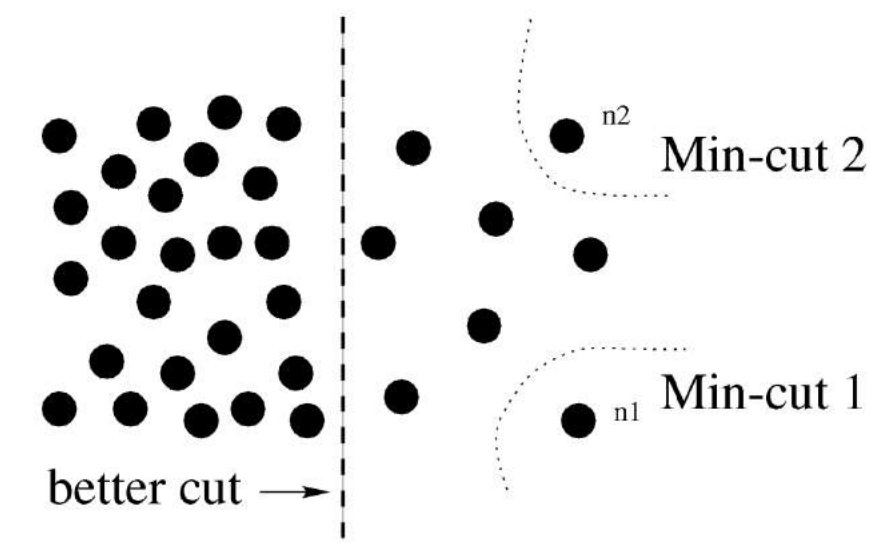
MinCut clustering

Good news

Solvable in polynomial time

Bad news

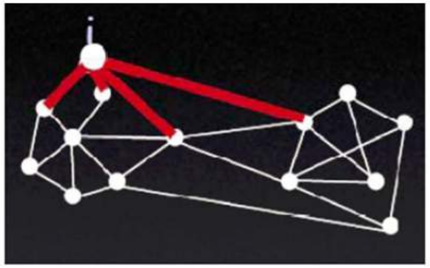
Favors highly unbalanced clusters (often with isolated vertices)

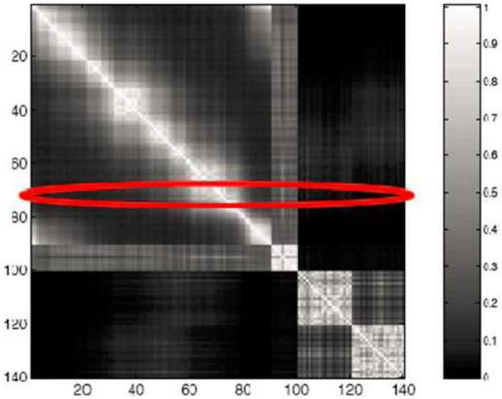


Graph terminology

Degree of nodes

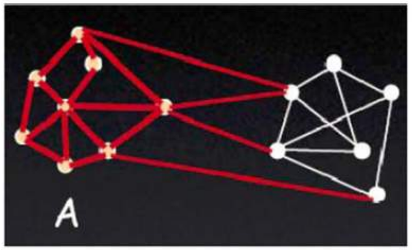
$$d_i = \sum_j w_{i,j}$$

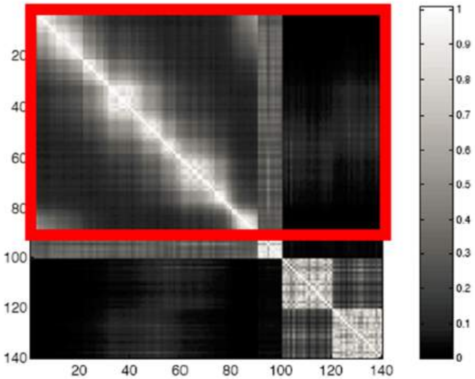




Volume of a set

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

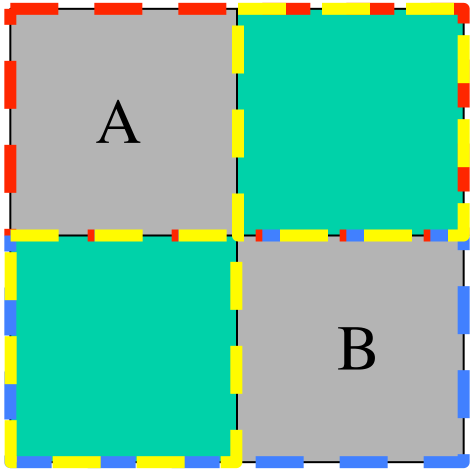




Adapted from D. Sontag

Normalized Cut

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



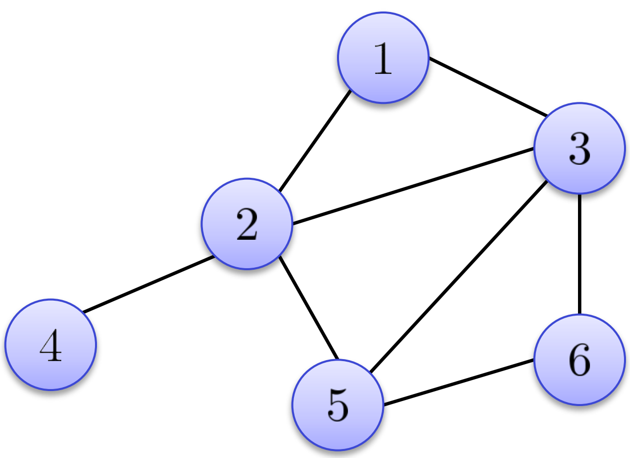
Adapted from D. Sontag

Graph Laplacian (unnormalized)

Defined as

$$L = D - W$$

Example:



$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 \\ -1 & -1 & 4 & 0 & -1 & -1 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 3 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix}$$

Assume the weights of edges are 1

Key fact

For all vectors f in $\mathbf{R}^n$, we have:

$$f^\top L f = \frac{1}{2} \sum_{ij=1}^{n} w_{ij} (f_i - f_j)^2$$

Indeed:

$$\begin{aligned} f^\top L f &= f^\top D f - f^\top W f \\ &= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\ &= \frac{1}{2} \left(\sum_i (\sum_j w_{ij}) f_i^2 - 2 \sum_{ij} f_i f_j w_{ij} + \sum_j (\sum_i w_{ij}) f_j^2 \right) \\ &= \frac{1}{2} \sum_{ij} w_{ij} (f_i - f_j)^2 \end{aligned}$$

Properties

- L is symmetric (by assumption) and positive semi-definite:

  $$f' L f \geq 0$$

  for all vectors f (by "key fact")
- Smallest eigenvalue of L is 0; corresponding eigenvector is **1**
- Thus eigenvalues are: $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

First relation between spectrum and clusters:

- Multiplicity of eigenvalue $\lambda_1 = 0$ is the number of connected components of the graph
- eigenspace is spanned by the characteristic functions of these components (so all eigenvectors are piecewise constant)

Normalized graph Laplacians

- Row sum (random walk) normalization:

$$
\begin{aligned}
L_{\mathrm{rw}} &= D^{-1} L \\
&= I - D^{-1} W
\end{aligned}
$$

- Symmetric normalization:

$$
\begin{aligned}
L_{\mathrm{sym}} &= D^{-1/2} L \, D^{-1/2} \\
&= I - D^{-1} W \, D^{-1/2}
\end{aligned}
$$

Spectral properties of both matrices similar to the ones of L.

Solving Ncut

Any cut (A, B) can be represented by a binary indicator vector x:

$$x_i = \begin{cases} +1 & \text{if } i \in A \\ -1 & \text{if } i \in B \end{cases}$$

It can be shown that:

Rayleigh quotient

$$\min_x Ncut(x) = \min_y \frac{y'(D-W)y}{y'Dy}$$

subject to the constraint that $y'D\mathbf{1} = \sum_i y_i d_i = 0$ (with $y_i \in \{1, -b\}$).

This is **NP-hard!**

Ncut as an eigensystem

If we *relax* the constraint that y be a discrete-valued vector and allow it to take on real values, the problem

$$\min_y \frac{y'(D-W)y}{y'Dy}$$

is equivalent to:

$$\min_y y'(D-W)y \quad \text{subject to} \quad y'Dy = 1$$

This amounts to solving a *generalized* eigenvalue problem:

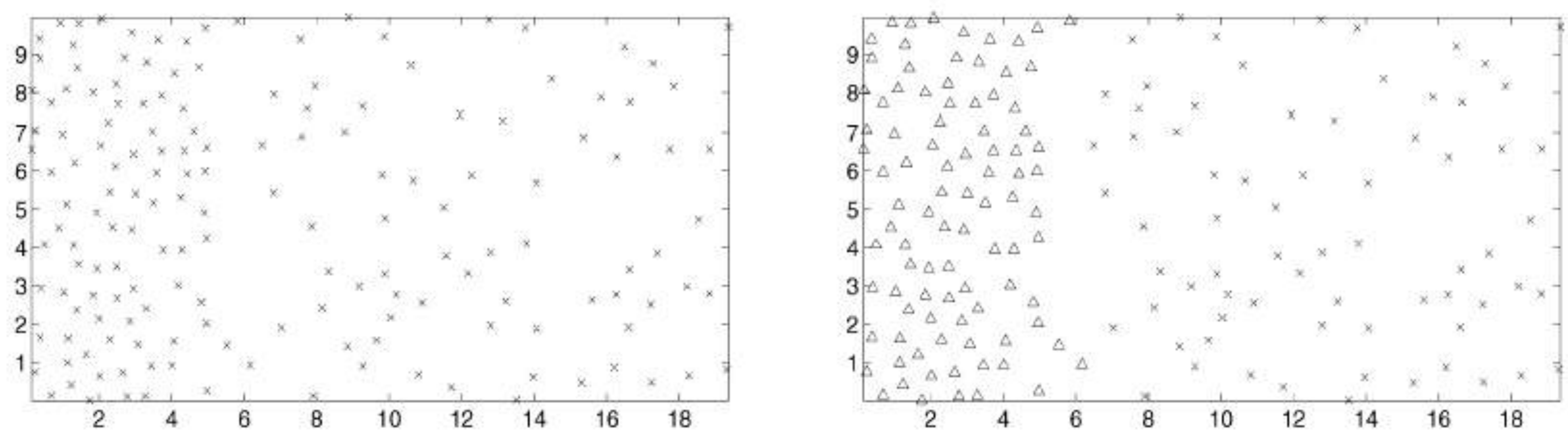
Laplacian ← $$(D-W)y = \lambda D y$$

Note: Equivalent to a *standard* eigenvalue problem using the normalized Laplacian: $L_{\mathrm{rw}} = D^{-1} L = I - D^{-1} W$.

2-way Ncut

1. Compute the affinity matrix W, compute the degree matrix D
2. Solve the generalized eigenvalue problem $(D - W)y = \lambda Dy$
3. Use the eigenvector associated to the second smallest eigenvalue to bipartition the graph into two parts.

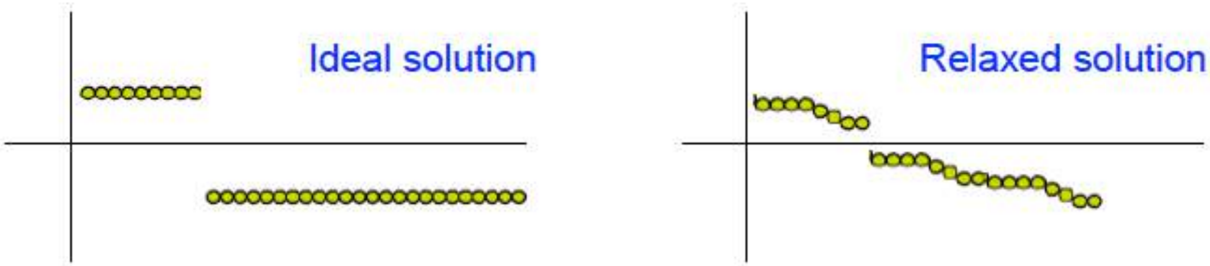
Why the *second* smallest eigenvalue?
Remember, the smallest eigenvalue of Laplacians is always 0
(corresponds to the trivial partition $A = V, B = \{\}$)

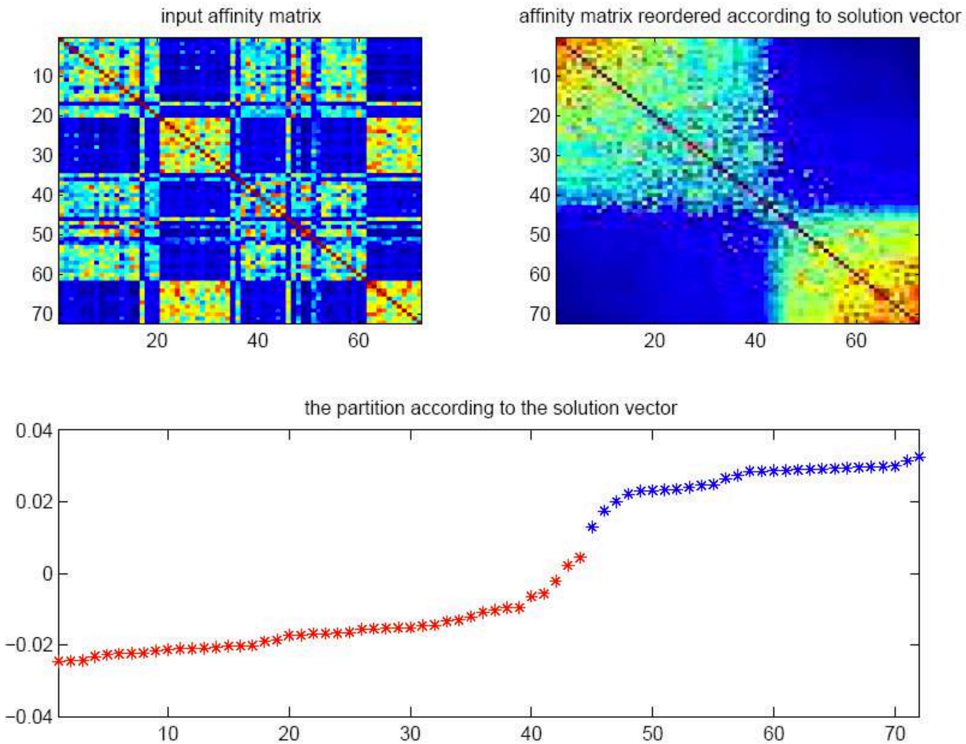


(a) (b)

Fig. 5. (a) Point set generated by two Poisson processes, with densities of 2.5 and 1.0 on the left and right clusters respectively, (b) $\triangle$ and $\times$ indicate the partition of point set in (a). Parameter settings: $\sigma_X = 5$, $r = 3$.

The effect of relaxation





How to choose the splitting point?

- Pick a constant value (0 or 0.5)
- Pick the median value as splitting point
- Look for the splitting point that has minimum Ncut value:
  1. Choose n possible splitting points
  2. Compute Ncut value
  3. Pick minimum

Random walk intepretation

Problem: Finding a cut (A, B) in a graph G such that a random walk does not have many opportunities to jump between the two clusters.

This is equivalent to the *Ncut* problem due to the following relation:

$$Ncut(A, B) = P(A \mid B) + P(B \mid A)$$

(Meila and Shi, 2001)

Ncut: More than 2 clusters

Approach #1: Recursive two-way cuts

1. Given a weighted graph $G = (V, E, w)$, summarize the information into matrices W and D

2. Solve $(D - W)y = \lambda Dy$ for eigenvectors with the smallest eigenvalues

3. Use the eigenvector with the second smallest eigenvalue to bipartition the graph by finding the splitting point such that Ncut is minimized

4. Decide if the current partition should be subdivided by checking the stability of the cut, and make sure Ncut is below the prespecified value

5. Recursively repartition the segmented parts if necessary

Note. The approach is computationally wasteful; only the second eigenvector is used, whereas the next few small eigenvectors also contain useful partitioning information.

Ncut: More than 2 clusters

Approach #2: Using first k eigenvectors

1. Construct a similarity graph and compute the unnormalized graph Laplacian L.
2. Compute the k smallest **generalized** eigenvectors $u_1, u_2, \cdots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.
3. Let $U = [\, u_1 \; u_2 \; \cdots \; u_k \,] \in \mathbb{R}^{n \times k}$.
4. Let $y_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of U.

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nk} \end{bmatrix} = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix}$$

5. Thinking of y_i's as points in $\mathbb{R}^k$, cluster them with k-means algorithms.

Fig. 2. A gray level image of a baseball game.

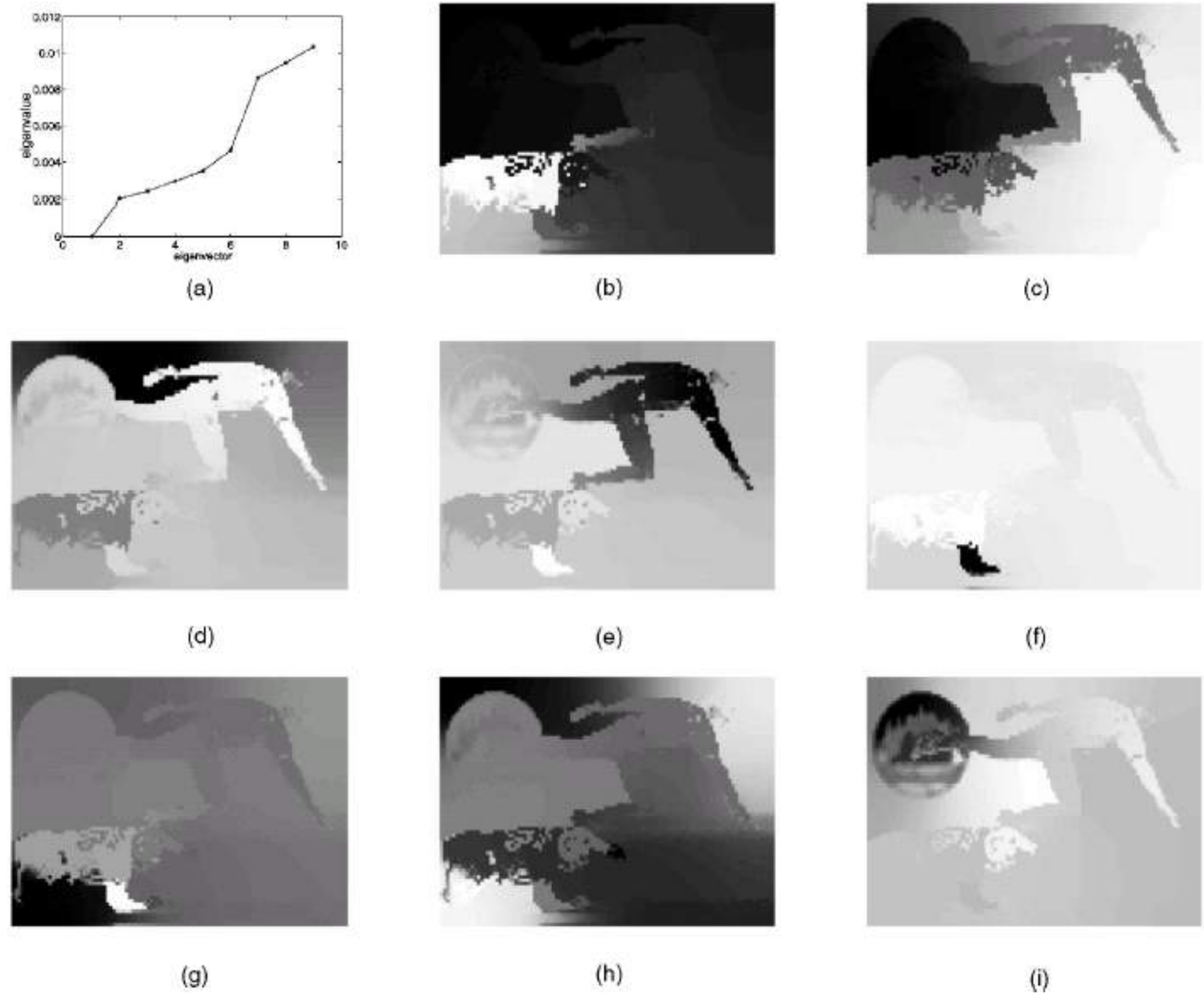


Fig. 3. Subplot (a) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplots (b)-(i) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

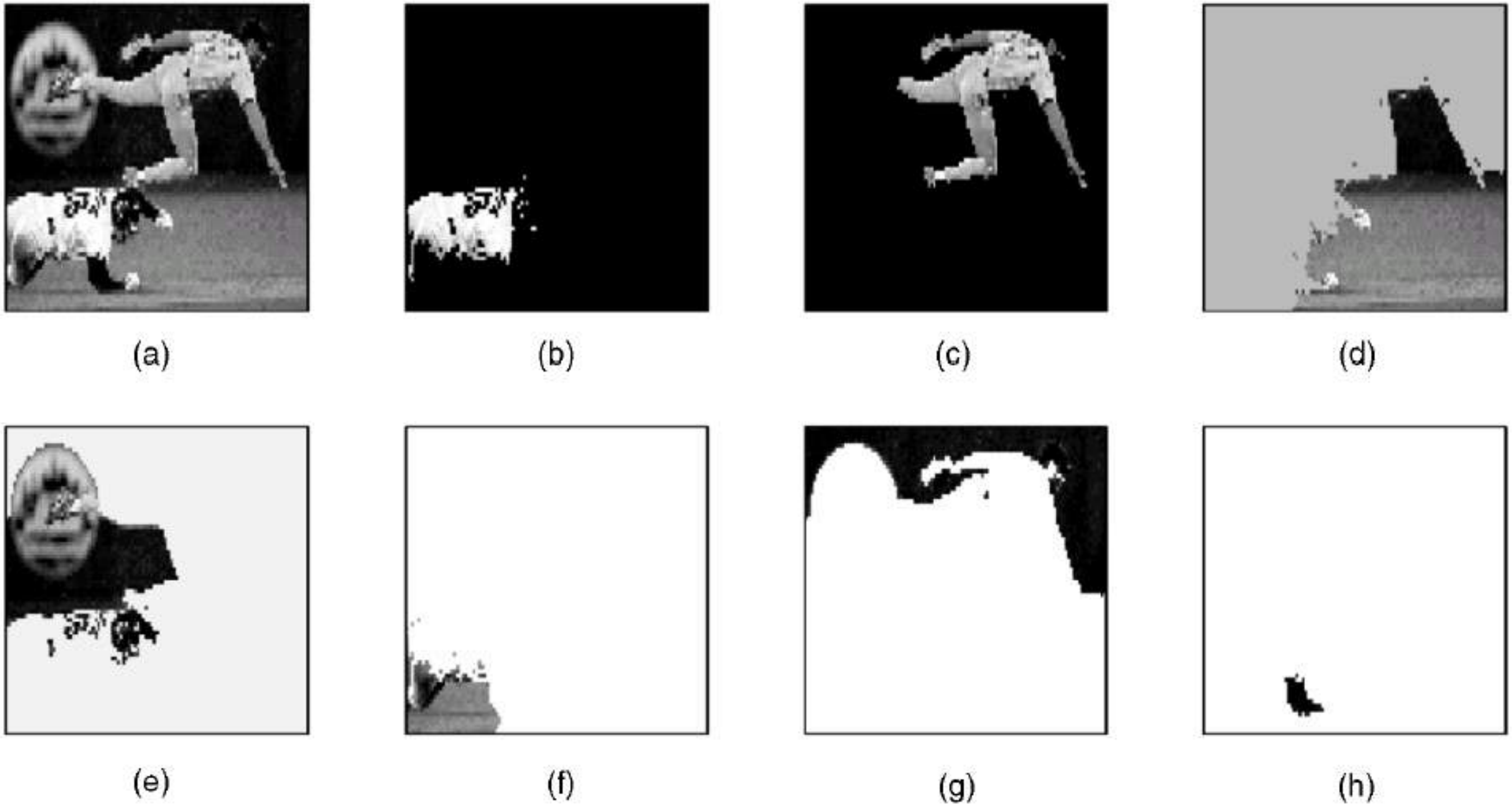


Fig. 4. (a) shows the original image of size 80×100. Image intensity is normalized to lie within 0 and 1. Subplots (b)-(h) show the components of the partition with $Ncut$ value less than 0.04. Parameter setting: $\sigma_I = 0.1$, $\sigma_X = 4.0$, $r = 5$.

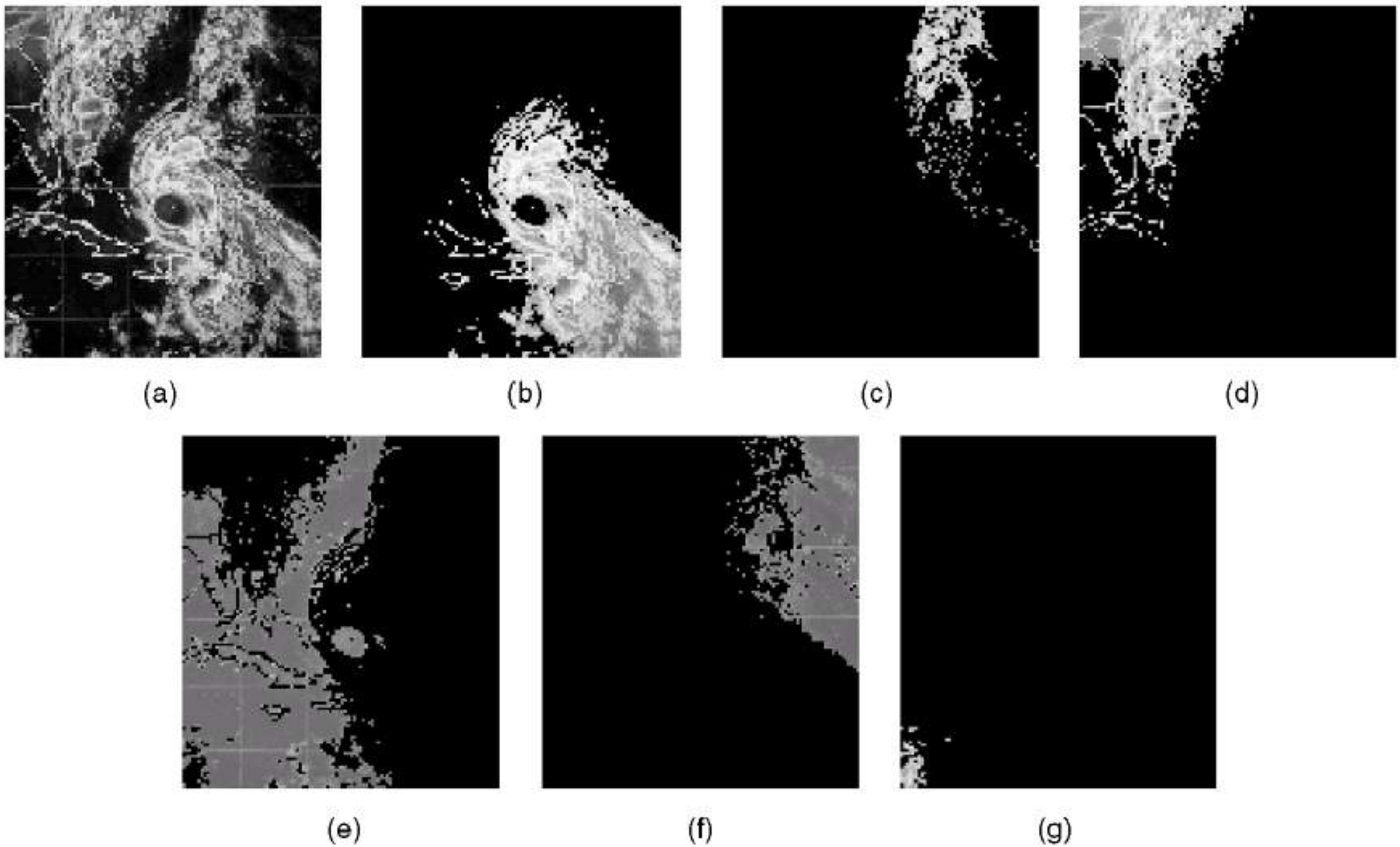


Fig. 8. (a) shows a 126×106 weather radar image. (b)-(g) show the components of the partition with $Ncut$ value less than 0.08. Parameter setting: $\sigma_I = 0.007$, $\sigma_x = 15.0$, $r = 10$.

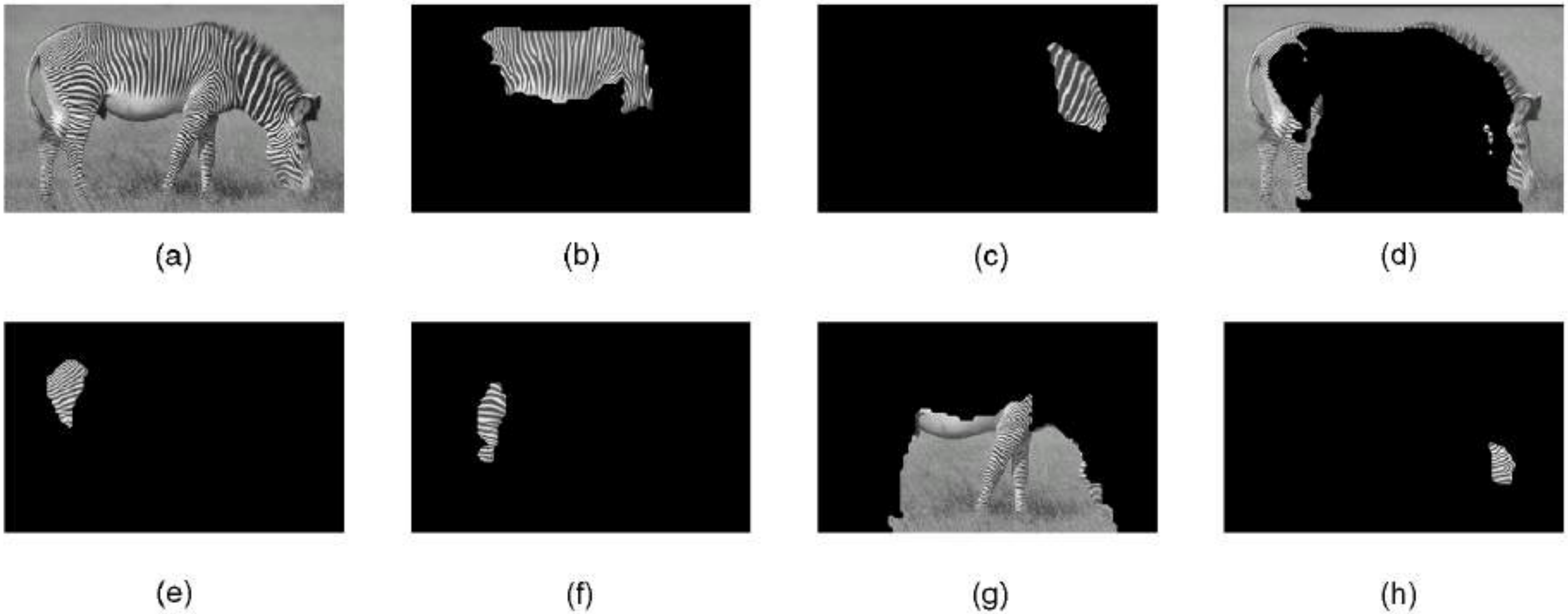


Fig. 10. (a) shows an image of a zebra. The remaining images show the major components of the partition. The texture features used correspond to convolutions with DOOG filters [16] at six orientations and five scales.

Spectral clustering

1. Construct a similarity graph and compute the normalized graph Laplacian L_{sym}.
2. Compute the k smallest eigenvectors $u_1, u_2, \cdots, u_k$ of L_{sym}.
3. Let $U = [\, u_1 \; u_2 \; \cdots \; u_k \,] \in \mathbb{R}^{n \times k}$.
4. Normalized the rows of U to norm 1.

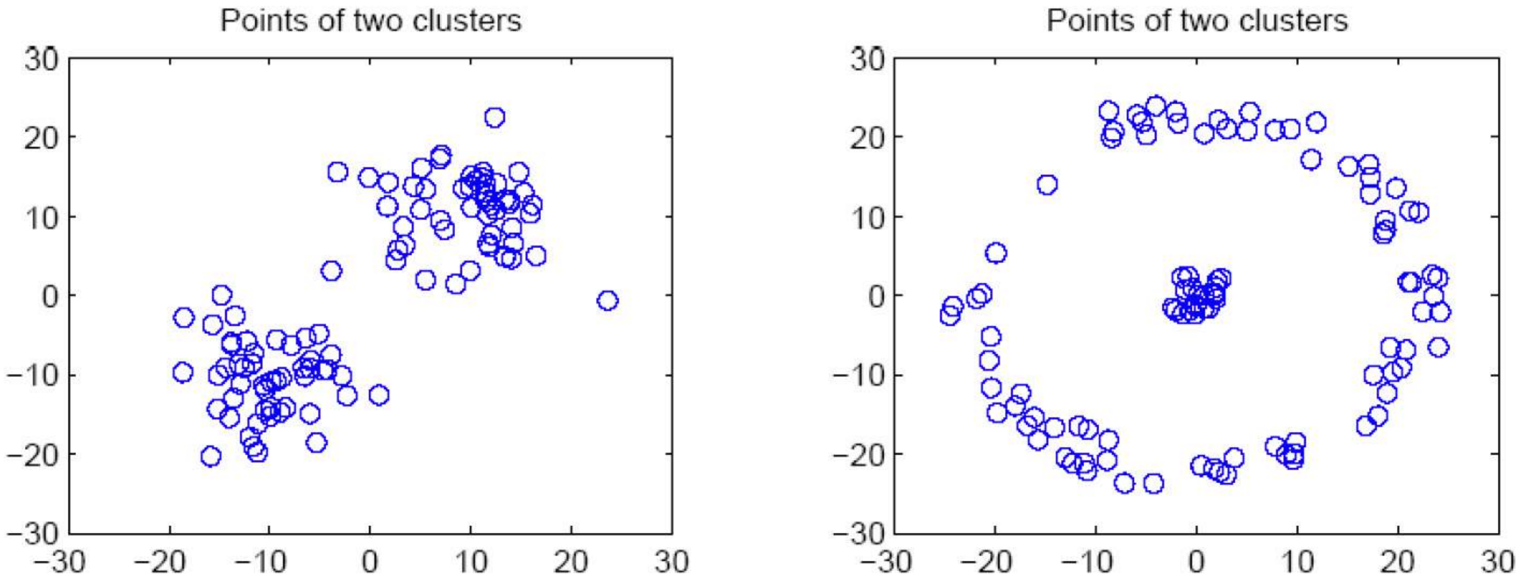
$$U_{ij} \leftarrow \frac{U_{ij}}{(\sum_k U_{ik}^2)^{1/2}}$$

5. Let $y_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of U.
6. Thinking of y_i's as points in $\mathbb{R}^k$, cluster them with k-means algorithms.

Ng, Jordan and Weiss (2002)

K-means *vs* Spectral clustering

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.



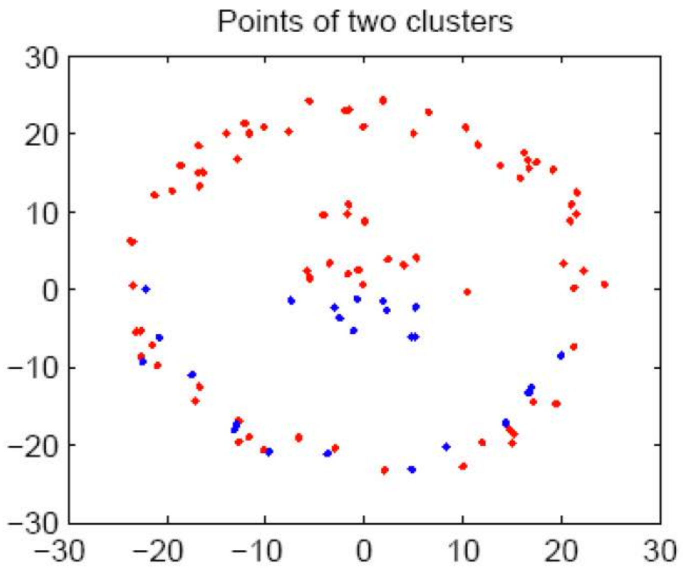
Both perform same

Spectral clustering is superior

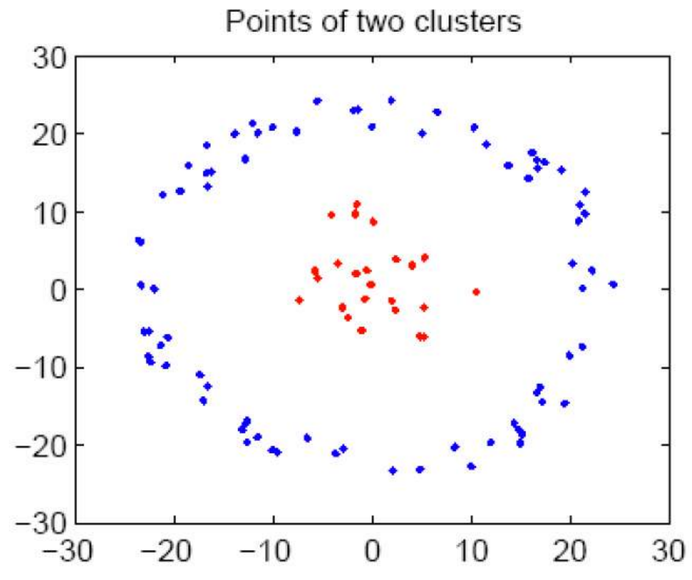
Adapted from A. Singh

K-means *vs* Spectral clustering

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.



k-means output

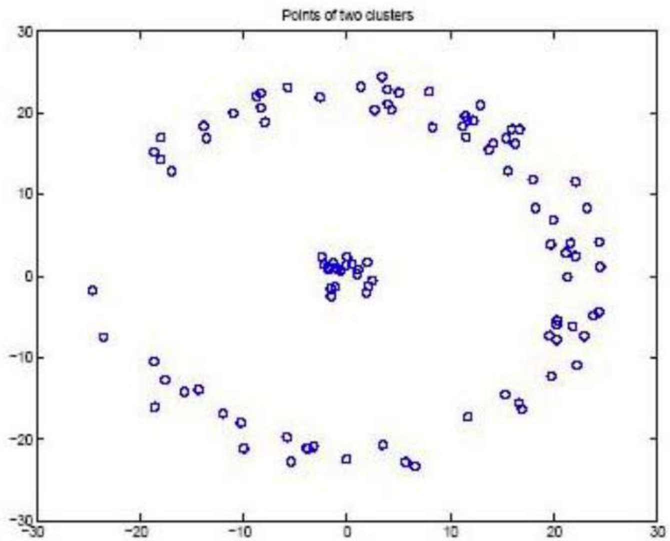


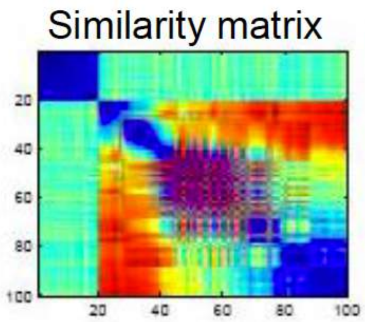
Spectral clustering output

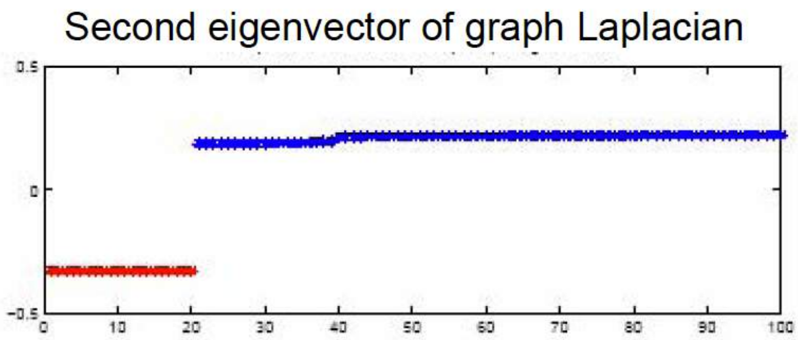
Adapted from A. Singh

K-means *vs* Spectral clustering

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.

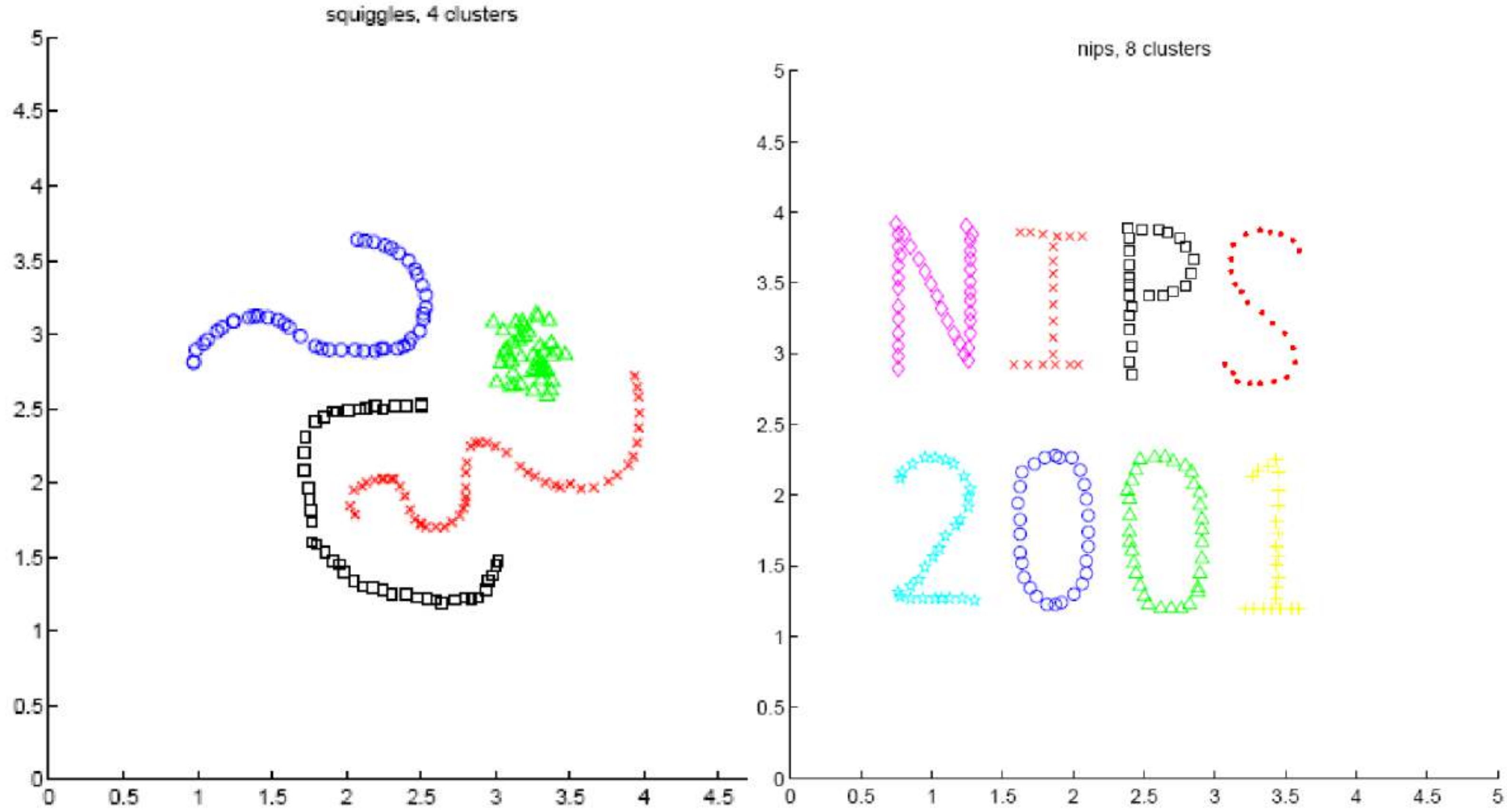




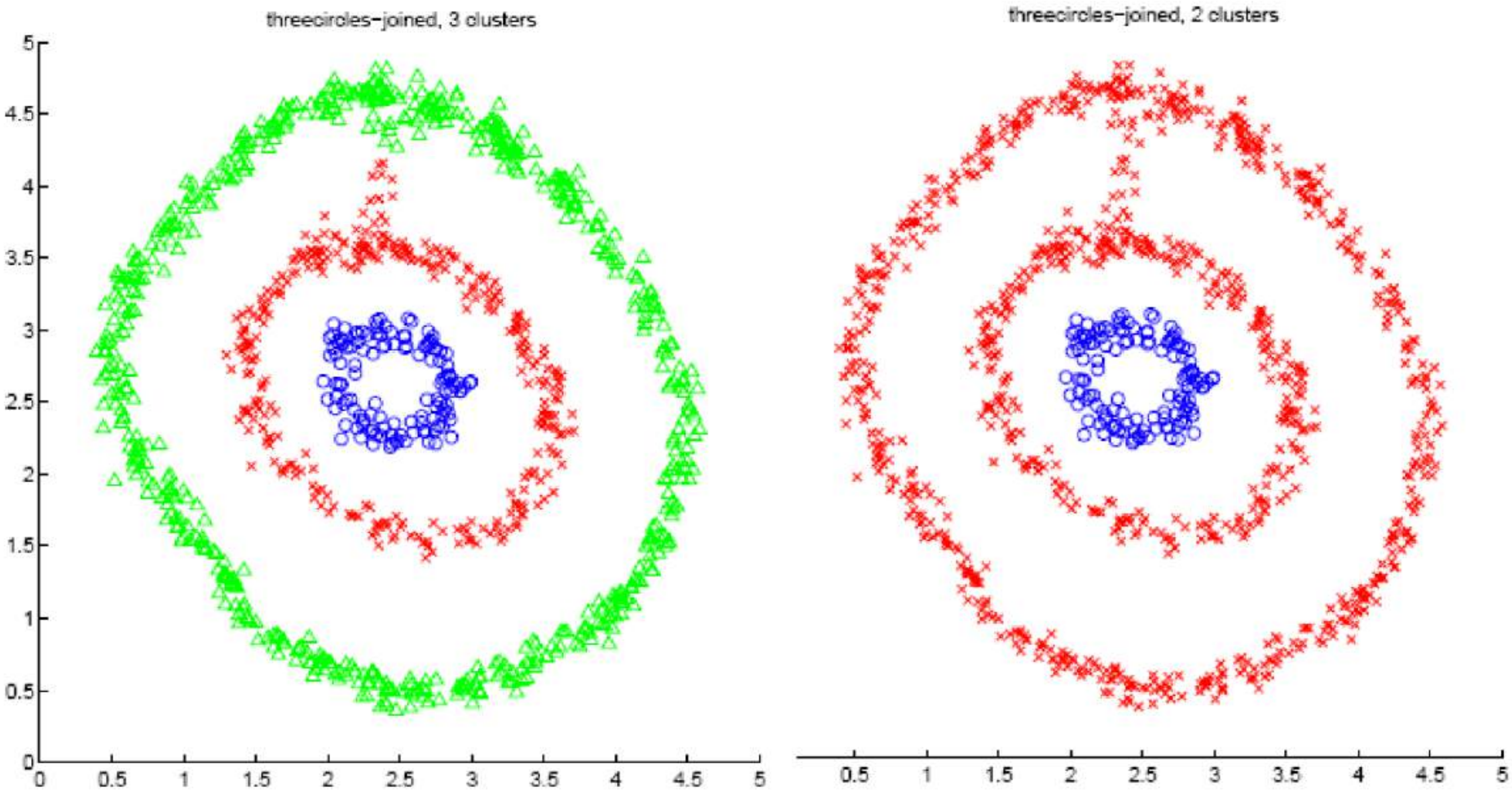


Adapted from A. Singh

Examples

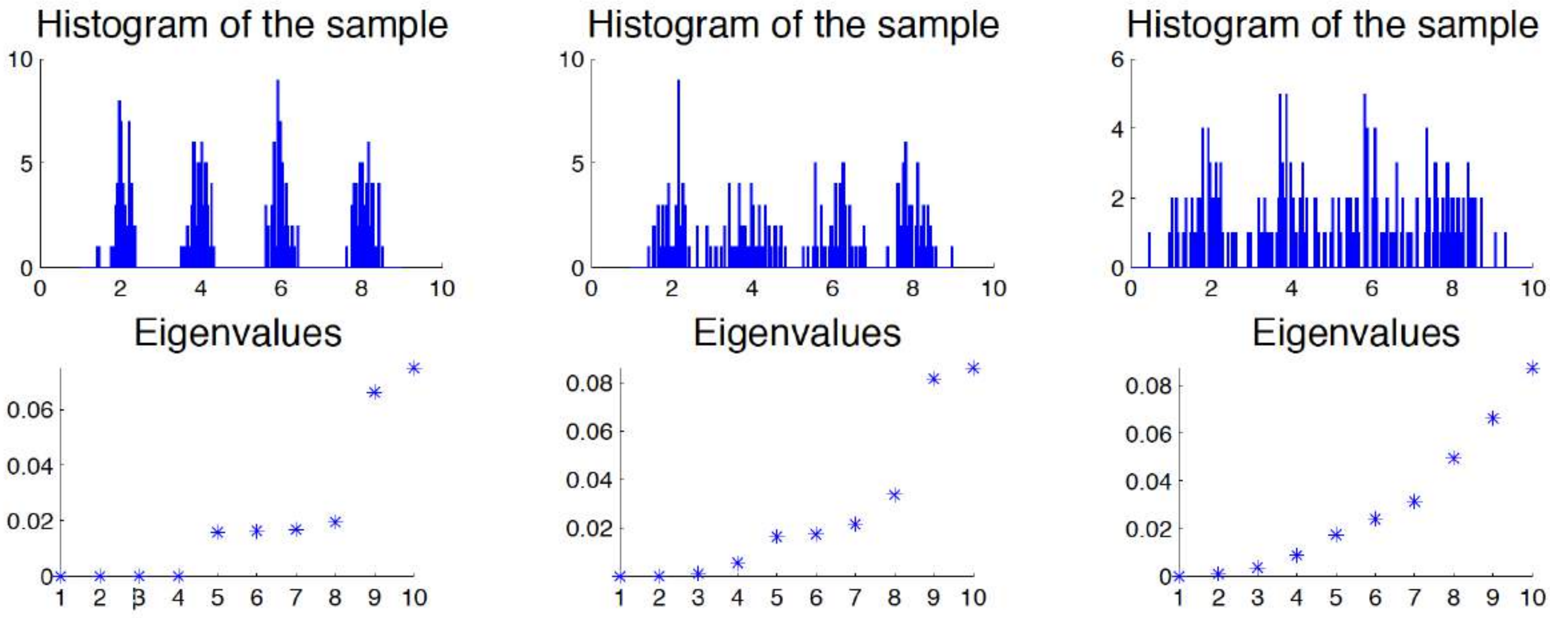


Examples (choice of k)



Choosing k

The eigengap heuristic: Choose k such that all eigenvalues $\lambda_1, \cdots, \lambda_k$ are very small, but λ_{k+1} is relatively large



Four 1D Gaussian clusters with increasing variance and corresponding eigevalues of L_{rw} (von Luxburg, 2007).

References

- J. Shi and J. Malik, Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8): 888-905 (2000).

- M. Meila and J. Shi. A random walks view of spectral segmentation. *AISTATS* (2001).

- A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and analgorithm. *NIPS 14* (2002).

- U. von Luxburg, A tutorial on spectral clustering. *Statistics and Computing* 17(4) 395-4*16* (2007).

- A. K. Jain, Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31(8):651-666 (2010).

Graph-based Methods in Computer Vision: Recent Advances

Marcello Pelillo

Ca' Foscari University of Venice, Italy

European Centre for Living Technology

Huawei Video Intelligence Forum, Dublin, Ireland, October 23, 2018

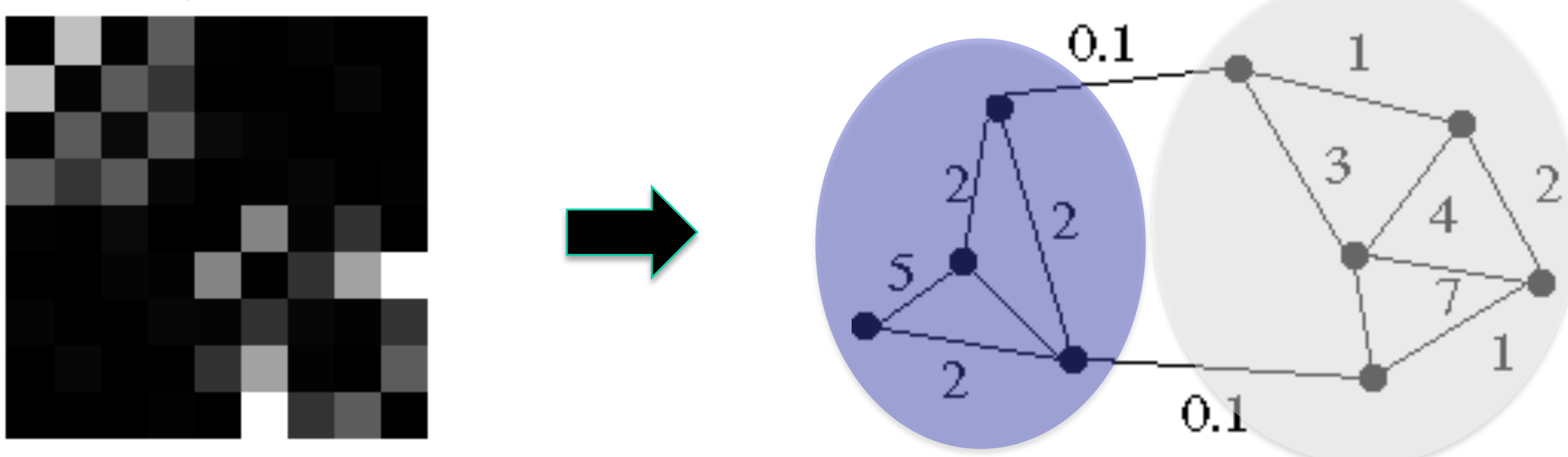
Clustering on Graphs

Given:

- a set of *n* "objects"
- an *n* × *n* matrix A of pairwise similarities

} = an edge-weighted graph

Goal: Group the the input objects (the vertices of the graph) into maximally homogeneous classes (i.e., clusters).



What is a Cluster?

No universally accepted (formal) definition of a “cluster” but, informally, a cluster should satisfy two criteria:

Internal criterion

all “objects” inside a cluster should be highly similar to each other

External criterion

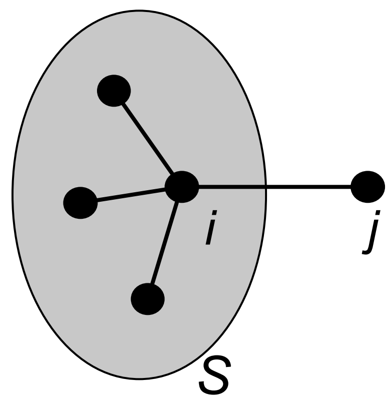
all “objects” outside a cluster should be highly dissimilar to the ones inside

How to formalize these criteria?

Basic Definitions

Let $S \subseteq V$ be a non-empty subset of vertices, and $i \in S$.

The **(average) weighted degree** of *i* w.r.t. *S* is defined as: $\text{awdeg}_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij}$



Moreover, if $j \notin S$, we define:

$$\phi_S(i,j) = a_{ij} - \text{awdeg}_S(i)$$

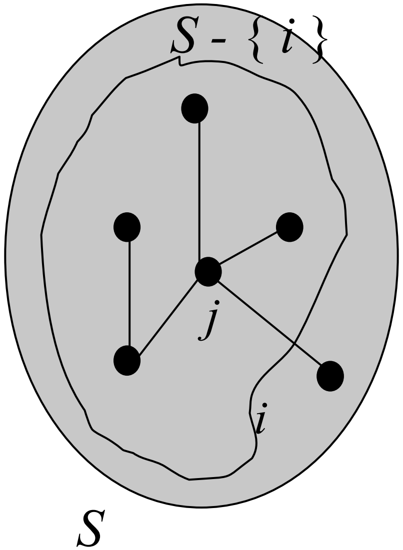
Intuitively, $\Phi_S(i,j)$ measures the similarity between vertices *j* and *i*, with respect to the (average) similarity between vertex *i* and its neighbors in *S*.

Basic Definitions

Let $S \subseteq V$ be a non-empty subset of vertices, and $i \in S$.

The **weight** of *i* w.r.t. *S* is defined as:

$$w_S(i) = \begin{cases} 1 & \text{if } |S| = 1 \\ \sum_{j \in S - \{i\}} \phi_{S-\{i\}}(j,i) w_{S-\{i\}}(j) & \text{otherwise} \end{cases}$$

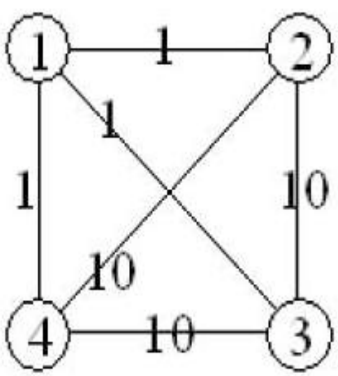


Further, the **total weight** of *S* is defined as:

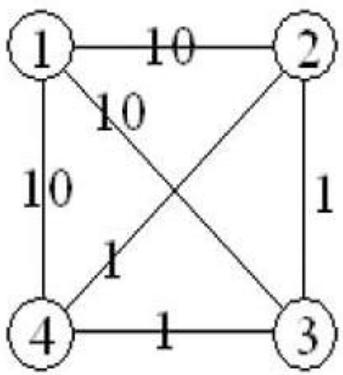
$$W(S) = \sum_{i \in S} w_S(i)$$

Interpretation

Intuitively, $w_S(i)$ gives us a measure of the overall (relative) similarity between vertex *i* and the vertices of S \ {*i*} with respect to the overall similarity among the vertices in S \ {*i*}.



$w_{\{1,2,3,4\}}(1) < 0$



$w_{\{1,2,3,4\}}(1) > 0$

Dominant Sets

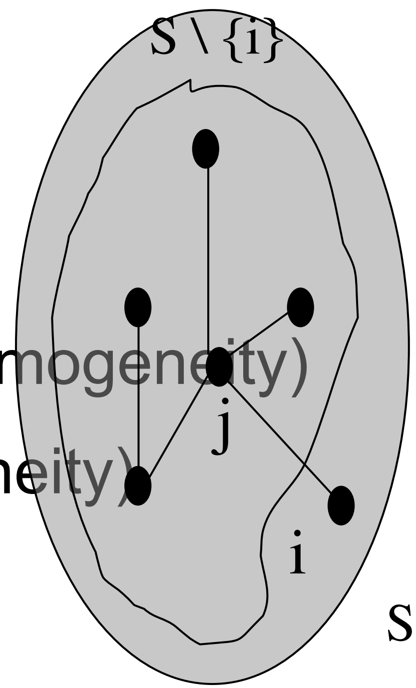
Let $S \subseteq V$ be a subset of vertices of a graph G and $i \in S$.

Define a measure for the similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall internal similarity of $S \setminus \{i\}$.

Call it $w_S(i)$.

S is said to be a **dominant set** if:

1. $w_S(i) > 0$, for all $i \in S$ (internal homogeneity)
2. $w_{S \cup \{i\}}(i) < 0$, for all $i \notin S$ (external homogeneity)



M. Pavan and M. Pelillo. Dominant sets and pairwise clustering (PAMI 2007)

Dominant Sets

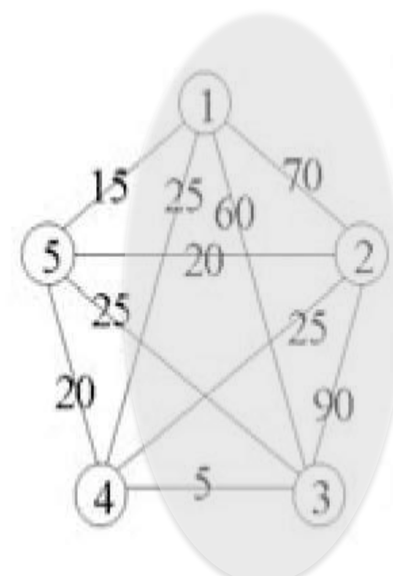
Let $S \subseteq V$ be a subset of vertices of a graph G and $i \in S$.

Define a measure for the similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall internal similarity of $S \setminus \{i\}$.

Call it $w_S(i)$.

S is said to be a **dominant set** if:

1. $w_S(i) > 0$, for all $i \in S$ (internal hoı
2. $w_{S \cup \{i\}}(i) < 0$, for all $i \notin S$ (external homoger



M. Pavan and M. Pelillo. Dominant sets and pairwise clustering (PAMI 2007)

The Many Facets of Dominant Sets

Dominant sets have intriguing connections wth:

- **Game theory**
  Nash equilibria of "clustering games"
- **Optimization theory**
  Local maximizers of (continuous) quadratic problems
- **Graph theory**
  Maximal cliques
- **Dynamical systems theory**
  Stable attractors of evolutionary game dynamics

See Rota Bulò and Pelillo (EJOR 2017) for a a review

Using Symmetric Affinities

Given a symmetric affinity matrix A, consider the following continuous quadratic optimization problem (QP):

$$\begin{array}{ll} \text{maximize} & f(\mathrm{x}) = \mathrm{x}' A \mathrm{x} \\ \text{subject to} & \mathrm{x} \in \Delta \end{array}$$

where Δ is the standard simplex (probability space).

The function $f(x)$ provides a measure of cohesiveness of a cluster.

Dominant sets are in one-to-one correspondence to (strict) local solutions of QP

Note. In the 0/1 case, dominant sets correspond to **maximal cliques**

Finding Dominant Sets

Replicator dynamics from evolutionary game theory are a popular and principled way to find DS's.

$$x_i(t+1) = x_i(t)\frac{A\big(x(t)\big)_i}{x(t)^T A x(t)}$$

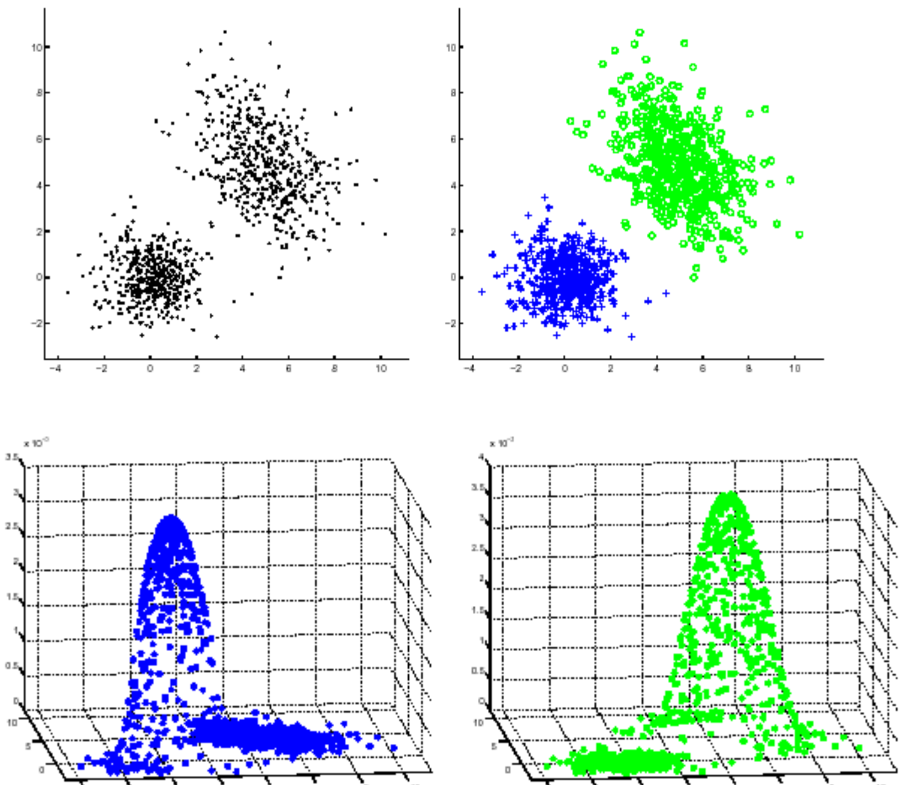
MATLAB implementation

```
distance=inf;
while distance>epsilon
    old_x=x;
    x = x.*(A*x);
    x = x./sum(x);
    distance=pdist([x,old_x]');
end
```

Faster dynamics available!
(See Rota Bulò and Pelillo, 2017)

Measuring Cluster Membership

The components of the converged vector **x** give us a measure of the participation of the corresponding vertices in the cluster, while the value of the objective function measures the cluster's cohe



Useful for ranking the
elements in the cluster!

In a Nutshell

The dominant-set approach to clustering:

- ✓ does not require *a priori* knowledge on the number of clusters
- ✓ is robust against outliers
- ✓ allows to rank the cluster's elements according to "centrality"
- ✓ allows extracting overlapping clusters *(ICPR'08)*
- ✓ generalizes naturally to hypergraph clustering problems *(PAMI'13)*
- ✓ makes no assumption on the structure of the similarity matrix, (works also with asymmetric and even negative

Some Computer Vision Applications

- Image and video segmentation
- Anomaly detection
- Video summarization
- Feature selection
- Image matching and registration
- 3D reconstruction
- Human action recognition
- Content-based image retrieval
- …

But also in neuroscience, bioinformatics, medical image analysis, etc.

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Detecting conversational groups in images and sequences: A robust game-theoretic approach☆

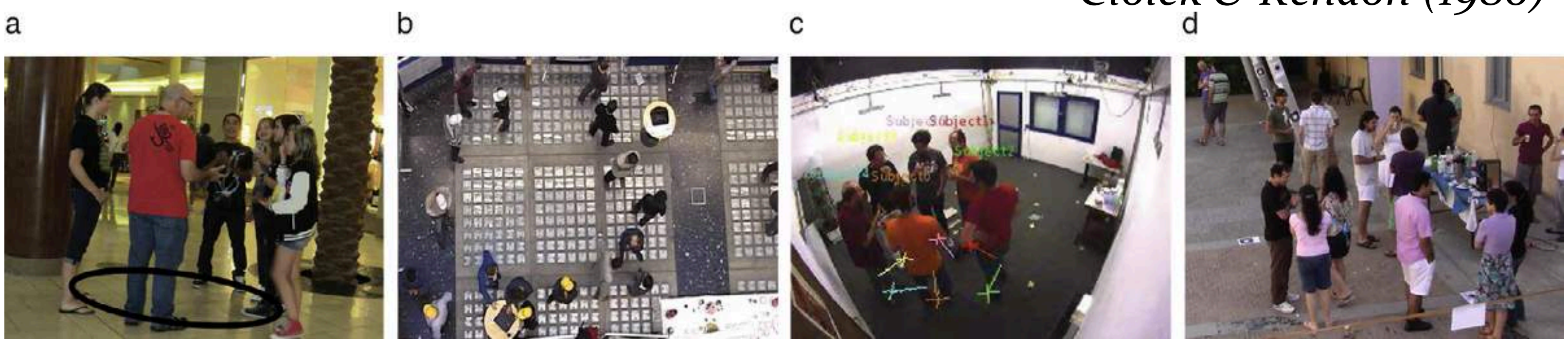
Sebastiano Vascon[a,*], Eyasu Z. Mequanint[b], Marco Cristani[a,c], Hayley Hung[d], Marcello Pelillo[b], Vittorio Murino[a,c]

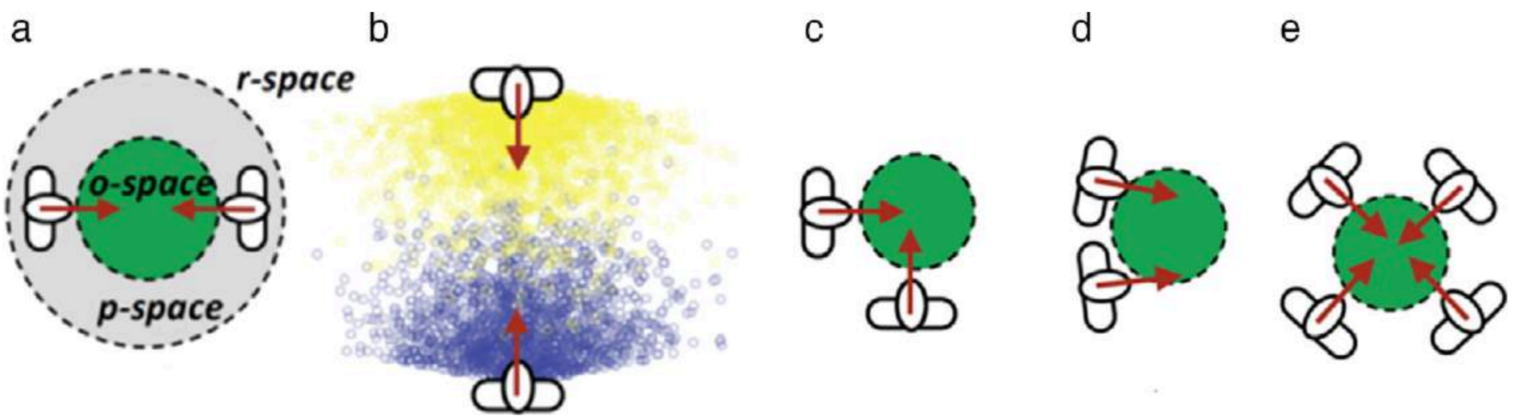
[a] Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia, Genova, Italy
[b] Deptartment of Environmental Sciences, Informatics and Statistics, University Ca' Foscari of Venice, Italy
[c] Deptartment of Computer Science, University of Verona, Italy
[d] Faculty of Electrical Engineering, Mathematics and Computer Science, Technical University of Delft, Netherlands

F-formations

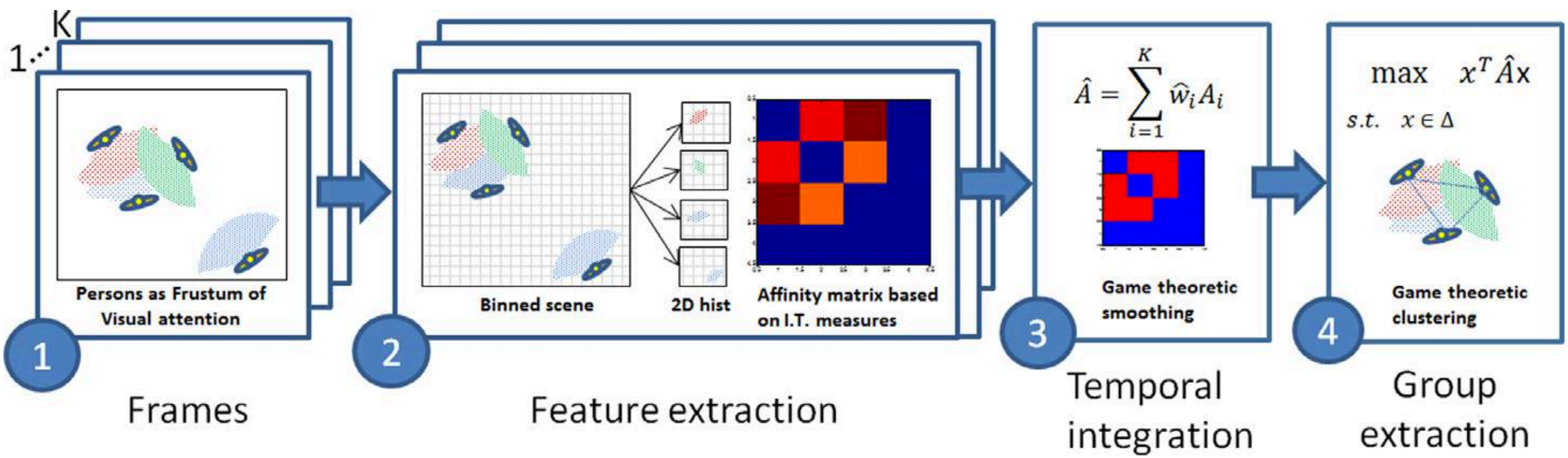
"Whenever two or more individuals in close proximity orient their bodies in such a way that each of them has an easy, direct and equal access to every other participant's transactional segment"

Ciolek & Kendon (1980)





System Architecture



Frustrum of visual attention

- A person in a scene is described by his/her position *(x,y)* and the head orientation θ
- The frustum represents the area in which a person can sustain a conversation and is defined by an aperture and by a length

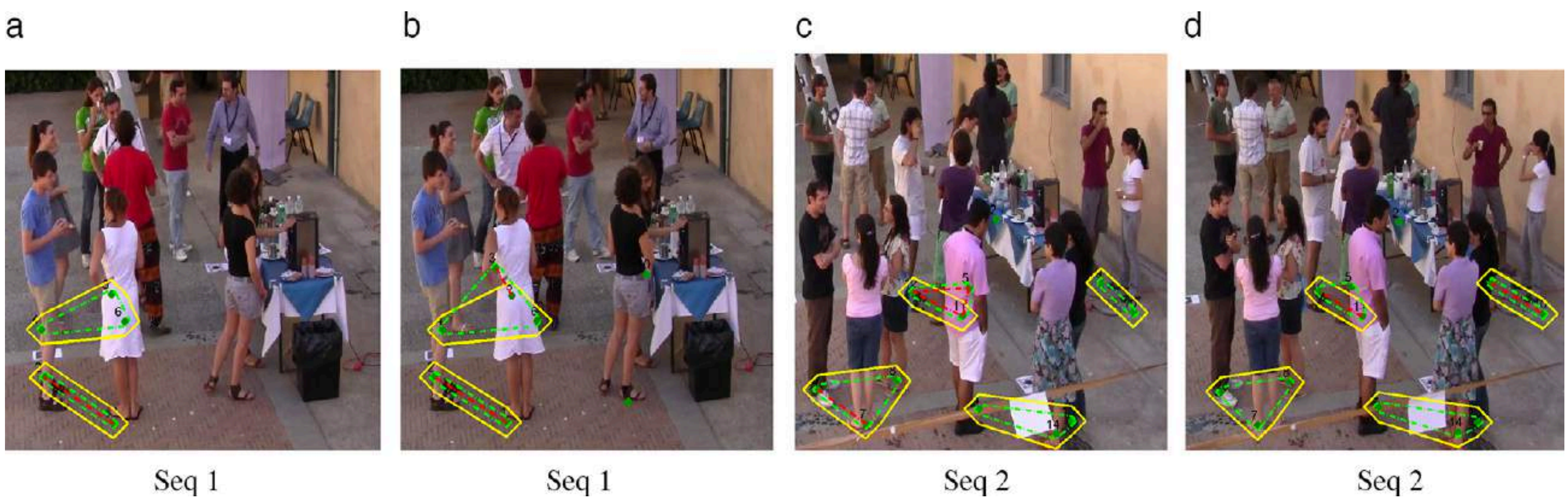
Results

Method ⋆	CoffeeBreak (S1+S2)			PosterData			Gdet		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
IRPM [61],[22]	0.60	0.41	0.49	–	–	–	–	–	–
HFF [22]	0.82	0.83	0.82	**0.93**	**0.96**	**0.94**	0.67	0.57	0.62
DS ([6], [22])∗	0.68	0.65	0.66	**0.93**	0.92	0.92	–	–	–
MULTISCALE [46]	0.82	0.77	0.80	–	–	–	–	–	–
GTCG [47] KL	0.80	0.84	0.82	0.90	0.94	0.92	**0.76**	0.75	0.75
GTCG [47] JS	0.83	**0.89**	0.86	0.92	**0.96**	**0.94**	**0.76**	**0.76**	**0.76**
R-GTCG SC	0.52	0.59	0.55	0.26	0.27	0.26	0.75	0.75	0.75
R-GTCG	**0.86**	0.88	**0.87**	0.92	**0.96**	**0.94**	**0.76**	**0.76**	**0.76**
	$\sigma = 0.2$, $l = 145$			$\sigma = 0.25$, $l = 115$			$\sigma = 0.7$ $l = 180$		

Method ⋆	**Cocktail Party**			**Synth**		
	Prec	Rec	F1	Prec	Rec	F1
IRPM [22,61]	–	–	–	0.71	0.54	0.61
HFF ([7], [46])	0.59	0.74	0.66	0.73	0.83	0.78
MULTISCALE [46]	0.69	0.74	0.71	0.86	0.94	0.90
GTCG [47] KL	0.85	0.81	0.83	**1.00**	**1.00**	**1.00**
GTCG [47] JS	0.86	**0.82**	**0.84**	**1.00**	**1.00**	**1.00**
R-GTCG SC	0.77	0.72	0.74	0.40	0.90	0.56
	σ=0.6 , l=170			σ=0.1 , l=75		
R-GTCG	**0.87**	**0.82**	**0.84**	**1.00**	**1.00**	**1.00**

Spectral Clustering

Results



Qualitative results on the CoffeeBreak dataset compared with the state of the art HFF.

Yellow = ground truth
Green = our method
Red = HFF.

Dominant Sets for “Constrained” Image Segmentation

Eyasu Zemene *, *Member, IEEE,* Leulseged Tesfaye Alemu *, *Member, IEEE*
and Marcello Pelillo, *Fellow, IEEE*

Abstract—Image segmentation has come a long way since the early days of computer vision, and still remains a challenging task. Modern variations of the classical (purely bottom-up) approach, involve, e.g., some form of user assistance (interactive segmentation) or ask for the simultaneous segmentation of two or more images (co-segmentation). At an abstract level, all these variants can be thought of as “constrained” versions of the original formulation, whereby the segmentation process is guided by some external source of information. In this paper, we propose a new approach to tackle this kind of problems in a unified way. Our work is based on some properties of a family of quadratic optimization problems related to *dominant sets*, a graph-theoretic notion of a cluster which generalizes the concept of a maximal clique to edge-weighted graphs. In particular, we show that by properly controlling a regularization parameter which determines the structure and the scale of the underlying problem, we are in a position to extract groups of dominant-set clusters that are constrained to contain predefined elements. In particular, we shall focus on interactive segmentation and co-segmentation (in both the unsupervised and the interactive versions). The proposed algorithm can deal naturally with several types of constraints and input modalities, including scribbles, sloppy contours and bounding boxes, and is able to robustly handle noisy annotations on the part of the user. Experiments on standard benchmark datasets show the effectiveness of our approach as compared to state-of-the-art algorithms on a variety of natural images under several input conditions and constraints.

Index Terms—Interactive segmentation, co-segmentation, dominant sets, quadratic optimization, game dynamics.

Constrained Dominant Sets

Given $S \subseteq V$ and a parameter $\alpha > 0$, define the following parameterized family of quadratic programs:

$$\begin{array}{ll} \text{maximize} & f_S^\alpha(\mathbf{x}) = \mathbf{x}'(A - \alpha \hat{I}_S)\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array}$$

where I_S is the diagonal matrix whose elements are set to 1 in correspondence to the vertices outside S, and to zero otherwise:

$$\hat{I}_S = \begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

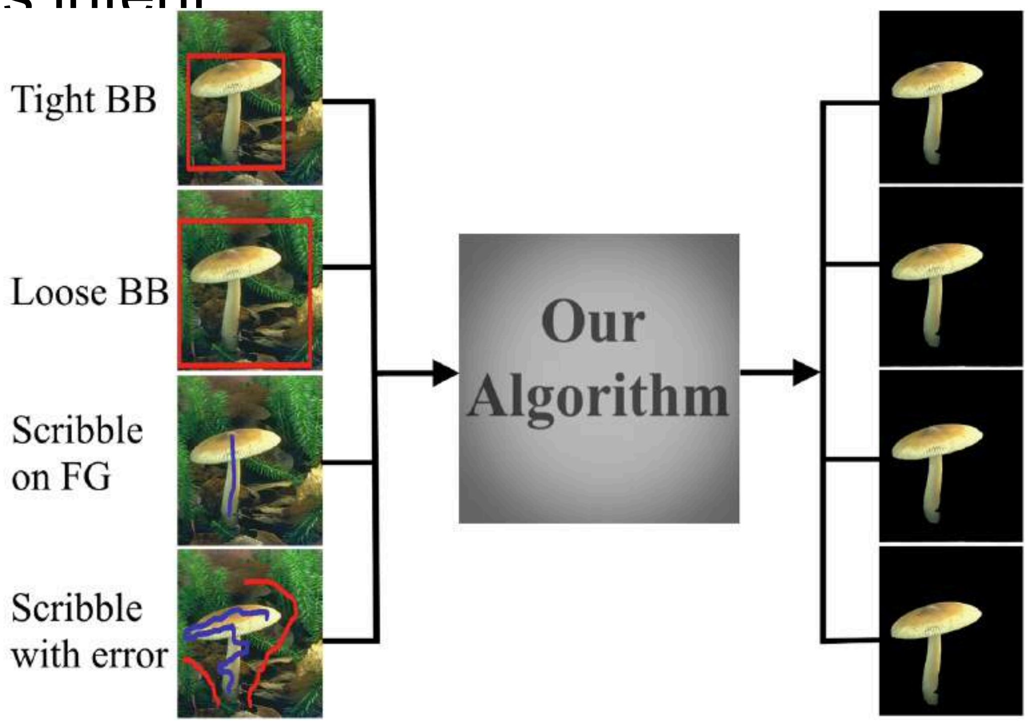
Property. By setting:

$$\alpha > \lambda_{\max}(A_{V \setminus S})$$

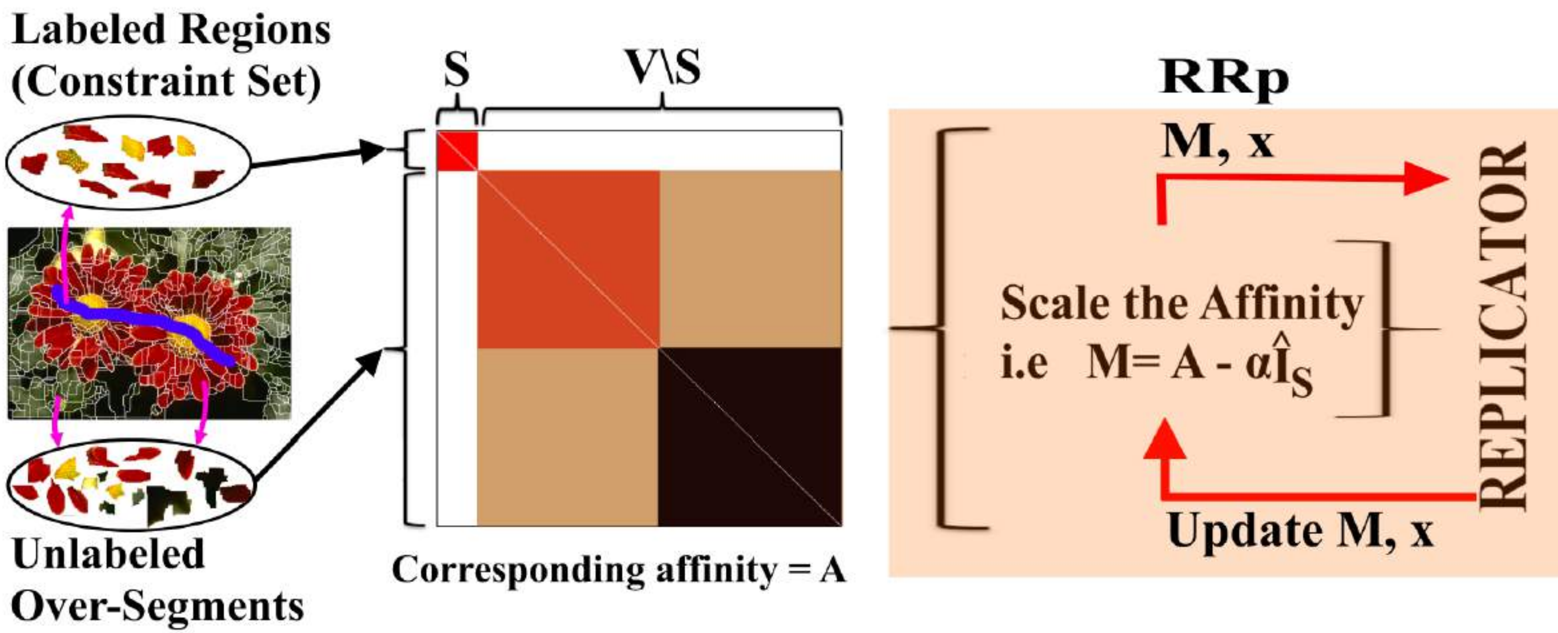
all local solutions will have a support containing elements of S.

Interactive Image Segmentation

Given an image and some information provided by a user, in the form of a scribble or of a bounding box, to provide as output a foreground object that best reflects the user's intent.



System Overview



Left: Over-segmented image with a user scribble (blue label).

Middle: The corresponding affinity matrix, using each over-segments as a node, showing its two parts: S, the constraint set which contains the user labels, and V n S, the part of the graph which takes the regularization parameter .

Right: RRp, starts from the barycenter and extracts the first dominant set and update x and M, for the next extraction till all the dominant sets which contain the user labeled regions are extracted.

Results

Table 1. Error rates of different scribble-based approaches on the Grab-Cut dataset

Methods	Error rate
Graph Cut [7]	6.7
Lazy Snapping [5]	6.7
Geodesic Segmentation [4]	6.8
Random Walker [33]	5.4
Transduction [34]	5.4
Geodesic Graph Cut [30]	4.8
Constrained Random Walker [31]	4.1
CDS_Self Tuning (Ours)	**3.57**
CDS_Single Sigma (Ours)	**3.80**
CDS_Best Sigma (Ours)	2.72

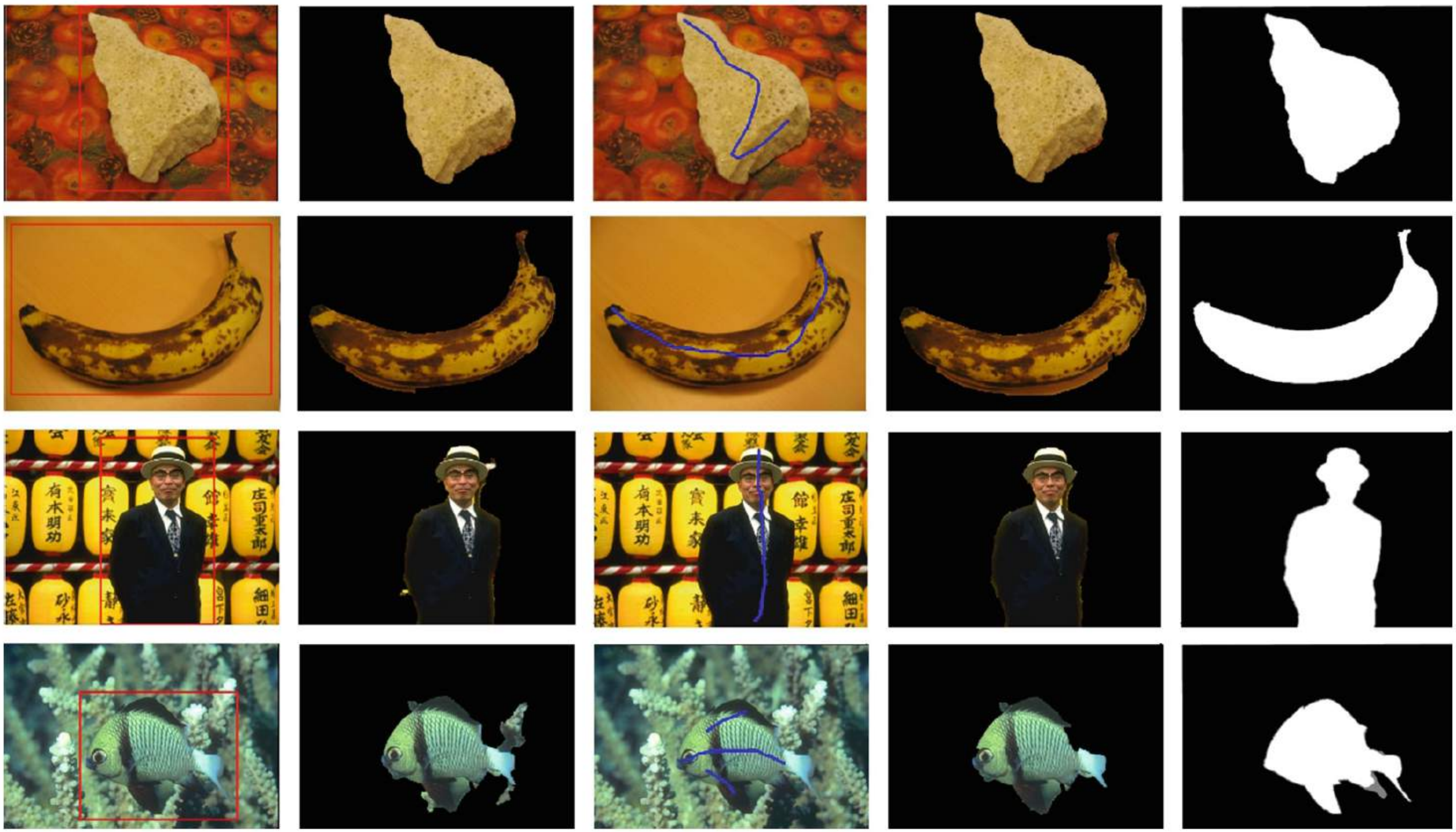
Table 2. Jaccard index of different approaches – first 5 bounding-box-based – on Berkeley dataset

Methods	Jaccard index
MILCut-Struct [3]	84
MILCut-Graph [3]	83
MILCut [3]	78
Graph Cut [1]	77
Binary Partition Trees [35]	71
Interactive Graph Cut [7]	64
Seeded Region Growing [36]	59
Simple Interactive O.E [37]	63
CDS_Self Tuning (Ours)	**93**
CDS_Single Sigma (Ours)	**93**
CDS_Best Sigma (Ours)	95

Results

Bounding box | Result | Scribble | Result | Ground truth

Results



Bounding box | Result | Scribble | Result | Ground truth

Large-scale Image Geo-Localization Using Dominant Sets

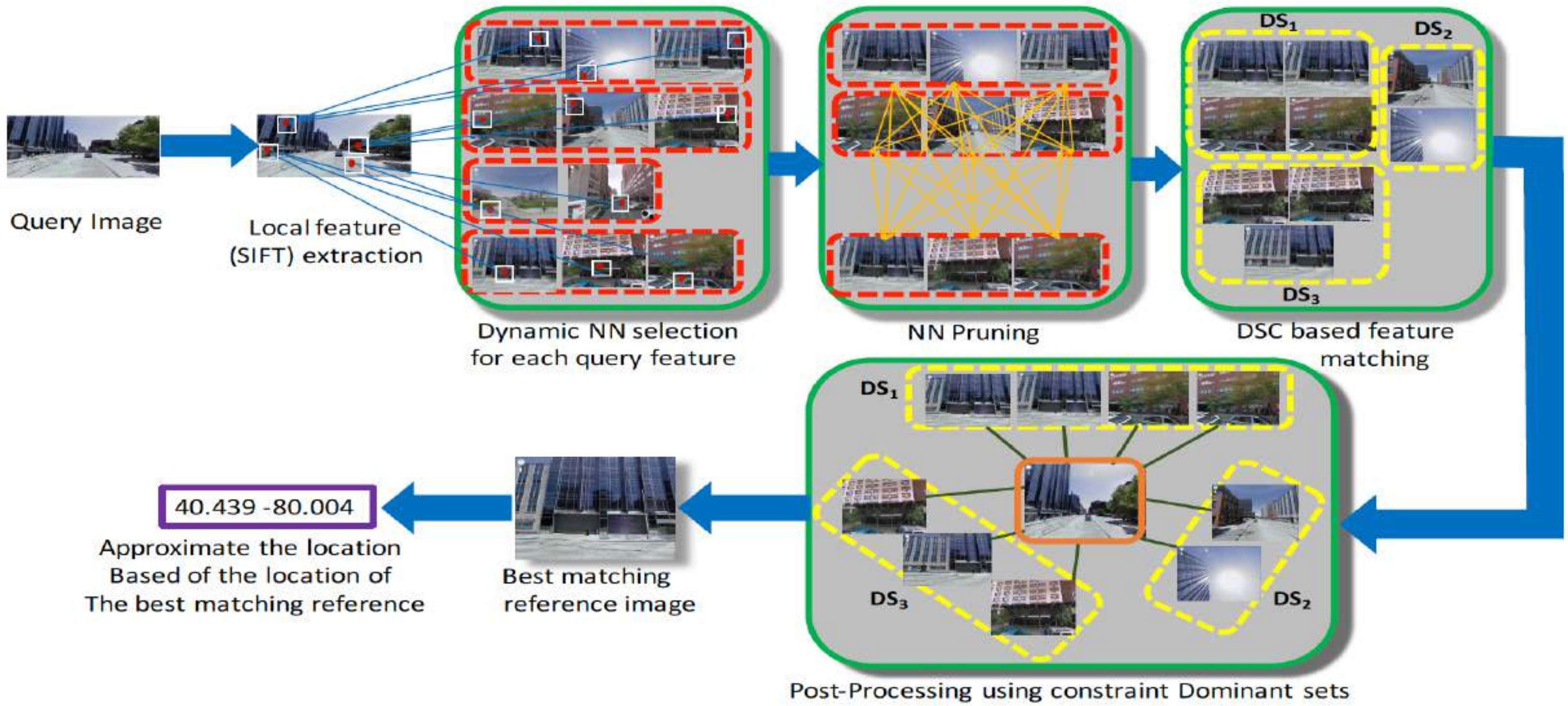
Eyasu Zemene*, *Student Member, IEEE,* Yonatan Tariku Tesfaye*, *Student Member, IEEE,* Haroon Idrees, *Member, IEEE,* Andrea Prati, *Senior member, IEEE,* Marcello Pelillo, *Fellow, IEEE,* and Mubarak Shah, *Fellow, IEEE*

Abstract—This paper presents a new approach for the challenging problem of geo-localization using image matching in a structured database of city-wide reference images with known GPS coordinates. We cast the geo-localization as a clustering problem of local image features. Akin to existing approaches to the problem, our framework builds on low-level features which allow local matching between images. For each local feature in the query image, we find its approximate nearest neighbors in the reference set. Next, we cluster the features from reference images using Dominant Set clustering, which affords several advantages over existing approaches. First, it permits variable number of nodes in the cluster, which we use to dynamically select the number of nearest neighbors for each query feature based on its discrimination value. Second, this approach is several orders of magnitude faster than existing approaches. Thus, we obtain multiple clusters (different local maximizers) and obtain a robust final solution to the problem using multiple weak solutions through constrained Dominant Set clustering on global image features, where we enforce the constraint that the query image must be included in the cluster. This second level of clustering also bypasses heuristic approaches to voting and selecting the reference image that matches to the query. We evaluate the proposed framework on an existing dataset of 102k street view images as well as a new larger dataset of 300k images, and show that it outperforms the state-of-the-art by 20% and 7%, respectively, on the two datasets.

Index Terms—Geo-localization, Dominant Set Clustering, Multiple Nearest Neighbor Feature Matching, Constrained Dominant Set

Image Geo-localization

A new approach for the problem of geo-localization using image matching in a structured database of city-wide reference images with known GPS coordinates.



200x time faster + 20% accuracy improvement w.r.t previous approach

Datasets:

- Datasets one:
  - Reference images:
    - 102K Google street view images from **Pittsburgh, PA** and **Orlando, FL**
  - Test Set:
    - 521 GPS-Tagged unconstrained images
    - Downloaded From Flickr, Panoramio, Picasa, …

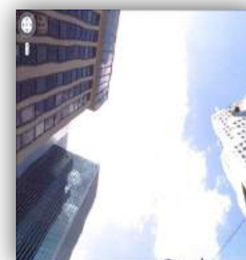
- ***WorldCities*** Datasets (NEW)*:
  - Reference images:
    - 300K Google street view images
      - 14 different cities from **Europe, N. America and Australia**
  - Test Set:
    - 500 GPS-Tagged unconstrained images
    - Downloaded From Flickr, Panoramio, Picasa, …

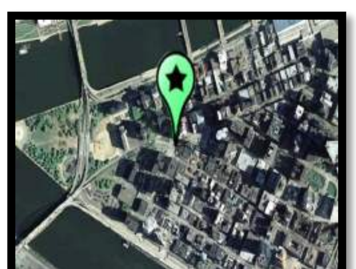
Google Maps Street View Datasets:

For each location: 4 side views and 1 top view is collected

Side Views

top View

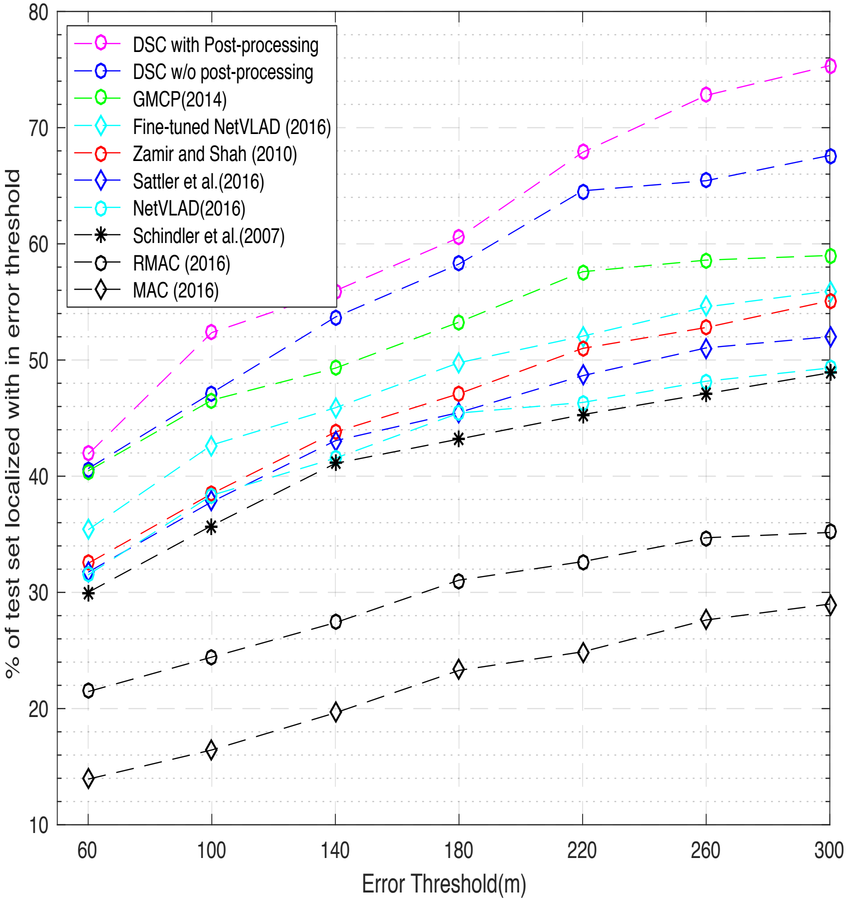




Pittsburgh, PA   Longitude = 40.44146° Latitude=-80.0037°

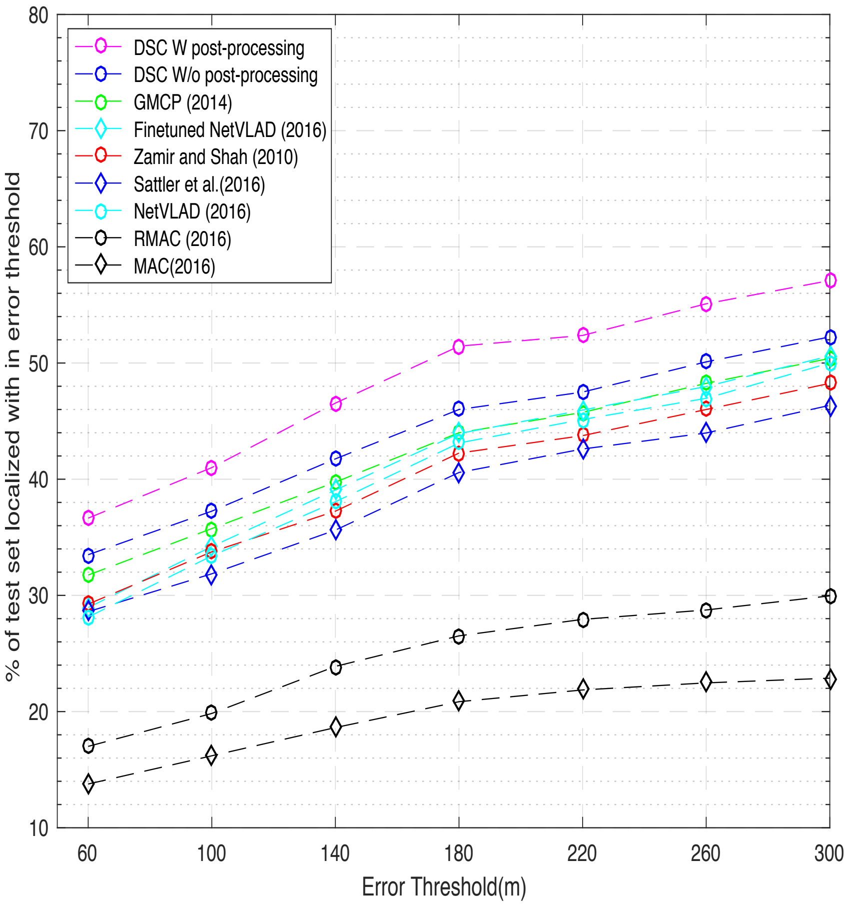
Overall Result

- Dataset 1: 102K Google street view images (Orlando and Pittsburg area)

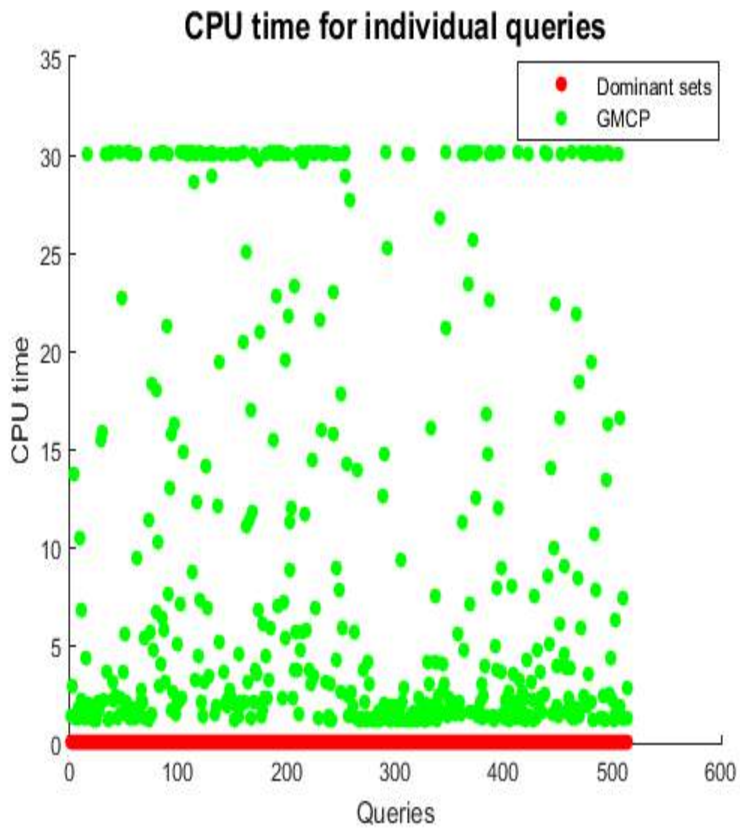


Overall Result

- Dataset 2: WorldCities (14 different cities from Europa, North America, Australia)



Computational Time



Qualitative Results

Query

Match – Error: 70.01 m

Query

Match – Error: 5.4 m

Query

Match – Error: 10.4 m

Query

Match – Error: 7.5 m

Query

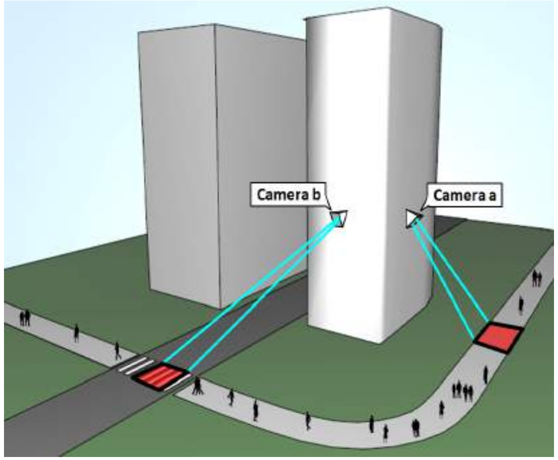
Match – Error: 62.7 m

Multi-Target Tracking in Multiple Non-Overlapping Cameras using Fast-Constrained Dominant Sets

Yonatan Tariku Tesfaye · Eyasu Zemene · Andrea Prati · Marcello Pelillo · Mubarak Shah

Submitted

Person Re-identification



- Recognize an individual over different non-overlapping cameras.
- Given a gallery of person images we want to recognize (between all of them) a new observed image, called probe.

Video-based Person Re-ID

Traditional methods focus on:

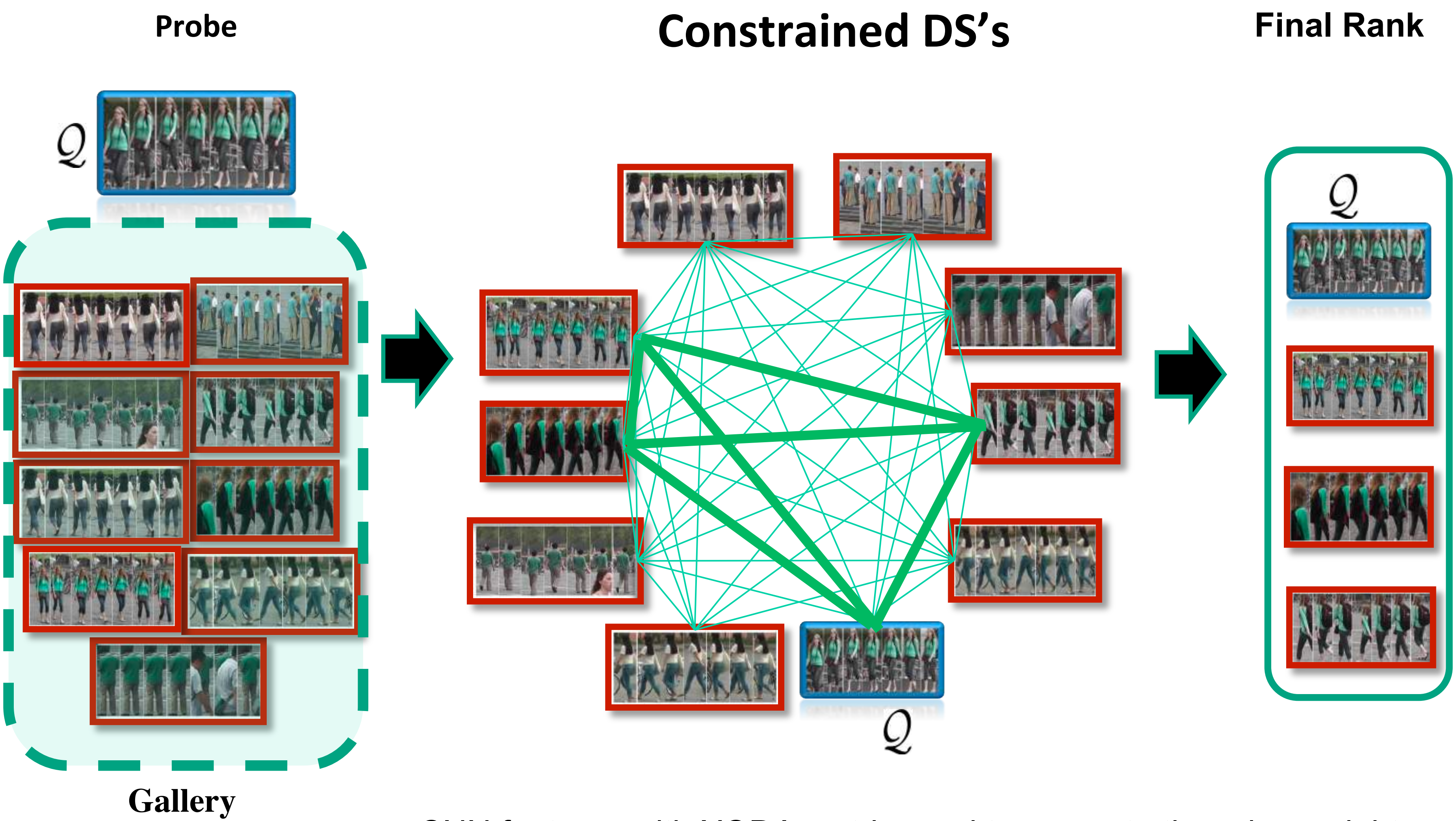
- Building better feature representation of objects
- Building a better distance metric
- Finally rank images from gallery based on the pairwise distances from the query

In our approach

- We use standard features and distance metric
- Extract constrained dominant sets for each query
- Perform ranking over shortlisted clips NOT over the whole set

We take into account **both** the relationship between query and elements in the gallery and elements in the gallery.

Re-ID with Constrained DS's



CNN features with XQDA metric used to compute the edge weights

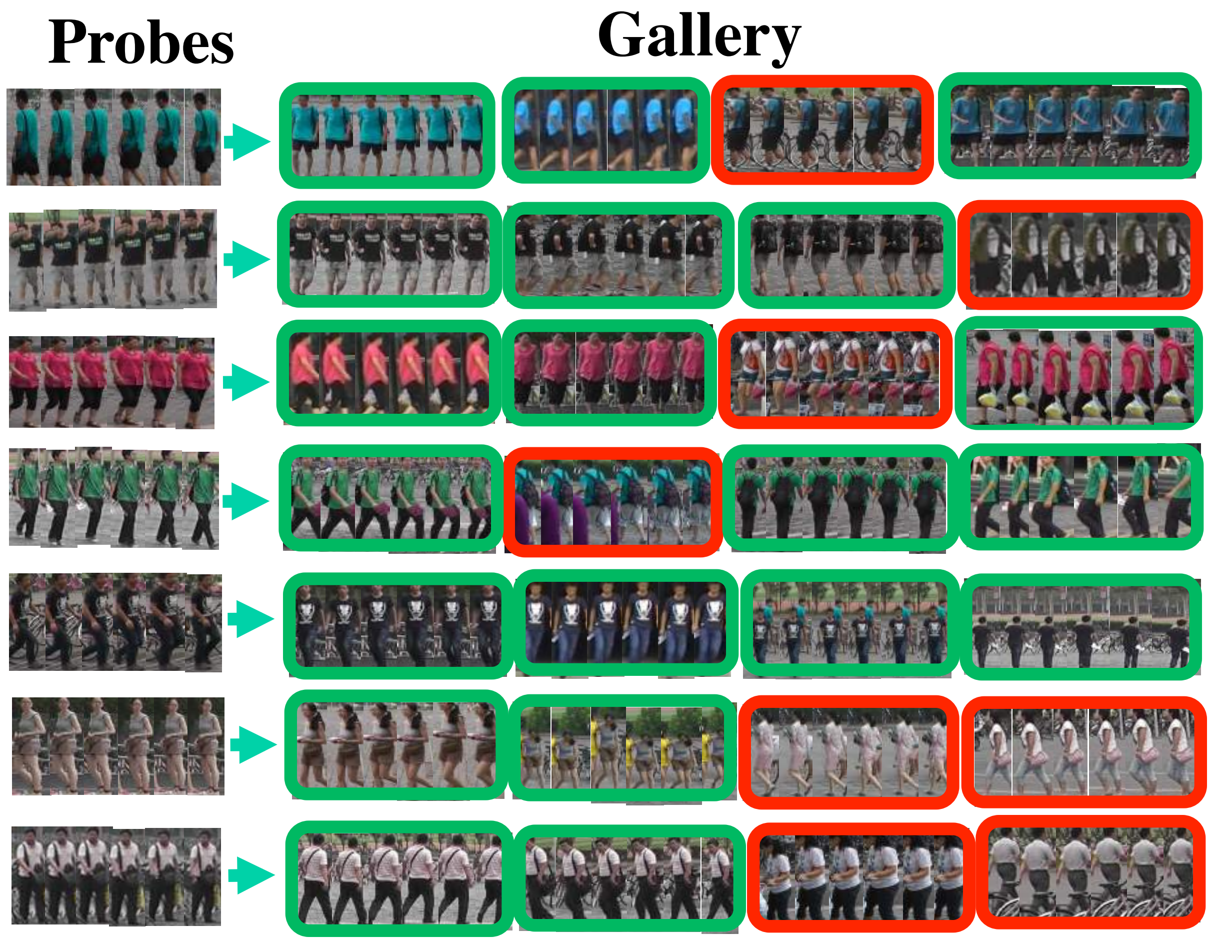
Results on MARS Dataset

- Largest video Re-ID dataset (2016)
- 6 near-synchronized cameras
- 1,261 identities
- 3,248 distractors
- tracklets are of 25-30 frames long

Methods	rank 1
HLBP [40] + XQDA	18.60
BCov [24] + XQDA	9.20
LOMO [20] + XQDA	30.70
BoW [49] + KISSME	30.60
SDALF [8] + DVR	4.10
HOG3D [16] + KISSME	2.60
CNN + XQDA [48]	65.30
CNN + KISSME [48]	65.00
Ours	**68.22**

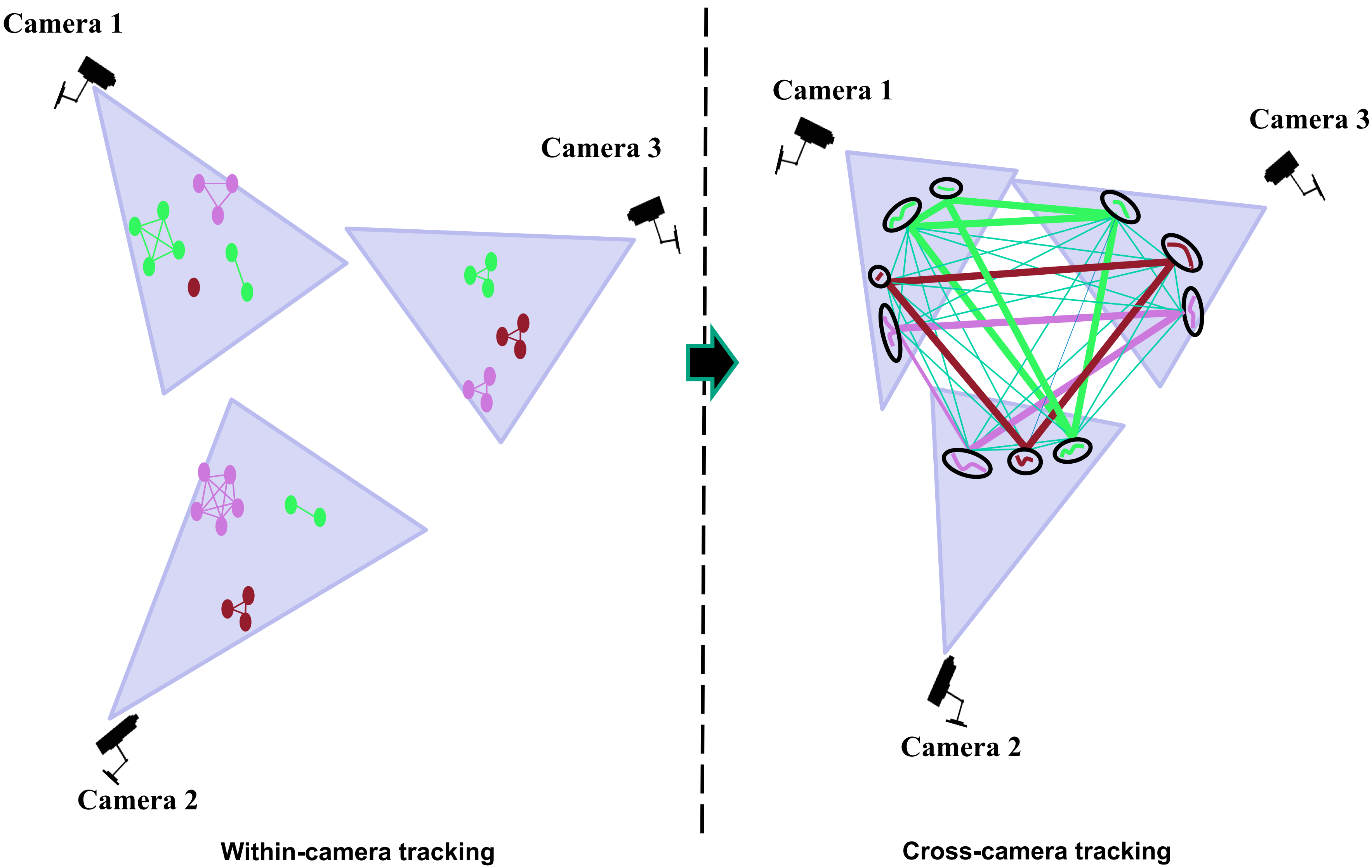
[8] M. Farenzena et al. Person re-identification by symmetry-driven accumulation of local features (*CVPR 2010)*
[16] A. Klaser et al. A spatio-temporal descriptor based on 3D-gradients (*BMVC 2008)*
[20] S. Liao et al. Person re-identification by local maximal occurrence representation and metric learning (*CVPR 2015)*
[24] B. Ma et al. Covariance descriptor based on bio-inspired features for person re-identification and face verification *(Image Vision Comput 2014)*
[40] F. Xiong et al. Person re-identification using kernel-based metric learning methods (*ECCV 2014)*
[48] L. Zheng et al. MARS: A video benchmark for large-scale person re-identification (*ECCV 2016)*
[49] L. Zheng et al. Scalable person re-identification: A benchmark *(ICCV 2015)*

Examples

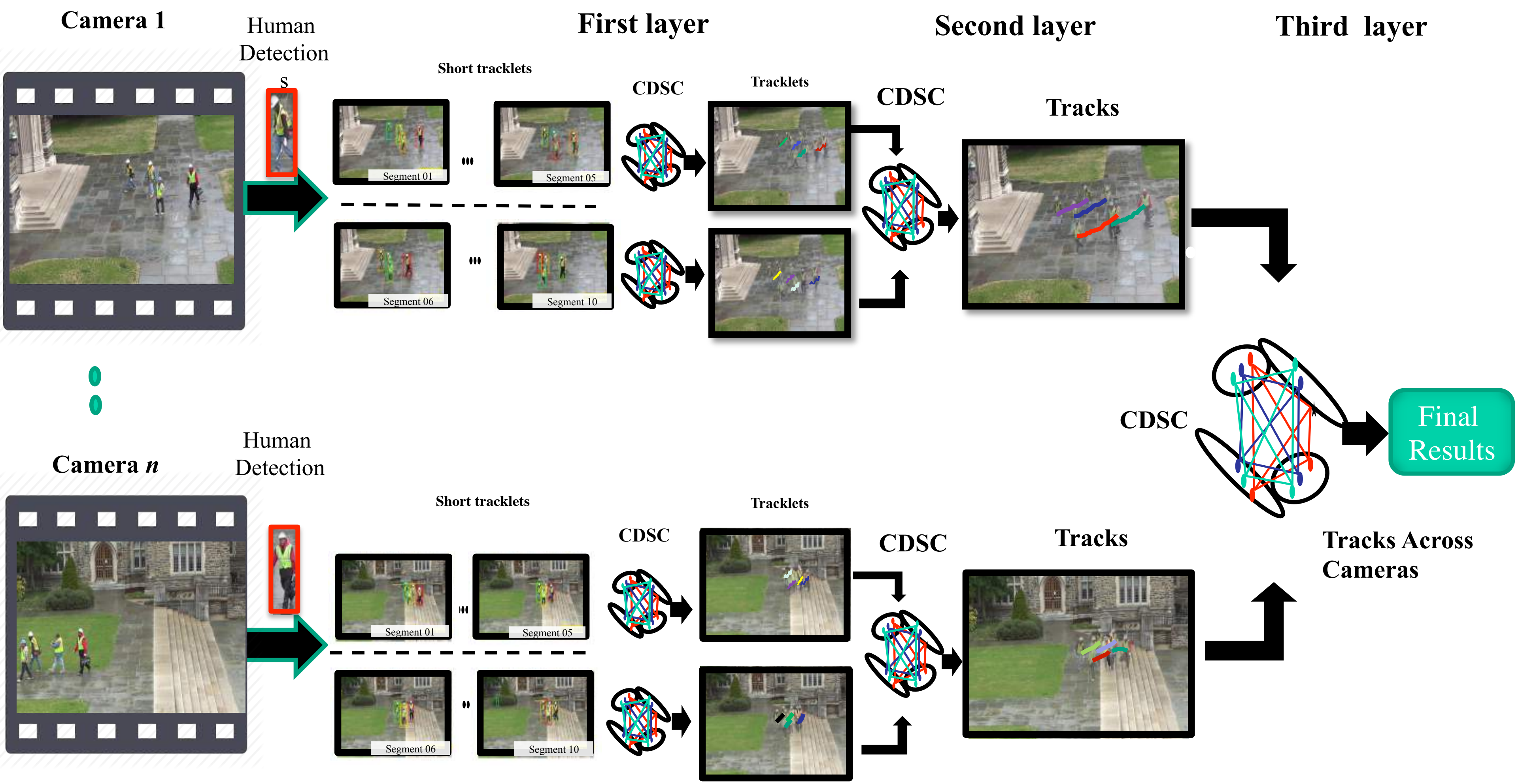


The green and red boxes denote the same and different persons with the probes, respectively
Gallery images are ordered based on their membership score (highest -> lowest).

Multi-target Multi-camera Tracking

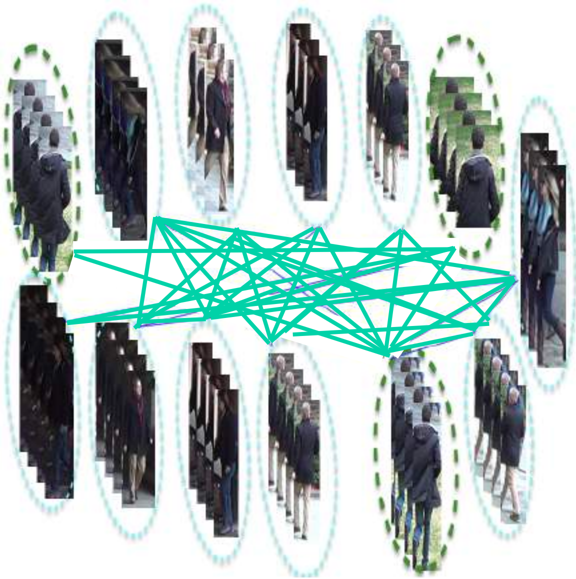


Pipeline



Camera 1
Human Detections
First layer
Second layer
Third layer
Short tracklets
CDSC
Tracklets
CDSC
Tracks
Segment 01
Segment 05
Segment 06
Segment 10
CDSC
Final Results
Camera n
Human Detection
Short tracklets
CDSC
Tracklets
CDSC
Tracks
Segment 01
Segment 05
Segment 06
Segment 10
Tracks Across Cameras

Layer 1: Tracklet Extraction

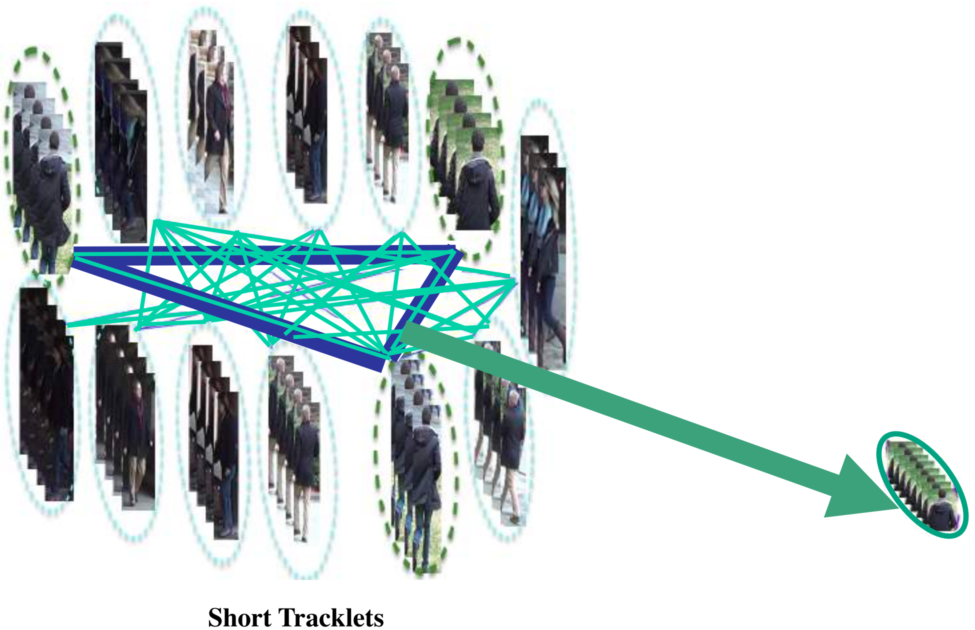


Short Tracklets

Edge weights combine appearance and motion

- Appearance = CNN features
- Motion = Constant velocity

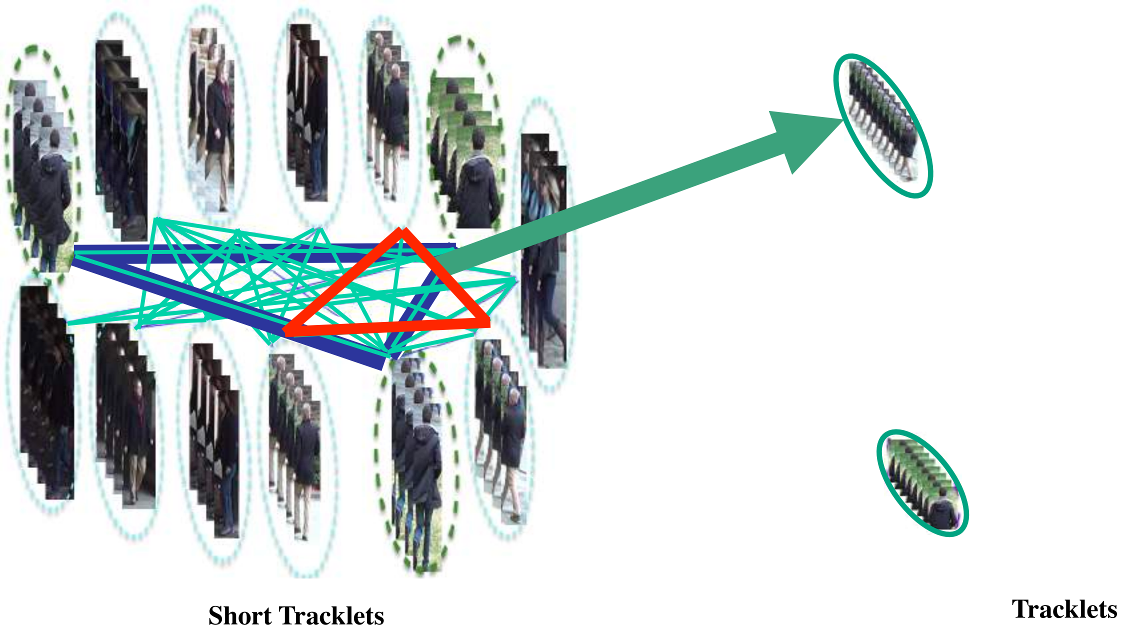
Layer 1: Tracklet Extraction



Edge weights combine appearance and motion

- Appearance = CNN features
- Motion = Constant velocity

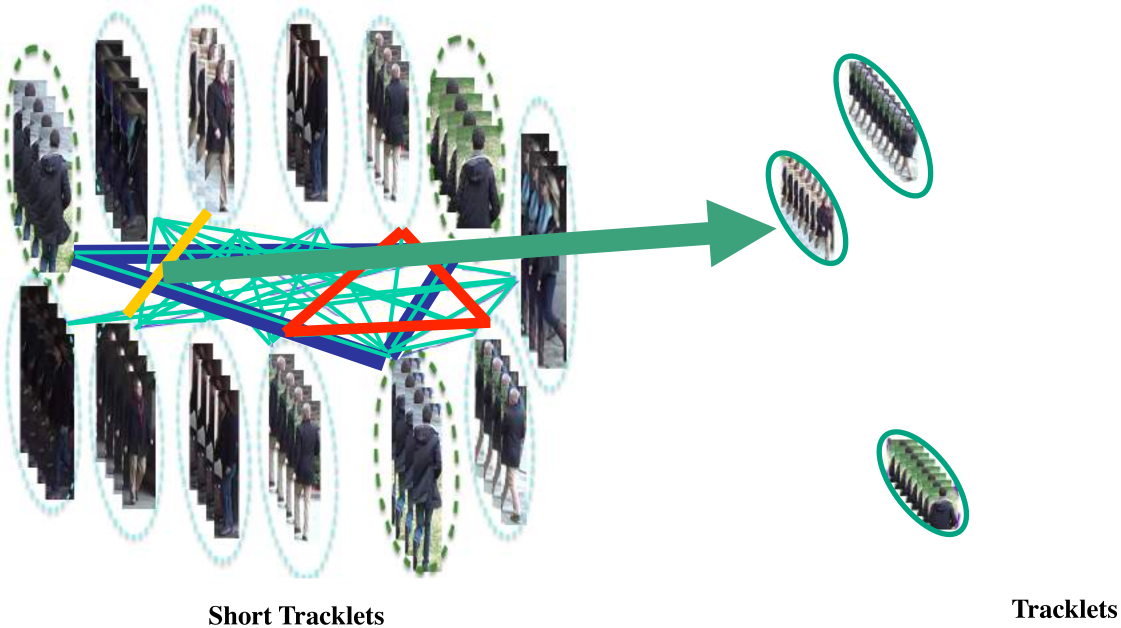
Layer 1: Tracklet Extraction



Edge weights combine appearance and motion

- Appearance = CNN features
- Motion = Constant velocity

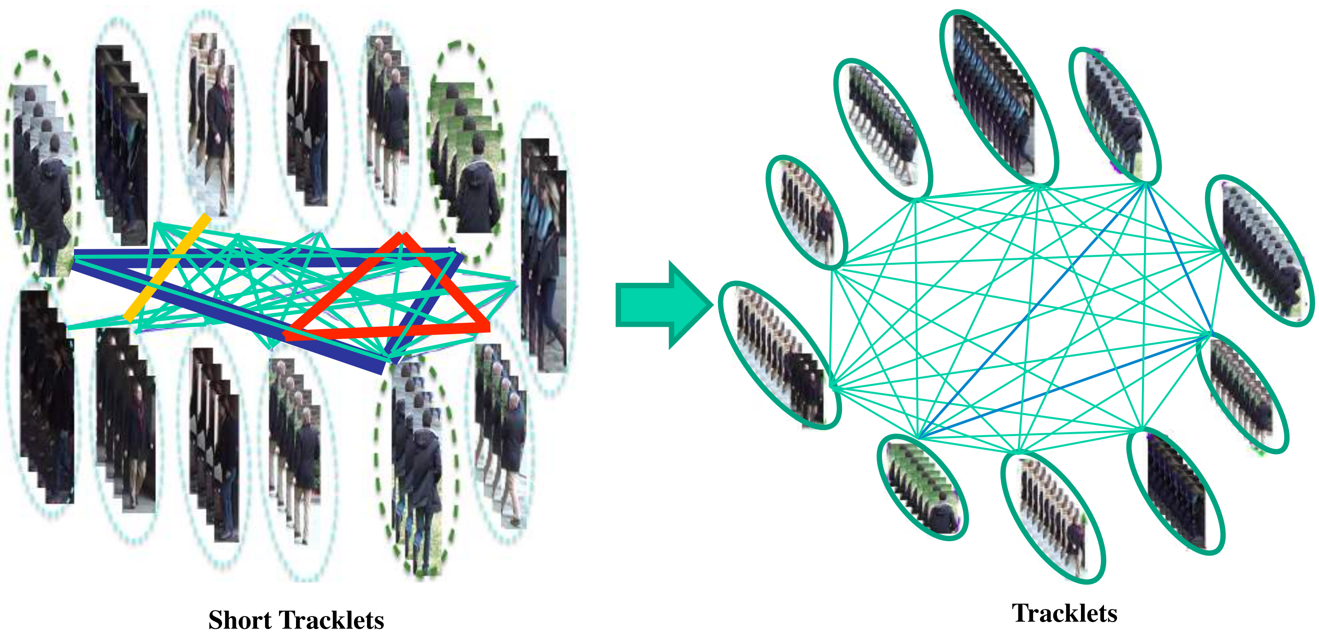
Layer 1: Tracklet Extraction



Edge weights combine appearance and motion

- Appearance = CNN features
- Motion = Constant velocity

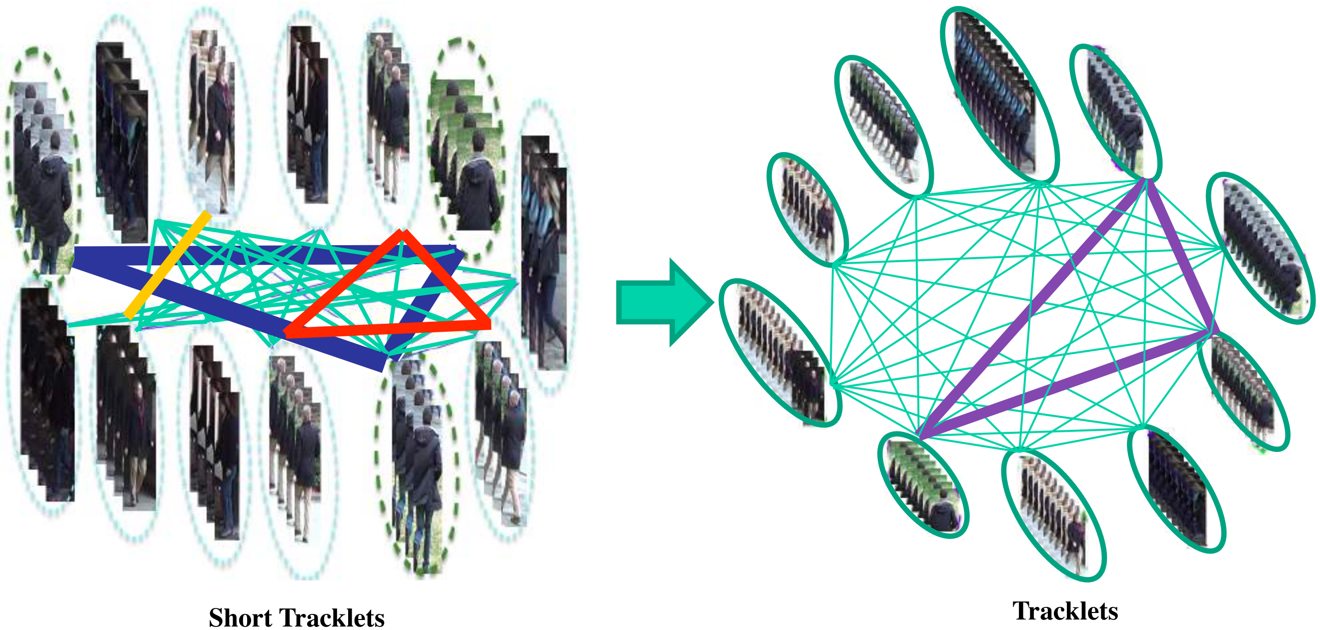
Layer 1: Tracklet Extraction



Edge weights combine appearance and motion

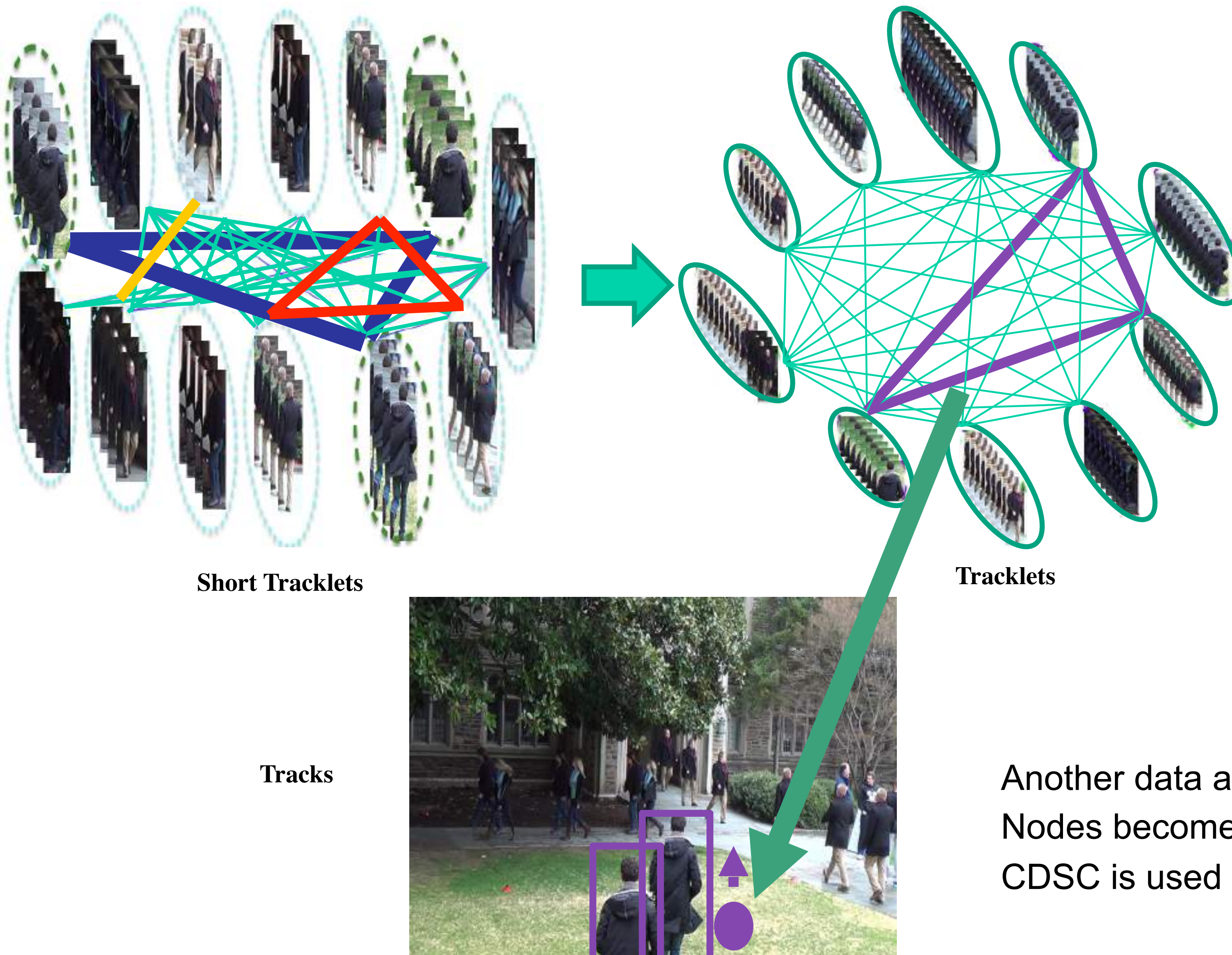
- Appearance = CNN features
- Motion = Constant velocity

Layer 2: Track Extraction



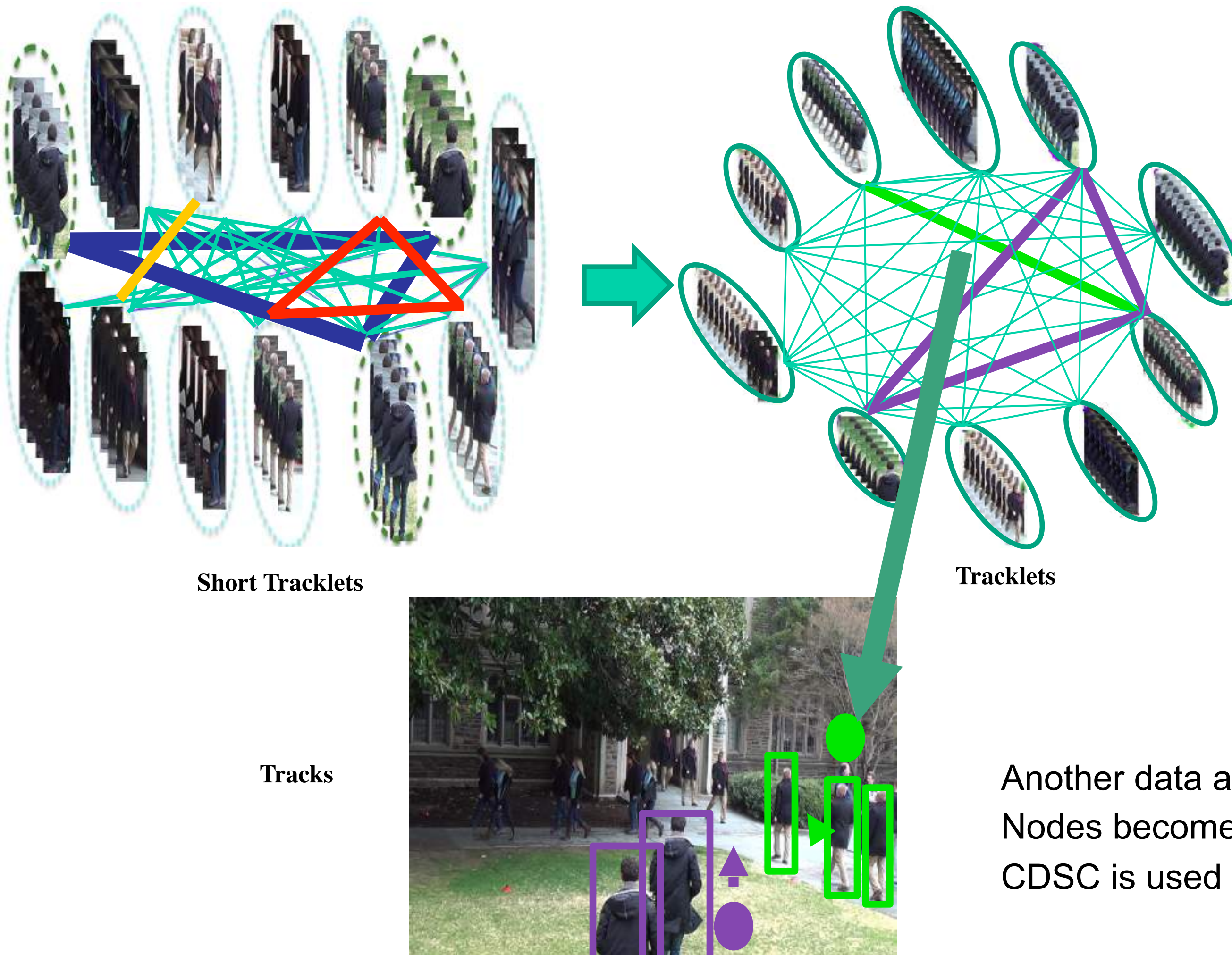
Another data association problem
Nodes become tracklets
CDSC is used to stitch tracklets

Layer 2: Track Extraction



Another data association problem
Nodes become tracklets
CDSC is used to stitch tracklets

Layer 2: Track Extraction

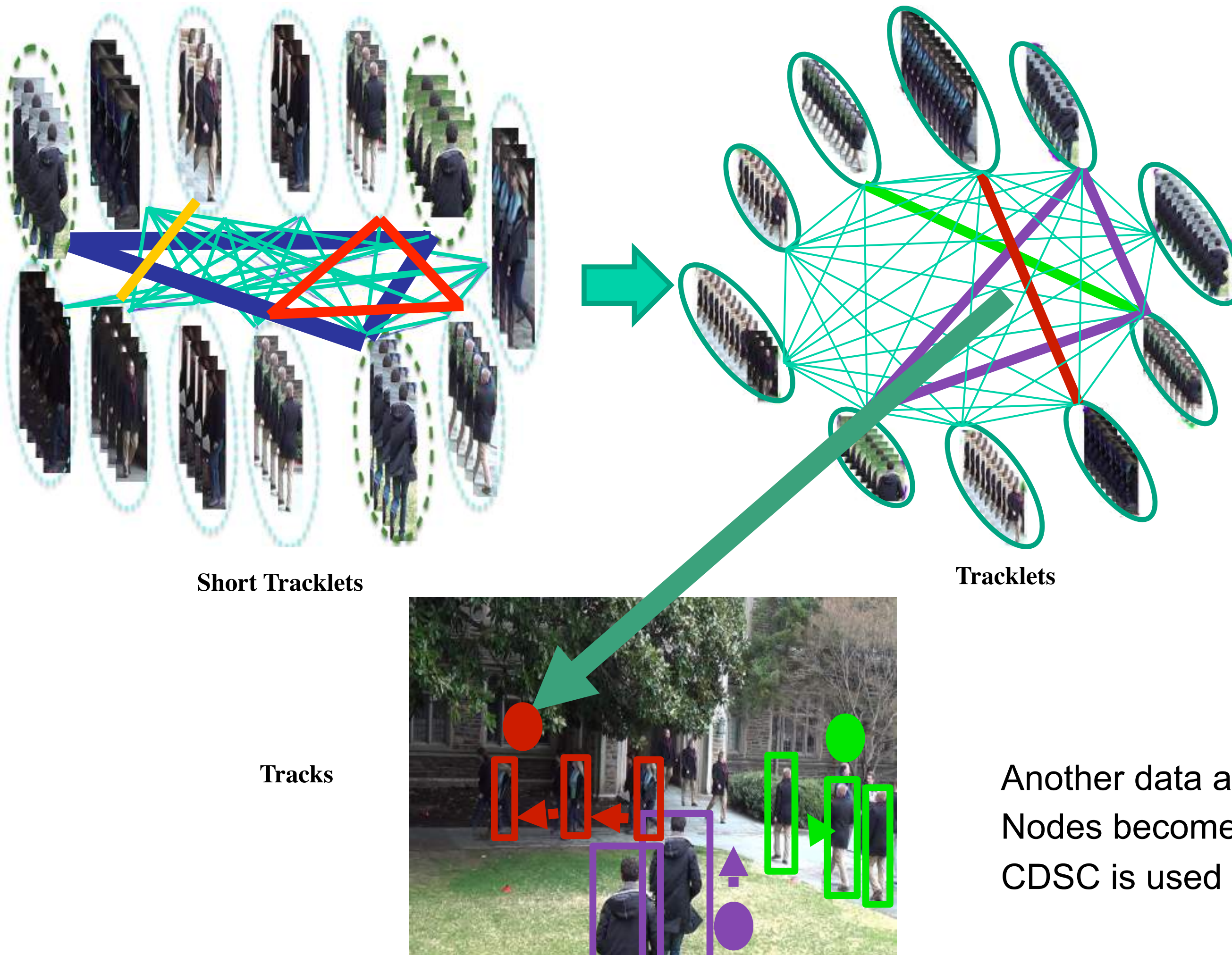


Another data association problem

Nodes become tracklets

CDSC is used to stitch tracklets

Layer 2: Track Extraction

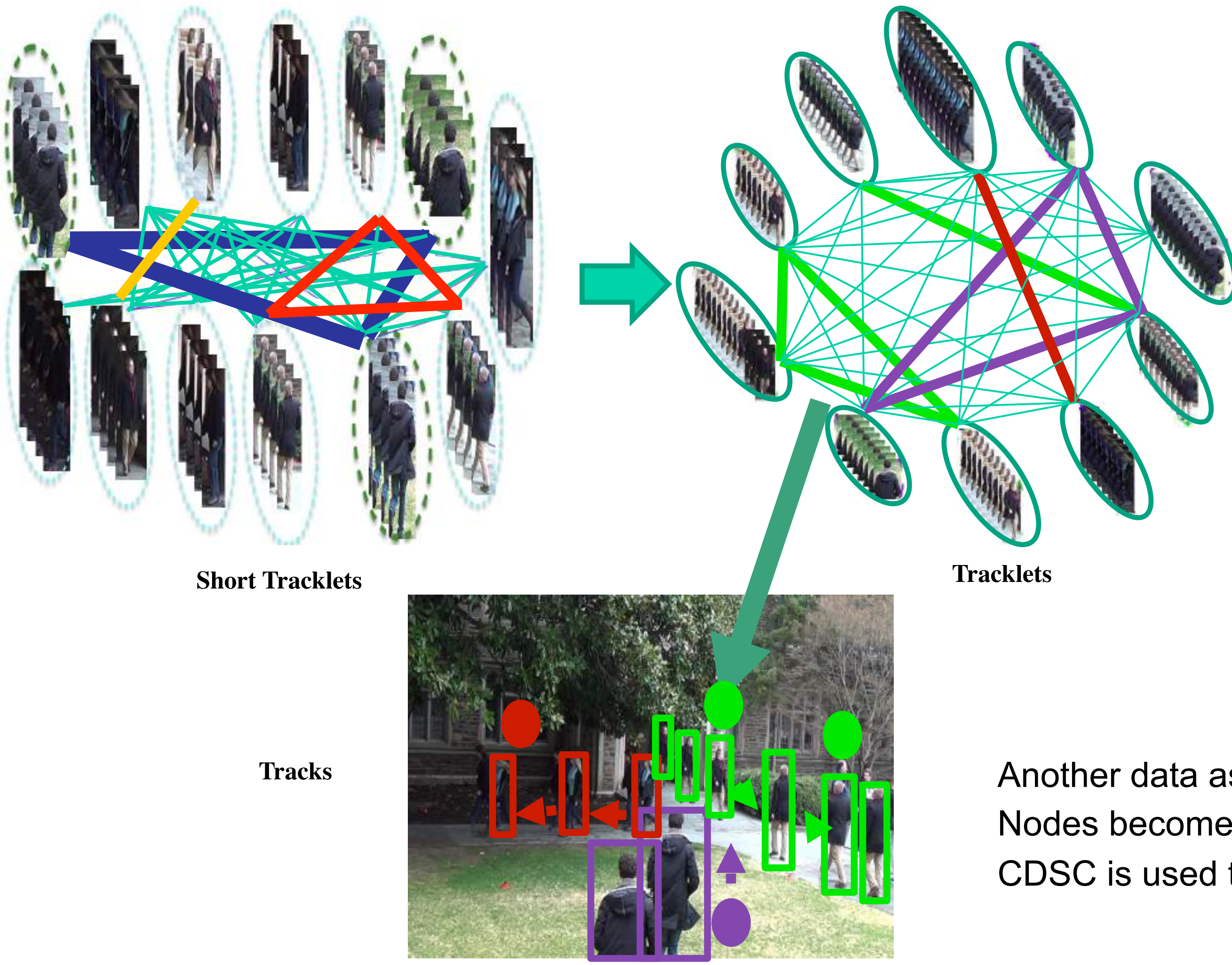


Another data association problem

Nodes become tracklets

CDSC is used to stitch tracklets

Layer 2: Track Extraction

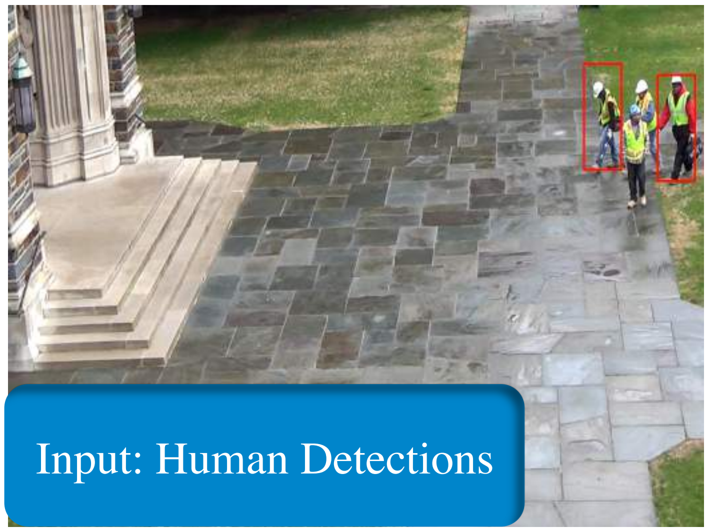


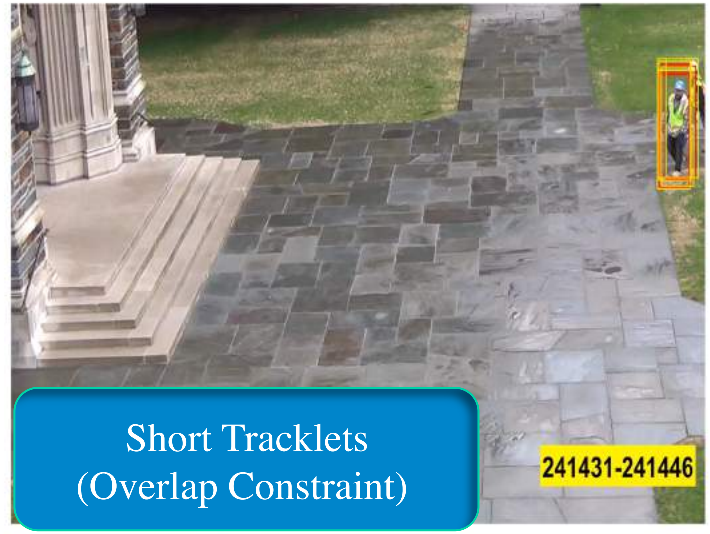
Another data association problem

Nodes become tracklets

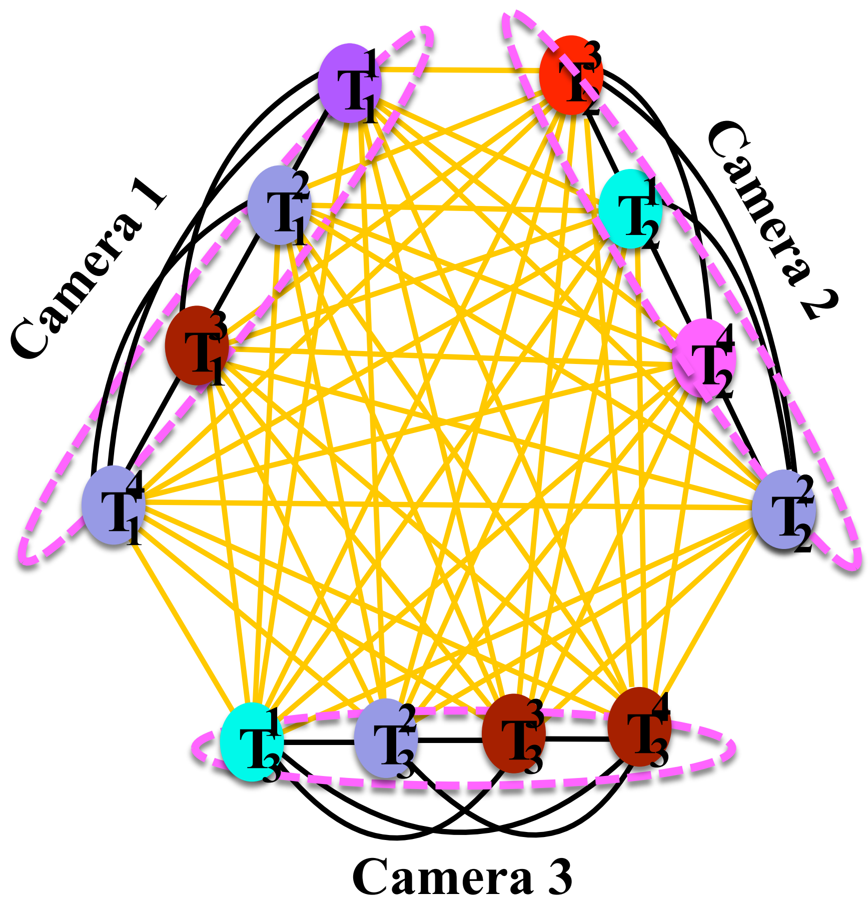
CDSC is used to stitch tracklets

Within-Camera Tracking



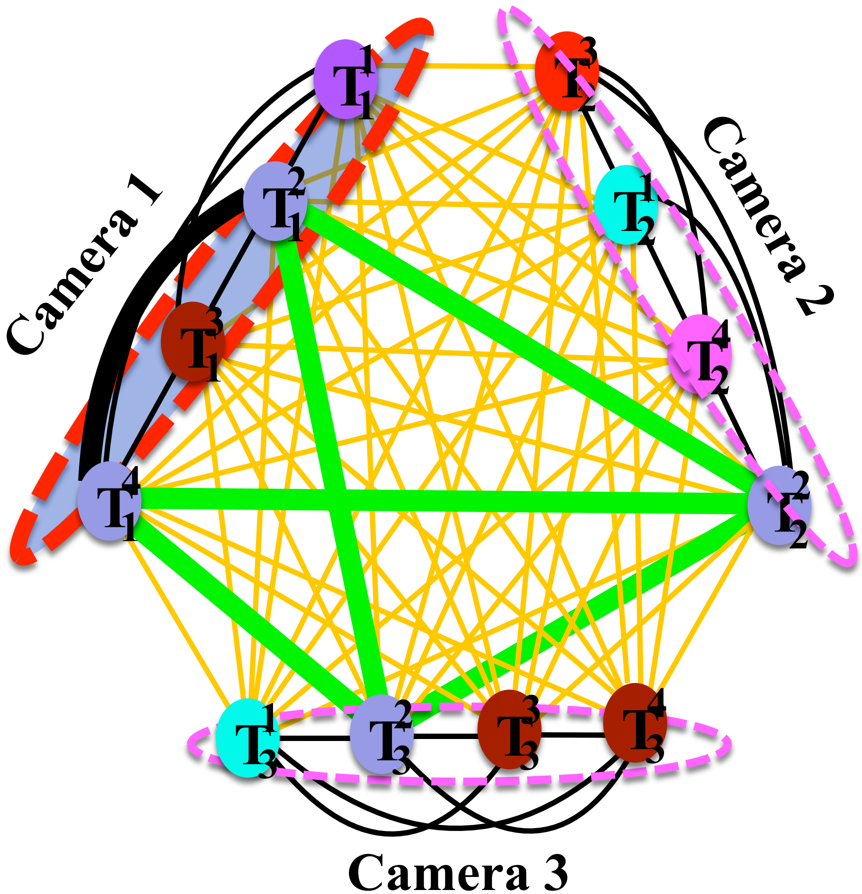


Layer 3: Cross-Camera Association



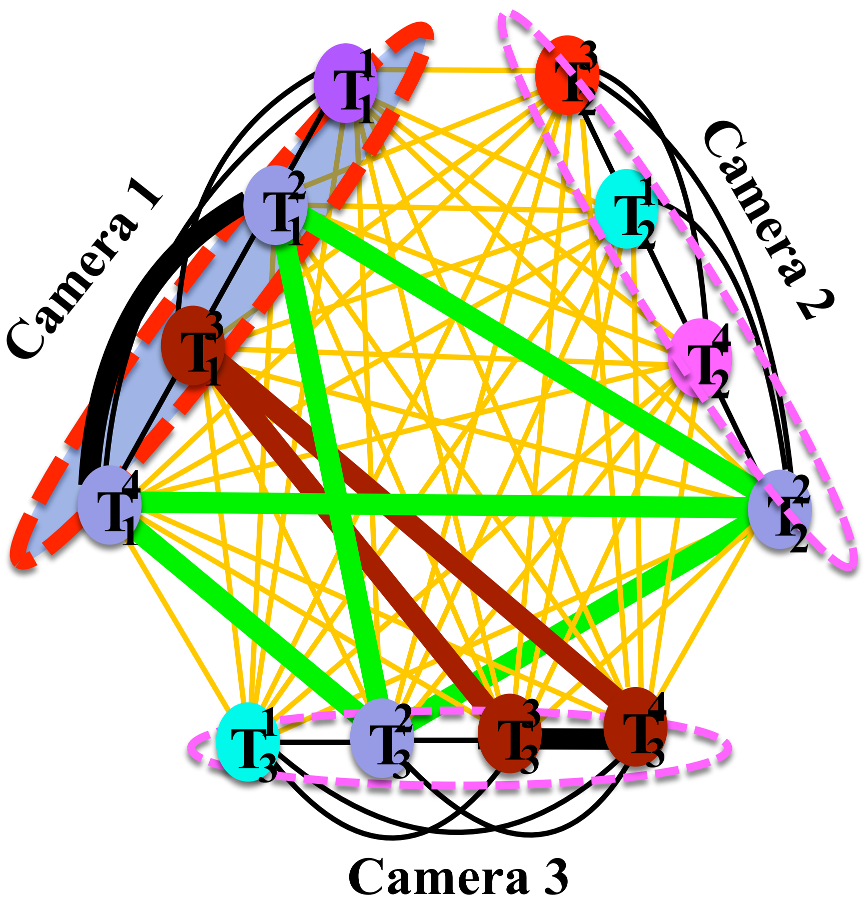
Tracks are nodes
Cameras as constraints

Layer 3: Cross-Camera Association



Tracks are nodes
Cameras as constraints

Layer 3: Cross-Camera Association



Tracks are nodes
Cameras as constraints

Results on DukeMTMC

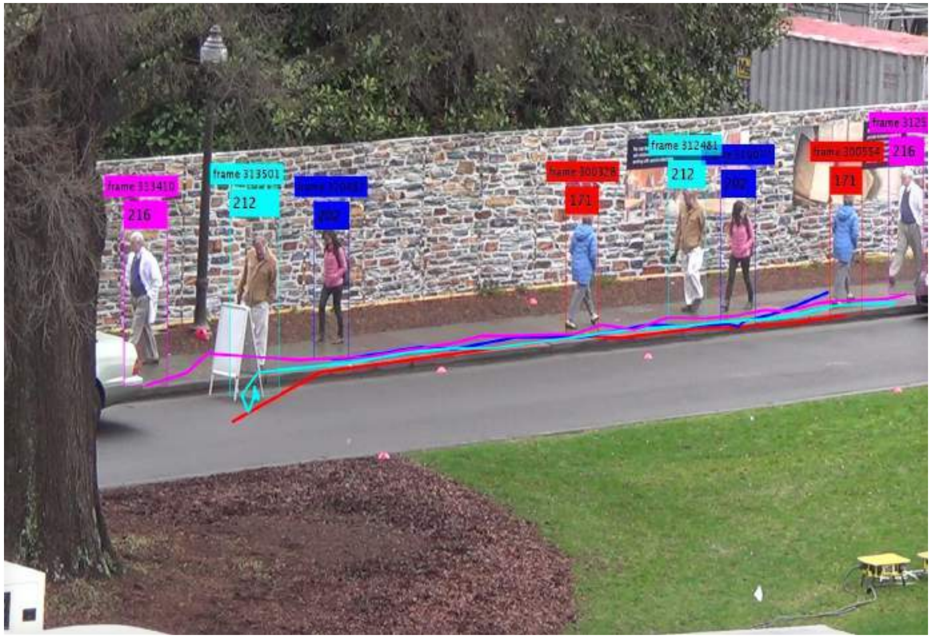
- Largest MTMC dataset (2016)
- 8 fixed synchronized cameras
- More than 2 million frames
- 0 to 54 persons per frame
- 2,700 Identities

	Methods	IDF1↑	IDP↑	IDR↑
Multi-Camera Test-easy	[33]	47.3	59.6	39.2
	[26]	32.9	41.3	27.3
	Ours	**50.9**	**63.2**	**42.6**

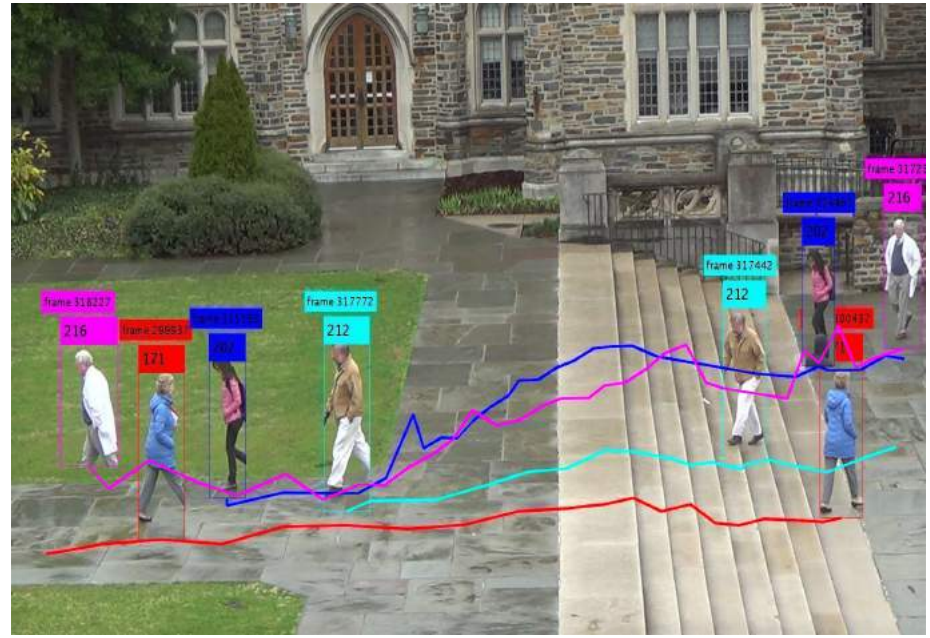
	Methods	IDF1↑	IDP↑	IDR↑
Multi-Camera Test-hard	[33]	56.2	67.0	48.4
	[26]	34.9	41.6	30.1
	Ours	**60.0**	**68.3**	**53.5**

IDP = Fraction of computed detections that are correctly identified
IDR = Fraction of ground-truth detections that are correctly identified
IDF1 = Ratio of correctly identified detections over the average number of ground-truth and computed detections

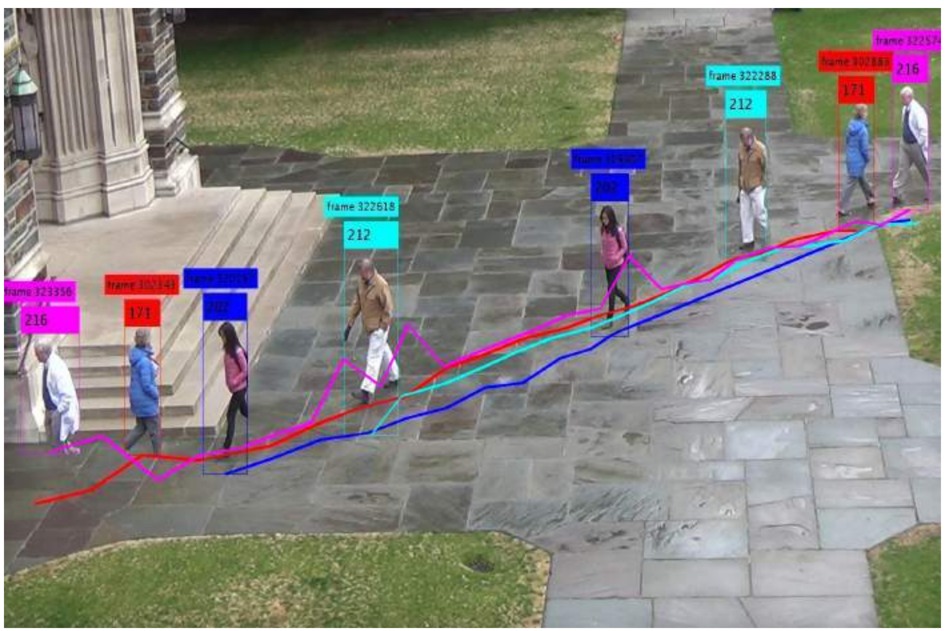
[33] E. Ristani et al. Performance measures and a data set for multi-target multi-camera tracking *(ECCV 2016)*
[26] A. Maksai et al. Non-Markovian globally consistent multi-object tracking *(ICCV 2017)*



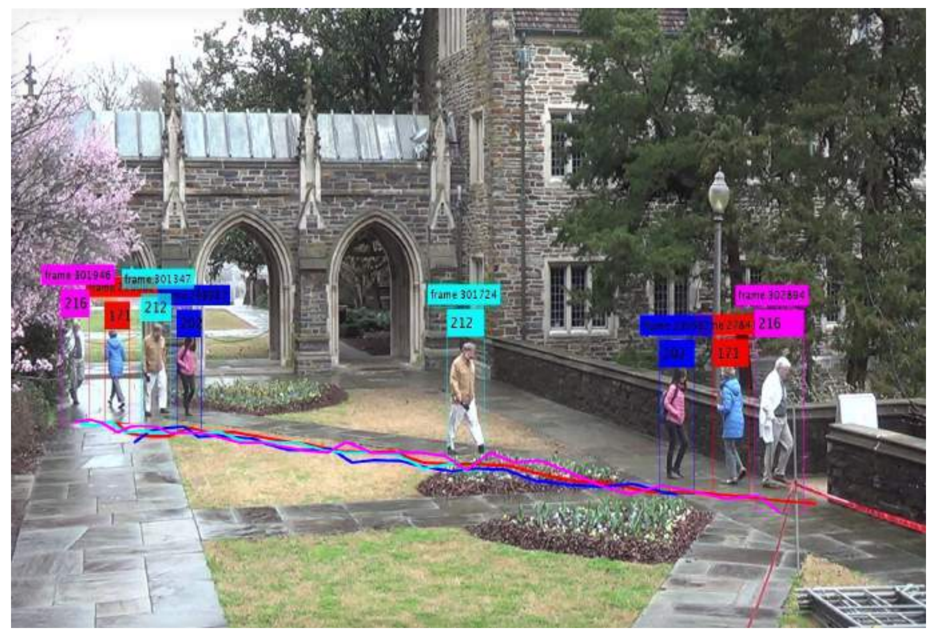
Camera 1



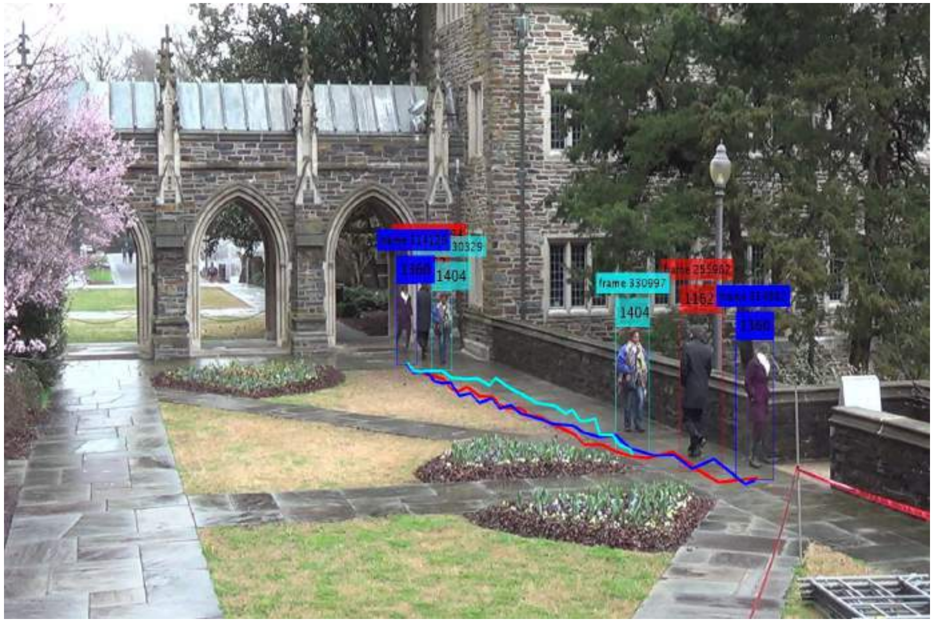
Camera 2



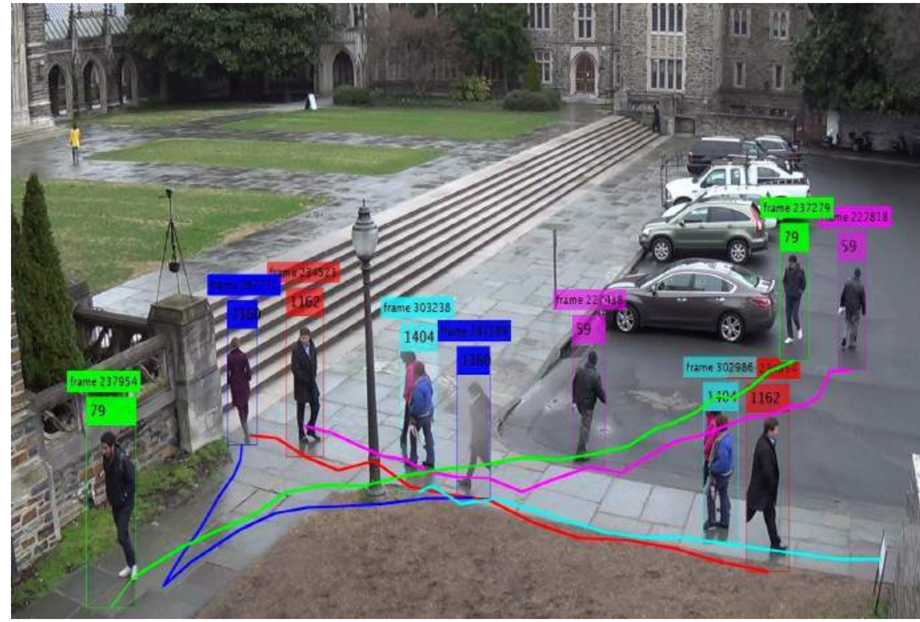
Camera 5



Camera 6



Camera 6



Camera 8

Camera 1

Camera 7


#009
11
19

Conclusions

Dominant sets and related concepts shown to be a powerful notion for attacking a variety of computer vision problems, e.g.,

- Interactive image segmentation and cosegmentation
- Geo-localization
- Group detection in image and videos
- Person re-identification
- Multi-target tracking
- …

On-going work focuses on combining deep learning and DS's for improving performances.

References

M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *PAMI* (2007)

S. Rota Bulò and M. Pelillo. A game-theoretic approach to hypergraph clustering. *PAMI* (2013)

E. Zemene, L. Tesfaye and M. Pelillo. Dominant sets for "constrained" image segmentation. *PAMI* (2018)

E. Zemene *et al.* Large-scale image geo-localization using dominant sets. *PAMI* (2018)

S. Vascon *et al.* Detecting conversational groups in images and sequences: A robust game-theoretic approach. *CVIU* (2016)

Y. Tariku *et al.* Multi-target tracking in multiple non-overlapping cameras using fast-constrained dominant sets. *arXiv:1706.06196*

S. Rota Bulò and M. Pelillo. Dominant-set clustering: A review. *EJOR* (2017).