Graph Based Shape Representations for Object Detection

Jeffrey Byrne
University of Pennsylvania
WPE-II Presentation
April 24, 2012

Outline

- Introduction and Motivation
  - What is shape and why is it important?
  - Why graphs to represent shape?
  - Scope
- Graph Based Shape Representations
  - Problem definition
  - Geometric representations
  - Topological representations
- Analysis
  - Task independent comparison
  - Is object detection geometric?
  - Is object detection topological?
- Summary and Conclusions

Why Shape?

Shape is the foundation for basic level categorization

Shape is a strong feature even when other local features are unrecognizable

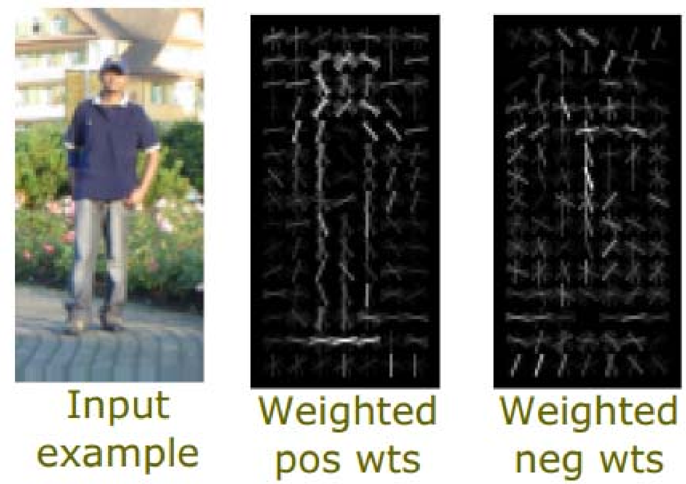
What is Shape?

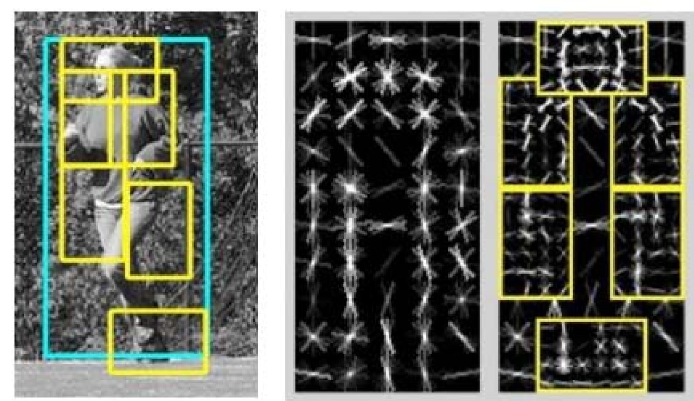
Discipline	Shape is ____?
Cognitive Science	Objective shape and shape equivalence
Statistics	Statistical shape models
Algebraic Geometry	Equivalence class under a group of transformations
Algebraic Topology	Homeomorphism
Computer Vision	Task specific representations $(n \rightarrow \infty)$…

Representation and generalization of shape is a primary problem in computer vision

Shape Representations in Vision

- Geometric Representations
  - templates vs. **graphs**
- Task specific
  - Low, mid and **high level** vision tasks
  - Object detection
- Why graphs for object detection?
  - Holisic part based representations
  - Detection as *similarity by alignment to prototypes*
  - Supported by contemporary prototype and exemplar theories of categorization based on “family resemblances”





[1] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in In CVPR, 2005, pp. 886–893.
[2] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in CVPR, 2008.

Organization

- *Graph based shape representations for object detection*
- **Scope (✓)**
  - Geometric and topological graphical shape representations
  - Exemplar similarity without explicit category models
  - Similarity by alignment using exact graph matching
  - Detection of basic level categories
  - 2D not 3D
- **Scope (X)**
  - No category models: Generative models, global discriminative classifiers, *graph kernels*
  - Overlap with recent shape surveys
  - Non-geometric attributes
  - Indexing and retrieval
  - Other vision tasks

Outline

- Introduction and Motivation
  - What is shape and why is it important?
  - Why graphs to represent shape?
  - Scope
- Graph Based Shape Representations
  - Problem definition
  - Geometric representations
  - Topological representations
- Analysis
  - Task independent comparison
  - Is object detection geometric?
  - Is object detection topological?
- Summary and Conclusions

Graphical Representations of Shape

- Problem Definition
  - A *graph* G is a pair of sets (V, E) satisfying $E \subseteq V \times V$
  - *Graph matching* is the problem of finding exact or inexact correspondences between two graphs such that the relational structure is preserved
- Graph representation
  - Detection as similarity by alignment
  - Categorization as nearest neighbor
  - Taxonomy: structure, embedding, attributes, matching

Structure	Embedding	Attributes	Matching
General, tree, bipartite, star	Implicit, explicit	Patches, local feature descriptors, discriminative templates, inflection points, contours, regions, parts, weight matrix…	Quadratic assignment, linear assignment, graph edit distance, tree edit distance, dynamic programming, Generalized Hough transform

Quadratic Assignment Problem

$$
\begin{aligned}
x^*_{QAP} = \arg\max \quad & x^T W x \\
\text{s.t.} \forall (i,j) \quad & \mathbb{1}^T x_j = m \\
& x_i^T \mathbb{1} = n \\
& x_{ij} \in \{0,1\}
\end{aligned}
$$

- **Problem:** Given two attributed graphs G, G', find optimal assignment matrix X^*_{QAP} that satisfies mapping constraints
  - W is $|V||V'| \times |V||V'|$ edge compatibility matrix
  - Mapping constraints: one to one (n=m=1), many to one (n>m=1), many to many (n≥m>1)
  - If $m = n = 1$ and $|V| = |V'|$ then X^* is a *permutation matrix* Π, a square orthogonal binary matrix that is *doubly stochastic and* $y = \Pi x$ is a permutation of elements of x
- **Solution**: Quadratic assignment
  - Max weight edge preserving matching $(u, v) \in E \leftrightarrow \big(X(u), X(v)\big) \in E'$
  - Example: Facilities localization

Graduated Assignment

- Relaxation of quadratic assignment problem by graduated non-convexity
  - Max weight edge preserving matching for general graphs
  - Sequence of optimal solutions to linearized assignment problem
  - Example of optimization using *deterministic annealing*
- Key ideas
  - Softassign
  - Graduated non-convexity
- Graph based shape representation

Structure	**Embedding**	**Attributes**	**Matching**
General	Explicit	Weight matrix	Deterministic annealing

[1] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, 1996.

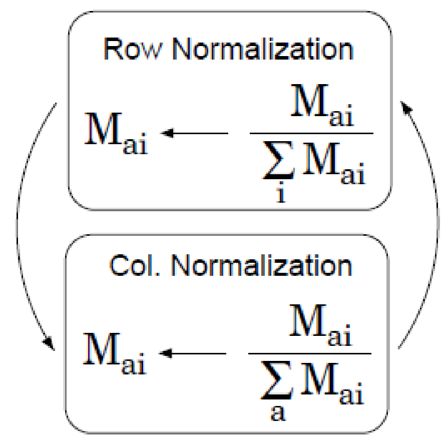
Softassign

- **Softmax**
  - **Problem**: Find the maximum real number in a set X, as represented by an indicator vector m
  - **Solution:** Iteratively update m_j increasing control parameter β, and as $\beta \rightarrow \infty$ the maximum m_j approaches one, all others zero

$$X = \{x_1, \ldots, x_n \mid x_i \in \boldsymbol{R}\}$$
$$m_i \in \{0,1\}, \mathbf{1}^T m = 1$$
$$m_j = 1 \leftrightarrow j = argmax(X_i)$$

$$m_j = \frac{\exp(\beta X_j)}{\sum_{i=1}^{I} \exp(\beta X_i)}$$

- **Softassign**
  - **Approach:** Softmax with Sinkhorn normalization to enforce bistochastic assignment constraints
  - **Sinkhorn normalization:** Any square matrix with strictly positive entries will converge to a doubly stochastic matrix by iteratively normalizing rows and columns



Graduated Assignment

- **Problem:** Given two attributed graphs G, G' and an edge relatability cost C, find a permutation matrix M that minimizes E_{ga}

$$E_{ga}(M) = -\frac{1}{2}\sum_{a=1}^{A}\sum_{i=1}^{I}\sum_{b=1}^{A}\sum_{j=1}^{I} M_{ai}M_{bj}C_{aibj}$$

$$E_{ga}(M) = -\frac{1}{2}m^T Cm$$

- **Solution:** Linearization of quadratic objective results in a linear assignment problem, for which matching constraints on Q are enforced by *softassign*.

$$E_{ga}(M) = m^T Cm$$

$$E_{ga}(M) \approx m_0^T Cm_0 - Q^T(m - m_0)$$

$$Q = -\frac{\partial E_{ga}}{\partial M_{ai}}\bigg|_{M=M_0} = \sum_{b=1}^{A}\sum_{j=1}^{I} M_{bj}^0 C_{aibj}$$

$$\underset{m}{\operatorname{argmin}}\ E_{ga} = \underset{m}{\operatorname{argmax}}\ Q^T m$$

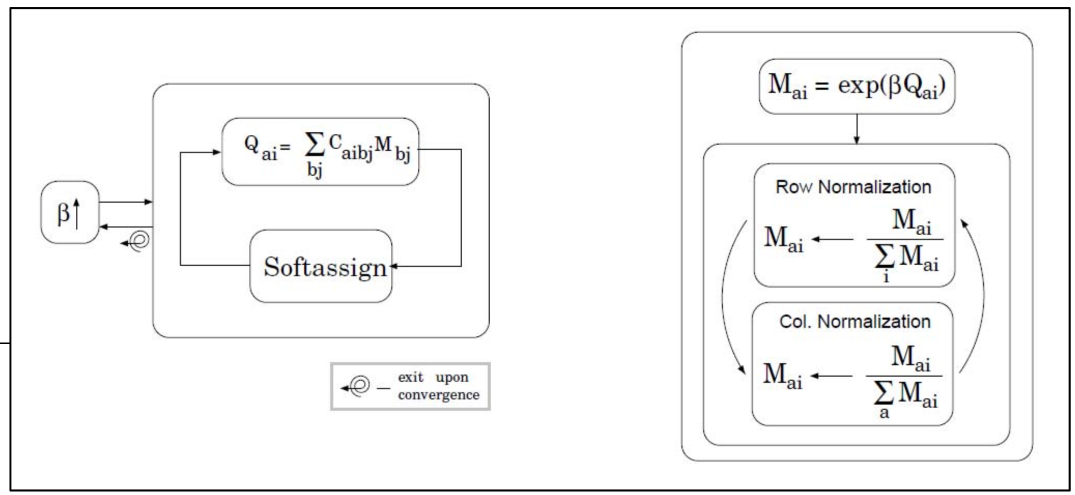
Linear assignment problem

Graduated Assignment Algorithm

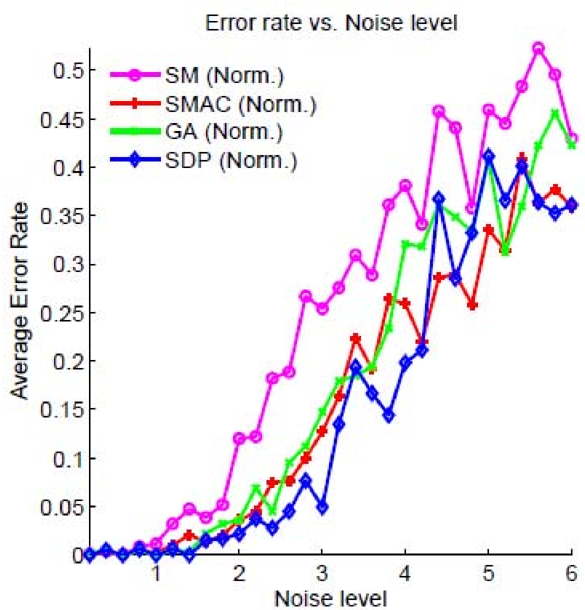
Algorithm 1: Graduated assignment for weighted graph matching

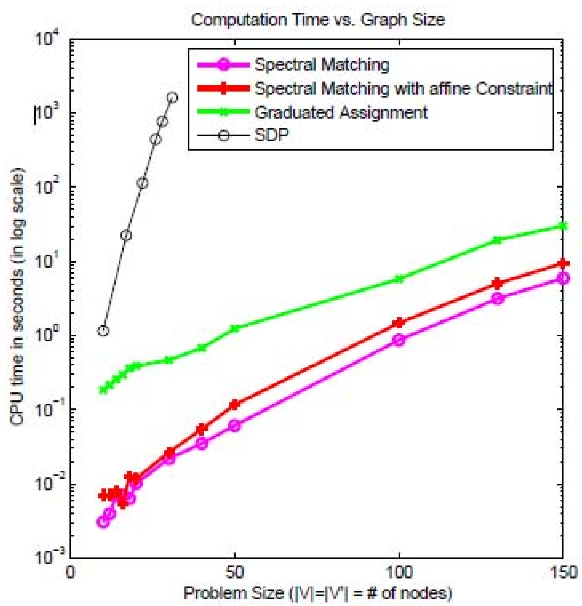
Input: $\beta \leftarrow \beta_0$, $\hat{M}_{ai} \leftarrow (1+\varepsilon)$
Output: $\Pi = f_h(M)$
while $\beta \leq \beta_f$ **do**
  while $I < I_0$ **do**
    $Q_{ai} \leftarrow -\frac{\partial E}{\partial M_{ai}}$
    $M^0_{ai} \leftarrow exp(\beta Q_{ai})$
    while $J < J_0$ **do**
      Sinkhorn normalization:
      $\hat{M}^1_{ai} \leftarrow \frac{\hat{M}^0_{ai}}{\sum_{i=1}^{I+1} \hat{M}^0_{ai}}$
      $\hat{M}^0_{ai} \leftarrow \frac{\hat{M}^1_{ai}}{\sum_{i=1}^{A+1} \hat{M}^1_{ai}}$
    end
    $\beta \leftarrow \beta_r \beta$
  end
end

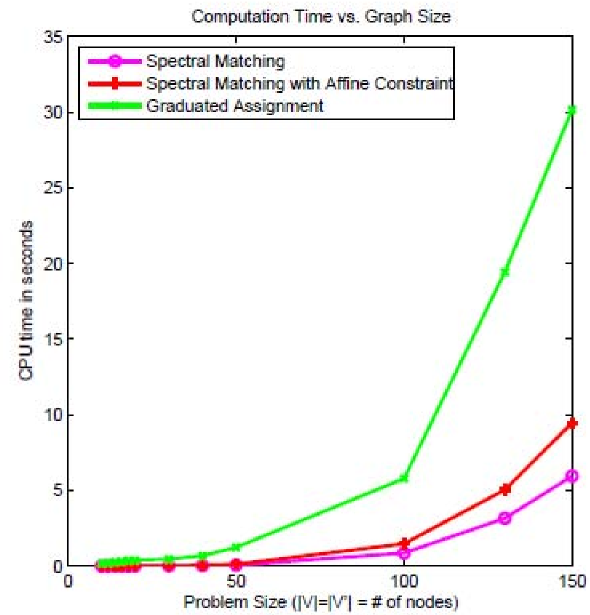
Complexity: O(|E||E'|)



Graduated Assignment Results







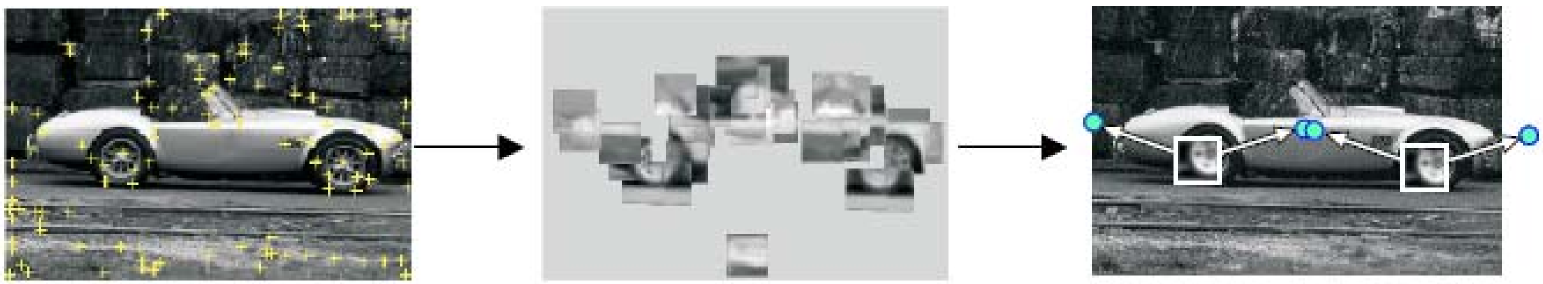
- Results
  - Task independent analysis: random graph matching
  - Compared with spectral matching
- Challenges: final discretization heuristic, optimal annealing schedule

[1] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in Advances in Neural Information Processing Systems (NIPS), 2006.

Outline

- Introduction and Motivation
  - What is shape and why is it important?
  - Why graphs to represent shape?
  - Scope
- Graph Based Shape Representations
  - Problem definition
  - Geometric representations
  - Topological representations
- Analysis
  - Task independent comparison
  - Is object detection geometric?
  - Is object detection topological?
- Summary and Conclusions

Tree Based Shape Representation



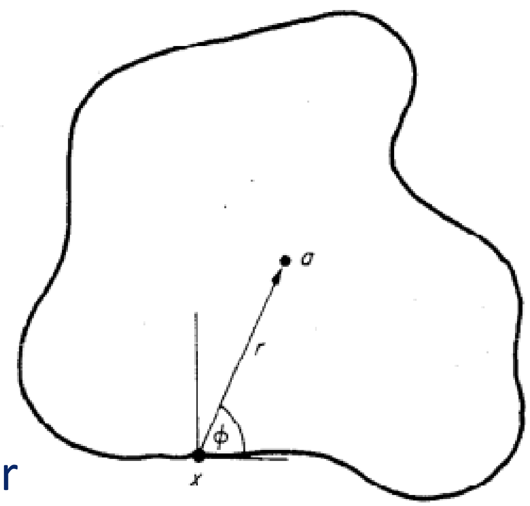
- **Problem**: Part based object detection
- **Solution**: *Implicit shape model* for a basic level category is a non-parametric spatial distribution of patch prototypes
  - Geometry captured in relative spatial distribution of parts
  - Implicit → No category model (!)
  - Graph based shape representation

Structure	Embedding	Attributes	Matching
Star	Implicit	Patch prototypes	Generalized Hough transform

[1] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," IJCV vol. 77, pp. 259–289, 2008.

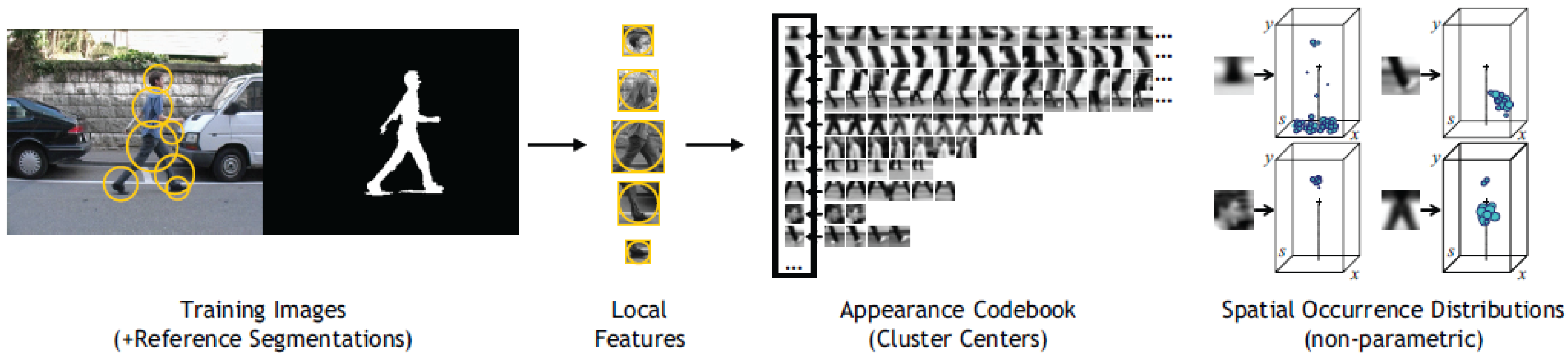
Generalized Hough Transform

- Hough transform
  - Detection of parameterized shapes by voting
  - Construct discretized parameter voting space
  - Accumulate votes by image support
  - Detection hypotheses are local maxima
- Generalized Hough Transform
  - Non-parametric extension to arbitrary shapes
  - Vote for *pose*, not shape parameters
  - Simplest voting space is translation only, vote for centroid relative to feature
  - *Challenges*: discretization, multiple local maxima, unmodelled pose variations, deformable shapes, star graph assumption

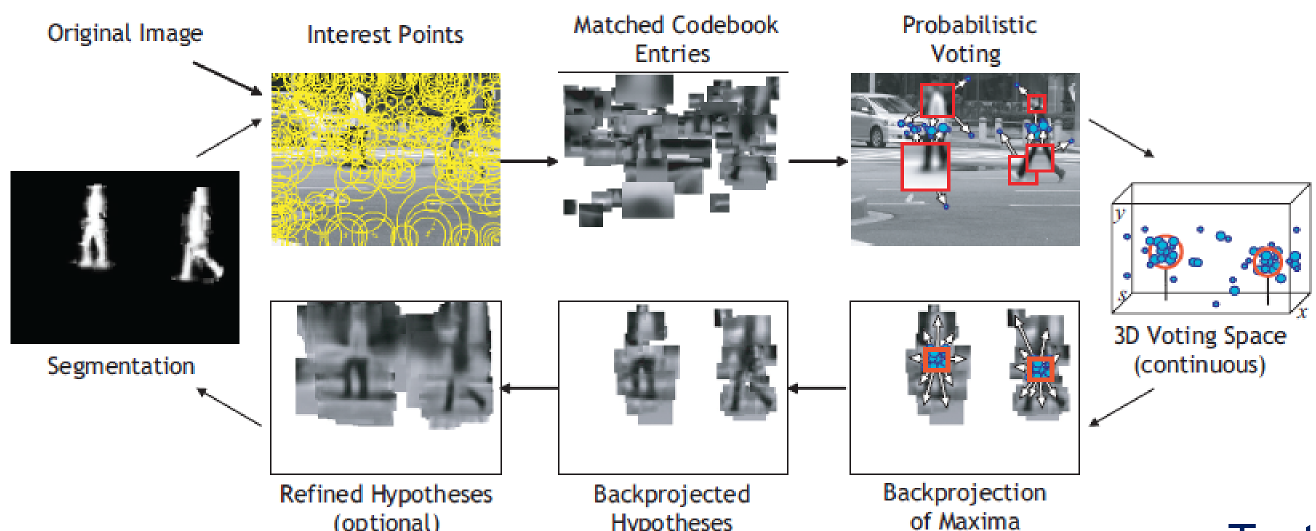


[1] D.H. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” Pattern Recognition, vol. 13, pp. 111–122, 1981.

ISM Training and Testing



Training



Testing

Implicit Shape Model

$$p(o_n,x) = \sum_i p(o_n,x,f_i,l_i)$$
$$p(o_n,x) = \sum_i p(o_n,x|f_i,l_i)p(f_i,l_i)$$
$$p(o_n,x) \propto \sum_i p(o_n,x|f_i,l_i)$$
$$p(o_n,x) \propto \sum_{i,j} p(o_n,x|C_i,f_j,l_j)p(C_i|f_j,l_j)$$
$$p(o_n,x) \propto \sum_{i,j} p(x|o_n,C_i,l_j)p(o_n|C_i,l_j)p(C_i|f_j)$$

Var	Description
o_n	Object category label
x	Detected object position
f	Observed feature
l	Observed feature location
C_i	Codebook prototype

Probabilistic foundation for vote weights of the generalized Hough transform

ISM Detection Results

State of the art performance (~2004)

Dataset	Training (#)	Testing (#)	Equal Error Rate (%)
UIUC cars, single scale	50	200	97.5
UIUC cars, multiscale	50	139	95
Caltech cars	126	1896	70.9
TUD motorbikes	153	125	87
VOC motorbikes	153	227	48
TUD pedestrians	210	595	80

- Analysis
  - *Clustering*: Performance degrades gracefully as number of prototypes is decreased
  - *Training set size:* Diminishing returns at ~30 training images
  - *Scale invariant interest points*: Hessian-Laplace and DoG perform similarly
  - *Features*: SIFT and Shape context attributes outperformed grayscale patches
- Challenges: partial occlusions, in-plane rotation, background clutter

Outline

- Introduction and Motivation
  - What is shape and why is it important?
  - Why graphs to represent shape?
  - Scope
- Graph Based Shape Representations
  - Problem definition
  - Geometric representations
  - Topological representations
- Analysis
  - Task independent comparison
  - Is object detection geometric?
  - Is object detection topological?
- Summary and Conclusions

Topological Representations

- Topology is qualitative geometry
  - Invariant properties when considering stretching and bending but not tearing or gluing
  - Non-metric properties: “Connected”, “boundary”, “surrounded”, “interior”…
  - Higher order simplexes can capture complex relations
- Simplicial homology
- Persistent homology
- Homologous cycle matching
  - Graph based shape representations

Structure	Embedding	Attributes	Matching
Simplical complex	Explicit	Weighted cycles	Homology preserving matching

Simplicial Complex

Definition 4.1. A p-dimensional simplex or p-simplex is the convex hull of $p+1$ affinely independent vertices $v \in \mathbb{R}^D$

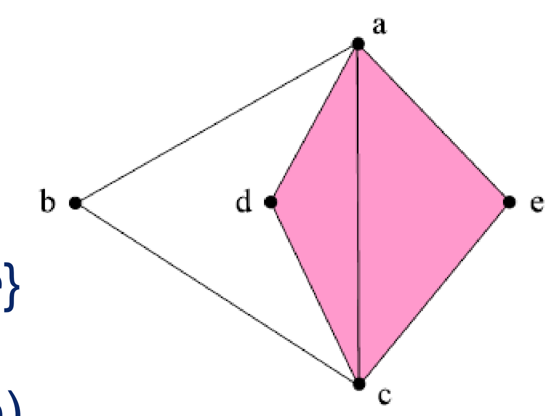
Definition 4.2. A *face* of a p-simplex σ is a non-empty subset of vertices of σ.

Definition 4.3. A simplicial complex K is a set of simplices that satisfies the following closure conditions

(i) Any face of a simplex in K is a simplex in K

(ii) The intersection of any two simplexes $\sigma_i, \sigma_j \in K$ is a face of both σ_i, σ_j

- 0-simplexes “vertices”: {*a*,*b*,*c*,*d*,*e*}
- 1-simplexes “edges”: {*ab*,*ac*,*ad*,*ae*,*cd*,*ce*}
- **2-simplexes “triangles”:** {*acd*,*ace*}
- Examples
  - K = {*a*,*b*,*c*,*d*,*e*,*ab*,*ac*,*ad*,*ae*,*cd*,*ce*,*acd*,*ace*}
  - Faces of 1-simplex *ab* are *a* and *b*
  - *ac* is a face of both *acd* and *ace* (closure)



[1] A. Hatcher, Algebraic Topology, Cambridge University Press, 2002.
[2] D. Freedman and C. Chen, “Algebraic topology for computer vision,” Nova Science, 2011.

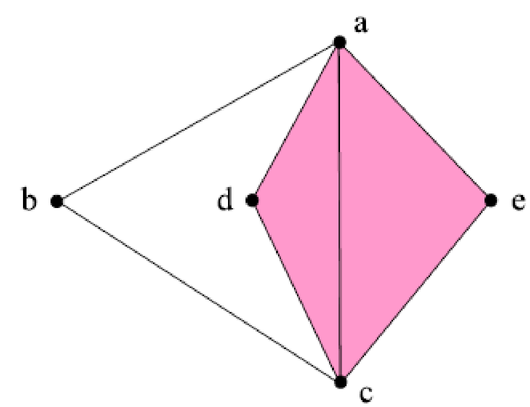
Chain Group

Let K be a finite simplicial complex of dimension p, such that all simplexes $\sigma \in K$ have dimension at most p. A simplicial k-chain is a finite formal sum of k-simplices

$$\sum_{i=1}^{N} c_i \sigma_i, \; c_i \in \mathbb{Z}_2, \; \sigma_i \in K \quad (23)$$

where $c_i \in \{-1, 0, 1\}$ are binary valued coefficients. For each $k \geq 0$, k-chains along with the modulo-2 addition operator form the *chain group* $C_k(K)$.

Closure	$c_{abc} = c_{ab} + c_{bc}$
Associativity	$c_{ab} + (c_{bc} + c_{ac}) = (c_{ab} + c_{bc}) + c_{ac}$
Identity	$c_{ab} = c_{ab} + c_0$
Invertibility	$c_{ab} - c_{ab} = -c_{ab} + c_{ab} = c_0$



A k-chain is an indicator vector for included simplexes

Boundary Group

Definition 4.4. The boundary operator $\partial_k \colon C_k \to C_{k-1}$ is a homomorphism between chain groups such that

$$\partial_k(\sigma) = \sum_{i=0}^{k} (-1)^i \langle v^0, \ldots, v^{i-1}, \hat{v^i}, v^{i+1}, \ldots, v^k \rangle \qquad (24)$$

Lemma 4.6. *Given a boundary homomorphism* ∂,

(i) The boundary of a boundary is zero, $\partial_k \partial_{k+1} d = 0$, for every integer k and every $(k+1)$-chain d.

(ii) A k-cycle c is a k-chain with zero boundary $\partial_k c = 0$

(iii) The boundary of every 0-simplex is zero.

(iv) The cycle group $Z_k = ker(\partial_k) = \{x \in C_k(K) : \partial_k x = 0\}$

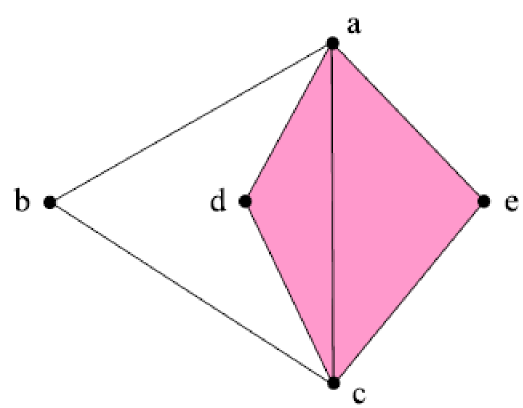
(v) The boundary group $B_k = im(\partial_{k+1}) = \{x \in C_k(K) : \exists\, y\ s.t.\ x = \partial_{k+1} y\}$.

ab	ac	ad	ae	bc	cd	ce

$$\partial_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

acd	ace

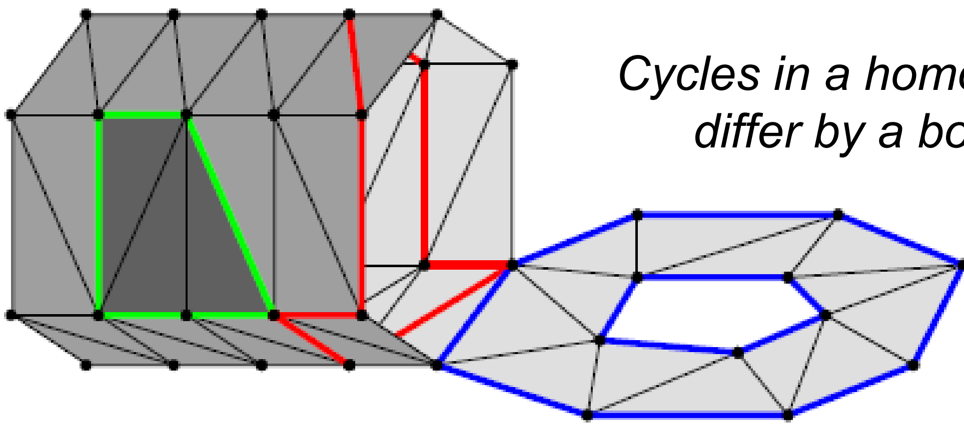
$$\partial_2 = \begin{bmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$



$$c_{abca} = \partial_1 [1\ 1\ 0\ 0\ 1\ 0\ 0\ ]^T = \mathbf{0}^T$$

Homology Classes

Definition 4.7. An *homology class* is an equivalence class of cycles such that for a fixed representative cycle z_0, $\{z | z = z_0 + \partial_{k+1} c, c \in C_{k+1}(K)\}$, where equivalent cycles of the same homology class are *homologous* and denoted $c \sim c'$.



Cycles in a homology class differ by a boundary

- **Green cycle** is a boundary of dark grey 2-simplexes (▲)
- **Blue cycles** are homologous, not homologous to **red cycle**

Homologous Cycle Matching

- **Problem**: Given a p-chain c in simplicial complex K, find an optimal homologous cycle x such that $c \sim x$ and $\|Wx\|_1$ is minimized.

$$
\begin{aligned}
(x^*, y^*) = \arg\min \quad & \|Wx\|_1 \\
\text{s.t.} \quad & x = c + \partial_{p+1} y \\
& x \in \mathbb{Z}^m, y \in \mathbb{Z}^n
\end{aligned}
$$

- **Solution**: Convert weighed $L1$-norm minimization to equivalent convex linear program by adding slacks
  - Linear inequality constraints $Ax \geq b$ are *totally unimodular* (TUM) if boundary matrix ∂_{p+1} is TUM
  - Therefore, linear program has an optimal integer solution. And…
  - $x \in \{-1,0,1\} \leftrightarrow 0 \leq (x^+, x^-) \leq 1$ for $x^+, x^- \in \boldsymbol{Z}^m$

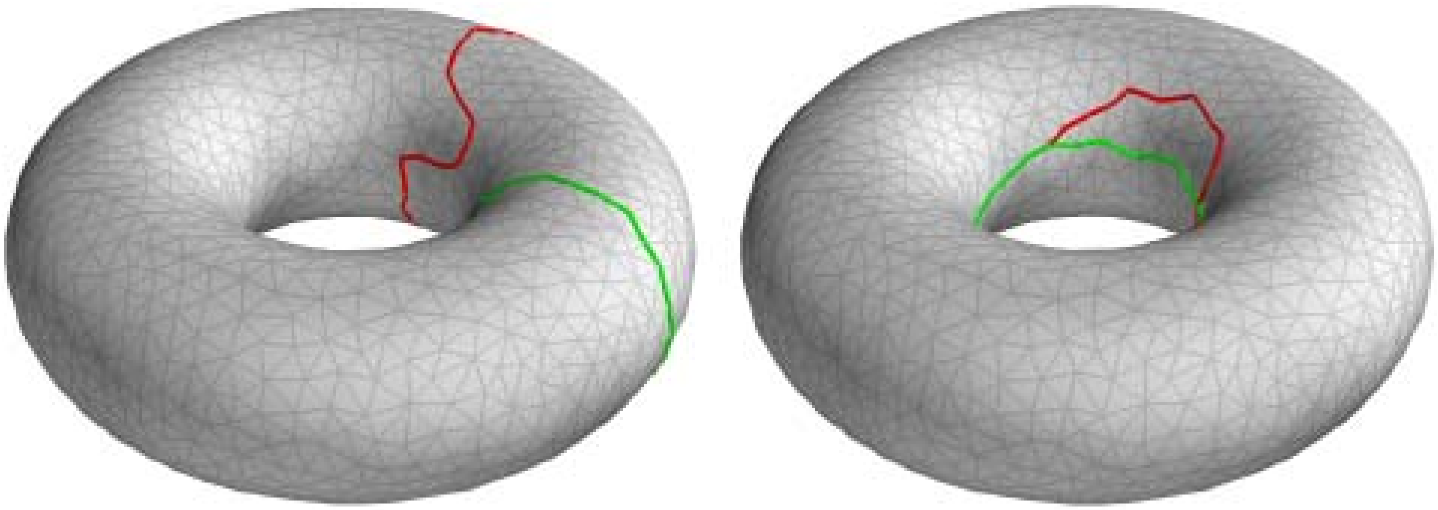
$$
\begin{aligned}
(x^+, x^-, y^+, y^-)^* = \arg\min \quad & \textstyle\sum_i w_i (x_i^+ + x_i^-) \\
\text{s.t.} \quad & x^+ - x^- = c + \partial_{p+1}(y^+ - y^-) \\
& y^+, y^- \geq 0 \\
& x^+, x^- \leq 1 \\
& x^+, x^- \geq 0
\end{aligned}
$$

$$
A = \begin{bmatrix} -I & I & \partial_{p+1} & -\partial_{p+1} \\ I & -I & -\partial_{p+1} & \partial_{p+1} \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad b = \begin{bmatrix} -c \\ c \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

LP has optimal and efficient integer solution

[1] T. K. Dey, A. Hirani, and B. Krishnamoorthy, “Optimal homologous cycles, total unimodularity, and linear programming,” STOC 2010

Homologous Cycle Matching Results



- *Given a* ***reference cycle*** *for a given homology group, optimal homologous cycle matching finds the* ***minimum weight homologous cycle****.*
- *Challenge:* how to determine reference cycles and weights?

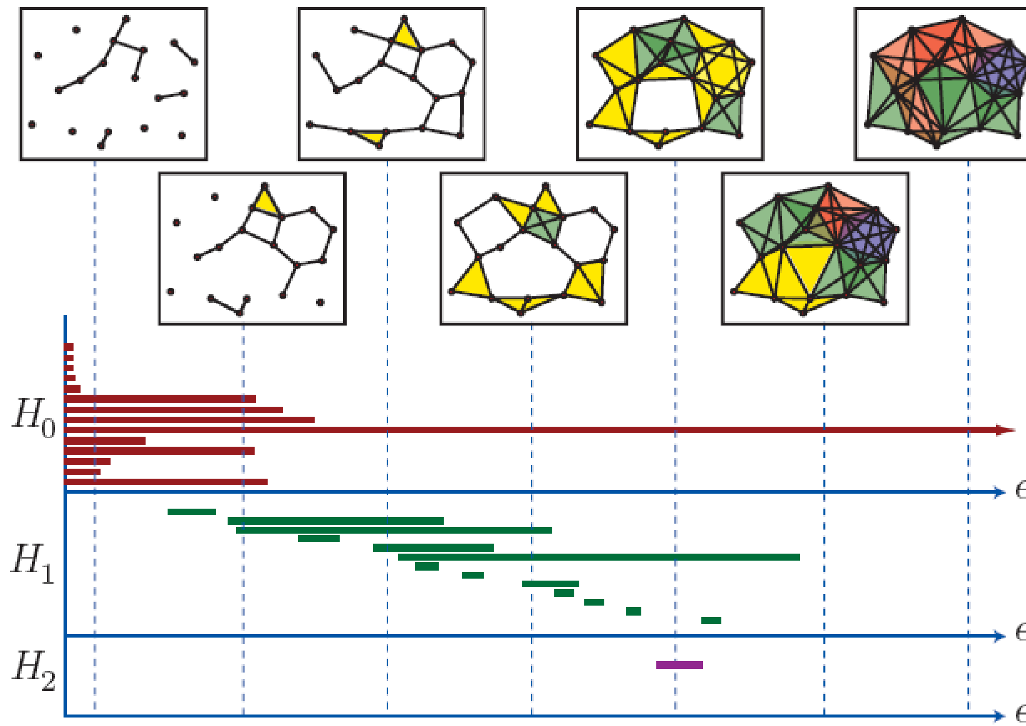
Persistent Homology

- **Problem**: For real data, what is signal and what is noise?
- **Solution**: Construct many simplicial complexes of increasing scale to determine which topological invariants *persist.*

Betti numbers β capture topological invariants

$$\beta_k = rank(L_k)$$

$$L_k = \partial_k^T \partial_k + \partial_{k+1} \partial_{k+1}^T$$



[1] R. Ghrist, "Barcodes: the persistent topology of data," Bull. Amer. Math. Soc, vol. 45, pp. 61–75, 2008.

Outline

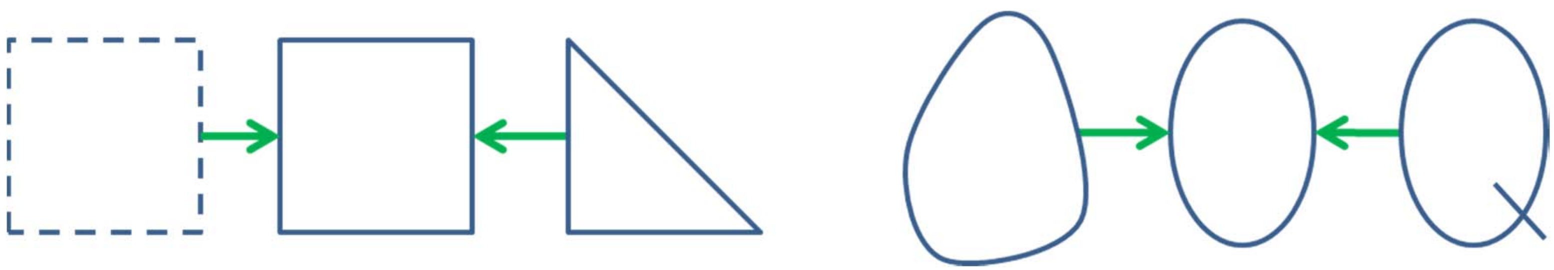
- Introduction and Motivation
  - What is shape and why is it important?
  - Why graphs to represent shape?
  - Scope
- Graph Based Shape Representations
  - Problem definition
  - Geometric representations
  - Topological representations
- Analysis
  - Task independent comparison
  - Is object detection geometric?
  - Is object detection topological?
- Summary and Conclusions

Analysis

Graph	Matching	Attributes	Explicit?	Reoresentative example
star	GHT	prototypes	no	ISM [26, 50]
star	DP	discriminative templates	no	[20]
star	linear assignment	contours	no	[30]
star	GHT	regions	no	[70]
bipartite	perfect matching	local descriptors	yes	shape context [28]
tree	tree edit distance	weight matrix	yes	[64][63][62]
tree	tree isomorphism	inflection points	yes	Shock graph [29]
general	QAP	weight matrix	yes	Annealing [51], spectral [59]
general	TPS	local descriptors	yes	[27]
simplex	Homologous cycles	weight matrix	yes	[52]

- Representational power vs. computational tractability
- Invariance vs. specificity
- Explicit embedding is noisy, implicit preferred
  - Marr's principal of least commitment
- Discriminative classifiers as the final step for detection
  - Outperforms matching with nearest neighbor classification

Is Object Detection Geometric?



Alignment only does not capture all perceptual similarities

Is Object Detection Topological?

- No, not by itself.
  - Bagels are not coffee cups
  - Invariance vs. specificity for generalization
- Invariances from algebraic topology can capture abstract global shape properties
  - Connected
  - Interior
  - Boundary
  - Surrounded
- Topology is almost nonexistent in vision today, but perhaps a combination of geometric and topological properties could provide improved generalization

Classical Categorization Revisited?

- Classical categorization is *definitional*
  - Example: triangles
  - Counter examples: dogs, metals, games…
  - Widely discredited as a categorization model
- Feature space classifiers provide *global* classification rules
  - Example: Bag of words attributes, one-vs-all SVM
  - Classifier learns optimal function mapping attributes to classes, and this function is a rule (if $w_1x_1+\ldots+w_nx_n > 0 \rightarrow$ category A)
- … so, feature space classifiers are classical categorization
  - Why so popular? Wrong representation, but best performing so far
  - Feynman's problem solving
- Alternatives
  - Non-parametric, data driven classifiers, local distance learning

Summary and Conclusions

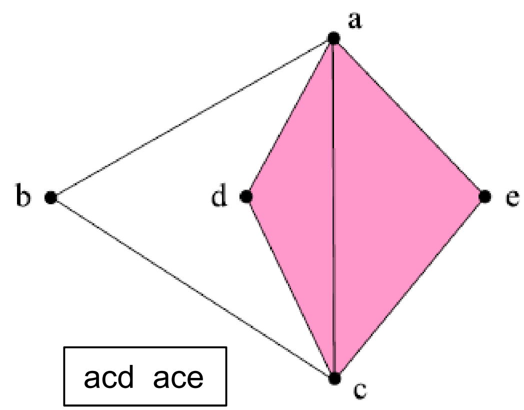
- Graph based shape representations do provide geometric and topological similarity by matching
  - … but not equivalent to perceptual similarity as shown by counterexamples
  - Computational complexity vs. representational expressivity
  - Invariance vs. specificity
  - Simpler graphs with overlapping attributes and a final discriminative classifier are a practical compromise

More work is needed to cross the representational gap for generalization of shape

ARCHIVE

Simplicial Homology - Example

- 0-simplexes: {a,b,c,d,e}
- 1-simplexes: {ab,ac,ad,ae,cd,ce}
- **2-simplexes:** {acd,ace}
- $c_{abcd} \sim c_{abc}$ are *homologous*



ab	ac	ad	ae	bc	cd	ce

$$\partial_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

acd	ace

$$\partial_2 = \begin{bmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$c = [1\ 0]^T$$

$$\partial_2 c = [0\ 1\ {-1}\ 0\ 0\ 1\ 0]^T$$

$$c_{abc} + \partial_2 c = [1\ 1\ 0\ 0\ 1\ 0\ 0]^T + [0\ {-1}\ 0\ 0\ 1\ 0]^T$$
$$= [1\ 0\ {-1}\ 0\ 1\ 1\ 0]^T = c_{abcd}$$

Implicit Shape Model - Segmentation

$$p(u_{fig}|o_n,x) = \sum_{p\in(f,l)} p(u_{fig}|o_n,x,f,l)p(f,l|o_n,x) \quad (8)$$

$$p(u_{fig}|o_n,x) = \sum_{p\in(f,l)} \sum_i p(u_{fig}|o_n,x,C_i,l)\frac{p(o_n,x|C_i,l)p(C_i|f)p(f,l)}{p(o_n,x)} \quad (9)$$

$$p(u_{fig}|o_n,x) \propto \sum_{p\in(f,l)} \sum_i p(u_{fig}|o_n,x,C_i,l)p(o_n,x|C_i,l)p(C_i|f) \quad (10)$$

Marginalization over all features that contain pixel u

Graph Definitions

Definition 2.5. A *cycle* in G is a path P such that $u_0 = u_k$.

A graph is *acyclic* if it contains no cycles.

Definition 2.6. A graph $G = (V, E)$ is a tree T if and only if any of the following properties hold.

- Any two vertices of T are linked by a unique path in T
- T is minimally connected, such that removing any edge e results in a disconnected graph
- T is maximally acyclic, such that T contains no cycles and adding an edge e between any two non-adjacent vertices will introduce a cycle

Note that definition 2.6 is in fact a standard theorem of graph theory and not a definition, however we state it here without proof and refer the reader to [54]. The *diameter* of a tree is the maximum length of the shortest path between any two vertices.

Definition 2.7. A tree is a *star* if and only if either the following two properties holds:

- The tree has diameter 2.
- The tree $T = (V, E)$ is a complete bipartite graph such that nodes are partitioned into disjoint sets $V_1 = \{v_0\}$ and $V_2 = V \backslash v_o$, and the cardinality $|V_1| = 1$.

Definition 2.8. Graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if and only if there exists a bijection $\phi : V \rightarrow V'$ with $(u, v) \in E \Leftrightarrow (\phi(u), \phi(v)) \in E'$ for all $u, v \in V$

where ϕ is a functional mapping from nodes in G to nodes in G' that is one to one and onto, and preserve incidence.

Definition 2.9. Graphs $G = (V, E)$ and $G' = (V', E')$ are *homomorphic* if and only if there exists an surjection $\phi : V \rightarrow V'$ with $(u, v) \in E \Leftrightarrow (\phi(u), \phi(v)) \in E'$

where ϕ is a functional mapping from nodes in G to nodes in G' that is onto, and preserve incidence.

Total Unimodularity

Lemma 4.10. *If a matrix A is totally unimodular (TUM) then a matrix A' obtained from A by any of the following operations is also TUM*

- $A' = A^T$
- $A' = [A, I]$
- *A' is obtained from gauss jordan pivoting*
- *Adding one or more rows or columns with all zeros and a single one.*
- *Removing a row or column from A*
- *Adding to A one or more rows or columns already in A*
- *Multiplying a row or column by -1*
- *Permuting rows or columns*

Lemma 4.11. *If ∂_{p+1} is totally unimodular, then the constraint matrix A in (29) is totally unimodular.*

Proof: The proof uses the properties for totally unimodularity described in lemma 4.10. Let B be of size $M \times N$. If B is TUM, then $[B\ B]$ is TUM by applying property five to repeat columns of B. If $[B\ B]$ is TUM, then $[B\ -B]$ is TUM by applying property seven to multiply the first N columns by -1. If $[B\ -B]$ is TUM then $[I\ I\ B\ -B]$ is TUM by applying property four. If $[I\ I\ B\ -B]$ is TUM then $[-I\ I\ B\ -B]$ is TUM by applying property seven. Observe that this form is the same as the first block row of A in (29). Let this block row be A_1. If A_1 is TUM, then $[A_1^T\ -A_1^T]^T$ is TUM by applying properties one, six and seven. Let this matrix be A_{12}. If A_{12} is TUM, then A is TUM by applying properties four and six. Therefore, since $B = \partial_{p+1}$, if ∂_{p+1} is TUM, then A is TUM. □

Boundary Matrix TUM

Lemma 4.11. *If ∂_{p+1} is totally unimodular, then the constraint matrix A in (29) is totally unimodular.*

Theorem 5.2. *$[\partial_{p+1}]$ is totally unimodular if and only if $H_p(L, L_0)$ is torsion-free, for all pure subcomplexes L_0, L of K of dimensions p and $p+1$ respectively, where $L_0 \subset L$.*

Graph Definitions

Definition 2.1. A graph G is a pair (V, E) of sets satisfying $E \subseteq V \times V$

Elements E are 2-element subsets of V formed from the Cartesian product of V. The elements of set V are *vertices* and elements of E are *edges*. Two vertices (u, v) are *adjacent* if there exists an edge $(u, v) \in E$. Vertices (u, v) are *incident* with an edge e if $(u, v) \in E$. The *order* of G is $|V|$ or the number of vertices. The *degree* of a node $v \in V$ is the number of incident edges. A graph G' is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$.

Definition 2.2. An *attributed graph* is a graph $G = (V, E, \alpha)$ that has been augmented with a set of node and edge attributes $\alpha = \{\alpha_V, \alpha_E\}$ such that $v \in V, (u, v) \in E$ there exists node attributes $\alpha_V(v)$ and edge attributes $\alpha_E(u, v)$.

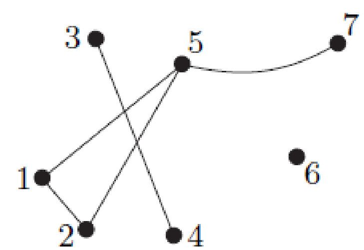
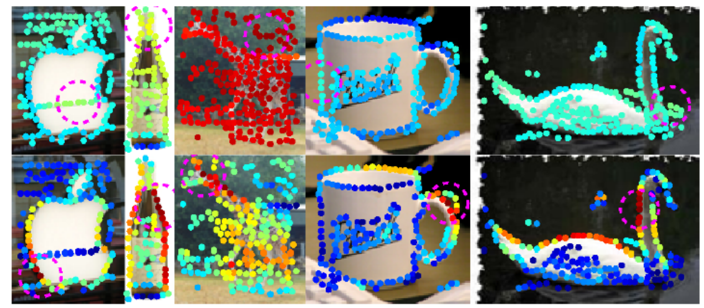
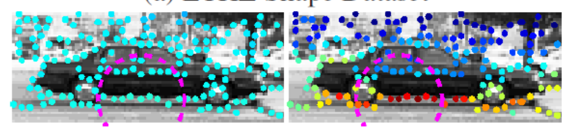


Fig. 1.1.1. The graph on $V = \{1, \ldots, 7\}$ with edge set $E = \{\{1,2\}, \{1,5\}, \{2,5\}, \{3,4\}, \{5,7\}\}$

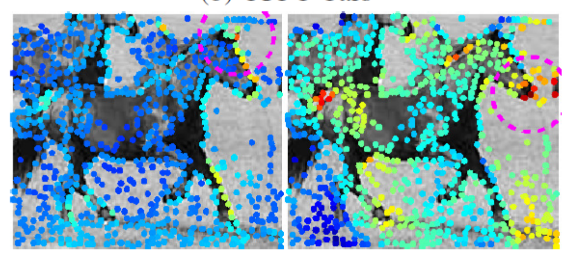
Max Margin Hough Transform



(a) ETHZ Shape Dataset



(b) UIUC Cars



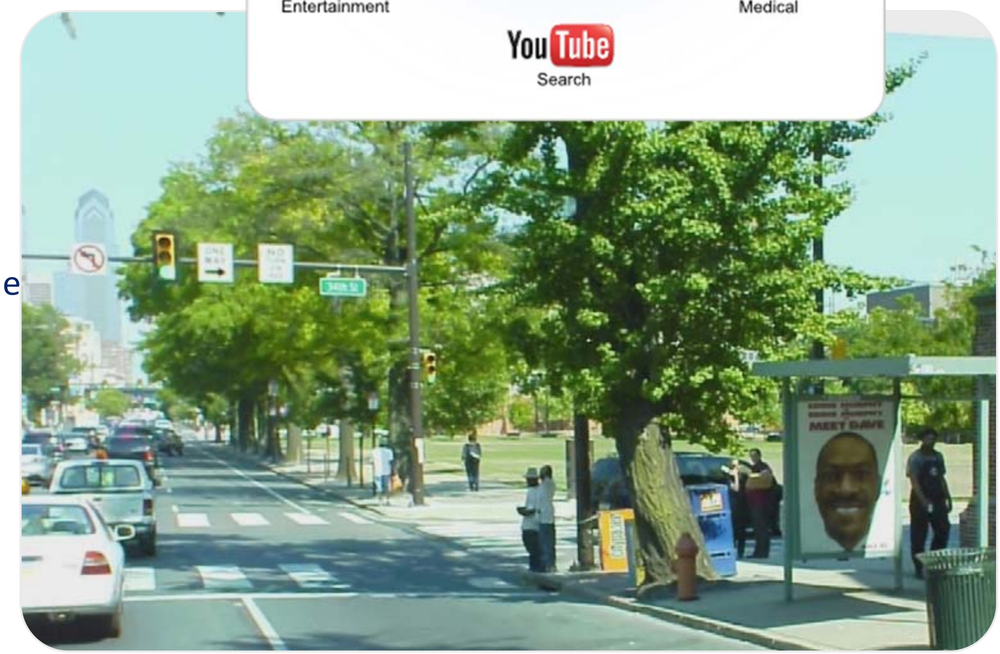
(c) INRIA Horses

ISM extension optimizing vote weights on repeatable and stable parts

[1] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in CVPR, 2009, pp. 1038–1045

Overview

- The long tail of computer vision…
- Vision is "Big Data"
  - Persistent surveillance
  - Mobile cameras
  - Visual learning
  - Search
- Vision is "Big, *Fast* Data"
  - Real time, streaming imagery
  - Actionable time constraints
- Vision is *"*Big, Fast, *Diverse* Data"
  - Complex image, many functions
  - The long tail requires a big library
  - Imperfect performance is often acceptable to prioritize big data



Computer vision is a "big, fast and diverse" problem and needs a new platform suitable for computation, sharing and distribution