Canonical Sets in Graphs

Ali Shokoufandeh,

Department of Computer Science, Drexel University

Motivation

- Many problems in computer vision share the following common theme:
  - "Given a large data set select a subset of its elements that best represent the original set."
- This reduction is usually driven by the requirements of a particular application or domain:
  - Reducing the space complexity of the data
  - Improving the performance of the algorithms
  - Dealing with oversampled, noisy data

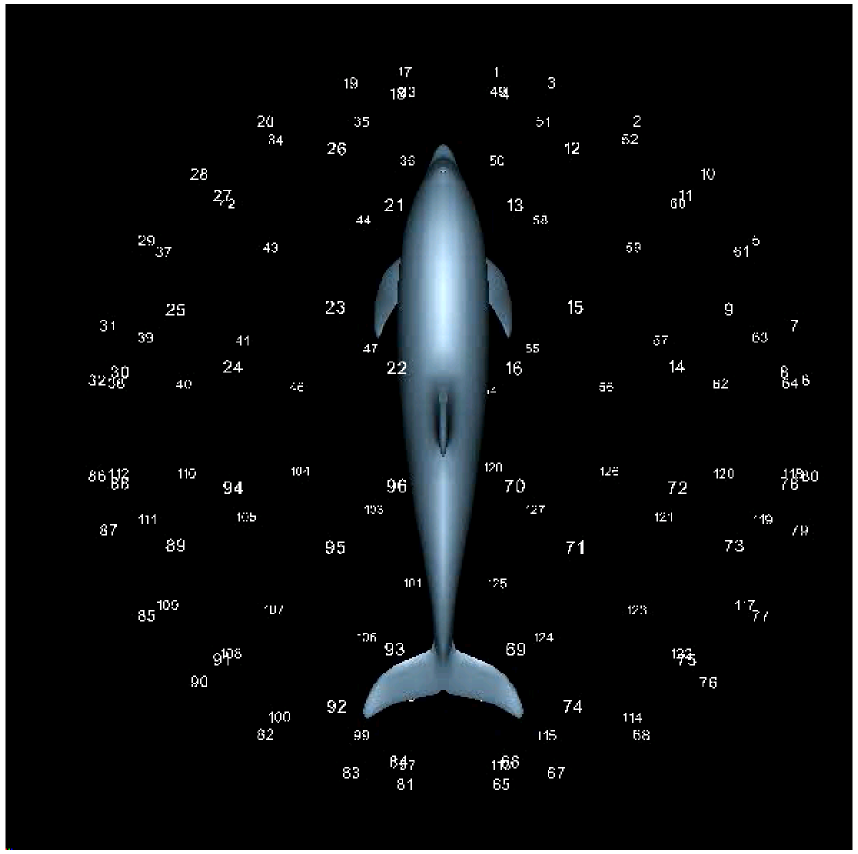
Overview

- Sample applications
  - How clustering algorithms help
  - Need for direct feature selection
- How optimization helps?
  - Discrete formulation
  - Complexity of discrete problems
  - Approximate algorithms for subset selection

A Typical Scenario:
View-based 3D Recognition

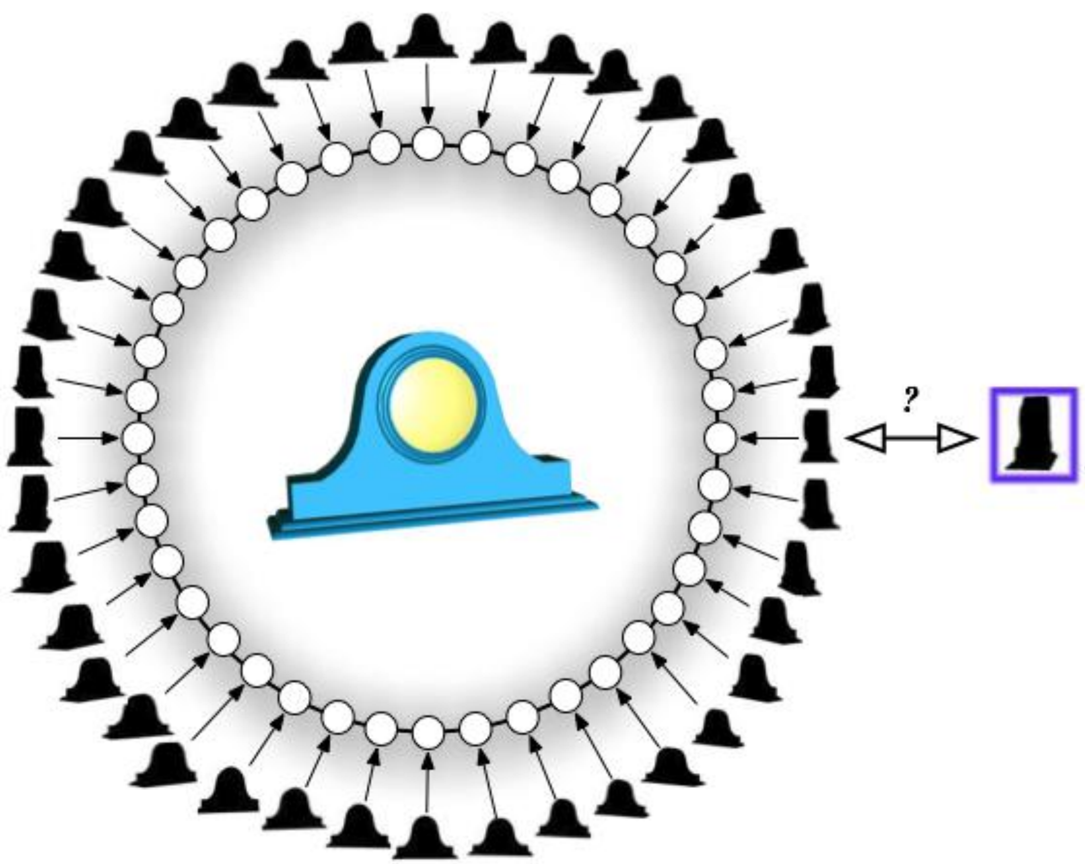
Representation model

- A 3D object will be represented with a set of 2D views.
- This results in significant reduction in dimensionality when comparing objects.



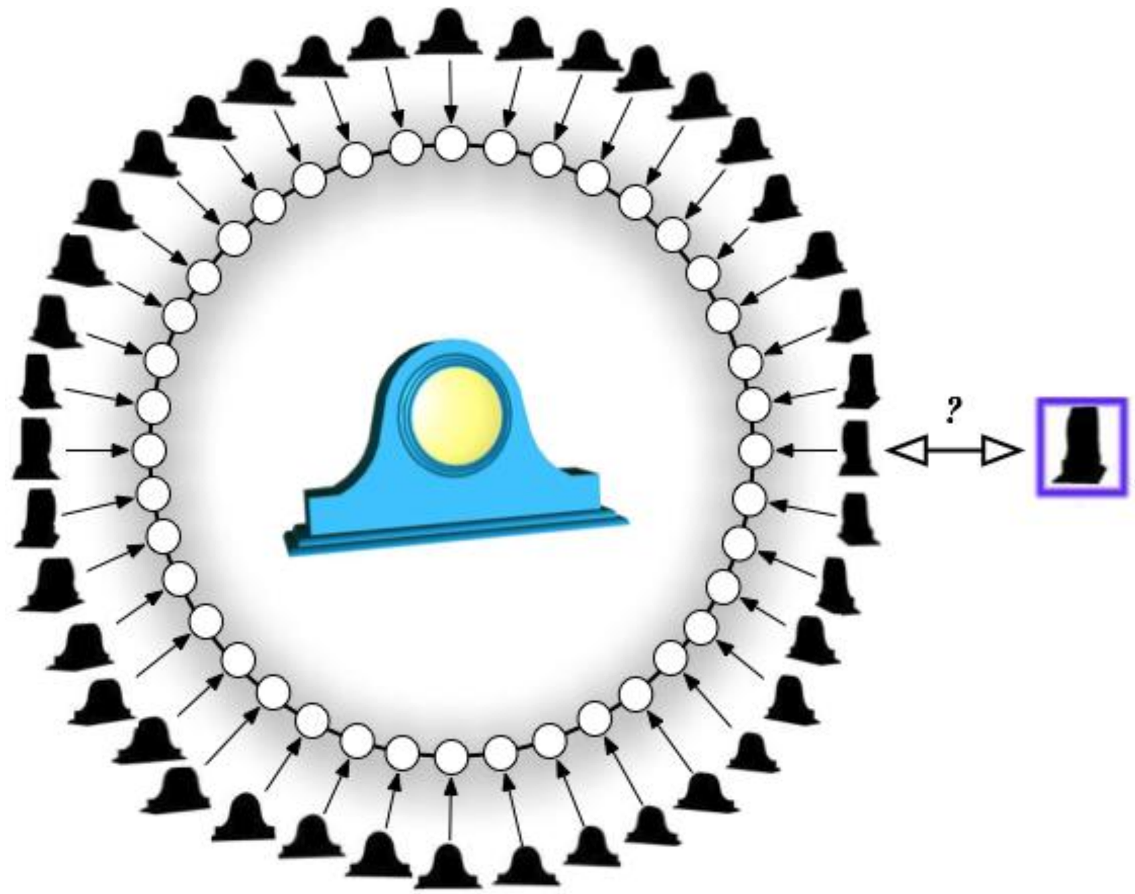
Downside of View-based Recognition

The recognition algorithm compares 2D views rather than comparing 3D objects.



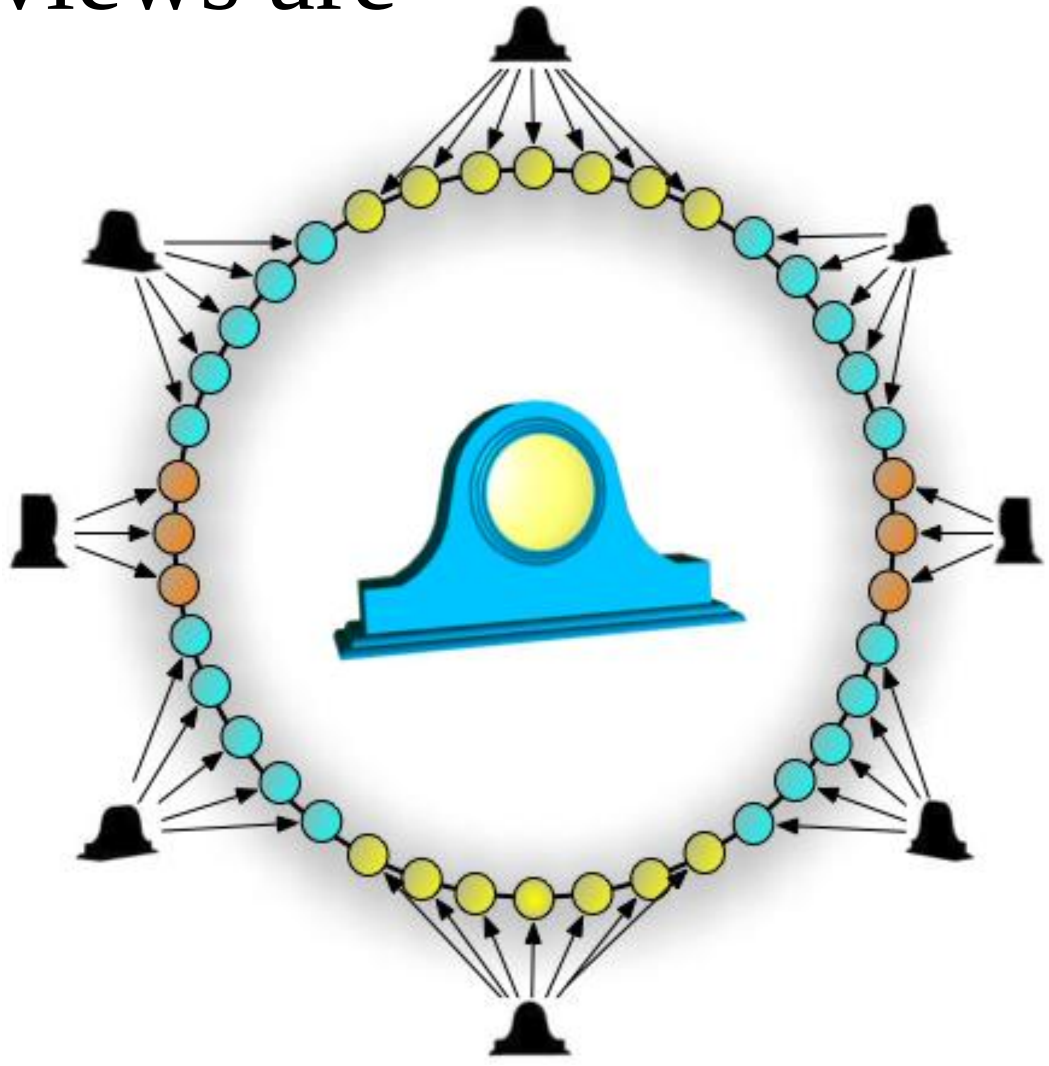
A Typical Scenario: View-based 3D Recognition

- All the gain in dimensionality reduction seems to have vanished as a result of number of necessary comparisons.



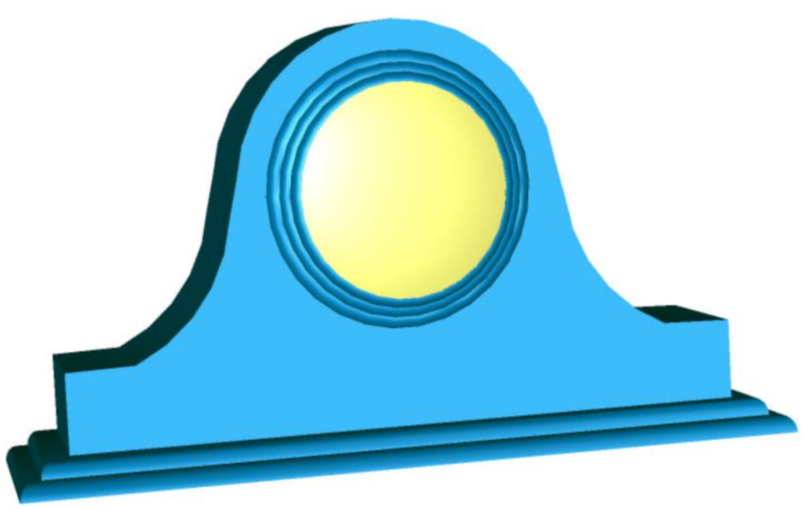
Redundancy Helps

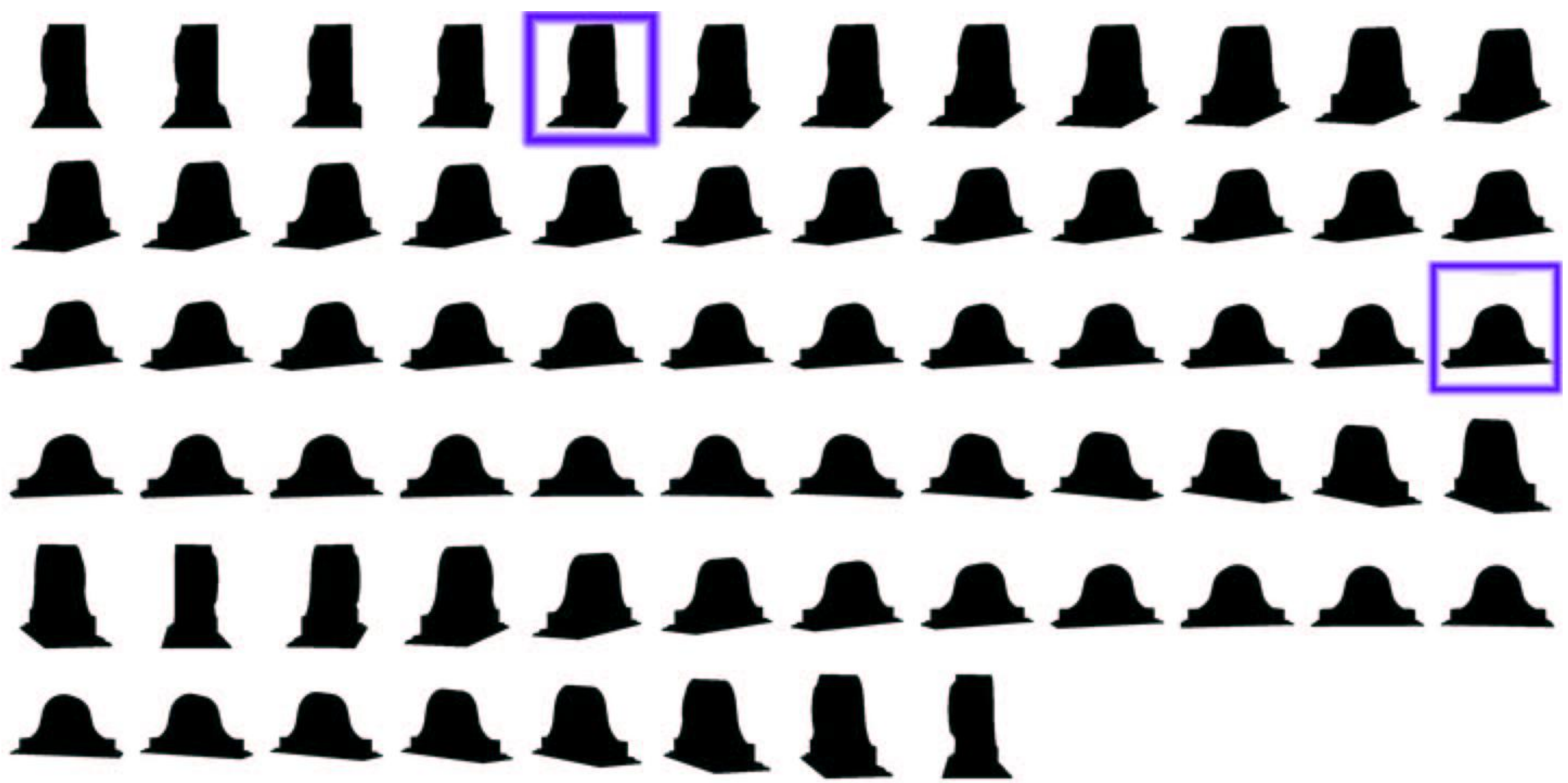
- The need for efficiency forces us to use a minimal set of views for representation; views are redundant to some degree.
- The reduced set is the result of a process such as clustering.
- The representative elements are centroid of clusters.



2D View Selection

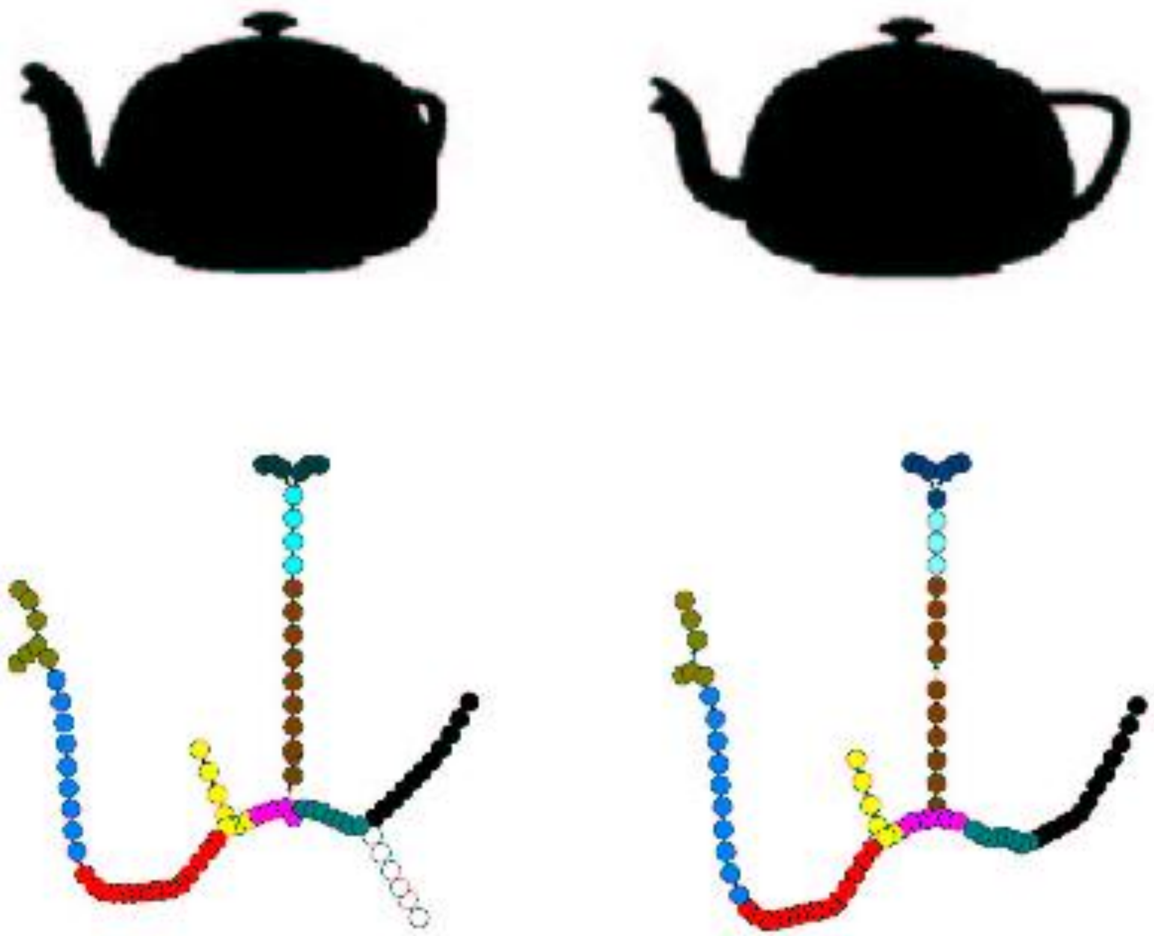
- Is it necessary to use clustering to select highly informative 2D views of a 3D object





Distance Measure

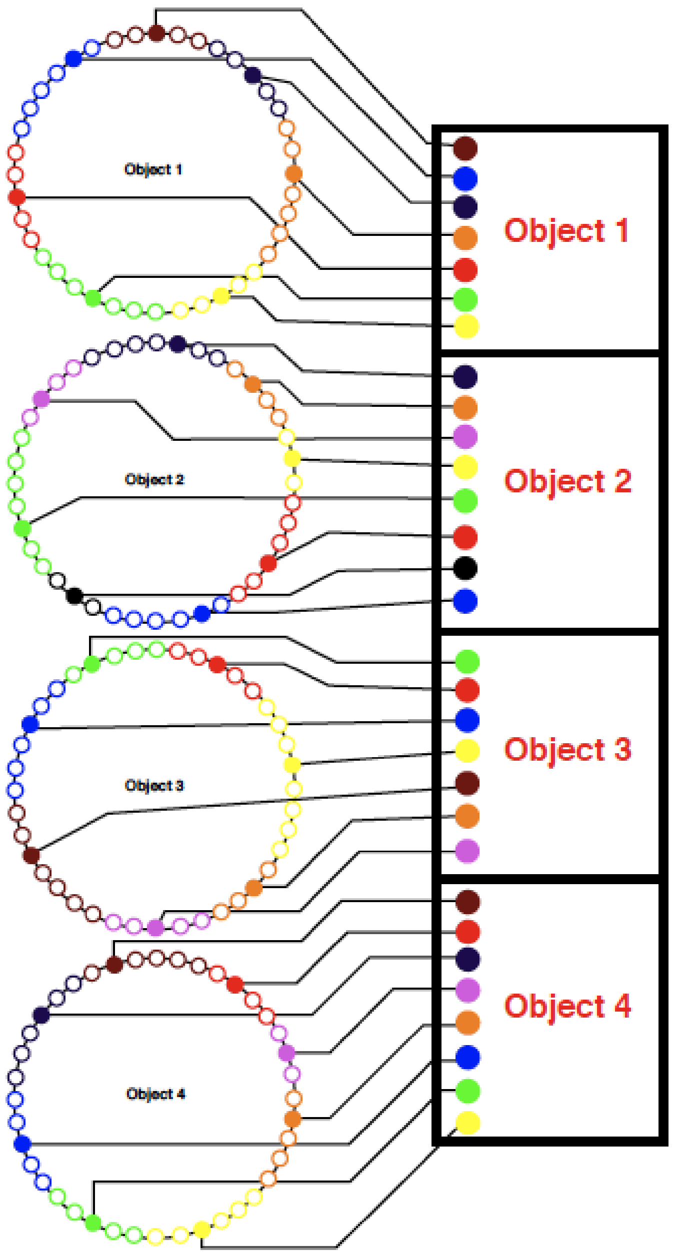
We assume there exists a similarity measure to compare the 2D views.



M. F. Demirci, A. Shokoufandeh, and S. Dickinson. cviu 2011

Discriminating among Multiple Objects

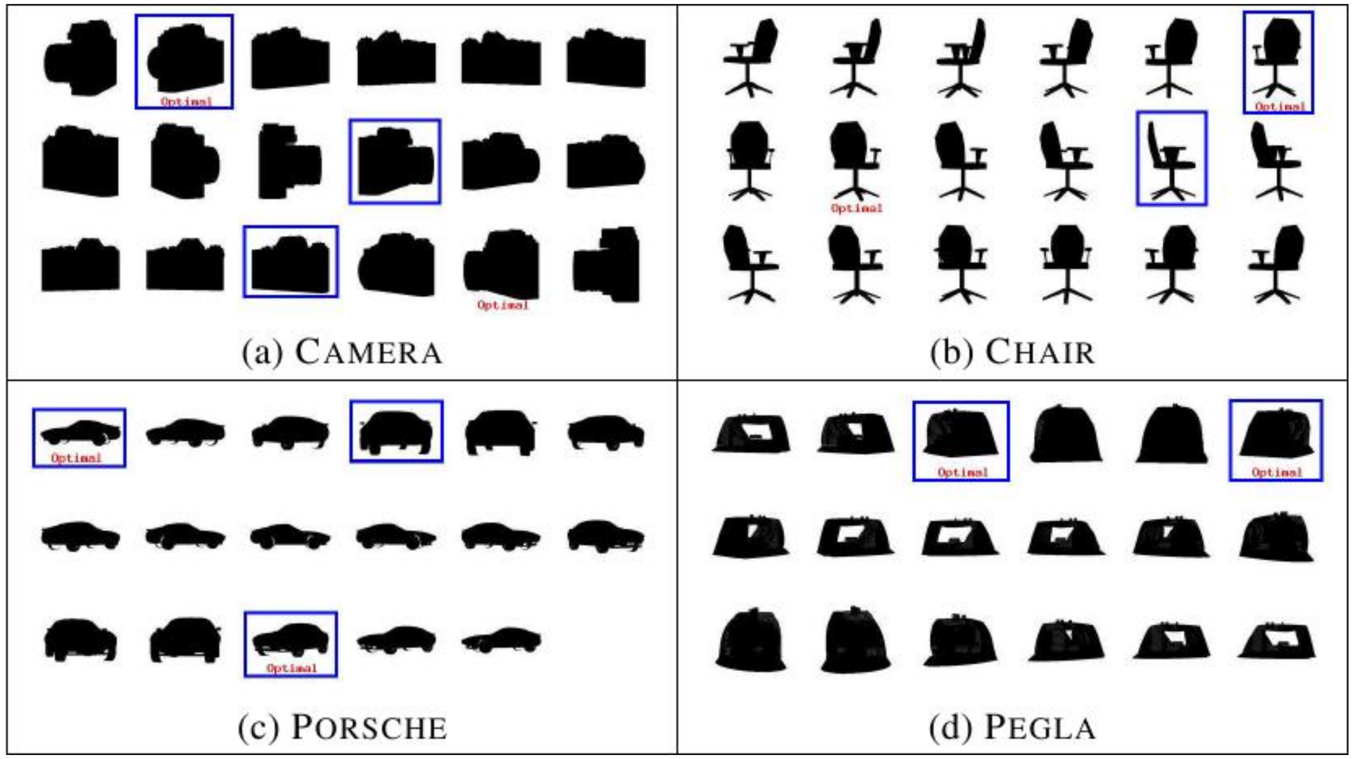
- Selecting views for recognition will become subset selection for class discrimination.



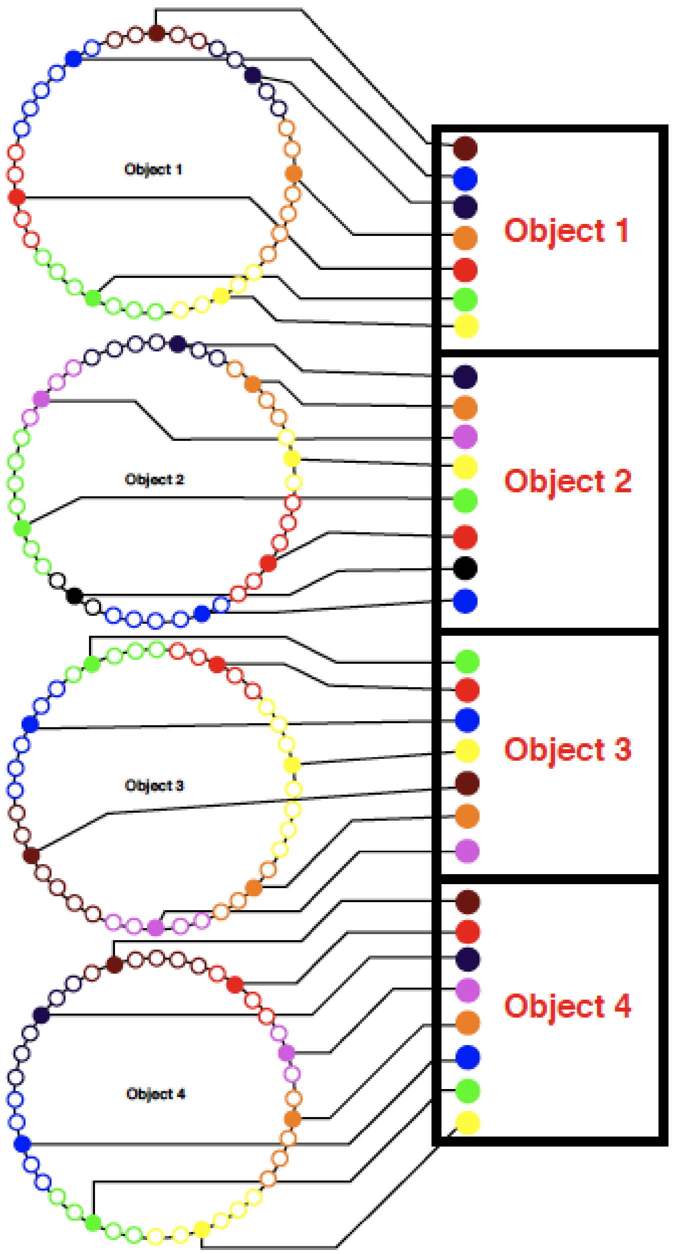
C. M. Cyr and B. B. Kimia, A similarity-based aspect-graph approach to 3D object recognition, IJCV, vol. 57, pp. 5-22, April 2004.

Discriminating among Multiple Objects

- Selecting views for recognition will become subset selection for class discrimination.
- Techniques such as LDA and FDA are more relevant for subset selection, i.e., selecting subsets directly.



T. Denton, Shokoufandeh, CVPR 2005.



C. M. Cyr and B. B. Kimia, A similarity-based aspect-graph approach to 3D object recognition, IJCV, vol. 57, pp. 5-22, April 2004.

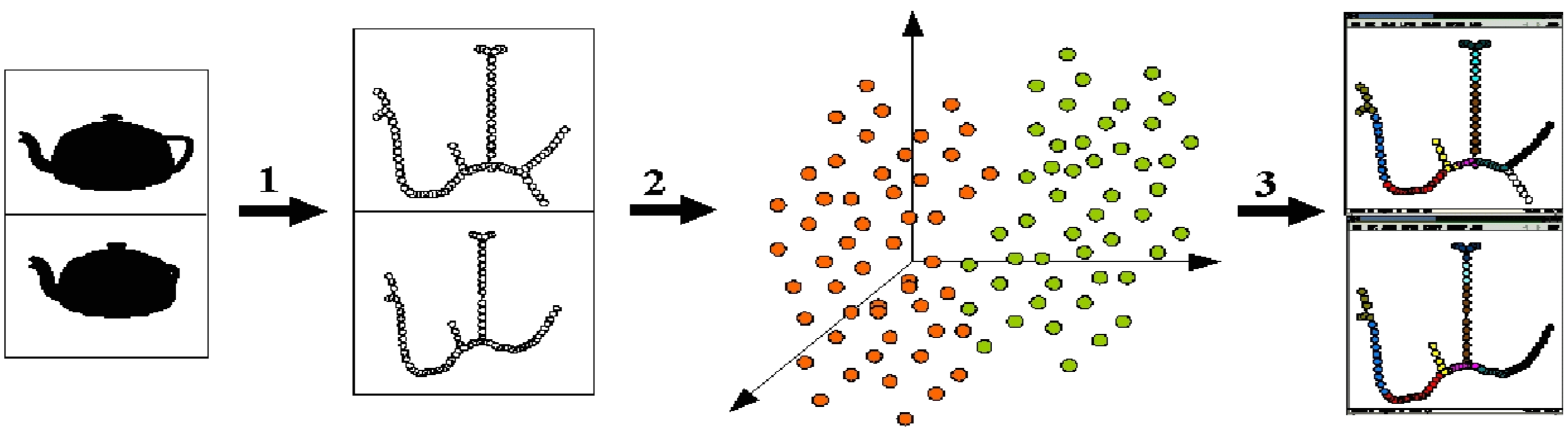
Appearance-based Representation

- Rely on the affinity of the projected intensity image among neighboring views and use some form of PCA on images, to determine the principal direction of variations.
- Only a subset of information will be retained as advocated by the eigenmodels.

M. Tarr and D. Kriegman. "What defines a view?" Vision Research, 2001.

Distance Measure

- We do need a distance measure to compare 2D views:

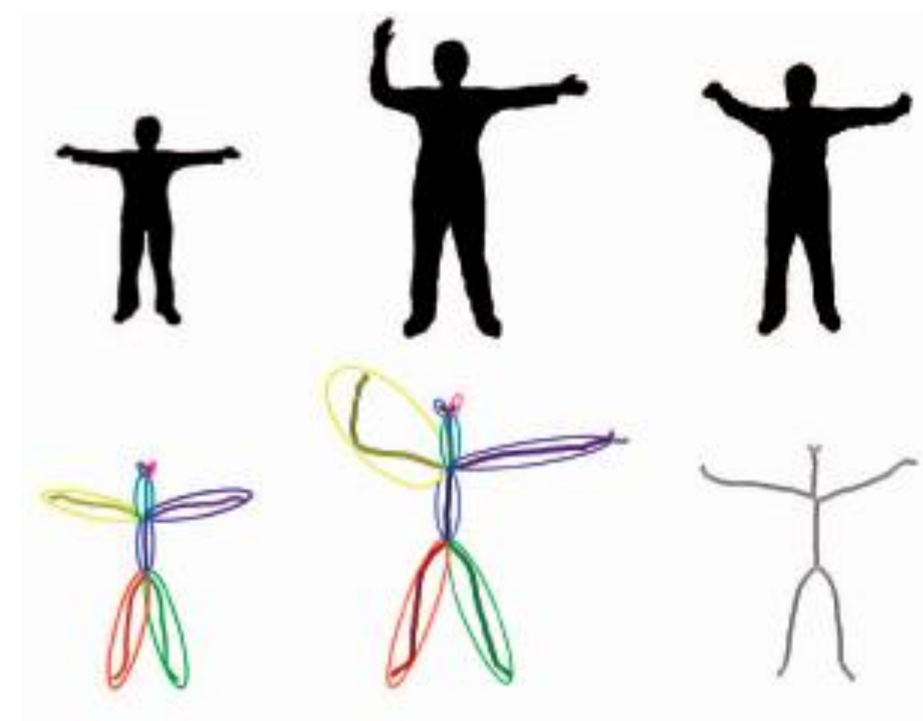


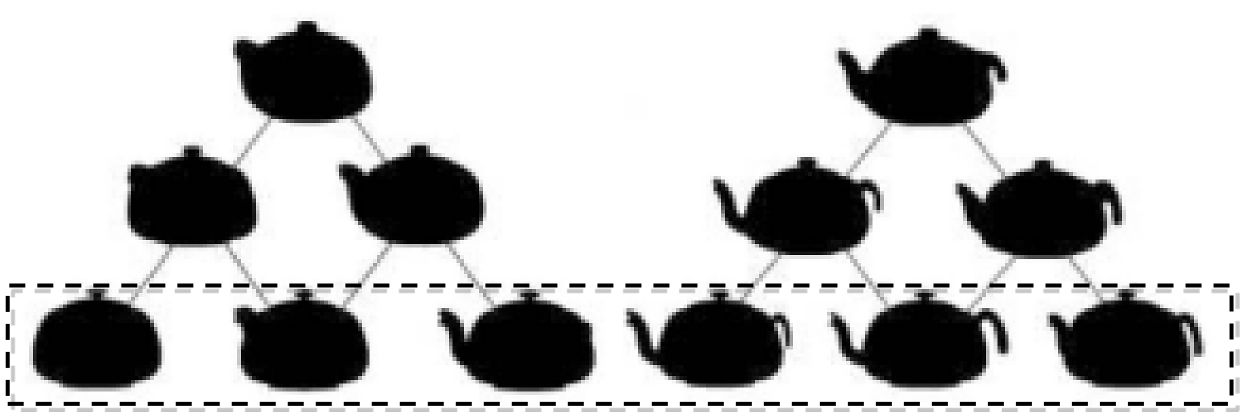
1. Silhouettes are converted to graphs
2. Graphs are embedded into *d*-dimensional Euclidean space
3. Distribution based metric similar to EMD is used to calculate the distance between weighted point sets

M. F. Demirci, A. Shokoufandeh, and S. Dickinson. cviu 2011

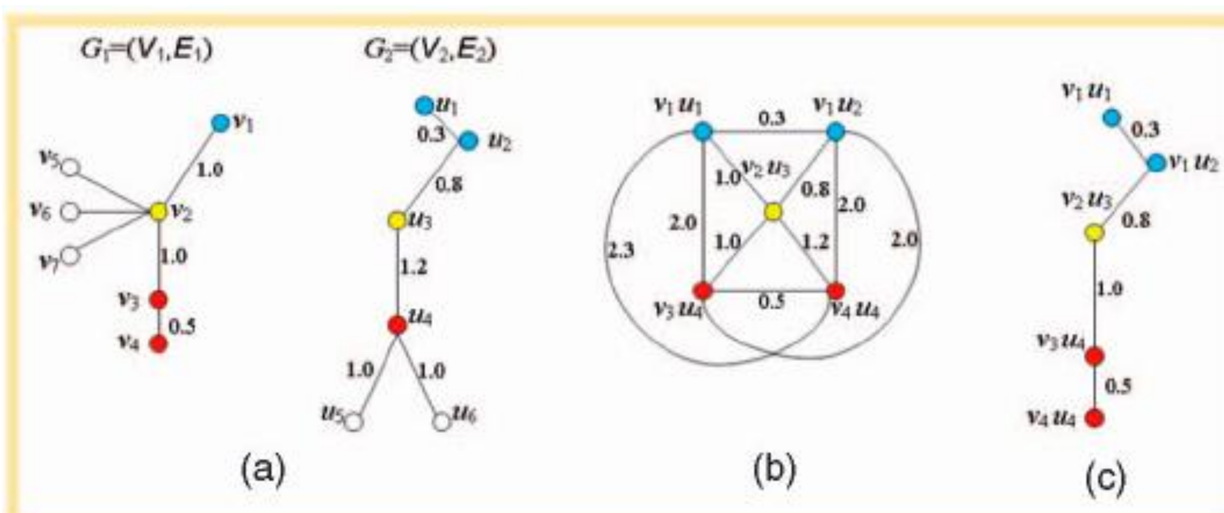
Shape Averaging:

- Generating a new views out of existing ones.
- The pair-wise average shapes in a cluster can be utilized to organize views for efficient



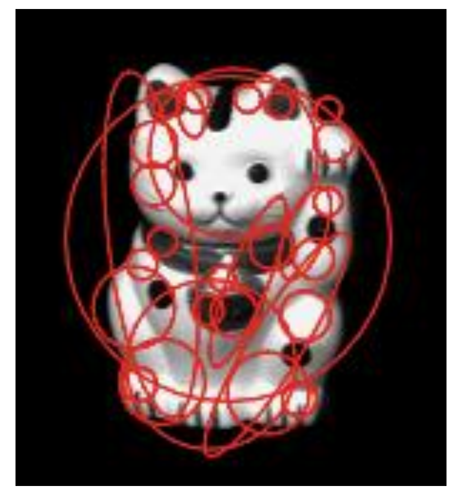


Demirci, Shokoufandeh, Dickinson, Member, Skeletal Shape Abstraction from Examples, IEEE PAMI 2009.

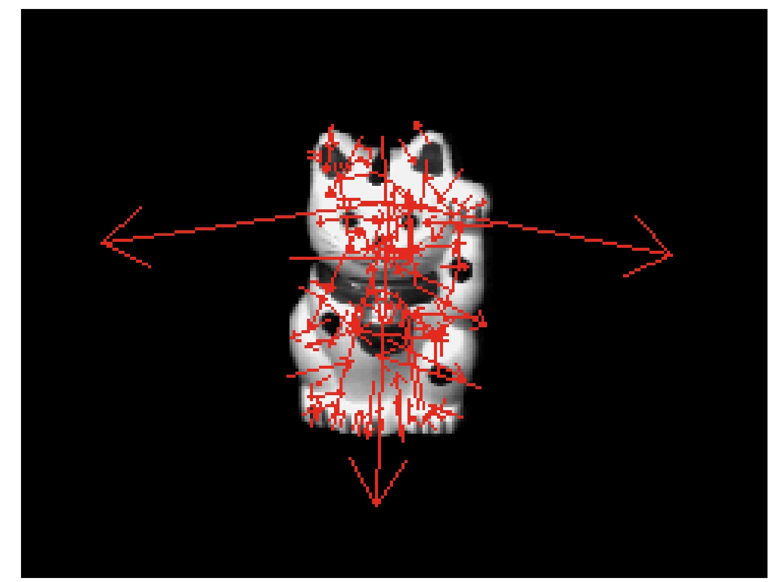


Another Scenario: Feature Selection for Recognition

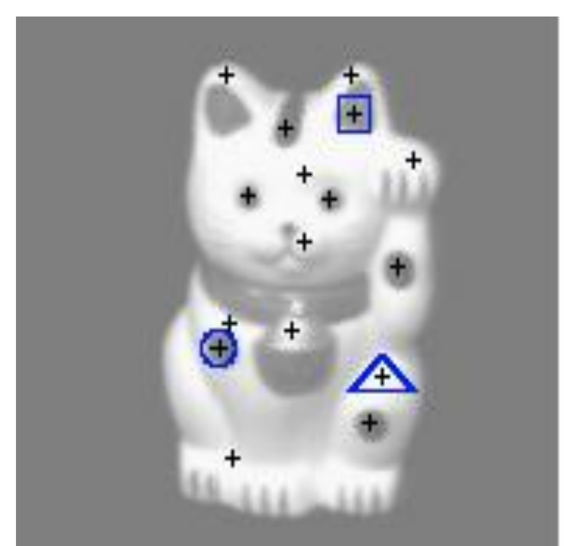
- Given a set of appearance features associated with different views of an object.
- Find a small subset of features, invariant under minor changes in view points that ***best*** characterize the views.

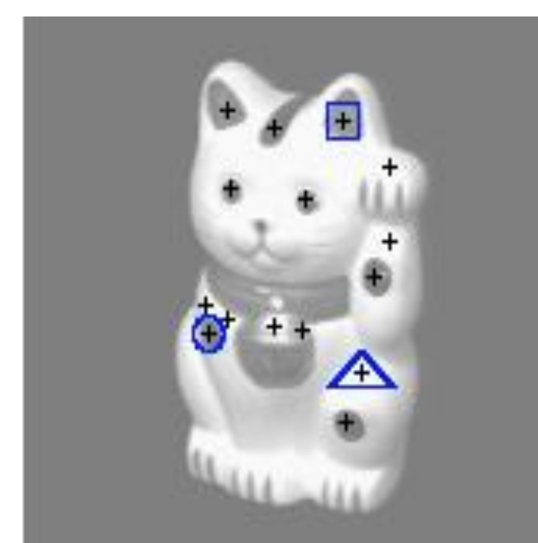


Blobs and ridges detected in scale-space



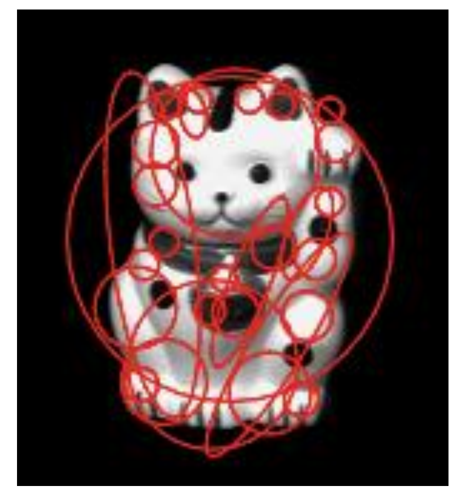
SIFT features (red) showing orientation direction and scale

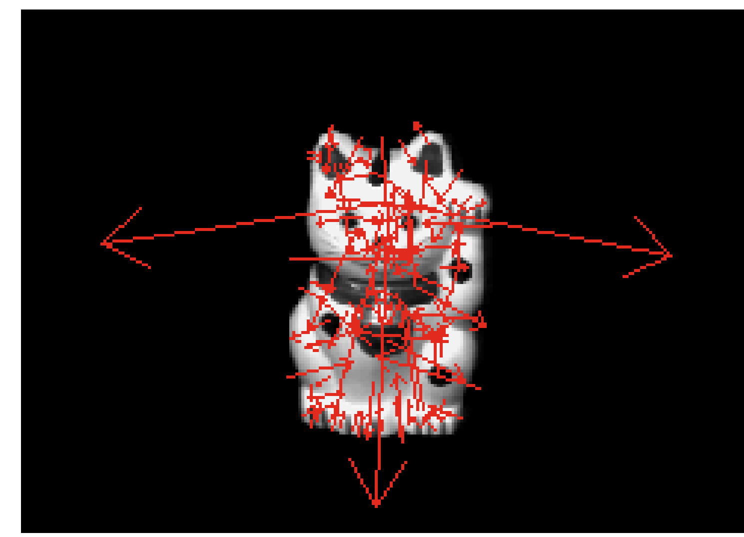




Feature selection for recognition

- Discrimination across objects.





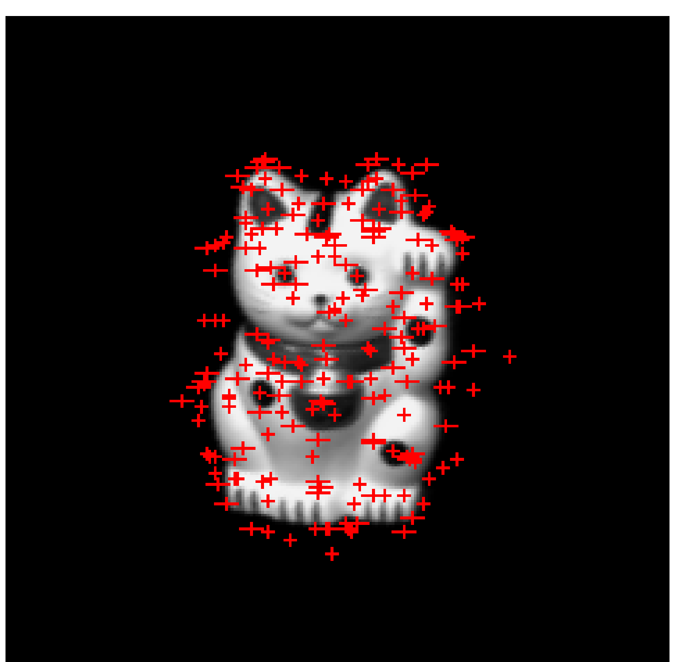
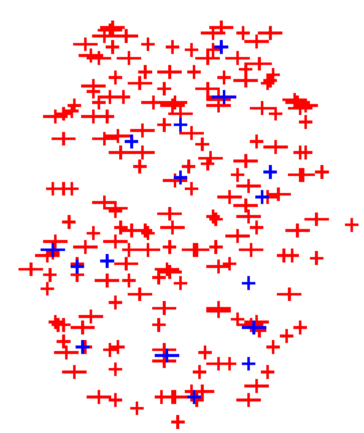
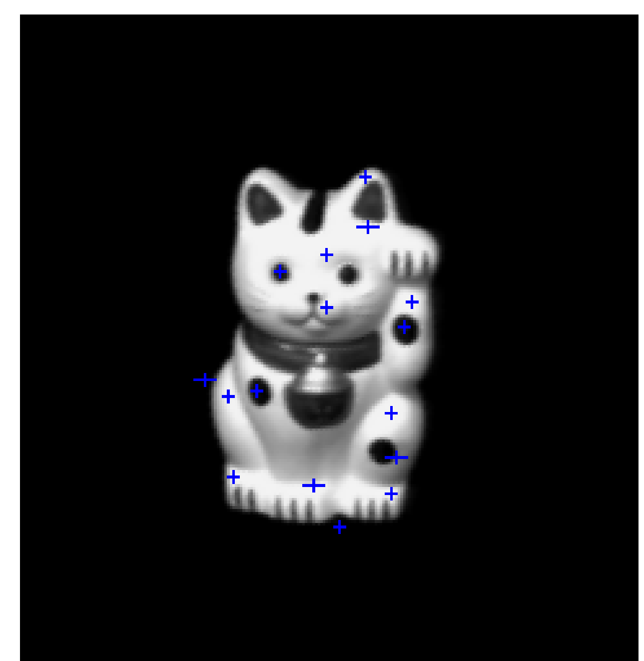
Flexibility in Imposing Constraints

- The feature selection process should allow for imposing constrains:
  - Spatial constraints to deal with occlusion
  - Stability constraints to deal with noise.

Denton, Shokoufandeh, Novatnack, and Nishino. (2008).

Setting up the Optimization Problem

- Given
  - Dataset *P*
  - Pair-wise similarity function (similarity between related pairs of data points can be determined)
- Find a subset *P** that ***best*** represents *P*

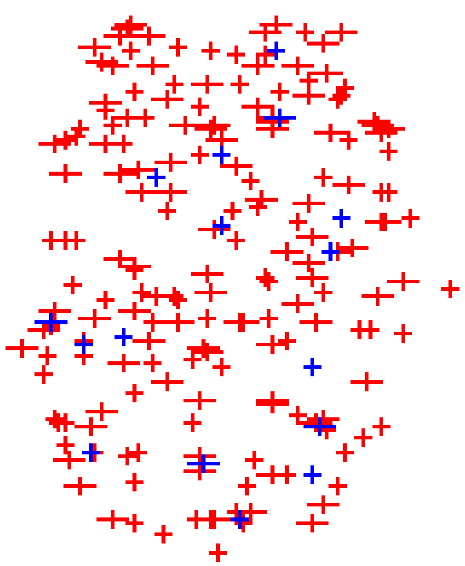
Constraints to Optimize

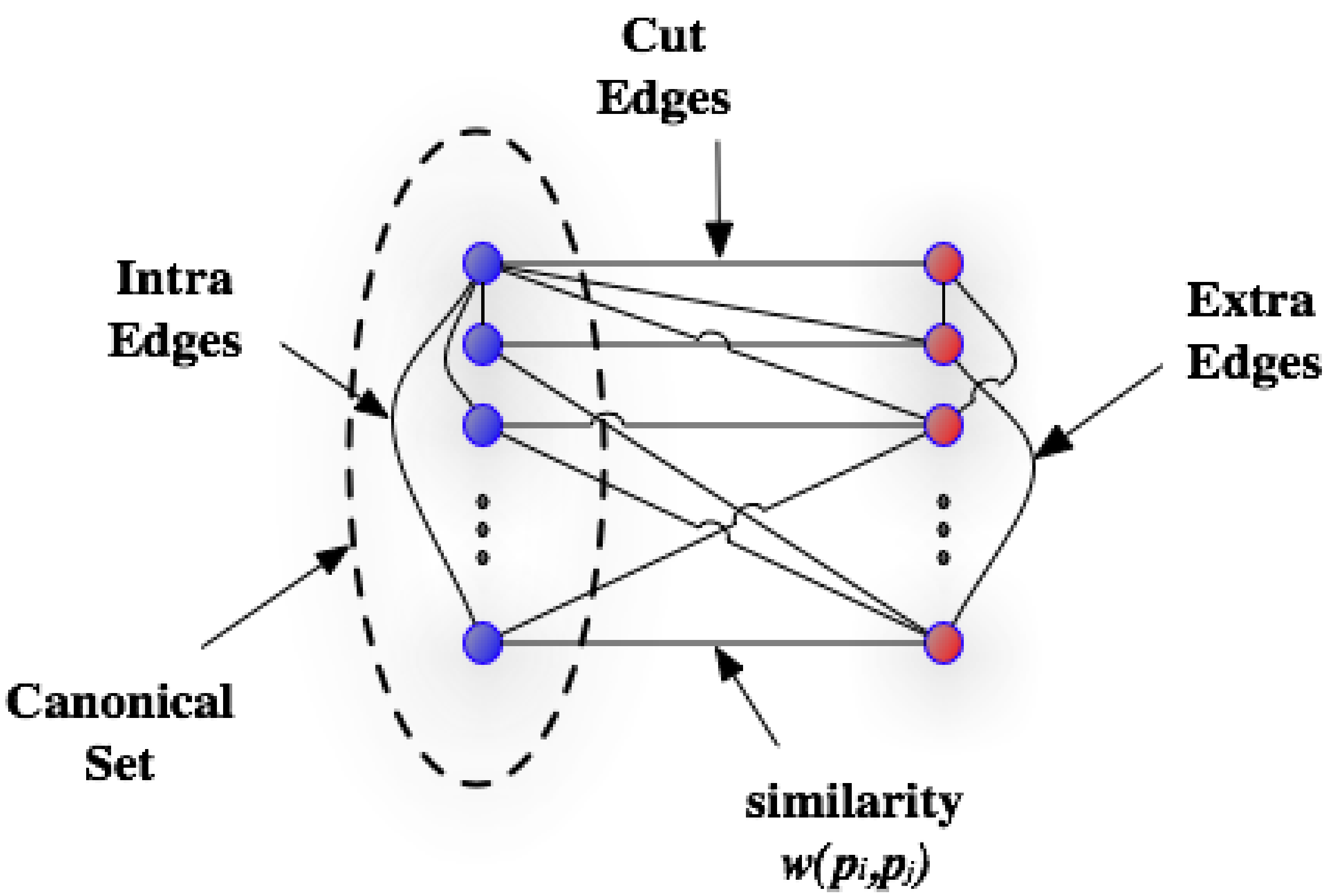
- The combinatorial properties of algorithms for subset selection:
  - Generates a compact form of original data
  - Highly representative subset
  - Less sensitive to outliers
  - No requirement for the number of clusters
  - Easy to incorporate domain knowledge such as stability, spatial distribution, etc.

Discrete representation

Represent the set as graph:

- Vertices: data points
- Edges: similarity of vertices
  - Intra
  - Cut
  - Extra





Indicator Variables and Property Formulation

- For each element p_i, create an indicator variable:

$$y_i = \begin{cases} 1 & \text{if } p_i \in P^* \\ -1 & \text{if } p_i \notin P^* \end{cases}$$

Indicator Variables and Property Formulation

- For each element p_i, create an indicator variable:

$$y_i = \begin{cases} 1 & \text{if } p_i \in P^* \\ -1 & \text{if } p_i \notin P^* \end{cases}$$

- Using the indicator variables we can define properties such as Cut(P*) the sum of the weights of the cut edges as

$$\frac{1}{4} \sum_{i,j} w_{ij}(1 - y_i y_j)$$

Properties

- Size($P*$) : Number of vertices in canonical set
- Cut($P*$): Sum of cut edge weights
- Intra($P*$): Sum of intra edge weights
- Extra($P*$) : Sum of extra edge weights
- Stability($P*$): Sum of vertex weights in canonical set

Property	**Formulation**
Size($\mathcal{P}^*$)	$\frac{1}{2}\sum_{i=1}^{n}(1 + y_i y_{n+1})$
Stability($\mathcal{P}^*$)	$\frac{1}{2}\sum_{i=1}^{n} t_i(1 + y_i y_{n+1})$
Cut($\mathcal{P}^*$)	$\frac{1}{4}\sum_{i,j} \mathcal{W}_{ij}(1 - y_i y_j)$
Intra($\mathcal{P}^*$)	$\frac{1}{4}\sum_{i,j} \mathcal{W}_{ij}(1 + y_i y_{n+1})(1 + y_j y_{n+1})$
Extra($\mathcal{P}^*$)	$\frac{1}{4}\sum_{i,j} \mathcal{W}_{ij}(1 - y_i y_{n+1})(1 - y_j y_{n+1})$

Inverse Properties

- We can also define inverse properties:
  - Such as Cut^{-1}(P*) : Sum of uncut edge weights
  - Minimizing Cut(P*) is the same as maximizing Cut^{-1}(P*)

- Inverse properties designed for switching between minimization and maximization for ms:

Property	Formulation	Description
$\text{Size}^{-1}(\mathcal{P}^*)$	$\frac{1}{2}\sum_{i=1}^{n}(1-y_i y_{n+1})$	$\lvert P \setminus P^* \rvert$
$\text{Stability}^{-1}(\mathcal{P}^*)$	$\frac{1}{2}\sum_{i=1}^{n} t_i(1-y_i y_{n+1})$	Stability of $P \setminus P^*$
$\text{Cut}^{-1}(\mathcal{P}^*)$	$\frac{1}{4}\sum_{i,j}\mathcal{W}_{ij}(1+y_i y_j)$	Sum of uncut edge weights
$\text{Intra}^{-1}(\mathcal{P}^*)$	$\frac{1}{4}\sum_{i,j}\mathcal{W}_{ij}(1-y_i y_j) + \frac{1}{4}\sum_{i,j}\mathcal{W}_{ij}(1-y_i y_{n+1})(1-y_j y_{n+1})$	Sum of non-intra edge weights
$\text{Extra}^{-1}(\mathcal{P}^*)$	$\frac{1}{4}\sum_{i,j}\mathcal{W}_{ij}(1-y_i y_j) + \frac{1}{4}\sum_{i,j}\mathcal{W}_{ij}(1+y_i y_{n+1})(1+y_j y_{n+1})$	Sum of non-extra edge weights

Some Possible strategies for Selecting subsets

- Bounded Canonical Set
- Stable Bounded Canonical Set

Bounded Canonical Set (BCS)

- BCS
  - Elements in canonical set are minimally similar
  - Elements in canonical set are maximally similar to elements not in canonical set
  - Size of canonical set is at least k_{min} and at most k_{max}

$$\begin{aligned} \text{Minimize} \quad & \mathbf{Intra}(\mathcal{P}^*), \\ \text{Maximize} \quad & \mathbf{Cut}(\mathcal{P}^*), \\ \text{Subject to} \quad & \frac{1}{2}\sum_{i=1}^{n}(1+y_i y_{n+1}) - k_{min} \geq 0, \\ & k_{max} - \frac{1}{2}\sum_{i=1}^{n}(1+y_i y_{n+1}) \geq 0, \\ & y_i \in \{-1,+1\}, \ \forall\, 1 \leq i \leq n+1. \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \text{Maximize} \quad & \mathbf{Intra}^{-1}(\mathcal{P}^*), \\ \text{Maximize} \quad & \mathbf{Cut}(\mathcal{P}^*), \\ \text{Subject to} \quad & \frac{1}{2}\sum_{i=1}^{n}(1+y_i y_{n+1}) - k_{min} \geq 0, \\ & k_{max} - \frac{1}{2}\sum_{i=1}^{n}(1+y_i y_{n+1}) \geq 0, \\ & y_i \in \{-1,+1\}, \ \forall\, 1 \leq i \leq n+1. \end{aligned}$$

Stable Bounded Canonical Set (SBCS)

- Stability value associated with each data point
  - Data points in canonical set are minimally similar
  - Data points in canonical set are maximally similar to data points not in canonical set
  - Data points in canonical set are maximally stable
  - Size of canonical set is at least $\boldsymbol{k}_{\mathbf{min}}$ and at most $\boldsymbol{k}_{\mathbf{max}}$

$$\text{Maximize} \quad \frac{1}{4}\sum_{i,j} \mathcal{W}_{ij}(1 - y_i y_j) + \frac{1}{4}\sum_{i,j} \mathcal{W}_{ij}(1 - y_i y_{n+1})(1 - y_j y_{n+1}),$$

$$\text{Maximize} \quad \frac{1}{4}\sum_{i,j} \mathcal{W}_{ij}(1 - y_i y_j),$$

$$\boxed{\text{Maximize} \quad \frac{1}{2}\sum_{i=1}^{n} t_i(1 + y_i y_{n+1}),}$$

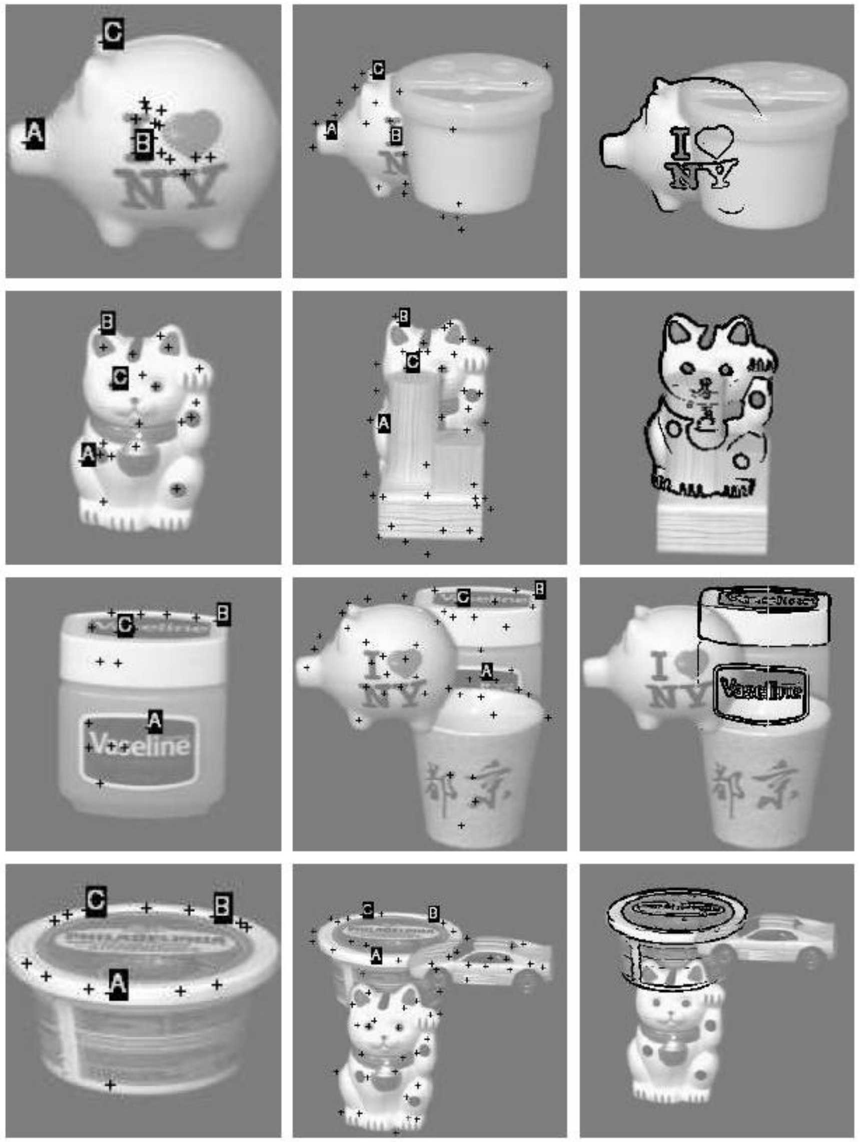
$$\text{Subject to} \quad \frac{1}{2}\sum_{i=1}^{n}(1 + y_i y_{n+1}) - k_{min} \geq 0,$$

$$k_{max} - \frac{1}{2}\sum_{i=1}^{n}(1 + y_i y_{n+1}) \geq 0,$$

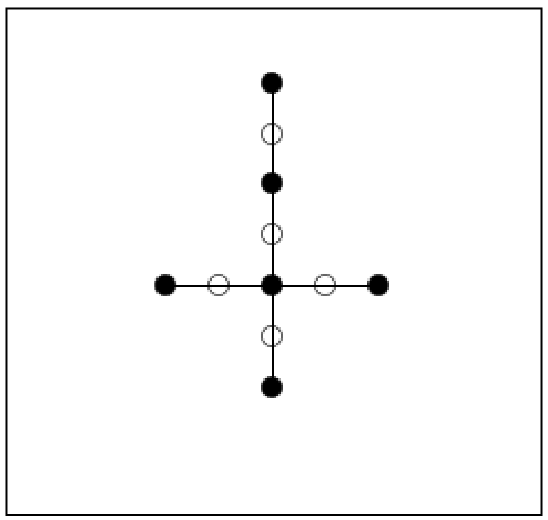
$$y_i \in \{-1, +1\}, \ \forall\, 1 \leq i \leq n+1.$$

Stable Bounded Canonical Set (SBCS)

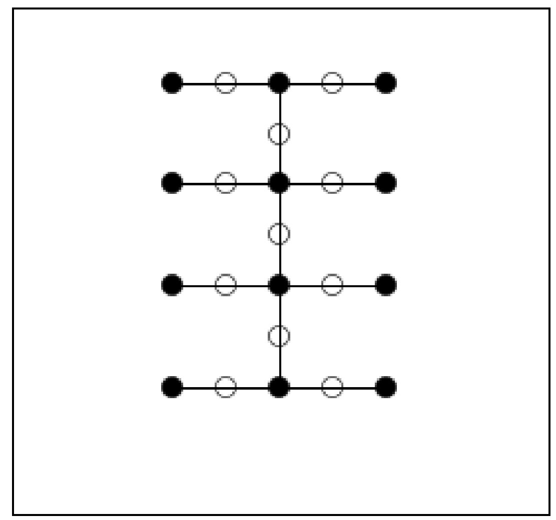
- Stability value associated with each data point
  - Data points in canonical set are minimally similar
  - Data points in canonical set are maximally similar to data points not in canonical set
  - Data points in canonical set are maximally stable
  - Size of canonical set is at least $\boldsymbol{k}_{\mathbf{min}}$ and at most $\boldsymbol{k}_{\mathbf{max}}$



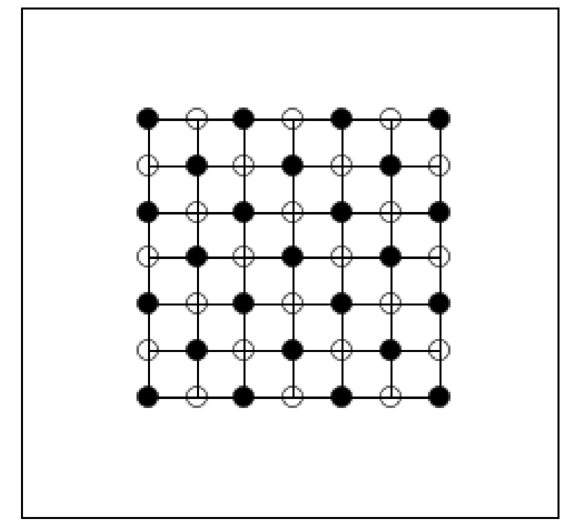
Effect of Distance Measure on BCS



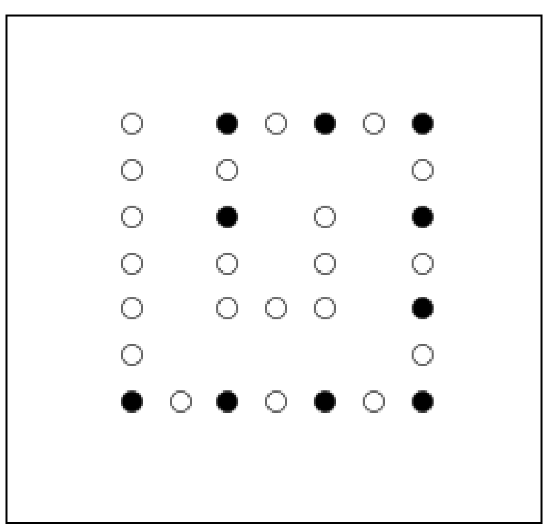
Tree 1



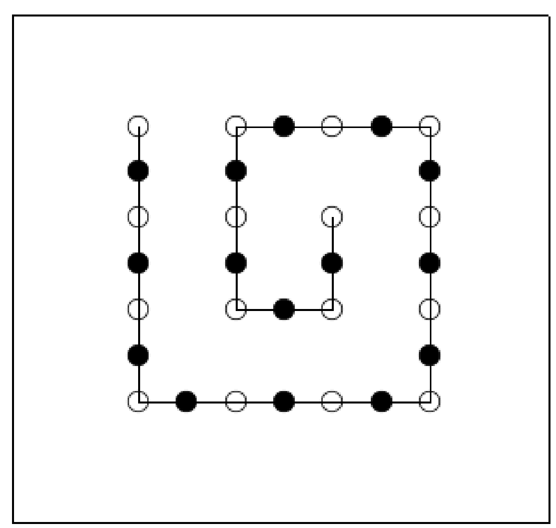
Tree 2



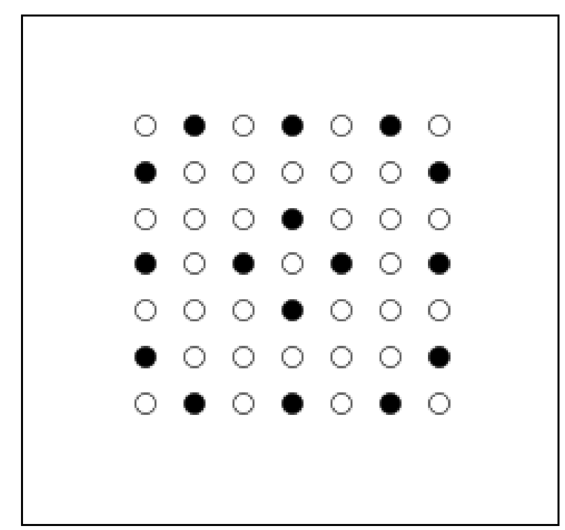
Lattice



Spiral



Spiral Manifold



Fully Connected Grid

Combining Objective Functions

The functions are all convex and may be combined

using Pareto optimality (Essentially a weighted combination):

- Solution is a Pareto optimal point for a given set of weights
- Solutions for different weightings might not be comparable

Pareto weighting parameters

$$
\begin{aligned}
&\text{Maximize} && \lambda_1\left[\mathrm{Intra}^{-1}(P^*)\right] + \lambda_2\left[\mathrm{Cut}(P^*)\right] \\
&\text{Subject to} && \mathrm{Size}(P^*) - k_{\min} \geq 0, \\
& && k_{\max} - \mathrm{Size}(P^*) \geq 0, \\
&\text{Where} && \lambda_1 + \lambda_2 = 1
\end{aligned}
$$

Intractability

- Using a simple Karp reduction it can be shown that the BCS is NP-Hard;
  - Reduction to the Bounded Canonical Set from *dominating set problem*
  - The minimum dominating set problem is known to be NP-Hard

Approximation Methods

- BCS problem is NP-hard!
- Approximate solution can be found using
  - Semidefinite programming (SDP)
    - Primal/Dual Method of SDP.
  - Quadratic programming (QP)

SDP Approximations

- Max-Cut
  - Goemans and Williamson 1994 used SDP to generate approximation to Max-Cut that is at least 0.878 of optimal
- Subgraph matching (Schellewald *et al. 2003*)
- Image segmentation (Keuchel *et al. 2004*)
- Shape from shading (*Zhu et al. 2006*)

Vector Relaxation

- To prepare for SDP we perform a relaxation where we replace each integer variable $\mathbf{y}_i$ with a unit length vector $x_i \in \Re^{n+1}$

Vector Relaxation

- To prepare for SDP we perform a relaxation where we replace each integer variable $\mathbf{y}_{\mathrm{i}}$ with a unit length vector $\boldsymbol{x_i} \in \Re^{n+1}$
- Define the matrix $\boldsymbol{V}$ to be the matrix obtained from concatenating the column vectors $\boldsymbol{x_i}$

Vector Relaxation

- To prepare for SDP we perform a relaxation where we replace each integer variable $\mathbf{y}_\mathrm{i}$ with a unit length vector $\boldsymbol{x_i} \in \Re^{n+1}$
- Define the matrix $\boldsymbol{V}$ to be the matrix obtained from concatenating the column vectors $\boldsymbol{x_i}$
- Define the matrix $\boldsymbol{X}$ to be $\boldsymbol{V^T V}$

Vector Relaxation

- To prepare for SDP we perform a relaxation where we replace each integer variable $\boldsymbol{y_i}$ with a unit length vector $\boldsymbol{x}_i \in \Re^{n+1}$
- Define the matrix $\boldsymbol{V}$ to be the matrix obtained from concatenating the column vectors $\boldsymbol{x_i}$
- Define the matrix $\boldsymbol{X}$ to be $\boldsymbol{V^T V}$
- $\boldsymbol{X}$ is semidefinite

Semidefinite Programming (SDP)

- Optimization of a matrix inner product form

$$\begin{aligned} &\text{Maximize} \quad C \cdot X \\ &\text{Subject to} \quad A_j \cdot X \le b_j, \quad \forall\ j \in \{1, \dots, m\}, \\ &\qquad\qquad\quad X \text{ is positive semidefinite} \end{aligned}$$

Fast Implementation:

- Instead of using an SDP solver we can establish a primal dual SDP pair:

$$\begin{aligned} \max \ \mathbf{C} \bullet \mathbf{X} & \\ \forall j \in [m]: \ \mathbf{A}_j \bullet \mathbf{X} &\leq b_j \\ \mathbf{X} &\succeq 0 \end{aligned} \qquad\qquad \begin{aligned} \min \ \mathbf{b} \cdot \mathbf{y} & \\ \textstyle\sum_{j=1}^{m} \mathbf{A}_j y_j &\succeq \mathbf{C} \\ \mathbf{y} &\geq 0 \end{aligned}$$

  - starts with a trivial candidate for a primal solution (possibly infeasible), viz. $X^{(1)} = (1/\boldsymbol{n})\, I$.
  - iteratively generates primal solutions $X^{(2)}, X^{(3)}, \ldots$
  - $X^{(\boldsymbol{t}+1)}$ is obtain $X^{(\boldsymbol{t})}$, through auxiliary Oracle.

Primal Dual Oracle

- The Oracle tries to certify the validity of the current $X^{(t)}$.
- Oracle searches for a vector $\boldsymbol{y}$ from the polytope $\boldsymbol{D}_{\boldsymbol{\alpha}}=\{\boldsymbol{y}:\boldsymbol{y} \geq 0, \boldsymbol{b}\cdot\boldsymbol{y} \leq \boldsymbol{\alpha}\}$ such that

$$\sum_{j=1}^{m}(\mathbf{A}_j \bullet \mathbf{X}^{(t)})y_j - (\mathbf{C} \bullet \mathbf{X}^{(t)}) \;\geq\; 0.$$

- if Oracle succeeds in finding such a $\boldsymbol{y}$ then we claim $X^{(t)}$ is either primal infeasible or has value $\boldsymbol{C}\bullet X^{(t)} \leq \alpha$.
- if there is no vector $\boldsymbol{y}$ in $\boldsymbol{D}$, then it can be seen that $X^{(t)}$ must be a primal feasible solution of objective value.

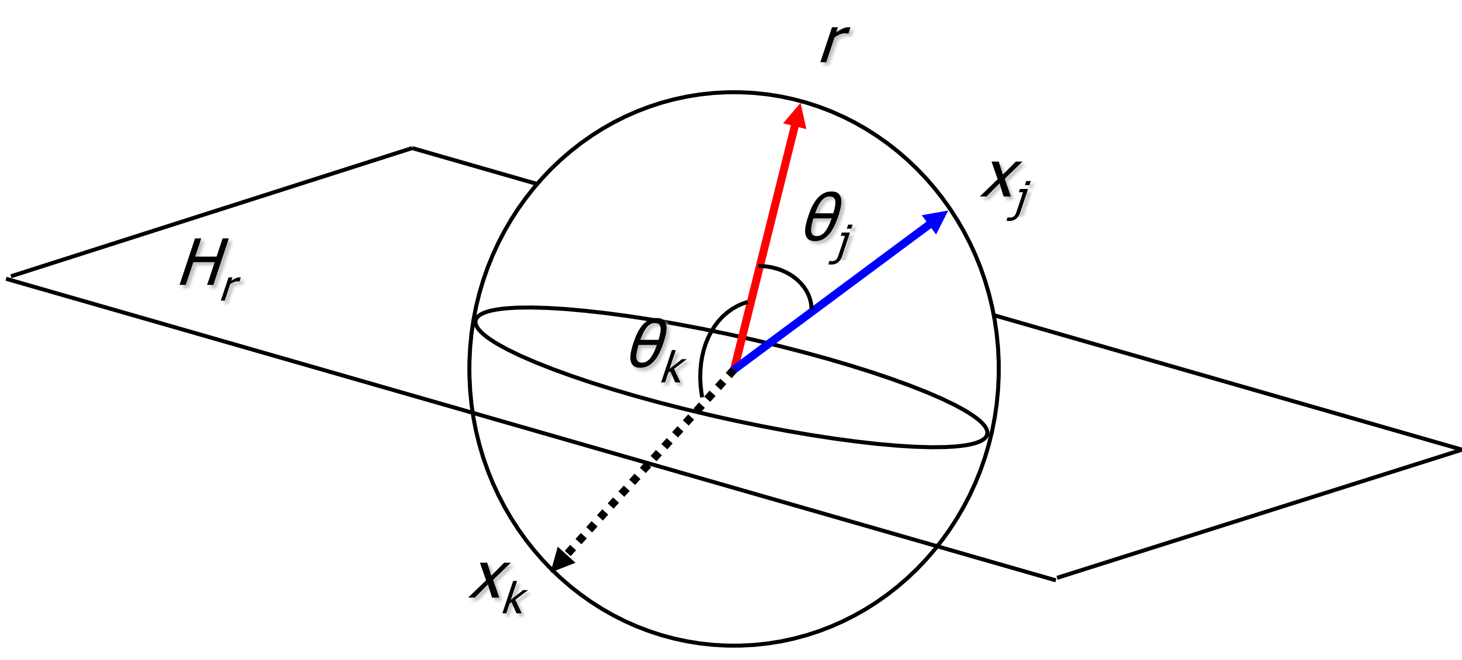
SDP Solution

- The output of the SDP solver is the matrix $\boldsymbol{X}$.
- We then use a Cholesky decomposition to obtain the matrix $\boldsymbol{V}$ ($\boldsymbol{X} = \boldsymbol{V}^{\mathbf{T}}\boldsymbol{V}$)
- The columns of matrix $\boldsymbol{V}$ are our unit length vectors $\boldsymbol{x}_i \in \Re^{n+1}$

Approximation

- Define a binary indicator vector $z \in \Re^{n+1}$
- We use a rounding technique to approximate z_i from the i^{th} column of matrix V.
- Pick a random vector r to be uniformly distributed on the unit sphere (V. Vazirani)
- For each column v_i of V

$$z_i = \begin{cases} 1 & \text{if } v_i \cdot r \geq 0 \\ -1 & \text{otherwis} \end{cases}$$



Finally

- If $z_i = z_{n+1}$ then vertex i is in the canonical set
- Thus the indicator vector z tells which of the vertices of P is in the canonical set: that is, the subset P' that best represents P
- This process can be de-randomized using the expectation maximization

Quadratic Programming (QP)

- Quadratic objective
- Linear constraints

$$
\begin{aligned}
&\text{Maximize} && \lambda_1\left[\mathrm{Intra}^{-1}(P^*)\right] + \lambda_2\left[\mathrm{Cut}(P^*)\right] \\
&\text{Subject to} && \mathrm{Size}(P^*) - k_{\min} \geq 0, \\
& && k_{\max} - \mathrm{Size}(P^*) \geq 0, \\
&\text{Where} && \lambda_1 + \lambda_2 = 1
\end{aligned}
$$

Quadratic Programming (QP)

- Quadratic objective
- Linear constraints

$$\text{Maximize} \quad \tfrac{1}{2} x^T H x + f^T x$$

$$\text{Subject to} \quad Ax \text{ £ } b$$

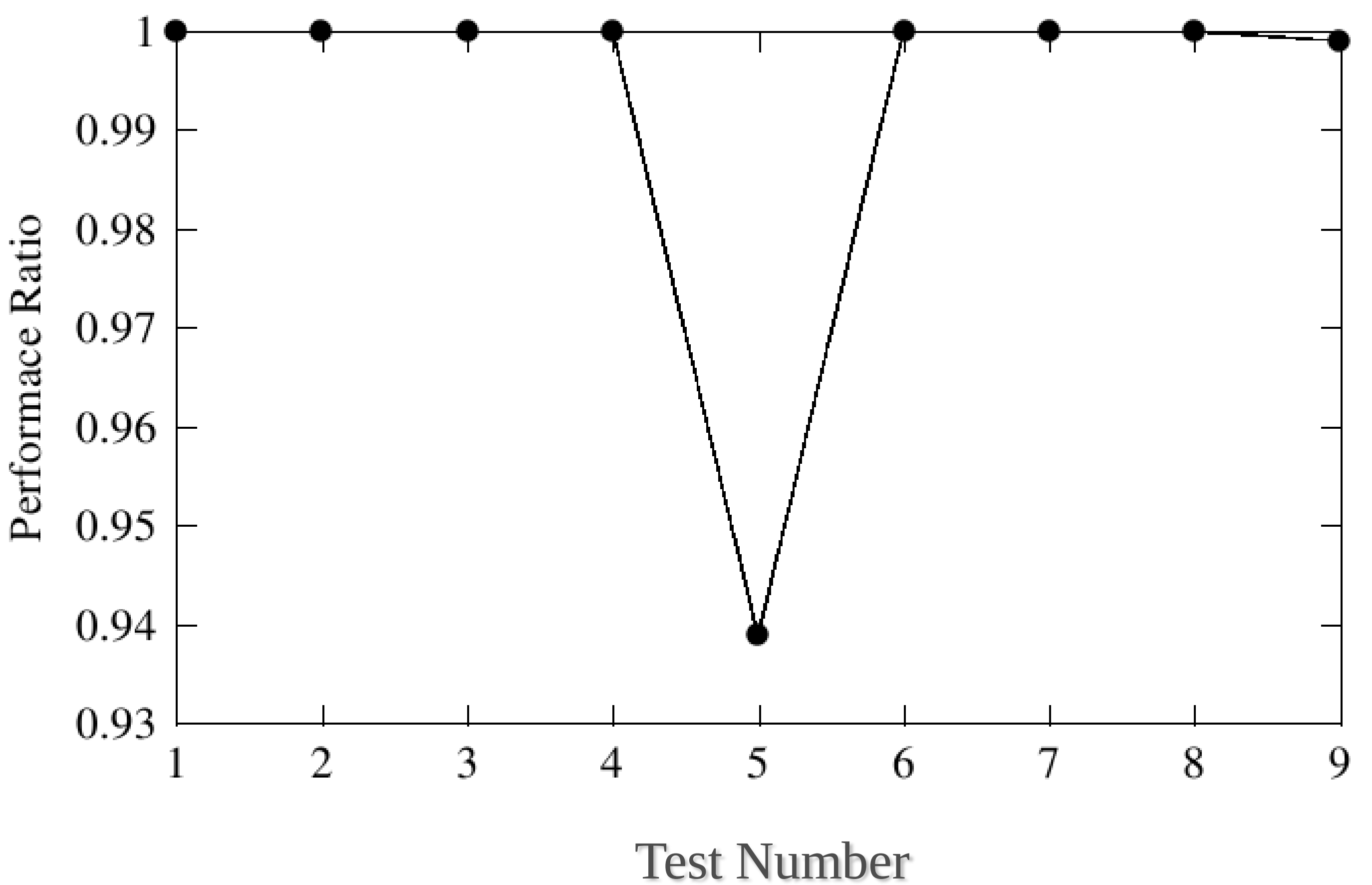
$$l \text{ £ } x \text{ £ } u$$

Bounds

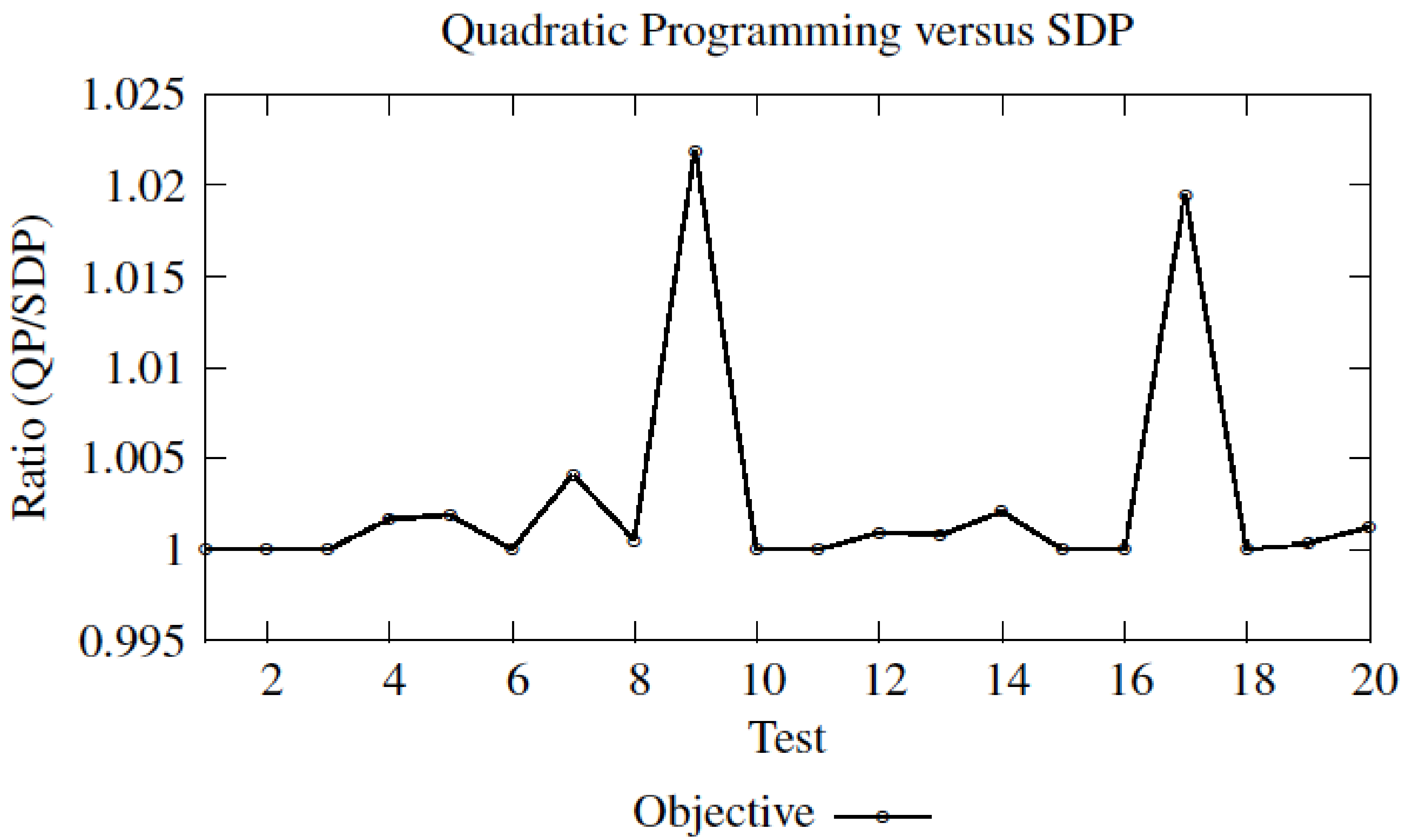
- It has been shown that (for any particular set of $\lambda_1$and λ_2) the expected value of the approximate solution is at least $0.878*OPT_{BCS}$.
- The guarantee only applies to the objective value.

Objective Performance

- Exhaustive search compared to approximation
- Experimental results confirm the quality of the approximation

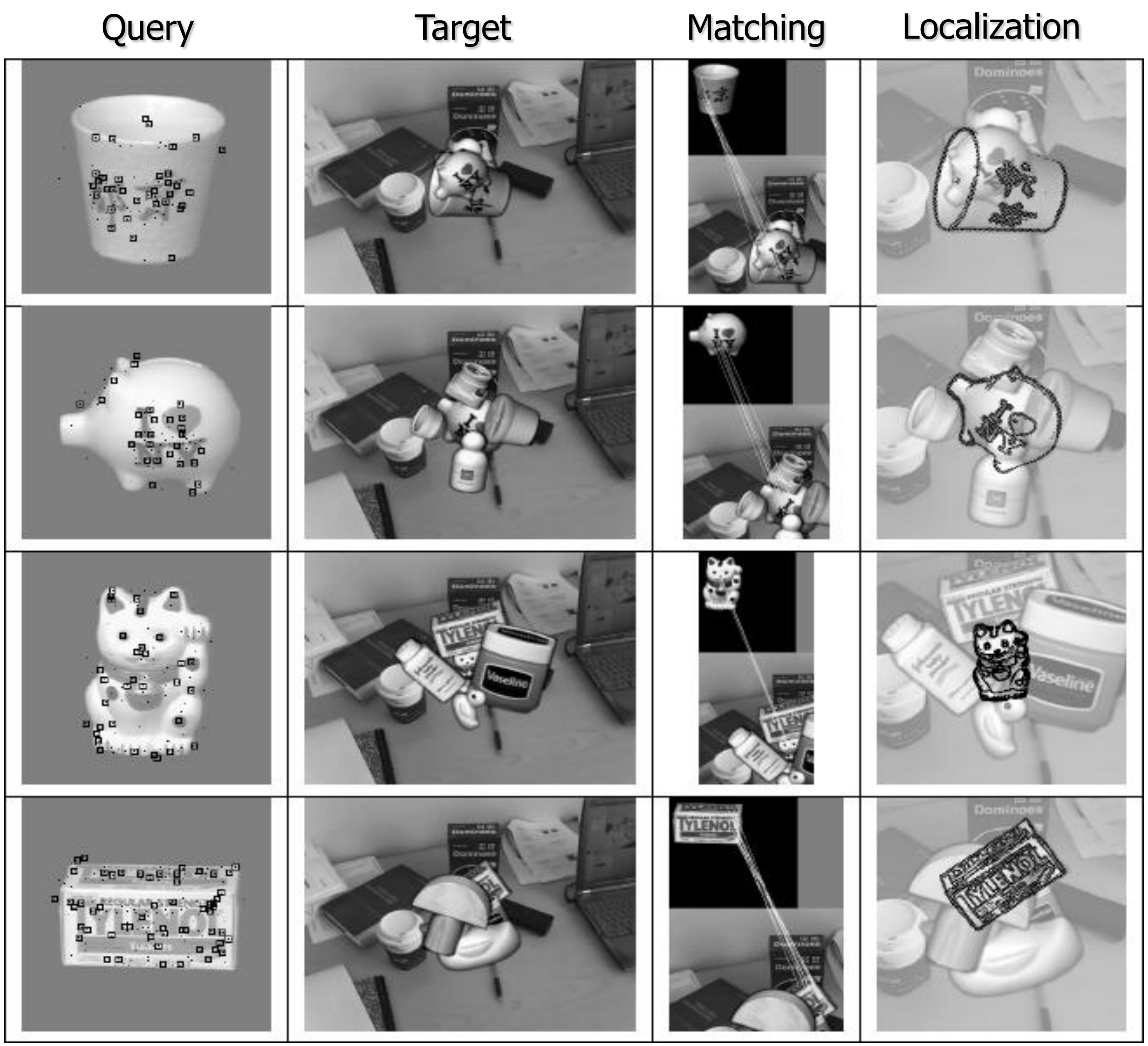


Comparing SDP vs. QP on Localization Task



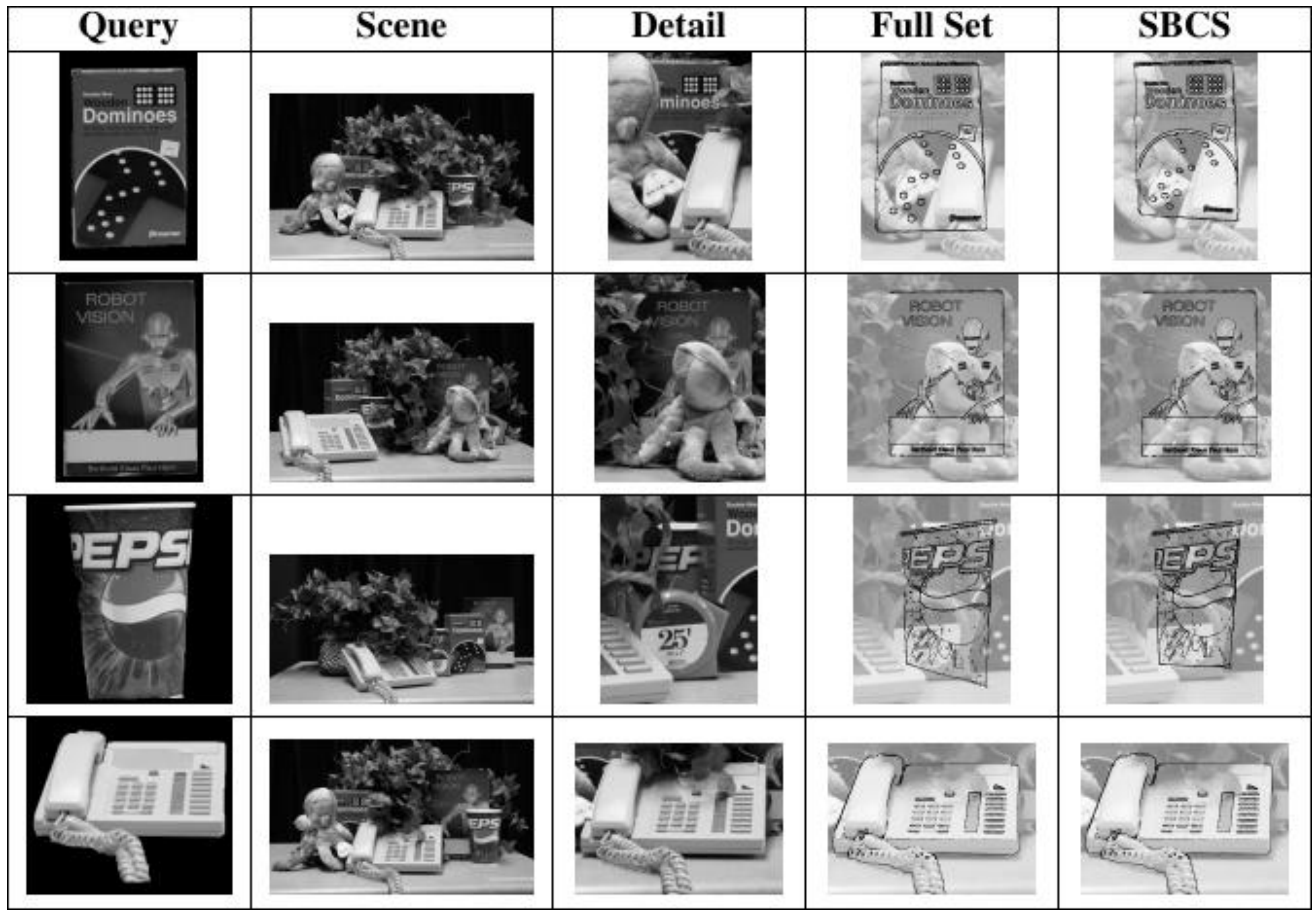
- The average ratio of objective values, QP/SDP was 1.003, and the ratio of execution times was 0.17.
- For technical reasons both algorithms were run as minimizations, thus lower values indicate better performance.

Localization Results



Denton, Shokoufandeh, Novatnack, and Nishino. CVIU (2008).

Object Localization in Occluded Scenes



Conclusion:

- **Bad news:**
  - Almost every nontrivial feature selection problem is computationally intractable.
- **Good news:**
  - Good approximation algorithms (objective performance guarantees of 0.878 of optimal) exist.
  - *Provided that constraints are in first order logic statement* ($\wedge, \vee, \neg$, *etc.*)
  - Objective Function can be complex
  - Size of approximate solutions (subsets) can be derived from optimization
  - Fast QP and primal-dual approximations exist

Acknowledgements

Collaborators

Jeff Abrahamson

Trip Denton

Lars Bretzner

M. Fatih Demirci

Sven Dickinson

Luc Florack

Frans Kanters

Ko Nishino

John Novatnack

Kaleem Siddiqi

Thank You.

Property	Formulation	Description
Size($\mathcal{P}^*$)	$\begin{bmatrix} \tilde{0} & \frac{1}{4}\mathbf{e} \\ \frac{1}{4}\mathbf{e}^T & \frac{1}{2}n \end{bmatrix}$	Cardinality ($\lvert\mathcal{P}^*\rvert$) of canonical set
Stability($\mathcal{P}^*$)	$\begin{bmatrix} \tilde{0} & \frac{1}{4}\mathbf{t} \\ \frac{1}{4}\mathbf{t}^T & \frac{1}{2}t_\Sigma \end{bmatrix}$	Stability of canonical set
Cut($\mathcal{P}^*$)	$\begin{bmatrix} -\frac{1}{4}\mathcal{W} & \hat{0} \\ \hat{0}^T & \frac{1}{4}w_\Sigma \end{bmatrix}$	Sum of cut edge weights
Intra($\mathcal{P}^*$)	$\begin{bmatrix} \frac{1}{4}\mathcal{W} & \frac{1}{4}\mathbf{d} \\ \frac{1}{4}\mathbf{d}^T & \frac{1}{4}w_\Sigma \end{bmatrix}$	Sum of intra edge weights
Extra($\mathcal{P}^*$)	$\begin{bmatrix} \frac{1}{4}\mathcal{W} & -\frac{1}{4}\mathbf{d} \\ -\frac{1}{4}\mathbf{d}^T & \frac{1}{4}w_\Sigma \end{bmatrix}$	Sum of extra edge weights

Table 3.1: Properties of canonical sets expressed in matrix form, where $\tilde{0}$ is an $n \times n$ matrix of zeros, $\mathbf{e}$ is the all-ones vector of order n, $\mathbf{t}$ is a column vector in $\mathbb{R}^n$ whose i^{th} entry is t_i, $t_\Sigma = \sum_{i=1}^{n} t_i$, $\mathcal{W}$ is the $n \times n$ similarity matrix, $w_\Sigma = \sum_{i,j} \mathcal{W}_{ij}$, $\hat{0}$ is an all zeros column vector in $\mathbb{R}^n$, and $\mathbf{d}$ is a column vector in $\mathbb{R}^n$ whose i^{th} entry has value $d_i = \sum_{j=1}^{n} \mathcal{W}_{ij}$.

$$\mathcal{D}_i = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & \dots & \mathcal{A}_{i,1} \\ 0 & 0 & \dots & 0 & 0 & \dots & \mathcal{A}_{i,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 & 0 & \dots & \mathcal{A}_{i,i+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & \mathcal{A}_{i,n} \\ \mathcal{A}_{i,1} & \mathcal{A}_{i,2} & \dots & 1 & \mathcal{A}_{i,i+1} & \dots & 2(d_i - 1) \end{bmatrix}.$$

$$\mathcal{D}_1 = \begin{bmatrix} 1 & \hat{0} \\ \hat{0}^T & \tilde{0} \end{bmatrix}, \mathcal{D}_{n+1} = \begin{bmatrix} \tilde{0} & \hat{0} \\ \hat{0}^T & 1 \end{bmatrix},$$

$$\mathcal{D} = \begin{bmatrix} \tilde{0} & \frac{1}{4}\mathbf{e} \\ \frac{1}{4}\mathbf{e}^T & k_{max} - \frac{1}{2}n \end{bmatrix}.$$

Exact Formulation

(BCS):

$$\begin{aligned} \text{Maximize} \quad & \mathcal{C} \bullet \mathcal{X} \\ \text{Subject to} \quad & \mathcal{D}_i \bullet \mathcal{X} \geq 0, \ \forall\, i = 1 \ldots, m, \\ & \mathcal{X} \succeq 0, \end{aligned}$$

where

$$\mathcal{C} = \lambda_1 \begin{bmatrix} \tilde{0} & -\frac{1}{4}\mathbf{d} \\ -\frac{1}{4}\mathbf{d}^T & \frac{1}{2}w_\Sigma \end{bmatrix} + \lambda_2 \begin{bmatrix} -\frac{1}{4}\mathcal{W} & \hat{0} \\ \hat{0}^T & \frac{1}{4}w_\Sigma \end{bmatrix},$$

Property	Formulation
Size($\mathcal{P}^*$)	$\frac{n}{2} + \frac{1}{2}\sum_{i=1}^{n} y_i$
Stability($\mathcal{P}^*$)	$\frac{1}{2}\sum_{i=1}^{n} t_i + \frac{1}{2}\sum_{i=1}^{n} t_i y_i$
Cut($\mathcal{P}^*$)	$\frac{1}{4}\sum_{i,j} \mathcal{W}_{ij} - \frac{1}{4}\sum_{i,j} \mathcal{W}_{ij} y_i y_j$
Intra($\mathcal{P}^*$)	$\frac{1}{4}\sum_{i,j} \mathcal{W}_{ij} + \frac{1}{2}\sum_{i=1}^{n} y_i \sum_{j=1}^{n} \mathcal{W}_{ij} + \frac{1}{4}\sum_{i,j} \mathcal{W}_{ij} y_i y_j$
Extra($\mathcal{P}^*$)	$\frac{1}{4}\sum_{i,j} \mathcal{W}_{ij} - \frac{1}{2}\sum_{i=1}^{n} y_i \sum_{j=1}^{n} \mathcal{W}_{ij} + \frac{1}{4}\sum_{i,j} \mathcal{W}_{ij} y_i y_j$

Table 3.5: Properties of canonical sets (QP) combined

Property	H	f	Description
Size($\mathcal{P}^*$)	$[\tilde{0}]$	$\frac{1}{2}\mathrm{e}$	Cardinality ($\lvert\mathcal{P}^*\rvert$) of canonical set
Stability($\mathcal{P}^*$)	$[\tilde{0}]$	$\frac{1}{2}\mathrm{t}$	Stability of canonical set
Cut($\mathcal{P}^*$)	$[-\frac{1}{2}\mathcal{W}]$	$\hat{0}$	Sum of cut edge weights
Intra($\mathcal{P}^*$)	$[\frac{1}{2}\mathcal{W}]$	$\frac{1}{2}\mathrm{d}$	Sum of intra edge weights
Extra($\mathcal{P}^*$)	$[\frac{1}{2}\mathcal{W}]$	$-\frac{1}{2}\mathrm{d}$	Sum of extra edge weights

BCS as QP

(BCS QP):

$$
\begin{aligned}
\text{Maximize} \quad & \frac{1}{2}x^T H x + f^T x \\
\text{Subject to} \quad & Ax \le b, \\
& l \le x \le u,
\end{aligned}
$$

$$
\begin{aligned}
H &= -\frac{\lambda_2}{2}[\mathcal{W}], \\
A &= \begin{bmatrix} -\mathbf{e}^T \\ \mathbf{e}^T \end{bmatrix}, \\
b &= \begin{bmatrix} n - 2k_{min} \\ 2k_{max} - n \end{bmatrix}, \\
f &= -\frac{\lambda_1}{2}\mathbf{d}, \\
l &= -\mathbf{e}, \\
u &= \mathbf{e},
\end{aligned}
$$

Primal Dual Oracle

- The Oracle tries to certify the validity of the current $X^{(t)}$.

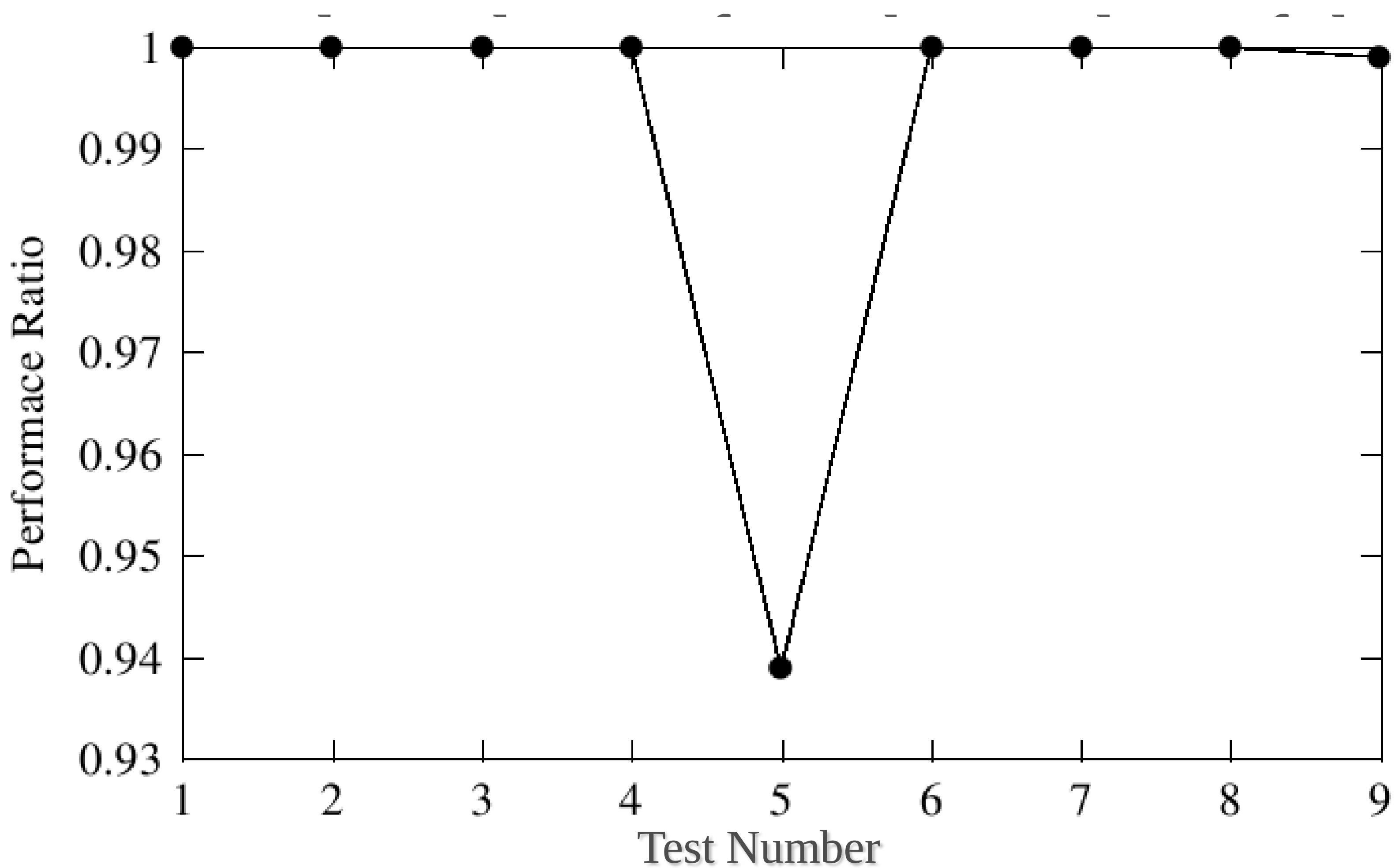
- Oracle searches for a vector $\boldsymbol{y}$ from the polytope $\boldsymbol{D}_{\alpha}=\{\boldsymbol{y}:\boldsymbol{y} \geq 0, \boldsymbol{b}\cdot\boldsymbol{y} \leq \boldsymbol{\alpha}\}$ such that

$$\sum_{j=1}^{m} (\mathbf{A}_j \bullet \mathbf{X}^{(t)}) y_j - (\mathbf{C} \bullet \mathbf{X}^{(t)}) \geq 0.$$

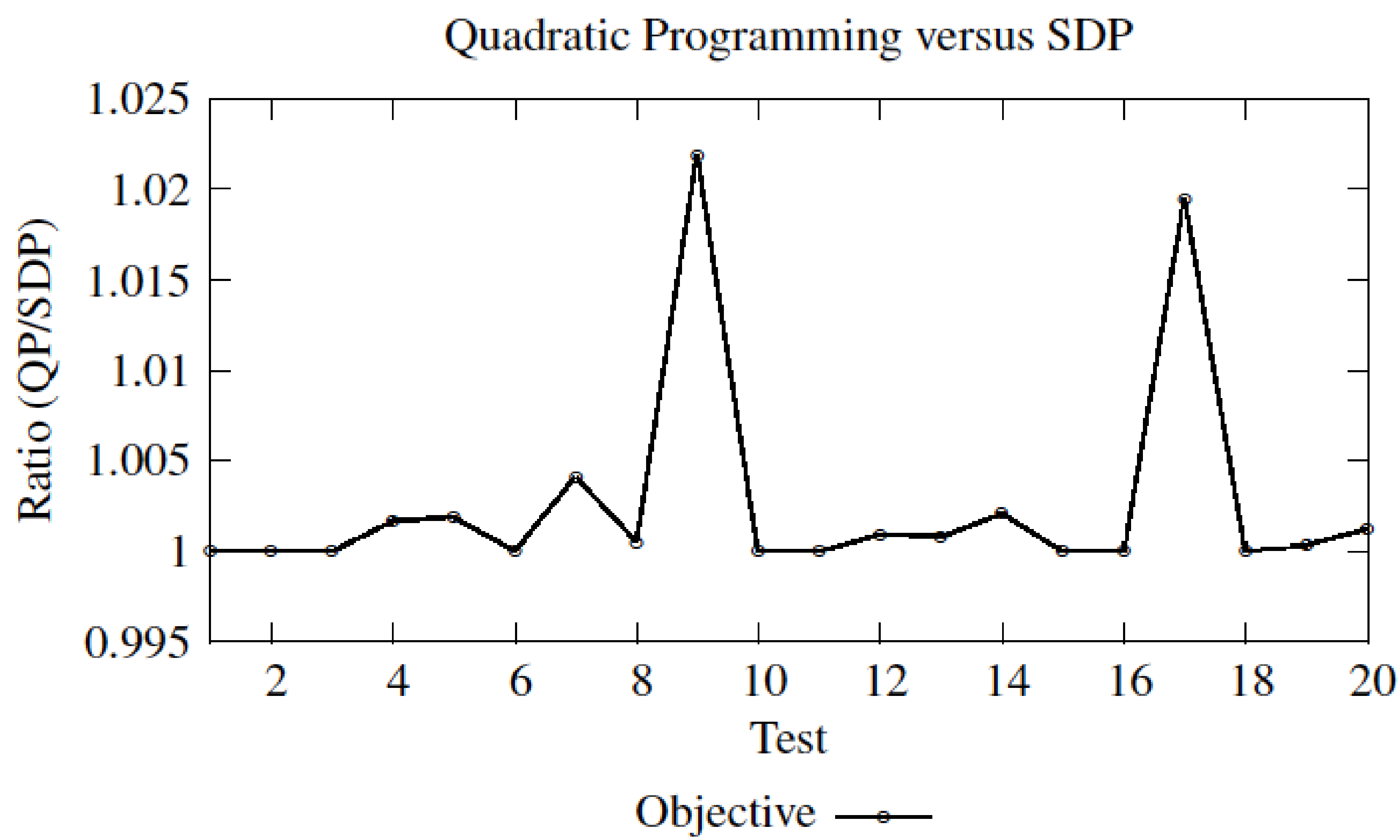
  - if Oracle succeeds in finding such a $\boldsymbol{y}$ then we claim $X^{(t)}$ is either primal infeasible or has value $\boldsymbol{C}\bullet X^{(t)} \leq \alpha$.
  - if there is no vector $\boldsymbol{y}$ in $\boldsymbol{D}$, then it can be seen that $X^{(t)}$ must be a primal feasible solution of objective value.

Objective Performance

- Exhaustive search compared to approximation
- Ex ap



Comparing SDP vs. QP on Localization Task



- The average ratio of objective values, QP/SDP was 1.003, and the ratio of execution times was 0.17.
- For technical reasons both algorithms were run as minimizations, thus lower values indicate better performance.

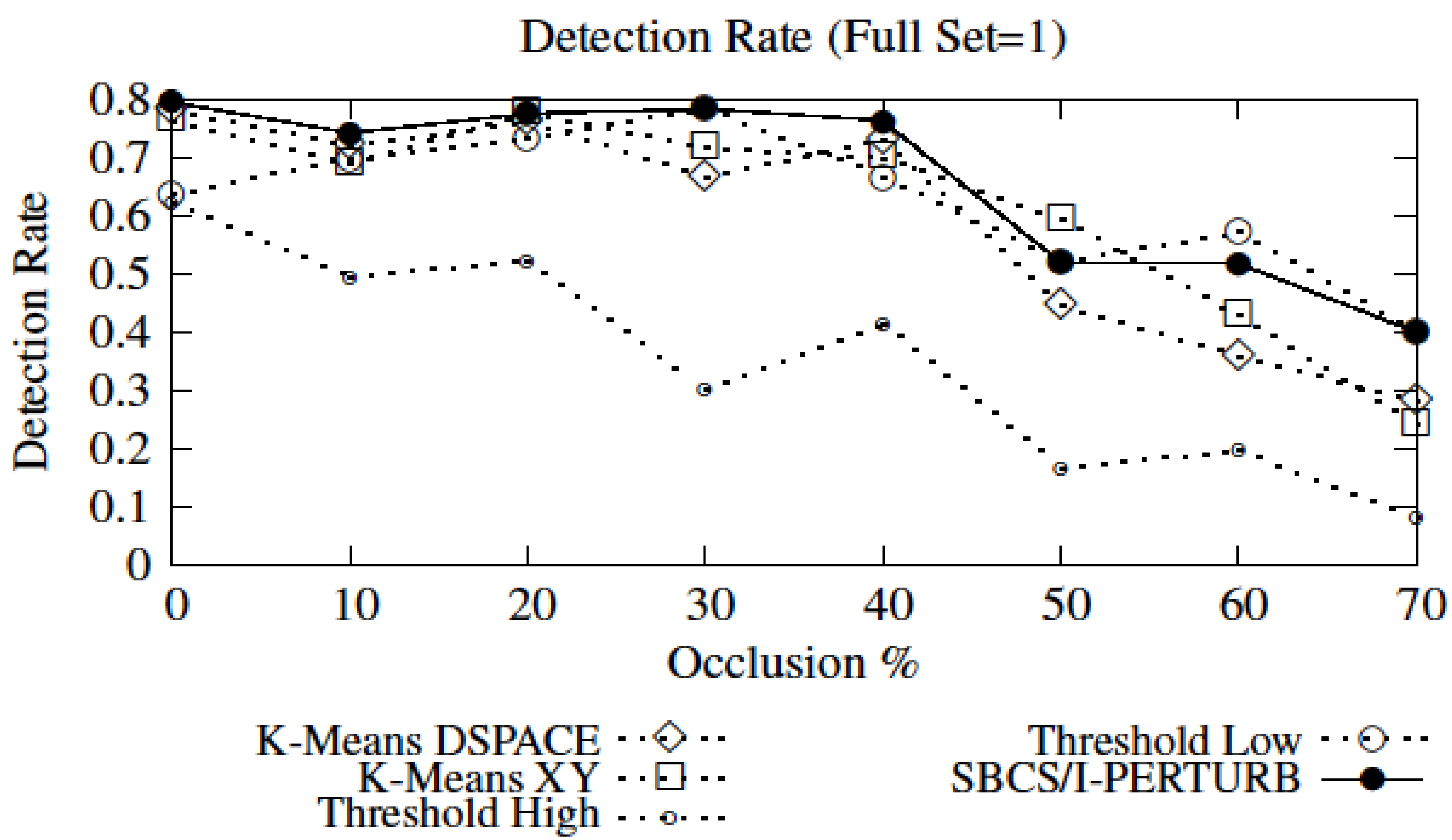


Figure 6.18: Comparison of detection rates of subset algorithms under varying amounts of occlusion. The detection rate is the number of detections divided by the number of detections obtained with the full set of features. SBCS/DSPACE/I-PERTURB has comparable performance over the entire range of occlusion. Data points include detections in a 10%