

# Approximation of Canonical Sets and Their Applications to 2D View Simplification

Trip Denton  
Drexel University

Jeff Abrahamson  
Drexel University

Ali Shokoufandeh  
Drexel University

## Abstract

Given a set of patterns and a similarity measure between them, we will present an optimization framework to approximate a small subset, known as a canonical set, whose members closely resemble the members of the original set. We will present a combinatorial formulation of the canonical set problem in terms of quadratic optimization integer programming, present a relaxation through semidefinite programming, and propose a bounded performance rounding procedure for its approximation solution in polynomial time. Through a set of experiments we will investigate the application of canonical sets for computing a summary of views from a dense set of 2D views computed for a 3D object.

## 1. Introduction & Background

Given a set of patterns  $\mathcal{P} = \{p_1, \dots, p_n\}$  and a similarity function  $\mathcal{S} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$ , the *canonical set* for  $\mathcal{P}$  is a subset  $\mathcal{P}' \subseteq \mathcal{P}$  that best characterizes the elements of  $\mathcal{P}'$  with respect to the similarity function  $\mathcal{S}$ . To motivate the problem, let us consider the 3D object in Figure 1(a). Assume we have also obtained a dense set of 2D views  $\mathcal{P}$  of this object over the viewing sphere, such as in Figure 1(b), which shows 68 silhouettes obtained from this 3D object. Our goal is to identify a small set of views that closely resembles the full set of views in  $\mathcal{P}$ .

In developing an approximation framework for computing canonical sets, we will take the novel view expressed by Cyr and Kimia [3] that “the shape similarity metric between object outlines endows the viewing sphere with a metric which can be used to cluster views into aspects, and to represent each aspect with a prototypical view.” In fact, our work on canonical sets of a general class of patterns is in large part motivated by novel ideas that were introduced in the context of aspect graph representations [13] and their relevance in identifying regions of “equivalent views” on the viewing sphere. Intuitively, in this graphical representation, each vertex represents an “aspect” of the 3D object,

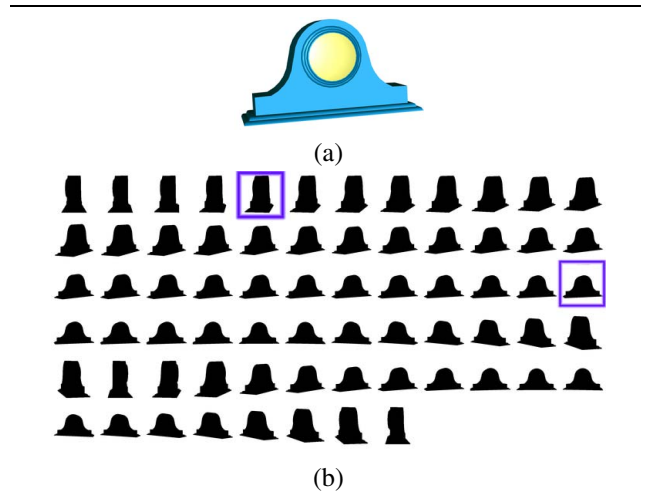


Figure 1. Motivation: Given the 3D model in (a) and a set of views in (b), and similarity function among the views, identify a small subset of views that best characterizes the object. The boxed views represent the canonical set obtained through our algorithm.

i.e., a maximally connected region on the viewing sphere. The edges of this graph will correspond to visual transitions between two neighboring general views. See Bowyer and Dyer [2] for a survey on aspect graphs and their applications. To reduce the complexity of generating the aspects, Eggert *et al.* [6] developed the notion of a scale-space aspect graph, and Dickinson *et al.* [4] used a hierarchical aspect graph system based on a finite set of primitives. Weinshall and Werman [22] studied the notions of view stability and view likelihood to establish a theory which defines the aspect graph.

In the broader context of pattern simplification, computing “equivalent views” for pattern class  $\mathcal{P}$  is closely related to that of the clustering problem. Namely, if the size  $k$  of the canonical set—the number of typical views—is more or less known, then one can use a clustering algorithm to partition  $\mathcal{P}$  into subsets  $\mathcal{P}_1, \dots, \mathcal{P}_k$ , and then choose an element

$c_i$  near the center of each cluster as the typical element of  $\mathcal{P}_i$  for each  $1 \leq i \leq k$ . This technique has, in fact, been used to define aspect graphs for polyhedra [18, 16], and solids of revolution [5]. Clearly, if the number of subsets  $k$  in the partition of  $\mathcal{P}$  is not known, then defining a centroid  $c_i$  based on the pair-wise similarity function  $\mathcal{S}$  will be extremely difficult.

But there are still several questions we would like to ask. For example, what if we do not know the size of the canonical set? How do we know if the clustering algorithm is, in fact, creating reasonable partitions? Or what if we change our notion of resemblance to a more complex predicate? For example, "I would like the elements of  $\mathcal{P}'$  to be as dissimilar to each other as possible and its size to be at least  $c_1$  but not more than  $c_2$ ." Alternatively, what if we change the definition of similarity between elements in  $\mathcal{P} \setminus \mathcal{P}'$ , the complement of  $\mathcal{P}'$ , and  $\mathcal{P}'$ ? Unfortunately, despite the simplicity and strength of such clustering methods, there is a clear need for further work.

This work is an attempt to address some of the above questions. We will present an original direct formulation of the canonical set problem as an integer programming optimization problem with a well defined set of objectives and a clear set of constraints. In view of the intractability of the integer programming problem, we present an original but natural reduction to a semidefinite programming problem (SDP) (section 2.3).

Since the seminal work of Goemans and Williamson [10] for the MAX-CUT problem in graphs, SDP relaxations have proven useful in obtaining improved approximation algorithms for several optimization problems. See Goemans [9] and Mahajan and Ramesh [14] for a survey of recent results and applications of SDP.

In order to preserve flexibility in the formulation, we will use a multi-objective formulation. To be able to solve the problem we will clearly need somehow to combine the objectives in a meaningful way. We will present the details of this setup in section 2.5. For an overview of the theory behind combining such objective functions, the reader is referred to Miettinen [15].

Given a solution of the semidefinite program, we will present a mechanism for constructing the canonical sets (section 2.6). The rounding scheme used in the construction of canonical sets is based on multivariate normal distribution rounding. See Mahajan and Ramesh [15] for an overview of rounding and its derandomization. Finally, we will carry out a set of experiments to evaluate the quality of solutions with respect to exhaustive search in the context of 2D view simplification (section 4).

## 2. Problem Formulation

In this section we define the notion of canonical set. This combinatoric problem is unfortunately intractable. We

therefore present a series of steps (integer programming, quadratic programming, semidefinite programming, and finally rounding) that allow us to compute good approximate solutions in polynomial time.

### 2.1. Definitions

Given a set of patterns  $\mathcal{P} = \{p_1, \dots, p_n\}$  and a similarity function  $\mathcal{S} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$ , we define a corresponding edge-weighted graph  $G = G(\mathcal{P})$  to capture the similarity structure among patterns in  $\mathcal{P}$ . The vertex set  $V$  consists of nodes corresponding to patterns  $p_1, \dots, p_n$ , and for every pair of patterns  $p_i, p_j \in \mathcal{P}$ , there is an edge  $(p_i, p_j) \in E$  of weight  $\mathcal{S}(p_i, p_j)$  if  $\mathcal{S}(p_i, p_j) \geq \sigma$  for a threshold  $\sigma$ . Thus, if the similarity is not sufficient, one pattern can never dominate the other. In a view-based setting, for example, the absence of an edge between two views implies that the views do not belong to the same class.

A set of vertices  $V' \subseteq V$  is a *dominating set* for  $\mathcal{P}$  if for every vertex  $u \in V \setminus V'$ , there exists a node  $v \in V'$ , with  $(u, v) \in E$ . Observe that the dominating set is different from a vertex cover in that for every vertex, either the vertex or one of its neighbors is in  $V'$ . The minimum dominating set problem is the problem of finding a dominating set  $V^*$  of minimum cardinality.

For a cut  $(V', V \setminus V')$  of  $G$ , let  $\delta(V') \subseteq E$  denote the set of cut edges with respect to  $V'$ , i.e., the edges with one end-point in  $V'$  and the other end-point in  $V \setminus V'$ . We define the cut-weight with respect to  $V'$  as:

$$\mathcal{S}(V') = \sum_{(p,q) \in \delta(V')} \mathcal{S}(p, q).$$

The *canonical set* for patterns  $\mathcal{P}$  (equivalently, its graph  $G$ ), is the minimum dominating set  $V^*$  with maximum cut-weight. Intuitively, such a set is a simultaneous solution to the dominating set and the maximum cut problems in graph  $G$ .

Our goal is to formulate the problem as a multi-objective optimization problem. To this end we need to define appropriate indicator variables, objective functions, and constraints. First we introduce some notation. We use  $I$  to denote the  $n \times n$  identity matrix and  $\mathbf{e}$  the all-ones vector of order  $n$ . We use  $d \in \mathbb{R}^n$  to denote the vector of degrees for  $G$ , i.e.,  $d_v$  is the degree of vertex  $v \in V$ . Let  $A \in \{0, 1\}^{n \times n}$  denote the adjacency matrix of graph  $G$

$$A_{i,j} = \begin{cases} 1 & \text{if } (p_i, p_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Let  $\mathcal{W} \in \mathbb{R}^{n \times n}$  denote the edge weight matrix of graph  $G$

$$\mathcal{W}_{i,j} = \begin{cases} \mathcal{S}(p_i, p_j) & \text{if } (p_i, p_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

## 2.2. Integer Programming Formulation

In this section, we present our formulation of the canonical set problem as a multi-objective integer optimization problem. For each pattern  $p_i$ ,  $1 \leq i \leq n$ , we introduce a binary indicator variable  $y_i$ . The variable  $y_i$  can have a value of  $+1$  or  $-1$ , indicating whether the corresponding pattern belongs to  $V^*$  or  $V \setminus V^*$ , respectively. Let  $y \in \{-1, +1\}^n$  be the vector  $[y_1, \dots, y_n]^t$ .

The first property a canonical set needs to satisfy is the dominating set property. That is, for every pattern  $p_i$ ,

$$(1 + y_i) + \sum_{j=1}^n A_{i,j}(1 + y_j) \geq 2. \quad (1)$$

This means that every pattern  $p_i$  is either in the dominating set, or has at least one of its immediate neighbors in the dominating set. To guarantee minimum cardinality of the set  $V^*$ , we formulate the constraint

$$\text{Minimize } \frac{1}{2} \sum_{i=1}^n (1 + y_i) = \frac{1}{2}(n + \mathbf{e}^t y). \quad (2)$$

or, equivalently, to maximize the cardinality of  $V \setminus V^*$

$$\text{Maximize } \frac{1}{2}(n - \mathbf{e}^t y). \quad (3)$$

Observe that the indicator variable  $y$  corresponding to a subset  $V^*$  satisfies  $1 - y_i y_j = 2$  if  $(p_i, p_j) \in \delta(V^*)$  and 0 otherwise.

Next, the maximization of similarity (cut weight) between patterns in  $V^*$  and those in  $V \setminus V^*$  can be formulated as

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

Finally, putting (1), (3), and (4) together, we have our integer programming formulation:

$$\begin{aligned} & \text{Maximize} && \frac{1}{2}(n - \mathbf{e}^t y) \\ & \text{Maximize} && \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \mathcal{W}_{i,j}(1 - y_i y_j) \\ & \text{Subject to} && (1 + y_i) + \sum_{j=1}^n A_{i,j}(1 + y_j) \geq 2, \\ & && y \in \{-1, +1\}^n \end{aligned}$$

## 2.3. Quadratic Integer Programming

The integer programming formulation described in section 2.2 is unfortunately known to be NP-hard [8]. In order to develop an approximation solution, our approach is to give a reformulation as a quadratic integer programming

problem and then to use vector labeling and a semidefinite programming (SDP) relaxation to obtain an approximate solution in polynomial time.

In its current form, the problem is heterogeneous, because equation (4) is quadratic and the others are linear. We therefore homogenize everything to a quadratic form to prepare for the SDP relaxation.

First we introduce a *set indicator* variable  $y_{n+1} \in \{-1, +1\}$ , that is,  $p_i \in V^*$ ,  $1 \leq i \leq n$ , if and only if  $y_i = y_{n+1}$ . This gives us

$$\begin{aligned} p_i \in V^* &\Leftrightarrow \frac{1 + y_i y_{n+1}}{2} = 1, \\ p_i \notin V^* &\Leftrightarrow \frac{1 - y_i y_{n+1}}{2} = 1. \end{aligned} \quad (5)$$

We can reformulate the cardinality objective function (2) as

$$\text{Minimize } \frac{1}{2} \sum_{i=1}^n (1 + y_i y_{n+1}) \quad (6)$$

or equivalently as

$$\text{Maximize } \frac{1}{2} \sum_{i=1}^n (1 - y_i y_{n+1}). \quad (7)$$

Similarly the dominating set constraint (1) can be rewritten as

$$(1 + y_i y_{n+1}) + \sum_{j=1}^n A_{i,j}(1 + y_j y_{n+1}) \geq 2. \quad (8)$$

Combining (4), (7), and (8) we arrive at the following quadratic formulation of the canonical set problem:

**(QIP):**

$$\begin{aligned} & \text{Maximize} && \sum_{i=1}^n \frac{1 - y_i y_{n+1}}{2} \\ & \text{Maximize} && \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \mathcal{W}_{i,j}(1 - y_i y_j) \\ & \text{Subject to} && (1 + y_i y_{n+1}) \\ & && + \sum_{j=1}^n a_{i,j}(1 + y_j y_{n+1}) \geq 2, \quad \forall 1 \leq i \leq n \\ & && y_i \in \{-1, +1\}, \quad \forall 1 \leq i \leq n + 1 \end{aligned}$$

## 2.4. Semidefinite Programming Relaxation

Since the quadratic formulation is still intractable, we use vector labeling to reformulate the problem as a semidefinite program. In canonical form, a semidefinite programming problem can be characterized as

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

where matrices  $C$ ,  $\mathcal{D}_i$ , and  $\mathcal{X}$  are all real symmetric matrices, and  $\mathcal{A} \bullet \mathcal{B}$  denotes the Frobenius inner product of matrices  $\mathcal{A}$  and  $\mathcal{B}$ , i.e.  $\mathcal{A} \bullet \mathcal{B} = \text{Trace}(\mathcal{A}^t \mathcal{B})$ .

As is conventional in SDP relaxations of quadratic integer programming problems, we introduce a vector for each indicator variable [20, 9]. That is, we replace each  $y_i$  with a vector  $x_i \in S_{n+1}$ , where  $0 \leq i \leq n$  and  $S_{n+1}$  is the unit sphere in  $\mathbb{R}^{n+1}$ . First we address equation (7), which maximizes the number of vertices that are *not* in  $V^*$ . Replacing each variable  $y_i$  with the vector  $x_i$ , we obtain

$$\text{Maximize } \frac{1}{2} \sum_{i=1}^n (1 - x_i^t x_{n+1}). \quad (9)$$

Define the matrix  $\mathcal{V} = [x_1, \dots, x_n, x_{n+1}]$  to be the matrix obtained by concatenating the column vectors  $x_i$ ,  $0 \leq i \leq n$ , and define  $\mathcal{X} = \mathcal{V}^t \mathcal{V}$ . Clearly,  $\mathcal{X}$  is a semidefinite matrix, since  $x^t \mathcal{X} x = x^t \mathcal{V}^t \mathcal{V} x = \|\mathcal{V} x\|_2^2$ ,  $\forall x \in \mathbb{R}^{n+1}$ . Observe that  $\sum_{i=1}^n x_i^t x_{n+1} = \mathcal{E} \bullet \mathcal{X} / 2$ , where  $\mathcal{E} = \begin{bmatrix} \tilde{\mathbf{0}} & \mathbf{e} \\ \mathbf{e}^t & 0 \end{bmatrix}$  and  $\tilde{\mathbf{0}}$  is an  $n \times n$  matrix of all zeroes. If we let  $\mathcal{J}$  be an  $(n+1) \times (n+1)$  all ones matrix, then we can rewrite the relaxed cardinality objective of (9) as

$$\text{Maximize } \frac{1}{2} \mathcal{E} \bullet (\mathcal{J} - \mathcal{X}) \quad (10)$$

or, equivalently, as

$$\text{Maximize } \frac{n}{2} - \frac{1}{2} \mathcal{E} \bullet \mathcal{X}. \quad (11)$$

This, in turn, can be stated in minimization form as

$$\text{Minimize } \frac{1}{2} \mathcal{E} \bullet \mathcal{X}. \quad (12)$$

Similarly, we rewrite the similarity objective in (4) as

$$\sum_{i=1}^n \sum_{j=1}^n \mathcal{W}_{ij} (1 - y_i y_j) = \tilde{\mathcal{W}} \bullet (\mathcal{J} - \mathcal{X}),$$

where  $\tilde{\mathcal{W}} = \begin{bmatrix} \mathcal{W} & \tilde{\mathbf{0}} \\ \tilde{\mathbf{0}}^t & 0 \end{bmatrix}$  and  $\tilde{\mathbf{0}}$  is an all zeroes vector in  $\mathbb{R}^n$ . Consequently, the second objective function becomes

$$\text{Maximize } \frac{1}{4} \tilde{\mathcal{W}} \bullet \mathcal{J} - \frac{1}{4} \tilde{\mathcal{W}} \bullet \mathcal{X},$$

or, equivalently,

$$\text{Minimize } \frac{1}{4} \tilde{\mathcal{W}} \bullet \mathcal{X}. \quad (13)$$

Lastly, the constraints for the dominating set property in (8) can be relaxed (for each  $1 \leq i \leq n$ ) as

$$(1 + x_i^t x_{n+1}) + \sum_{j=1}^n A_{i,j} (1 + x_j^t x_{n+1}) \geq 2,$$

which is the same as  $x_i^t x_{n+1} + \sum_{j=1}^n A_{i,j} (x_j^t) \geq 1 - d_i$ , where  $d_i$  is the degree of vertex  $i$ . Defining the coefficient matrix  $\mathcal{D}^i$  for every constraint  $i = 1, \dots, n$  as

$$\mathcal{D}^i = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & \dots & A_{i,1} \\ 0 & 0 & \dots & 0 & 0 & \dots & A_{i,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 & 0 & \dots & A_{i,i+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{i,1} & A_{i,2} & \dots & 1 & A_{i,i+1} & \dots & 2(d_i - 1) \end{bmatrix}$$

and using  $x_{n+1}^2 = 1$ , we get the following set of constraints in our relaxed SDP formulation

$$\mathcal{D}^i \bullet \mathcal{X} \geq 0, \quad \forall i = 1, \dots, n. \quad (14)$$

Observe that the integrality constraints  $y_i \in \{-1, +1\}$  are replaced by  $\text{diag}(\mathcal{X}) = \mathbf{e}$ , so the diagonal of  $\mathcal{X}$  is all ones. Putting together (12), (13), and (14) we get our semidefinite relaxation:

$$\begin{aligned} \text{(SDP):} \\ \text{Minimize } & \frac{1}{2} \mathcal{E} \bullet \mathcal{X} \\ \text{Minimize } & \frac{1}{4} \tilde{\mathcal{W}} \bullet \mathcal{X} \\ \text{Subject to } & \mathcal{D}^i \bullet \mathcal{X} \geq 0, \quad \forall i = 1, \dots, n, \\ & \text{diag}(\mathcal{X}) = \mathbf{e}, \\ & \mathcal{X} \succeq 0. \end{aligned}$$

Finally, note that both objectives in our SDP formulation are convex functions. This is due to the fact that Jacobians of both  $\mathcal{E} \bullet \mathcal{X}$  and  $\tilde{\mathcal{W}} \bullet \mathcal{X}$  are non-negative semidefinite matrices.

## 2.5. Combining Objectives

Our SDP formulation of the canonical set problem is a multi-objective optimization problem that seeks to find the optimal solution for a set of objective functions. The general set up of such optimization problems is as follows [7, 15]:

$$\begin{aligned} \text{Minimize } & \mathcal{F}(\mathcal{X}) = \{f_1(\mathcal{X}), f_2(\mathcal{X}), \dots, f_m(\mathcal{X})\} \\ \text{Subject to } & \mathcal{X} \in \Gamma, \end{aligned}$$

where  $\mathcal{F}(\mathcal{X}) = \{f_1(\mathcal{X}), \dots, f_m(\mathcal{X})\}$  is the set of objective functions and  $\Gamma$  is the feasible set.

Observe that the objective functions of the SDP formulation for the canonical set problem are somewhat competing, in that a solution that produces a canonical set of minimum cardinality does not necessarily maximize the similarity objective. In such a set up, a trade-off optimality condition known as *Pareto* optimality is used [7]. Specifically, a solution  $\mathcal{X}^*$  is called Pareto optimal if there is no  $\mathcal{X} \in \Gamma$  such that  $\mathcal{F}(\mathcal{X}) \leq \mathcal{F}(\mathcal{X}^*)$ ; that is,  $\mathcal{X}^*$  is lexicographically

optimal compared to any sub-optimal solution  $\mathcal{X}$ . Observe that such a solution is not necessarily unique. The set of all Pareto optimal solutions  $\mathcal{X}^* \in \Gamma$  is denoted by  $\Gamma_{Par}$ , the *Pareto set*. The Pareto set can be thought of as the boundary of the image of the feasible set, and a Pareto optimal solution is a point on the boundary.

If in a multi-objective optimization problem, the functions  $f_i(\mathcal{X})$ ,  $1 \leq i \leq m$ , are all convex functions, then the optimal solution  $\mathcal{X}^*$  of the following single-objective problem belongs to the Pareto set of problem  $\mathcal{F}(\mathcal{X})$  [15]:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^m \alpha_i f_i(\mathcal{X}) \\ \text{Subject to} \quad & \mathcal{X} \in \Gamma, \\ \text{where} \quad & \alpha_i \geq 0 \quad \forall 1 \leq i \leq m, \\ & \sum_{i=1}^m \alpha_i = 1. \end{aligned} \quad (15)$$

In fact, our first approach for combining the cardinality and similarity objectives of the multi-objective SDP formulation is based on using such a convex combination of the two objective functions. Specifically, we let  $\mathcal{C}_1 = \alpha\mathcal{E}/2$  and define the normalized weight matrix  $\mathcal{C}_2 = (1 - \alpha)\tilde{W}/\left(4\sum_{i,j} w_{ij}\right)$  for a convexity parameter  $\alpha \in [0, 1]$ . The normalization of the weight function has the effect of not allowing any combination of cut edges to outweigh a single vertex. The combined objective function then satisfies

$$\begin{aligned} \mathcal{C}_1 \bullet \mathcal{X} + \mathcal{C}_2 \bullet \mathcal{X} &= \text{Trace}(\mathcal{C}_1^t \mathcal{X}) + \text{Trace}(\mathcal{C}_2^t \mathcal{X}) \\ &= \text{Trace}((\mathcal{C}_1 + \mathcal{C}_2)^t \mathcal{X}). \end{aligned}$$

Letting  $\mathcal{C}(\alpha) = \mathcal{C}_1 + \mathcal{C}_2$ , our multi-objective SDP formulation is

$$\begin{aligned} \text{(CoSDP):} \\ \text{Minimize} \quad & \text{Trace}(\mathcal{C}^t \mathcal{X}) \\ \text{Subject to} \quad & \mathcal{D}^i \bullet \mathcal{X} \geq 0, \quad \forall i = 1, \dots, n \\ & \text{diag}(\mathcal{X}) = \mathbf{e}, \\ & \mathcal{X} \succeq 0. \end{aligned}$$

Finally, for every  $\alpha \in [0, 1]$  the optimal solution  $\mathcal{X}^* = \mathcal{X}^*(\alpha)$  can be computed in polynomial time using the algorithm [1] (for details on a SDP solver the reader is referred to [19]).

## 2.6. Rounding Scheme

The final stage in our approximation algorithm, once we have solved the semidefinite program, is to construct a feasible integer solution to the canonical set problem.

This process, also known as rounding, identifies the set of values for indicator variables  $y_1, \dots, y_n$  and the set indicator variable  $y_{n+1}$ . In our experiments, we have used a rounding scheme based on Cholesky decomposition and a

multivariate normal hyperplane method that can be effectively derandomized. (See [14] for details.)

Let  $\mathcal{X}^* = \mathcal{X}^*(\alpha)$  denote the optimal solution of the (CoSDP) for a given  $\alpha \in [0, 1]$ . Since  $\mathcal{X}^*$  is a symmetric positive semidefinite matrix, using Cholesky decomposition it can be represented as  $\mathcal{X}^* = \mathcal{V}^t \mathcal{V}$  [11]. This provides us with  $n+1$  vectors for the relaxed canonical set problem. Specifically, column  $x_i$ ,  $1 \leq i \leq n$ , of  $\mathcal{V}$  forms the vector associated with vertex  $v_i \in G$  in the optimal SDP relaxation of the canonical set problem, and column  $x_{n+1}$  corresponds to the set indicator variable.

Finally, we pick a random vector  $\rho \in \mathbb{R}^{n+1}$  with multivariate normal distribution, with 0 mean, and with covariance  $\mathcal{X}^*$ , to generate the indicator variables  $y_i$ ,  $i \leq 1 \leq n+1$  and the canonical set  $V^*$  as  $y_i = \mathbf{sign}(\rho^t x_i)$  and  $V^* = \{v_i \mid y_i = y_{n+1}, 1 \leq i \leq n\}$ .

Note that as a result of this rounding step, some of the constraints of (QIP) might be violated. Namely, for some vertex  $v$ , neither  $v$  nor any of its neighbors may belong to set  $V^*$ . Let  $\tilde{V}$  denote the set of all such vertices, and  $\ell = |\tilde{V}|$ . For every  $v \in \tilde{V}$  we define its potential  $\phi(v)$  as the sum of edge-weights of neighbors of  $v$  normalized by the size of

its neighborhood, i.e.,  $\phi(v) = \left( \sum_{u \in \text{adj}(v)} \mathcal{W}_{v,u} \right) / d_v$ , where

$\text{adj}(v)$  is the neighborhood of  $v$  and  $d_v$  is the degree of  $v$ . Next we sort the potential values of all vertices in  $\tilde{V}$ , obtaining a sequence  $\phi(v_1) \geq \phi(v_2) \geq \dots \geq \phi(v_\ell)$ . We iteratively add vertex  $v_1$  to  $V^*$ , remove  $v_1$  and all its neighbors in  $\text{adj}(v)$  from  $\tilde{V}$ , and recompute the potential of all remaining vertices in  $\tilde{V}$ . This greedy process tries to add vertices into  $V^*$  in increasing order of potential contribution to both cardinality and similarity and can be repeated at most  $\ell$  times. It is worth mentioning that such a greedy algorithm can be started from the beginning to try to find an approximation solution with  $V^*$  initially empty. Experiments comparing the greedy algorithm to our SDP formulation confirm that SDP produces better results.

Alternatively, we can run the rounding algorithm multiple times and keep the best result. For details of the rounding scheme and quality of the post-rounding solution the reader is referred to [21].

## 3. Final Algorithm

Our algorithm for computing the canonical set for a given class of patterns  $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$  under similarity function  $\mathcal{S}$  is a combination of the previous procedures. Specifically, given the pair  $(\mathcal{P}, \mathcal{S})$ , we form the weighted graph  $G(\mathcal{P})$ , and form the semidefinite programming formulation, and the combined objective functions. We next solve the resulting semidefinite optimization program to obtain the solution  $\mathcal{X}^*$ . To construct the vector-coloring of nodes in  $G(\mathcal{P})$ , we compute the Cholesky decomposition of

$\mathcal{X}^*$ . Finally, to form the canonical set  $\mathcal{V}^*$ , we use the rounding scheme and fix-up steps. We summarize our approach in Algorithm 1.

---

**Algorithm 1** Approximation of Canonical Set

---

- 1: Construct the graph  $G(\mathcal{P})$  according to section 2.
  - 2: Form the semidefinite program with combined objective (**CoSDP**) according to sections 2.3 and 2.5 .
  - 3: Solve (**CoSDP**) using the algorithm in [19], obtaining PSD matrix  $\mathcal{X}^*$ .
  - 4: Compute the Cholesky decomposition  $X^* = \mathcal{V}^t \mathcal{V}$ .
  - 5: Construct the indicator variables  $y_1, \dots, y_n, y_{n+1}$  and form the set set  $V^*$  according to section 2.6.
- 

Steps one, two, and five of algorithm 1 each have running time  $O(n^2)$ . Using Alizadeh’s adaption of the interior-point method [1], step three can be implemented in time  $O(\sqrt{n}(\log L + \log 1/\epsilon)n^3)$  where  $\log L$  is the size of the coefficient matrix  $\widetilde{W}$  and  $\epsilon$  is the error on the quality of the SDP solution. Finally, step four requires  $O(n^3)$ .

**4. Experiments**

In this section we present an overview of experiments we have performed to date to evaluate the optimization framework for computing canonical sets. Each pattern class in our experiments corresponds to a set of 2D views acquired from a 3D object. Specifically, we use 9 objects, each representing a single pattern class. For each object we have as many as 180 2D views acquired along a great circle of the viewing sphere, giving a total of 1620 views. A representative view of each object is shown in Figure 2.



**Figure 2. Sample views of the 3D objects used in our experiments.**

---

Next we use a distance function to measure the similarity among the patterns that constitute each pattern class. In our previous work [12], we have developed an algorithm for computing the many-to-many matching, as well as similarity measure, of 2D views represented as silhouettes. For a given view an object’s silhouette is first represented by an undirected, rooted, weighted graph, in which nodes represent *shocks* [17] (or, equivalently, skeleton points) and edges connect adjacent shock points. The shock graphs will, in turn, be represented in terms of shock trees using a minimum spanning tree of the weighted shock graph. (For details on the construction of this trees, see [12].)

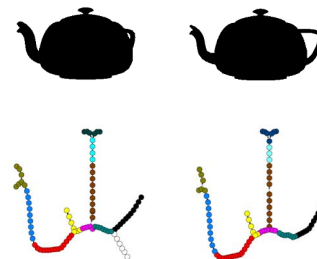
An illustration of this representation is given in Figure 3. The left portion shows the initial silhouette and its shock points (skeleton). The right portion depicts the constructed shock tree. Darker, heavier nodes correspond to fragments whose average radii are larger.



**Figure 3. Left: the silhouette and its medial axis. Right: the medial axis tree constructed from the medial axis. Darker nodes reflect larger radii.**

---

Figure 4 illustrates the many-to-many correspondences provided by the algorithm in [12] for two adjacent views (30° and 40°) of the TEAPOT. Corresponding clusters (many-to-many mappings) have been shaded with the same color (the extraneous branch in the left view was not matched in the right view), and the algorithm provides an overall measure of similarity between the two views as well.



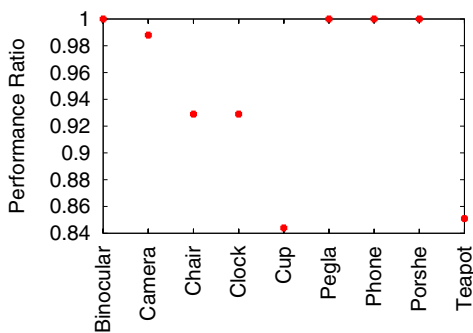
**Figure 4. Illustration of the correspondences produced by the matching algorithm to compute the similarity measure between objects.**

---

For the experiments, we compute the shock tree representation of every silhouette, and used the many-to-many matching algorithm of [12] to compute the distance values among 2D silhouettes corresponding to each 3D object (pattern class). The outcome of this procedure is distance matrices. To form similarity matrices, and thus the graph  $G(\mathcal{P})$ , we set to zero all elements whose distance is greater than the mean distance and use a similarity value of  $e^{-d}$  for distance  $d$  less than the mean.

Choosing the convexity parameter  $\alpha$  is a critical step in generating the combined objective function (see section 2.5). To this end we performed a set of experiments in which we measured the value of (CoSDP) as the value of  $\alpha$  was varied in the interval  $[0, 1]$ . We have observed that a balanced trade-off between the cardinality of the canonical set and the similarity objectives, i.e.,  $\alpha = 0.50$ , produces the most consistent results. This by no means is a general assertion, and certainly needs a closer investigation with respect to the similarity function  $S$ .

Next we created 9 smaller pattern classes from our original classes, with about 20 views per object. For each pattern class we searched the space of all possible subsets and identified the smallest canonical set whose value for the similarity objective is maximum. Next we used algorithm 1 to compute an approximation solution for (CoSDP) with balanced combined objectives. Figure 5 represents the performance ratio between the similarity objective of the canonical set obtained from algorithm 1 with that of the exhaustive search. In these cases results of algorithm 1 were within a factor of 0.84 of exhaustive search. Figure 6 illustrates the sample views of several pattern classes and the canonical sets computed for each class (outlined in blue).

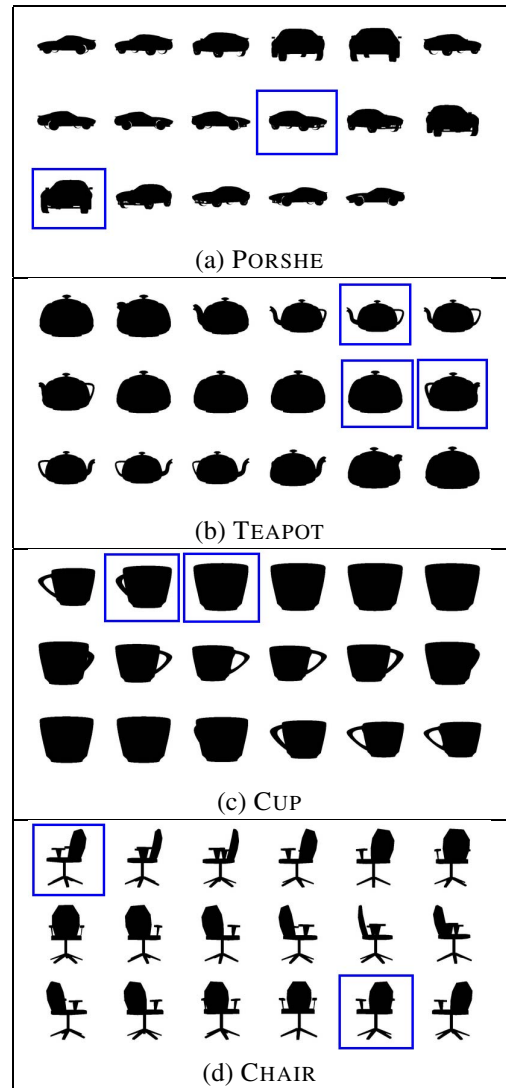


**Figure 5. Comparing the similarity objective for canonical sets obtained from Algorithm 1 and for exhaustive search.**

Examining figure 6 subjectively, we see that the results for PORshe and CUP are meaningful. For example, in the case of PORshe, the canonical set has a partial profile and partial frontal view. Similarly, CUP shows a view with handle and one without.

In the case of TEAPOT, on the other hand, visually one view appears to be redundant. Upon close investigation, we realize we are not currently minimizing the similarity between members of the canonical set.

In the case of CHAIR, we would have expected the bottom right chosen view to be instead the full frontal view to



**Figure 6. Canonical sets (blue rectangles) for four sets of views.**

its left. The algorithm did not produce as good results in this experiment. Unfortunately, at this point we do not have a complete understanding of the correspondence between the optimal canonical set and our psychophysical perception. We suspect our distance function is not properly modeling visual distance. Running the algorithm on a series of 90 views of the chair took 41 seconds (wall time) on a Pentium III running at 1133 MHz.

## 5. Summary and Future Work

We have developed an approximation framework for the canonical set problem based on SDP relaxation of an integer programming formulation. Through a series of experiments we evaluated this framework in the context of 2D

view simplification of 3D objects. Our results compared favorably to exhaustive search.

Our definition of canonical set was limited to a single pattern class. In reality we must always consider the application domain. For example, consider the problem of image indexing. Given multiple pattern classes, we may regard them as two classes: the class of interest and all others. We might ask for a canonical set of the class of interest, maximally similar to members of its own class while maximally dissimilar to patterns outside it.

Specifically, let  $\mathcal{P} = \{p_1, \dots, p_n\}$  denote the class of interest and  $\mathcal{Q} = \{q_1, \dots, q_m\}$  the set of patterns outside  $\mathcal{P}$ . We might ask for the canonical set  $\mathcal{P}^*$  for  $\mathcal{P}$  such that each view in  $\mathcal{P}^*$  is maximally dissimilar from the views in  $\mathcal{Q}$ . This corresponds to using the following objective function:

$$\text{Minimize } \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^m (1 - y_i y_{n+1}) \mathcal{S}(p_i, q_j).$$

We also expect that canonical sets may prove useful in clustering. The canonical set itself may provide proposed centroids of clusters, with a minimum distance classifier completing the task. This may prove useful for the segmentation problem.

In addition to these significant areas of further research, several questions about canonical sets remain open. We would like to establish performance guarantees for the canonical set framework. In addition, we would like to better understand the meaning of an optimal canonical set for our 2D view application, as well as to explore the effects of the similarity computation on the psychophysical quality of the canonical set.

Finally, we would like to explore the relationship between the sparsity of the similarity matrix and the cardinality of the canonical sets found.

## Acknowledgments

The authors would like to thank Sven Dickenson (University of Toronto), M. Fatih Demirci (Drexel University) and Yakov Keselman (DePaul CTI) for their insightful comments. Funded in part by grants from National Science Foundation (NSF/EIA 02-05178) and Office of Naval Research (ONR-N000140410363).

## References

- [1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.
- [2] K. Bowyer and C. Dyer. Aspect graphs: an introduction and survey of recent results. *IJIST*, 2:315–328, 1990.
- [3] C. M. Cyr and B. Kimia. 3d object recognition using shape similarity-based aspect graph. In *8<sup>th</sup> Inter. Conf. Comp. Vision*, pages 254–261, 2001.
- [4] S. Dickinson, A. Pentland, and A. Rosenfeld. A representation for qualitative 3-D object recognition integrating object-centered and viewer-centered models. Technical Report CAR-TR-453, Center for Automation Research, University of Maryland, 1989.
- [5] D. Eggert and K. Bowyer. Computing the orthographic projection aspect graph of solids of revolution. *Pattern Recognition Letters*, 11:751–763, 1990.
- [6] D. Eggert, K. Bowyer, C. Dyer, H. Christensen, and D. Goldgof. The scale space aspect graph. *IEEE PAMI*, 15(11):1114–1130, November 1993.
- [7] M. Ehrgott. *Multicriteria Optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, New York, 2000.
- [8] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman and Co., Baltimore, 1979.
- [9] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79:143–161, 1997.
- [10] M. X. Goemans and D. P. Williamson. .878-approximation algorithms for max cut and max 2sat. In *Twenty-sixth Annual ACM Symposium on Theory of Computing*, pages 422–431, New York, 1994.
- [11] G. Golub and C. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.
- [12] Y. Keselman, A. Shokoufandeh, M. Demirci, and S. Dickinson. Many-to-many graph matching via metric embedding. In *CVPR*, pages I–850–I–857 vol.1, June 2003.
- [13] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.
- [14] S. Mahajan and H. Ramesh. Derandomizing approximation algorithms based on semidefinite programming. *SIAM J. Comp.*, 28(5):1641–1663, 1999.
- [15] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.
- [16] I. Shimshoni and J. Ponce. Finite resolution aspect graphs of polyhedral objects. In *Proceedings, IEEE Workshop on Qualitative Vision*, pages 140–150, New York, NY, June 1993.
- [17] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *IJCV*, 30:1–24, 1999.
- [18] J. Stewman and K. Bowyer. Creating the perspective projection aspect graph of polyhedral objects. In *Proceedings, IEEE Second International Conference on Computer Vision*, pages 494–500, Tampa, FL, 1988.
- [19] K. C. Toh, M. J. Todd, and R. Tutuncu. SDPT3 — a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.
- [20] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.
- [21] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Germany, second edition, 2003.
- [22] D. Weinshall and M. Werman. On view likelihood and stability. *IEEE PAMI*, 19(2):97–108, February 1997.