Visibility Constraints on Features of 3D Objects

Ronen Basri*
Weizmann Institute and TTI-C

Pedro F. Felzenszwalb†
University of Chicago

Ross B. Girshick
University of Chicago

David W. Jacobs‡
University of Maryland

Caroline J. Klivans
University of Chicago

Abstract

To recognize three-dimensional objects it is important to model how their appearances can change due to changes in viewpoint. A key aspect of this involves understanding which object features can be simultaneously visible under different viewpoints. We address this problem in an image-based framework, in which we use a limited number of images of an object taken from unknown viewpoints to determine which subsets of features might be simultaneously visible in other views. This leads to the problem of determining whether a set of images, each containing a set of features, is consistent with a single 3D object. We assume that each feature is visible from a disk of viewpoints on the viewing sphere. In this case we show the problem is NP-hard in general, but can be solved efficiently when all views come from a circle on the viewing sphere. We also give iterative algorithms that can handle noisy data and converge to locally optimal solutions in the general case. Our techniques can also be used to recover viewpoint information from the set of features that are visible in different images. We show that these algorithms perform well both on synthetic data and images from the COIL dataset.

1. Introduction

To recognize three-dimensional objects it is important to model the variations in appearance that can occur due to changes in viewpoint. If we represent objects with the popular *bag-of-features* approach [14, 5] this variability primarily takes the form of visibility constraints. Different subsets of features are visible from different viewpoints, and there are constraints on the set of features that are simultaneously visible. For example, we might be able to see both eyes on a face, or only one eye, but there are no viewpoints from which both eyes are visible, and the nose is not visible.

This paper addresses the problem of inferring which sets of features in an object may be simultaneously visible based on a limited number of 2D views of the object taken from unknown viewpoints. We assume that an object contains a fixed set of features and when we see an image of the object we detect a subset of these features. The problem we tackle assumes that each feature is visible from a fixed but unknown set of viewpoints. For a subset of features we have not seen together, we wish to infer whether it is possible for this subset to be simultaneously visible.

The difficulty of this inference problem depends in part on the possible sets of viewpoints from which a feature can appear. If this set is arbitrary, no inference is possible.

Here we assume that each feature is visible from a disk in the viewing sphere. This is a natural model that can be motivated as follows. Suppose each feature comes from a particular point in an object, and we see the feature if the viewing direction is within some fixed angle of the object's surface normal at the feature location. This implies that the set of viewpoints from which a feature is visible form a disk on the viewing sphere. This disk is the intersection of the viewing sphere and a half-space defined by the surface normal and the angle constraint. For a smooth, convex object a feature could be visible if the viewing direction is within 90 degrees of the surface normal. Or we might imagine that we can reliably detect a feature when the viewing direction is within 30 or 45 degrees of the surface normal.

Note that while our visibility model is motivated in terms of point-like features we do not make any specific assumptions on how features are detected in practice. In particular, we can handle features that are not well spatially localized, as long as they satisfy the disk assumption.

Our results are complementary to geometric constraints that have been widely explored in the past (eg., [13, 6]). Features detected in multiple images may be geometrically consistent, but not satisfy the visibility constraints that we develop here. Similarly, our visibility constraints do not ensure geometric consistency in well-localized features. We note, though, that in the case of poorly localized features, visibility constraints may be used even when geometric con-

*Supported in part by the Israel Science Foundation Grant 628/08.
†Supported in part by NSF award 0746569.
‡Supported by ARO Grant #W911NF-08-1-0466.

straints are difficult to apply.

Our work resembles, but differs significantly from past work on aspect graphs (e.g., [4, 9]), which assumes prior knowledge of a 3D model. In particular, our approach is entirely image-based.

The disk assumption imposes strong constraints among subsets of features that are simultaneously visible from different viewing directions. Any object with n features will have $O(n^2)$ different subsets of features that are simultaneously visible, out of a total of 2^n subsets of features. This exponential gap demonstrates the potential value of visibility constraints. For example, in object recognition, we might detect a set of features in an image, and compare these to the features that can be seen in a specific object. It is much more likely that a distracter object will produce one of the 2^n subsets of features of a known object than one of the $O(n^2)$ subsets that could be simultaneously visible.

The main problem we consider here is the problem of deciding if a collection of subsets of features is consistent, in the sense that each subset could be generated by looking at the same object from different directions. When the subsets are consistent we also want to build a model of the visibility regions of each feature so that we can predict which features would be visible from an arbitrary viewpoint. In particular this allows us to estimate the viewpoint of an image based on the set of features that were detected.

Our main theoretical result is an efficient algorithm for a special case in which the viewpoints are coplanar. This includes the situation in which objects are observed from a camera at fixed height and distance. We also show that the problem is NP-hard in the general case, when the viewpoints are arbitrary.

Of course the disk assumption does not always hold in practice. In particular, feature detection can be a noisy process and features may not be visible due to self-occlusion. Thus we also consider the problem of building visibility models from "noisy" data. In this case we want to infer visibility regions that can capture a set of observed images as well as possible. We describe an effective iterative algorithm that finds locally optimal solutions, and illustrate that this algorithm works well both on real and synthetic data.

2. Visibility models

To capture the possible subsets of features that can be simultaneously visible in a single image of an object, we set up the following purely geometric problem.

Let $S^{d-1} = \{x = (x_1, \ldots x_d) : ||x|| = 1\} \subset \mathbb{R}^d$ be the (d-1)-dimensional hypersphere naturally embedded in $\mathbb{R}^d$ and $V = \{v_1, \ldots v_m\}$ a finite set of points on S^{d-1}. We are primarily concerned with situations in which $d \in \{2, 3\}$.

Furthermore, let $F = \{f_1, \ldots, f_n\}$ be a collection of half-spaces given by a normal direction and threshold; $f_i = (h_i, t_i)$, where $h_i \in S^{d-1}$, and $t_i \in \mathbb{R}$. If $t_i = 0$ the half-space f_i is called *central*. An important special case occurs when all the f_i are central, in which case we have a *central arrangement*.

We use the term *sign matrix* for an arbitrary $\{+1, -1\}$ matrix. We are interested in sign matrices M such that

$$M_{ij} = \begin{cases} +1 & \text{if } v_i^T h_j > t_j, \\ -1 & \text{if } v_i^T h_j < t_j \end{cases}$$

(or equivalently $M_{ij}(v_i^T h_j - t_j) > 0$), which record for each point $v_i \in V$ and each half-space $f_j \in F$ if v_i lies in f_j. To simplify the analysis we assume $v_i^T h_j \neq t_j$. Any matrix arising from a collection of viewpoints and half-spaces in this way will be called a *visibility matrix*.

In this setup S^{d-1} corresponds to the viewing sphere when $d = 3$ and a circle of the viewing sphere when $d = 2$. V is a set of viewpoints of an object that lies at the center of the sphere. The half-spaces F define the visibility regions for the features in the object. If a viewpoint v lies in the half-space f then the corresponding feature is thought to be visible from v. Thus the i-th row of M captures which features are visible from viewpoint v_i, and the j-th column captures which views see feature j.

The fundamental problems we consider in this paper are the following. *Given a sign matrix M, can we efficiently determine whether or not M is a visibility matrix? If yes, can we find a set of defining viewpoints V and half-spaces F? If no, can we find viewpoints V and half-spaces F that leads to the "closest" visibility matrix?*

The problem of identifying visibility matrices of central arrangements in $\mathbb{R}^d$ is equivalent to determining if the *sign-rank* of a matrix is at most d. The *sign-rank* of a matrix M is the minimum rank of a real-valued matrix $\tilde{M}$ with $\text{sign}(M_{ij}) = \text{sign}(\tilde{M}_{ij})$. The notion of sign-rank arises in various contexts including communication complexity and machine learning [7, 10, 12].

Let M be a visibility matrix generated from a set of viewpoints V and central half-spaces F in $\mathbb{R}^d$. Let $\tilde{V}$ be a $d \times m$ matrix whose columns are the viewpoints in V, and $\tilde{H}$ be a $d \times n$ matrix whose columns are the normals of the half-spaces in F. Note that $\tilde{M} = \tilde{V}^T \tilde{H}$ has rank at most d and $\tilde{M}_{ij} M_{ij} > 0$. Thus the sign-rank of M is at most d.

Now suppose M has sign-rank at most d. Then there exists a real-valued matrix $\tilde{M}$ with $M_{ij}\tilde{M}_{ij} > 0$ and the rank of $\tilde{M}$ is at most d. Now we can write $\tilde{M} = \tilde{V}^T \tilde{H}$ where $\tilde{V}$ is $d \times m$ and $\tilde{H}$ is $d \times n$. We can take V to be the columns of $\tilde{V}$ normalized to length one, and F to be hyperplanes with normals defined by the columns of $\tilde{H}$. Then M is the visibility matrix generated by V and F.

It is straightforward to show that M has sign-rank 1 iff it has rank 1. Our results establish that deciding if M has sign-rank at most 3 is NP-hard, answering a question posed

in [7]. In addition we provide an efficient algorithm for determining if the sign-rank of a matrix is at most 2.

3. Hardness

As a starting point we observe that determining whether a sign matrix M is a visibility matrix of a central arrangement in $\mathbb{R}^3$ is NP-hard. The key to this observation is a connection to the theory of oriented matroids [3].

An oriented matroid is a finite combinatorial object designed to capture the abstract notions of oriented linear dependence. One axiom system for oriented matroids is motivated by the geometry of a real central hyperplane arrangement. Consider a collection of hyperplanes $\mathcal{H} \in \mathbb{R}^d$ and the sign vectors induced by the cells of the arrangement. Namely, represent each hyperplane H_i by an orthogonal vector $a_i \in S^{d-1}$ and for each point $x \in \mathbb{R}^d \backslash \mathcal{H}$ form the vector $(\text{sign}(x^T a_1), \ldots, \text{sign}(x^T a_{|\mathcal{H}|}))$. Any collection of sign vectors formed as such defines an oriented matroid.

There exist collections of sign vectors which constitute an oriented matroid but which do not come from any real central hyperplane arrangement. These are known as non-realizable oriented matroids. A fundamental problem in the theory of oriented matroids asks, given an oriented matroid, is it realizable, i.e. does it come from some real hyperplane arrangement? The problem was shown to be NP-hard in dimension 3 by an equivalence to the existential theory of the reals [8]. Shor proves hardness via a reduction from 3-SAT using only planar incidence geometry [11].

Suppose we are given a sign matrix whose rows correspond to a collection of sign vectors of an oriented matroid. Then determining whether or not this matrix is a visibility matrix of a central arrangement in $\mathbb{R}^3$ is equivalent to determining whether or not the oriented matroid is realizable in dimension 3. Thus determining if a sign matrix M is a visibility matrix of a central arrangement in $\mathbb{R}^3$ is NP-hard. Therefore determining if the sign rank of a matrix is at most 3, and thus computing sign rank, is NP-hard.

The argument above does not directly apply to the case of non-central visibility matrices. However, hardness in this case may also be established from the oriented matroid framework (we omit the details here for lack of space).

4. 2D Arrangements

While the problem of deciding if M is a visibility matrix in three-dimensions is NP-hard, we have developed efficient algorithms for the case of two-dimensional arrangements. This case occurs in practice when the collection of viewpoints are coplanar.

First we consider the case of central arrangements, in which each of the visibility regions spans a range of $180°$. Next we address the case of arbitrary non-central arrangements. The added complexity of the non-central case can be seen in the structure of the *H1-graph* of an arrangement. Suppose we have an arrangement of n half-spaces in $\mathbb{R}^2$. The arrangement induces a decomposition of the circle into cells. Each cell can be labeled by the sign vector recording which side of each line the cell sits on. We define the *H1-graph* to be a graph with nodes representing the sign vectors obtained in this way and edges connecting every two sign vectors with Hamming distance one. In the central case this graph is simply a cycle.

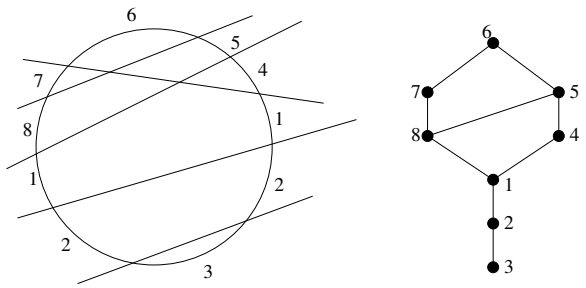

Figure 1. A non-central arrangement and its *H1-graph*

The case of non-central arrangements gives rise to more complex configurations because the graph identifies cells with the same sign vector. For example, two or more disconnected regions along the viewing circle may have the same sign vector and a pair of cells with Hamming distance one may not be neighbors on the circle, see Figure 1.

4.1. Algorithms

There are many arrangements, including combinatorially different ones, that lead to the same decomposition of the viewing circle into visibility cells. Figure 2 shows such an example. These arrangements produce the same sets of visible features. Our algorithms avoid distinguishing between them by working directly with the structure induced on the viewing sphere.

In two-dimensions a decomposition of the viewing circle into visibility cells can be encoded by a cyclic ordering of $I_n = \{1, 1', 2, 2', \ldots, n, n'\}$. The ordering specifies the clockwise order of intersection points between the lines defining visibility regions for each feature and the circle. We say that the i-th feature is visible in the region formed by going from i to i' in clockwise order. For example, the structure defined by the arrangements in Figure 2 is encoded by $(1, 3, 2', 4, 1', 3', 4', 2)$. Note that any decomposition of the viewing circle induced by an arrangement of oriented lines can be represented by such an ordering. Moreover, any ordering defines a decomposition that can be realized by an arrangement. An ordering defines a decomposition that can be realized by a central arrangement if and only if the position of i and i' differs by n.

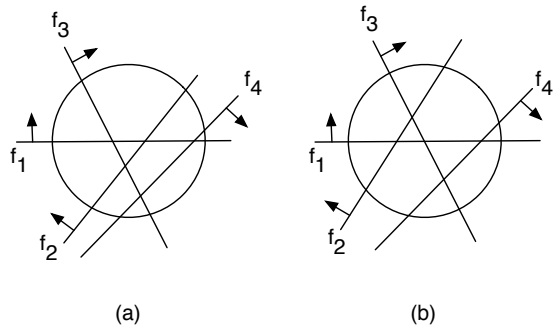

Figure 2. Two different arrangements that induce the same decomposition of the viewing sphere into visibility cells.

4.2. Central Arrangements in 2D

The input to the algorithm is an m by n sign matrix M, where each row is a sign vector representing the set of features visible in a particular image. We would like to (1) find a set of *central* oriented lines $F = \{f_1 \ldots, f_n\}$ and viewpoints $V = \{v_1 \ldots, v_m\}$ such that v_i is on the positive side of f_j if and only if $M_{ij} = 1$, or (2) decide that this is not possible. For the first case, the actual output of the algorithm is a cyclic ordering, A_n, of I_n, capturing the relative ordering of the intersection points of F with the viewing circle, and an assignment, ρ_n, of viewpoints to the visibility cells defined by such an ordering. A geometric arrangement of lines and viewpoints consistent with M can easily be produced from this data.

Our algorithm works in a greedy fashion, considering one feature at a time. At the i-th iteration we determine the relative position of f_i with respect to the previously considered features. Below, A_i is a cyclic ordering of I_i (defining an arrangement of the first i features) and ρ_i denotes a map from viewpoints to cells defined by A_i. We construct A_i and ρ_i by extending A_{i-1} and ρ_{i-1}. This is done so that at every iteration A_i and ρ_i represent an arrangement compatible with the first i columns of M.

If M contains two identical columns or columns that are opposites of each other we remove one of them. Furthermore for each row of M we add its opposite row if it was not already in M. The new matrix is a visibility matrix of a central arrangement if and only if the original matrix was. This transformation ensures that an arrangement realizing M has no empty cells. Because if there was an empty cell, the addition of opposite rows implies the opposite cell must also be empty. But a pair of opposite empty cells implies the existence of either equal or opposite columns.

1. Without loss of generality we let $A_2 = (1, 2, 1', 2')$. This decomposes the circle into four cells, according to the four possible sign configurations on the first two features. We define ρ_2 by assigning each sign vector to one of these cells according to its sign on f_1 and f_2.
2. At the i-th iteration, we look for a cell, C, induced by A_{i-1} which contains two vectors with opposite signs on f_i. If no such cell exists then M is not a central visibility matrix. Otherwise f_i must divide such a cell. Because we are looking for a central arrangement, f_i must also divide the opposite cell, $\bar{C}$. This defines two places in A_{i-1} where we should insert i and i'. All vectors in cells between C and $\bar{C}$ in clockwise order should have the same sign on f_i, and all vectors between $\bar{C}$ and C should have the opposite sign. This determines where i and i' must be inserted or leads to a contradiction, in which case we decide M is not a central visibility matrix. If there is no contradiction we place i and i' to obtain A_i, and update ρ_{i-1} by reassigning viewpoints in the two cells which are split by f_i according to their sign on f_i.

The correctness of the algorithm is due to the uniqueness of arrangements compatible with a subset of features, which we prove by induction. Note however that A_2 is only unique up to a choice of orientation of two features. There are two possible way to orient two features which simply lead to arrangements that are related by a mirror reflection.

Now suppose A_i and ρ_i are unique (up to the choice of A_2) and let A_{i+1} and ρ_{i+1} represent an arrangement of the first $i+1$ features compatible with the first $i+1$ columns of M. There is a natural arrangement of the first i features, A'_i, ρ'_i induced by A_{i+1}, ρ_{i+1} which simply restricts A_{i+1} to those elements of I_i and coarsens ρ_{i+1}. Each cell in A'_i is a union of cells of A_{i+1} and $\rho'_i(v)$ is the cell containing $\rho_{i+1}(v)$. By our uniqueness assumption, A'_i, ρ'_i must equal A_i, ρ_i. Furthermore, there is a unique way to extend A_i, ρ_i to A_{i+1}, ρ_{i+1}, since all steps in the construction above are forced. Thus A_{i+1} and ρ_{i+1} constructed by the algorithm represent the unique arrangement compatible with the first $i+1$ columns of M. By induction we conclude A_n and ρ_n represent the unique arrangement compatible with M (up to mirror reflection, depending on the choice of A_2).

4.3. Non-Central Arrangements

Now we consider the case of non-central arrangements. The input to the algorithm is an $m \times n$ sign matrix M and we would like to find a set of arbitrary (possibly non-central) oriented lines $F = \{f_1 \ldots, f_n\}$ and viewpoints $V = \{v_1 \ldots, v_m\}$ such that v_i is on the positive side of f_j if and only if $M_{ij} = 1$, or (2) decide that this is not possible. Again the actual output is a cyclic ordering capturing the relative ordering of intersection points of F with the viewing circle, and an assignment of viewpoints to cells.

The algorithm works in three phases. First, we identify sets of features that must cross each other inside the viewing circle. We then construct sub-arrangements for sets of features that form "connected components" of the crossing

relationship. In the last phase we glue the sub-arrangements together to form a complete arrangement.

Phase 1 We start by constructing a graph G where the nodes are the features and there is an edge (i, j) if features i and j must cross each other inside the viewing circle. Two features must cross exactly when we see all four possible sign patterns on columns i and j of M. Let $\mathcal{S}$ be the connected components of G. While some pairs of features in $S \in \mathcal{S}$ may not necessarily cross each other, there is an ordering of the features in S such that each feature must cross a feature that appears before it.

Phase 2 In the second phase we construct a representation of all valid arrangements of features and viewpoints for each connected component $S \in \mathcal{S}$. These are the arrangements compatible with the columns of M indexed by S.

In the case of non-central features, a set of observations might not fully constrain the ordering of intersection points between S and the viewing circle. So we will want to partially specify orderings, $A_{|S|}$ of $\{i_1, i'_1, \ldots, i_{|S|}, i'_{|S|}\}$, so that the relative position of some sets of consecutive intersection points is arbitrary. This can be captured by an ordering of subsets of intersection points, where each intersection point appears in exactly one subset. For example, $(\{1, 2'\}, \{3\}, \{1'\}, \{2\}, \{3'\})$ specifies two possible orderings: $(1, 2', 3, 1', 2, 3)$ and $(2', 1, 3, 1', 2, 3)$. We may think of this construction as initially allowing multiple features to intersect the viewing sphere at the same point. To form a fully prescribed arrangement, we slightly perturb the features (in any relative order) to create unique intersection points between features and the viewing sphere. The cells created between these features will all be empty.

Let $(i_1, \ldots, i_{|S|})$ be the features in S in an order such that each feature crosses a feature that appears before it.

1. Let $A_2 = (\{i_1\}, \{i_2\}, \{i'_1\}, \{i'_2\})$. This decomposes the circle into four cells, according to the four possible sign configurations on the first two features. We define ρ_2 by assigning each sign vector to one of these cells according to its sign on f_{i_1} and f_{i_2}.

2. At the j-th iteration, we want to determine where to place i_j and i'_j in A_{j-1}. There must be a feature f_{i_k} with $k < j$ such that f_{i_j} must cross f_{i_k}. This feature defines two intervals in A_{j-1}: i_k to i'_k (where f_{i_k} is visible) and i'_k to i_k (where f_{i_k} is not visible). We know f_{i_j} must intersect the viewing circle in both intervals, so that we can obtain all possible sign patterns on the pair f_{i_k} and f_{i_j}. Consider the interval from i_k to i'_k. There must be sign vectors that are $+$ and $-$ on f_{i_j} within this interval. There should be a single transition between $+$ and $-$ in the interval, and it can occur within a cell or as we go from one cell to another (otherwise M is not a visibility matrix). If the transition is within the cell between intersection points a and b we update A_{j-1} by inserting $\{i_j\}$ or $\{i'_j\}$ between a and b. If the transition occurs as we go from the cell defined by a and b to the cell defined by b and c we update A_{j-1} by adding i_j or i'_j to the set b. This fully determines where i_j and i'_j must be inserted or leads to a contradiction, in which case M is not a visibility matrix. If there is no contradiction we obtain A_j and update ρ_{j-1} by reassigning viewpoints in cells which are split by f_{i_j} according to their sign on f_{i_j}.

Phase 3 In the last phase we "glue" together the arrangements constructed for each connected component of features to obtain a complete arrangement compatible with M.

For any pair of distinct connected components S and T, we can think of T as lying entirely inside a cell, $C_{S,T}$, induced by S. Namely all intersection points of features in T with the viewing circle lie inside a single cell induced by S. And conversely, we can think of S as lying entirely within a cell, $C_{T,S}$, induced by T. For each pair of connected components we will identify these connector cells.

We form a complete arrangement by inserting components one at a time, at each step placing a component adjacent to a component already inserted into the final arrangement. Unlike the central case, this final arrangement will be only one of many consistent arrangements. In order to decide which components may be adjacent to each other we check if two components are separated by a third using a simple criterion on connector cells.

1. Let $S_i, S_j \in \mathcal{S}$ be two connected components of G. Consider each along with its ordering A_{S_i} and mapping ρ_{S_i} as constructed in Phase 2. $(C_{S_i,S_j}, C_{S_j,S_i})$ is the pair of connector cells between components S_i and S_j iff the following two conditions hold:

$$C_{S_i,S_j} \supseteq \{v \,|\, \rho_{S_j}(v) \notin C_{S_j,S_i}\}$$

$$C_{S_j,S_i} \supseteq \{u \,|\, \rho_{S_i}(u) \notin C_{S_i,S_j}\}$$

   where C_{S_i,S_j} (C_{S_j,S_i}) is a cell induced by S_i (S_j) and u, v are rows of M.

2. A pair of connected components can be adjacent in the complete arrangement if there are no components separating them. That is, $\nexists\ S_k$ s.t. $C_{S_k,S_i} \neq C_{S_k,S_j}$. If such a S_k did exist, then S_i and S_j must lie inside distinct cells of S_k and thus are separated by at least one feature of S_k.

3. Let $\mathcal{O} = (S_{i_1}, \ldots, S_{i_{|\mathcal{S}|}})$ be the components in $\mathcal{S}$ in an order such that each component can be adjacent to at

least one component that appears before it (in the sense defined by step 2).

The final arrangement is represented by an ordering, $\mathcal{A}_{|\mathcal{S}|}$, of the intersection points between all features in M and the viewing circle. We form this ordering iteratively. Let $\mathcal{A}_1 = A_{i_1}$. At step k, we insert A_{i_k} into $\mathcal{A}_{k-1}$. By construction we know that S_{i_k} can be adjacent to some previously considered component, say S_{i_j}. We insert A_{i_k} immediately following the appropriate boundary of the connector cell $C_{S_{i_j},S_{i_k}}$.

Again we note that the arrangement induced by $\mathcal{A}_{|\mathcal{S}|}$ is not unique. For example, there are many choices for the order $\mathcal{O}$, and this can lead to different arrangements that are compatible with M.

5. Dealing with Noise or Modeling Errors

Of course the assumption that each object feature is visible on a disk of the viewing sphere will not always hold in practice. There are several sources of "error" that will lead to inconsistent observations. First, feature detection is a noisy process and we will not always detect exactly the set of features that are visible in an image. Second, due to self-occlusions some features may have visibility regions that are not disks. Moreover, the disk assumption may be violated if an object has multiple copies of the same feature.

Suppose we have m images of an object with n features. This will lead to an m by n sign matrix M, where each row specifies the features that are visible in one image. Because of noise and modeling error M will typically not be a visibility matrix. A natural approach for dealing with this is to look for a model that agrees with M as well as possible. In particular, we can look for an arrangement of features and viewpoints that lead to a visibility matrix M' such that the Hamming distance between M and M' is minimized. We define our problem as follows.

Given a sign matrix M, can we find viewpoints V and visibility regions F that lead to a visibility matrix M' such that the Hamming distance between M and M' is as small as possible?

5.1. Iterative Local Search Algorithm

Let $F = \{f_1 \ldots, f_n\}$, with $f_i = (h_i, t_i)$, be a set of half-spaces defining the visibility regions for each feature, and $V = \{v_1 \ldots, v_m\}$ denote a placement of viewpoints on the viewing sphere.

Note that F and V define a visibility matrix that agrees with M on entry (i, j) iff $M_{ij}(v_i^T h_j - t_j) > 0$. Thus the error of F and V with respect to M is given by

$$\text{error}_M(F, V) = \sum_{ij} I(M_{ij}(v_i^T h_j - t_j) < 0),$$

where I is a 0/1 indicator function (equals 1 if the argument is true, and 0 otherwise).

While the problem of finding the pair (F, V) minimizing $\text{error}_M(F, V)$ seems to be hard, we have found that good solutions are produced by a simple local search algorithm.

We start with random placements of viewpoints on the viewing sphere and repeatedly update one of F or V to minimize the total error while the other set of values is fixed. Each update leads to independent optimization problems for each feature or each viewpoint:

1. Given a fixed set of viewpoints V, pick $f_j = (h_j, t_j)$ to minimize $\sum_i I(M_{ij}(v_i^T h_j - t_j) < 0)$.

2. Given a fixed set of features F, pick v_i to minimize $\sum_j I(M_{ij}(v_i^T h_j - t_j) < 0)$.

Both steps (1) and (2) could be solved optimally when $d \in \{2, 3\}$, but we have found it more convenient, and in practice just as good, to use the heuristics described below.

Note that (1) is equivalent to finding a set of linear classifiers with minimum 0/1 loss. That is, for each j we have labeled points (v_i, M_{ij}) in $\mathbb{R}^d$ and we would like to find a linear threshold function (a half-space) f_j that minimizes the number of misclassified points. In practice we use a variant of the perceptron algorithm to solve this problem.

Subproblem (2) is similar to (1), but it cannot be solved in the same way due to the constraint that $||v_i|| = 1$. In principle, we can construct and search the arrangement induced by F in polynomial time to find optimal placements for v_i, but instead we randomly sample points in S^{d-1} and pick the best sample point for each v_i. When $d \in \{2, 3\}$ a relatively small number of sample points suffices to get a good solution.

5.2. Synthetic Data

Figure 3 shows the results of using the local search algorithm described above on synthetic data. We generate random problems as follows. We start with a random arrangement of features and viewpoints to obtain a visibility matrix A. We considered both the case of arbitrary viewpoints on the viewing sphere (the 3D case) and the case where viewpoints are selected from a circle of the viewing sphere (the 2D case). We think of each row of A as a *visibility vector* that indicates which features are visible in one view. We add independent noise to the entries of A to obtain a sign matrix B. We then run the local search algorithm to obtain a visibility matrix C that is close to B. Each graph in Figure 3 shows experiments with an increasing amount of noise per visibility vector for a fixed number of views and features. We can see that on average the number of bits we need to flip to turn B into a visibility matrix (the distance from B to C) is smaller than the amount of noise introduced. This is an indication that the algorithm is working well. Moreover,

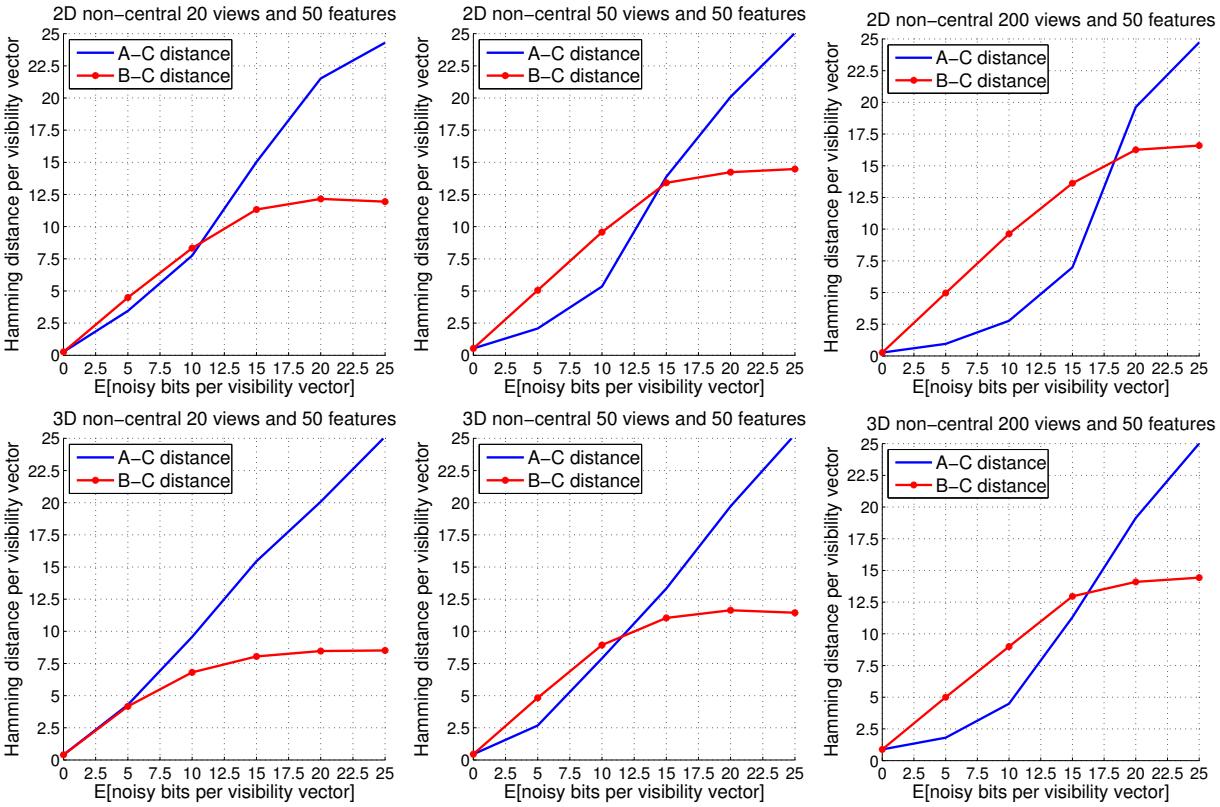

Figure 3. Experimental results with synthetic data. In each experiment we start with a set of compatible visibility vectors A and randomly flip each bit of each vector with some probability to get a set of (generally incompatible) visibility vectors B. We run our algorithm to obtain a set of vectors C that is similar to B but compatible. The distance from B to C indicates the number of bits we need to flip to get a set of compatible vectors. The distance from A to C indicates how well we are able to recover the initial set of visibility vectors.

if the amount of noise per vector is small, and the number of views is large, we obtain a matrix C that is close to the original visibility matrix A (see Figure 3).

6. Experiments with COIL images

We have also conducted experiments with images from the COIL dataset [2]. In each experiment we have 36 images of a single object taken from uniformly spaced viewpoints in a circle of the viewing sphere. We use half of the images to build a set of features and a visibility model. The feature set is constructed by clustering SIFT descriptors [1] detected in the 18 training images. We then use our local search algorithm to recover the visibility regions of each feature. Note that the visibility model is learned in an unsupervised manner since no image order information is used by the local search algorithm.

After building a visibility model we can place each of the 36 images, including those not used when building the model, in the visibility cell that best captures the features detected in that image. This lets us recover the relative viewpoint of each image. Figure 4 shows the results of this process on two different objects, where we display the images according to the order recovered by our algorithm. In both cases the order we recover is very good. There are only a few images that are slightly out of order. Note that if the same set of features are detected in two images it is impossible to distinguish among them using visibility constraints alone. In this case our algorithm will place the images in the same visibility cell in an arbitrary order.

In addition to recovering viewpoints the visibility model can also be used to find, and potentially correct, inconsistencies in the set of features detected in an image. Given a visibility model for an object we can place an image of the object in the visibility cell which best captures the features detected in that image. We can then check which detected features should not have been visible, and which features should have been visible but were not detected. This could be used to correct errors in the feature detection process. Figure 5 shows an example of this for a feature that intuitively corresponds to the back of the cat. The visibility model correctly identifies an image where the feature was not detected even though the back of the cat is visible, and several images where the feature was detected even though the back is not visible.

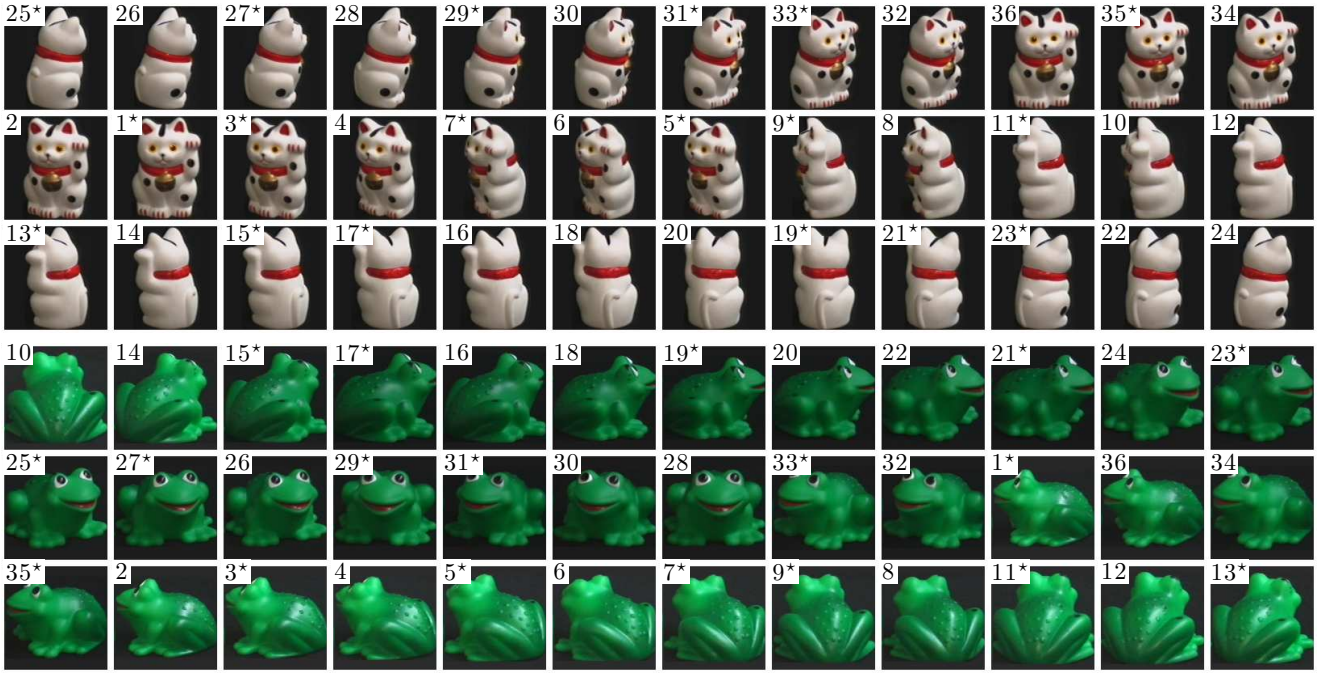

Figure 4. Experiments with objects from the COIL dataset. The label in the top left corner indicates the true viewpoint of each image. The order shown was recovered using our algorithm by building a visibility model from a subset of the images (labeled by $\star$), and then placing each image in the visibility cell which best matched it.

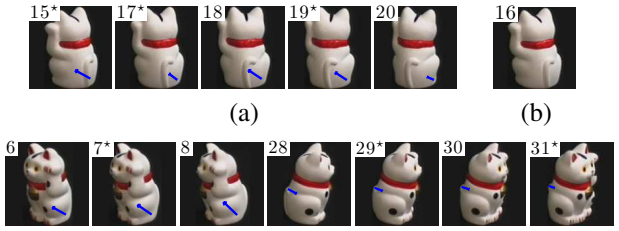

Figure 5. (a) and (c) show images where a particular feature (blue arrow) was detected. The visibility model learned for this object classifies the detections in (a) as correct, and the detections in (c) as inconsistent (the feature should not appear together with the other features detected in these images). The feature was not detected in (b) but should have been according to the model.

References

[1] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, vol. 60, no. 2, 2004.

[2] S. Nene, S. Nayar, H. Murase. Columbia Object Image Library. Columbia University, TR CUCS-006-96, 1996.

[3] A. Bjorner, M. Las Vergnas, B. Sturmfels, N. White, G. Ziegler. Oriented Matroids. Encyclopedia of mathematics, vol. 46, Cambridge University Press, 1993.

[4] Z. Gigus, J. Canny, R. Seidel, Efficiently computing and representing aspect graphs of polyhedral objects. PAMI, vol. 13, no. 6, 1991.

[5] K. Grauman, T. Darrel. Pyramid match kernels: discriminative classification with sets of image features. ICCV, 2005.

[6] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision. 2nd edition, Cambridge University Press, 2004.

[7] N. Linial, S. Mendelson, G. Schechtman, A. Shraibman. Complexity measures of sign matrices. Combinatorica, vol. 27, 2007.

[8] N.E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, ed., Topology and Geometry Rohlin Seminar, Vol. 1346, Lecture Notes in Math., Springer-Verlag, Berlin/Heidelberg, 1988.

[9] H. Plantinga, C. Dyer. Visibility, occlusion, and the aspect graph. IJCV, vol. 5, no. 2, 1990.

[10] A.A. Razborov, A.A. Sherstov. The sign-rank of AC0. FOCS, 2008.

[11] P. Shor. Stretchability of Pseudolines is NP-hard. DIMACS series in discrete mathematics and theoretical computer science, vol. 4, 1991.

[12] N. Srebro, N. Alon and T. Jaakkola, Generalization error bounds for collaborative prediction with low-rank matrices. NIPS, 2005.

[13] S. Ullman and R. Basri. Recognition by linear combination of models. IEEE Trans. PAMI, vol. 13, no. 10, 1991.

[14] C. Wallraven, B. Caputo, A. Graf. Recognition with local features: the kernel recipe. ICCV, 2003.