Reconstructing convex polygons and convex polyhedra from edge and face counts in orthogonal projections*

THERESE BIEDL†

Cheriton School of Computer Science
University of Waterloo, Ontario N2L 3G1, Canada
biedl@uwaterloo.ca

MASUD HASAN‡

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh
masudhasan@cse.buet.ac.bd

ALEJANDRO LÓPEZ-ORTIZ†

Cheriton School of Computer Science
University of Waterloo, Ontario N2L 3G1, Canada
alopez-o@uwaterloo.ca

We study the problem of reconstructing convex polygons and convex polyhedra given the number of visible edges and visible faces in some orthogonal projections. In 2D, we find necessary and sufficient conditions for the existence of a feasible polygon of size N and give an algorithm to construct one, if it exists. When N is not known, we give an algorithm to find the maximum and minimum size of a feasible polygon. In 3D, when the directions are covered by a single plane we show that a feasible polyhedron can be constructed from a feasible polygon. We also give an algorithm to construct a feasible polyhedron when the directions are covered by two planes. Finally, we show that the problem becomes NP-hard when the directions are covered by three or more planes.

Keywords: Polygon and polyhedra reconstruction; orthogonal projection; edge-colored planar graph; independent set; NP-hardness.

1. Introduction

Reconstructing polyhedra from projection information is an important field of research due to its applications in geometric modeling, computer vision, geometric tomography, and computer graphics. There are various sources of information from which polyhedra could be reconstructed, such as triangulations, line drawings, silhouettes, area/volume/shape of shadows, shading, texture, and reflection of light,

*A preliminary version appeared in Proc. FSTTCS 2007, LNCS 5431, Springer, 2007, pp. 400-411.
†Supported by NSERC.
‡Research done while the author was a PhD student at the University of Waterloo, Canada.

among others. Observe that in general a polyhedron may not be uniquely determined by the projection information.

Marlin and Toussaint [17] studied the problem of reconstructing convex polyhedra from triangulations of the shadow boundary. Given a convex polygon and its two distinct triangulations T_1 and T_2, they gave an $O(n^2)$-time algorithm for deciding whether a polyhedron exists for which the two opposite projections from the z-axis are T_1 and T_2, and constructing a polyhedron where possible. When T_1 and T_2 are isomorphic, Bereg [2] showed that the polyhedron can always be reconstructed. See Demaine and Erickson [6] for a collection of similar problems on reconstruction of polyhedra.

The problem of reconstructing polyhedra has been studied under specific application scenarios, where various types of projection information may be available. Among those, line drawings [15,16,19,20,23,24,27,28] are possibly the most common. Line drawings may be obtained from images, from geometric drawings of the designers [24] or may be freehand drawings [14,26]. The reconstruction algorithms differ for a single and multiple drawings. For multiple drawings there are two common approaches based on the representation of the polyhedra that are to be constructed: constructive solid geometry and boundary representation. Both approaches are used in engineering and product design such as designing complex mechanical parts and in CAD [12,27]. Comparatively, it is more difficult to reconstruct a polyhedron from a single drawing [24,27].

Reconstructing a polyhedron given the area and shape of projections has been considered in geometric tomography [8]. A related but more application oriented area of research is discrete tomography [10,11], where 2D and 3D objects are reconstructed from sectioning information such as the area of a plane section of the objects. Medical CAT scanning is an important application of discrete tomography where an image of the human body is reconstructed from X-ray information. The X-ray image gives the projection and thickness of different parts of an object.

Instead of whole projections, sometimes only silhouettes are used to reconstruct polyhedra [4,5,13,18]. In volume intersection, which is a well-known technique in computer vision, the only information available is a set of silhouettes [4,5,13], sometimes even with unknown view points [4,5].

1.1. *Our results*

Most reconstruction algorithms are based on fairly complex information such as triangulations, line drawings, silhouettes, and geometric measures of the projections, along with some non-geometric surface information such as shading, texture, and reflection of light. In contrast, we consider a very different and very limited type of information: we consider the number of visible edges for (2D) polygons and the number of visible faces for (3D) polyhedra in some orthogonal projections. Here we study reconstructing convex polygons and convex polyhedra from orthogonal projections only.

We only consider non-degenerate orthogonal projections where the view directions are not parallel to the edges (faces) of the polygon (polyhedron). A *direction-integer pair*, or simply a *d-n pair*, $\langle d, n \rangle$ consists of a direction vector d and a positive integer n, and expresses how many edges (faces) should be seen from the direction. A *d-n set* $\mathcal{R}$ is a set of d-n pairs where no two directions are the same or opposite to each other. (We assume this because we will ultimately generate and then use the d-n pairs for all opposite directions too. See also Section 2.) A convex polygon (polyhedron) P is *feasible* for $\mathcal{R}$ if, for each d-n pair $\langle d, n \rangle$ in $\mathcal{R}$, d is not parallel to edges (faces) of P and the number of visible edges (faces) from d is n. For a d-n set, a feasible polygon may or may not exist or it may exist for more than one size (i.e., number of edges) (see Fig. 1.)

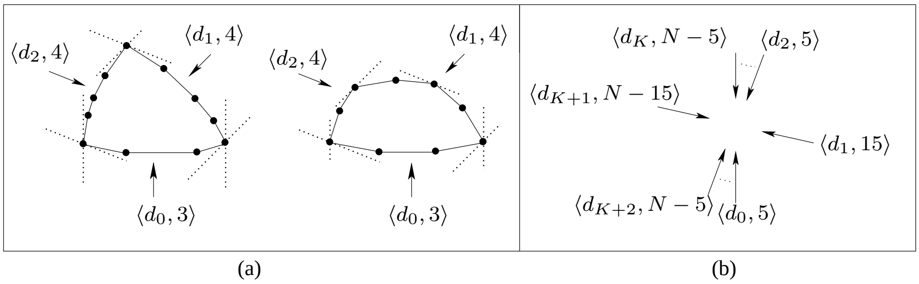


Fig. 1. (a) A d-n set with two feasible polygons of different size. (b) A proper d-n set containing these directions cannot have a feasible polygon. Also see Section 3, Page 5.

In this paper, we consider the problem of given a d-n set $\mathcal{R}$ and an integer N, create a feasible polygon (polyhedron) of size N for $\mathcal{R}$. We first give necessary and sufficient conditions for a feasible polygon to exist, which also gives an algorithm to construct the polygon, if it exists. With K directions, our algorithm runs in $O(K + N)$ time if $\mathcal{R}$ is ordered by direction vector, and in $O(K \log K + N)$ time otherwise. For unknown N, the above characterization gives an $O(K + v \log v)$-time algorithm to find the maximum and minimum size of a feasible polygon, where $1 \leq v \leq K$ is some parameter explained in detail later.

In 3D, we consider cases by the minimum number of planes that *cover* the directions, where “covering” means each direction lies in at least one plane. For directions covered by one plane, 2D results are easily transferred. For two planes, we give an algorithm to construct a feasible polyhedron, whenever it exists, except for one particular case. Finally, for three or more planes, we prove that testing the existence of a feasible polyhedron is NP-hard. (Throughout the paper we assume that the model of computation is the real RAM.)

For non-convex polygons and polyhedra, it is not straight-forward to formulate the reconstruction problem studied in this paper. This is because the definition of visible edges and faces are not straightforward as they may be partially visible or may not be visible at all. This also makes it impossible to relate the size of a feasible non-convex polygon/polyhedron to the visible edge/face counts from the given directions. The second author [9] studied reconstructing non-convex polygons in a special case where a partially visible edge is considered as invisible, the size of the polygon is not given, and each direction sees at least a certain number of edges.

Similar situation happens for perspective projections too, where simply constructing convex feasible polygons is not enough, because the feasibility of a convex polygon depends upon its position and size too [9]. More clearly, for orthogonal projections a feasible convex polygon is invariant to translation and scaling. But that is not true for perspective projections, since the visibility of a polygon from a view point differs if the size of the polygon is scaled up or down and if the polygon is translated. The second author [9] also studied the reconstruction of a convex polygon from perspective projections when the view point are in convex position and showed that it is always possible to construct a feasible polygon for this case.

Our reason for avoiding degenerate projections is that when an edge or a face of a convex polygon/polyhedron becomes parallel to a view direction, its visibility is not clearly defined, i.e., it can be counted as either visible or invisible, which makes it difficult to define the problem in terms of a given d-n set.

1.2. *Impact*

Our algorithm to test the feasibility of reconstruction can be useful as a preliminary step in applications in which other types of information is available and can be used for reconstruction purposes—the user can decide quickly the existence of possible resulting polyhedra before starting a rigorous reconstruction process.

Although from the applications point of view the problem of reconstructing polyhedra is more common than that of reconstructing polygons, surprisingly, the latter are themselves very rich and their solution techniques will serve as a foundation for solving the former.

2. Preliminaries

Recall that the input to our problem is a d-n set $\mathcal{R}$, and, sometimes, the size N of the convex polygon/polyhedron to be reconstructed. Observe that we have $N \geq 3$ in 2D and $N \geq 4$ in 3D, and N must be strictly larger than any integer of a d-n pair since at least one edge/face is visible from the opposite direction of any view direction. We assume this throughout.

Although our problems are defined in terms of a d-n set $\mathcal{R}$ having K d-n pairs, we will use a *proper d-n set* $\mathcal{S}$ which has $2K$ d-n pairs and is derived from $\mathcal{R}$ and N as follows: For each d-n pair $\langle d, n\rangle$ in $\mathcal{R}$, $\mathcal{S}$ has both $\langle d, n\rangle$ and $\langle d', N-n\rangle$, where d' is opposite to d. The d-n pairs $\langle d, n\rangle$ and $\langle d', N-n\rangle$ in $\mathcal{S}$ are called *opposite* to each other. Clearly, a convex polygon (polyhedron) P with N edges (faces) is feasible for $\mathcal{R}$ if and only if it is feasible for $\mathcal{S}$.

When the directions of $\mathcal{S}$ lie in one plane, $\mathcal{S}$ is represented as $\mathcal{S} = \{\langle d_0, n_0\rangle, \langle d_1, n_1\rangle, \ldots, \langle d_{2K-1}, n_{2K-1}\rangle\}$, where the d-n pairs are ordered by the counter-clockwise ordering of the directions. From now on indices of the terms related to $\mathcal{S}$ are taken modulo $2K$.

3. Reconstructing polygons

We first study the 2D case. Let P be a feasible polygon of size N for $\mathcal{S}$ and consider the sets of visible edges of P from the directions of $\mathcal{S}$. We shall perform a complete circular rotation of the view direction around P and count how the visibility of the edges change. During this rotation, while moving from direction d_i to d_{i+1}, some edges of P may become *newly visible*, i.e., become visible from invisible, and/or *newly invisible* to d_{i+1}. From n_i and n_{i+1} alone, it cannot be said exactly how many edges become newly visible or invisible to d_{i+1}. However, it is possible to lower bound these quantities. Observe that if an edge e becomes newly visible when going from d_i to d_{i+1}, then it becomes newly invisible when going from d_{i+K} to d_{i+K+1}. This implies that over the complete rotation, although the change in the visibility of each edge happens twice, the total change in the visibility of all edges can be counted by only considering their change from invisible to visible. (This use of opposite directions is the main motivation for considering $\mathcal{S}$ instead of $\mathcal{R}$.) Moreover, e is newly visible for exactly one direction of $\mathcal{S}$.

We now state the characterization formally. For each i, define $\delta_i = \max\{0, n_{i+1} - n_i\}$. We call δ_i the *i-th view difference*. Note that for any i, at least one of δ_i and δ_{i+K} is zero. There must be at least δ_i edges that become newly visible while moving from d_i to d_{i+1}. Therefore, if a feasible polygon for $\mathcal{S}$ exists, then $D := \sum_{i=0}^{2K-1} \delta_i \leq N$.

This can be used to show that some d-n sets do not permit a solution. For example, for any $\mathcal{S}$ having the three consecutive d-n pairs $\langle d_0, 5\rangle$,$\langle d_1, 15\rangle$,$\langle d_2, 5\rangle$ of Fig. 1(b), we have $\delta_0 = 10$, $\delta_1 = 0$, $\delta_K = 0$, and $\delta_{K+1} = 10$. But we also have $n_K = N - n_0 = N - 5$, and $\sum_{i=2}^{K-1} \delta_i \geq \sum_{i=2}^{K-1}(n_{i+1} - n_i) = n_K - n_2 = (N-5) - 5 = N - 10$. Since $\delta_i \geq 0$ for any i, we have $D = \sum_{i=0}^{2K-1} \delta_i = \delta_0 + \delta_1 + \sum_{i=2}^{K-1} \delta_i + \delta_K + \delta_{K+1} + \sum_{i=K+2}^{2K-1} \delta_i \geq 10 + 0 + (N-10) + 0 + 10 + \sum_{i=K+2}^{2K-1} \delta_i \geq N + 10$. Therefore, there is no feasible polygon for any N. Our main result here is that this argument works for all infeasible d-n sets, i.e., that this necessary condition is also sufficient. In particular, our first main result is the following theorem.

Theorem 1. *Given a proper d-n set $\mathcal{S}$ with all directions in one plane, and an integer N, a feasible convex polygon P of size N exists if and only if $D := \sum_{i=0}^{2K-1} \delta_i \leq N$.*

Before we prove this theorem, we will give an important lemma that will be used in the proof as well as in the rest of the paper.

Lemma 1. *For any i, $N - D = 2(n_i - \sum_{j=i+K}^{i+2K-1} \delta_j)$.*

Proof. We first observe that $n_{j+1} - n_j = \delta_j - \delta_{j+K}$, since $\delta_j = \max\{0, n_{j+1} - n_j\}$

and $\delta_{j+K} = \max\{0, n_{j+K+1} - n_{j+K}\} = \max\{0, (N-n_{j+1})-(N-n_j)\} = \max\{0, n_j - n_{j+1}\}$. Applying this K times gives $n_{i+K} - n_i = \sum_{j=i}^{i+K-1} \delta_j - \sum_{j=i+K}^{i+2K-1} \delta_j$. Using $n_{i+K} = N - n_i$ and subtracting $D = \sum_{j=0}^{2K-1} \delta_j$, this becomes $N - 2n_i - D = -2\sum_{j=i+K}^{i+2K-1} \delta_j$ as desired. □

We now prove Theorem 1.

Proof of Theorem 1. The idea is as follows. For each view direction d_i, choose δ_i edges, if $\delta_i > 0$, such that they are newly visible for d_{i+1}. The remaining $N - D$ edges are chosen in parallel pairs so that one becomes visible exactly when the other becomes invisible. Then by Lemma 1, d_i sees n_i edges.

Now we come to the exact details. To avoid constructing an unbounded polygon we have to be careful in how to chose edges. To simplify the description, we will not choose edges directly, and instead choose a normal point for each edge on a circle c centered at the origin o. The *normal point* of an edge e of P is the point of c that uniquely represents the outward normal vector of e, i.e., the intersection of the outward normal vector of e, when translated to the origin, and c. From these normal points, we can then reconstruct a polygon by computing the intersection of their tangent half-planes in $O(N)$ time if they are ordered.

For any direction d_i, denote by h_i the *visible half circle of* d_i, i.e., the (closed) half circle of c that is visible from d_i. Clearly, an edge e is visible from d_i if and only if e's normal point is strictly within h_i. Moreover, a polygon defined by normal points is bounded if and only if not all normal points are within a single open half circle.

The circular arc $\theta_i = h_{i+1} \backslash h_i$ is called the *i-th d-arc* ("d" for difference). Normal points in θ_i correspond to edges that are newly visible to d_{i+1}. Normal points will never be placed on the boundary of θ_i, and hence we will not distinguish as to whether θ_i is open or closed. Observe that θ_i and θ_{i+K} are the reflection of each other with respect to the origin and are called *opposite* to each other. Since d_i and d_{i+K} are opposite directions, we have $\bigcup_{j=i+K}^{i-1} \theta_j = h_i$ for all i. (See also Fig. 2(a)).

Now place δ_i arbitrary normal points strictly within each θ_i. If $N = D$, then we have no more normal points left. If $D < N$, then by Lemma 1, $N - D$ is even. Select $N - D - 2$ additional normal points in antipodal pairs arbitrarily (but not on end-points of any θ_i). The last two normal points p_1 and p_2 are placed within two opposite d-arcs, but chosen carefully as follows such that no half circle contains all normal points. Let p be one among the already selected normal points and let θ be the d-arc containing p. Place p_1 somewhere between p and the boundary of θ that is clockwise next to p. Let ε be the (circular) distance of p_1 from p. Let p' be the antipodal point of p. Place p_2 at the clockwise distance of $\varepsilon/2$ from p'. Clearly, p_1 and p_2 are within θ and its opposite d-arc respectively. See Fig. 2(b). Moreover, no half circle can contain all of p, p_1 and p_2.

Recall that $h_i = \bigcup_{j=i+K}^{i+2K-1} \theta_j$. The number of normal points strictly within h_i is hence $\sum_{j=i+K}^{i+2K-1} \delta_j + \frac{1}{2}(N - D)$, because d-arc θ_j initially gets δ_j normal points and

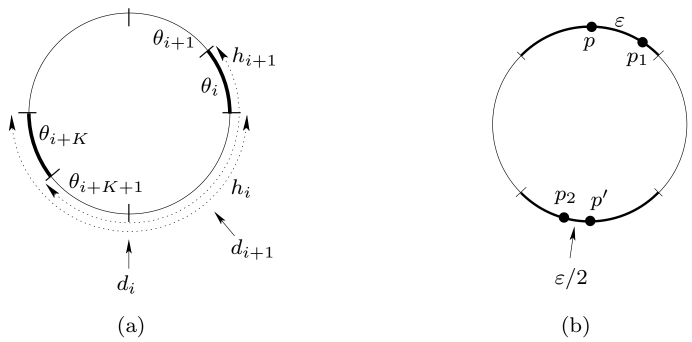


Fig. 2. (a) d-arcs and visible half circles. Two opposite d-arcs are in bold. $h_i = \bigcup_{j=i+K}^{i+2K-1} \theta_j$. (b) Selecting the last two normal points when $D < N$.

then exactly half of the additional $N - D$ points are placed within the half circle h_i. By Lemma 1, therefore, h_i contains n_i normal points as desired.

All that remains to show is that no half circle contains all normal points. This was already guaranteed if $D < N$, since the last two normal points p_1 and p_2 were chosen carefully. If $N = D$, then each d-arc θ_i gets exactly δ_i normal points. Any open half circle h contains $K - 1$ consecutive d-arcs, and let them be $\{\theta_j\}$, for $j = i+1, i+2, \ldots, i+K-1$. We claim that at least one of these d-arcs has positive view difference. Suppose not. Then $\sum_{j=i+1}^{i+K-1} \delta_j = 0$. But remember that at least one of δ_i and δ_{i+K} is 0, say $\delta_i = 0$. So, we have $\sum_{j=i}^{i+K-1} \delta_j = 0$. That means, $n_{i+K} = \sum_{j=i}^{i+K-1} \delta_j + \frac{1}{2}(N - D) = 0 + 0 = 0$, which by Lemma 1 means $n_i = N$. This is a contradiction to our assumption on the input that $N > n_i$ for any *d-n* pair $\langle d_i, n_i \rangle$. □

The above proof is algorithmic (constructive), and it is straightforward how to implement it. For the running time, if the d-arcs are not ordered then we need to sort them in $O(K \log K)$ time. Then, during the selection of normal points within the d-arcs, we can also maintain their ordering in a total of $O(N)$ time. Finally, we can construct P from these ordered normal points in $O(N)$ time. So the total time required is $O(N+K)$ if $\mathcal{S}$ is ordered, and $O(N+K \log K)$ otherwise. We summarize:

Theorem 2. *Given a d-n set $\mathcal{R}$ of size K with all directions in one plane, and given an integer N, a feasible convex polygon P with N edges can be computed, whenever it exists, in $O(N+K)$ time if $\mathcal{R}$ is ordered, or in $O(N+K\log K)$ time otherwise.*

3.1. *Maximum and minimum size convex polygon*

Using Theorem 1, we can also find out whether there exists a feasible polygon for a given *d-n* set $\mathcal{R}$ if N is unknown. In fact, we find both the maximum and minimum size of a feasible polygon. Observe that if $\mathcal{R}$ contains two opposite *d*-*n* pairs, then the sum of the two corresponding integers would give the value of N. Hence, once again it is assumed that no opposite *d*-*n* pair appears in $\mathcal{R}$.

Our algorithm is as follows. We compute as before a proper *d*-*n* set $\mathcal{S}(N)$ from $\mathcal{R}$, but this time the *d*-*n* pairs of $\mathcal{S}(N)$ will be functions of N—for each pair $\langle d, n\rangle$ in $\mathcal{R}$, the opposite pair $\langle d', N-n\rangle$ in $\mathcal{S}(N)$ contains the unknown N. Then we compute $\delta_i(N)$ and $D(N)$, which also become functions of N. Recall from Theorem 1 that a feasible polygon exists if and only if $D(N) \leq N$. Also remember that $\delta_i(N) = \max\{0, n_{i+1} - n_i\}$. For each δ_i, replacing n_i by $N - n_{i+K}$ if n_i is derived and n_{i+1} by $N - n_{i+K+1}$ if n_{i+1} is derived yields δ_i to be one of the following four in terms of N:

- $\delta_i(N) = \max\{0, n_{i+1} - n_i\}$, or
- $\delta_i(N) = \max\{0, n_{i+K} - n_{i+K+1}\}$, or
- $\delta_i(N) = \max\{0, N - (n_{i+K+1} + n_i)\}$, or
- $\delta_i(N) = \max\{0, n_{i+1} + n_{i+K} - N\}$

So, $\delta_i(N)$ is either constant or increasing/decreasing with slope ± 1. Moreover, δ_i is increasing/decreasing if and only if δ_{i+K} is decreasing/increasing, and the two meet (with $\delta_i(N) = 0$) at a place well-defined in terms of n_i and n_{i+1}. We call such a meeting place a *valley*. Hence the function $D(N)$, which is the sum of all $\delta_i(N)$, is convex and piecewise linear. Also see Fig. 3. So $D(N) = N$ has at most two solutions, and any N between them is feasible as long as $N \geq 3$ and $N > \max_i\{n_i\}$. (Observe that since some δ_i's keep increasing with the increase of N, $D(N)$ can be bigger than N. This is in contrast with Theorem 1 where N has no upper bound for fixed δ's.) The algorithm to compute this range of N will need, other than computing $D(N)$, to sort the valleys. So the algorithm takes $O(K + v \log v)$ time, where v is the number of valleys. Of course $v \in O(K)$, but v could be as small as one if all directions in $\mathcal{R}$ are spanned within a half-plane.

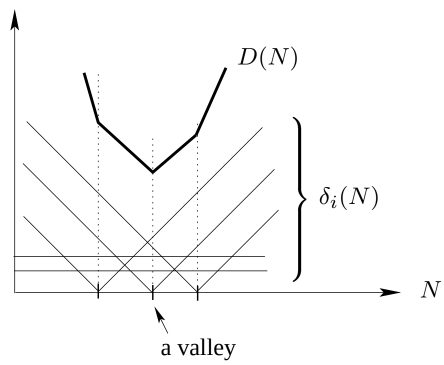


Fig. 3. δ_i and $D(N)$ against unknown N.

Theorem 3. *Given an ordered d-n set $\mathcal{R}$ of size K with all directions in one plane, the maximum and minimum size of a feasible convex polygon can be computed in $O(K + v \log v)$ time, where the number of valleys v is at most K and can be as small as one. If $\mathcal{R}$ is not ordered, then the algorithm takes $O(K \log K)$ time.*

4. Reconstructing polyhedra

Similar to 2D, in order to construct a feasible polyhedron P in 3D, we will compute the proper *d-n* set $\mathcal{S}$ from the given *d-n* set $\mathcal{R}$ and instead of choosing faces directly we will choose them implicitly by choosing normal points of the faces on the surface of an origin-centered sphere s. (The definition of the normal point of a face is analogous to that of an edge in 2D.) Then given such normal points, we can compute a polyhedron from them by computing the intersection of their tangent half-spaces in $O(N \log N)$ time [7,21].

A face f is visible from a direction d_i if and only if its normal point is strictly within the *visible hemisphere* h_i of d_i. Moreover, P is bounded if and only if not all normal points intersect a single open hemisphere.

4.1. *Directions covered by a single plane*

If all directions are in one plane, a solution to the 2D case implies an open cylinder in 3D which can easily be converted to a solution to the 3D case. The other direction is slightly less trivial; the following theorem gives a precise proof.

Theorem 4. *Given an ordered proper d-n set $\mathcal{S}$ of size $2K$, where all the directions lie in one plane π, and given $N \geq 4$, there exists a feasible convex polyhedron P of size N for $\mathcal{S}$ if and only if there exists a feasible polygon P' of size N for $\mathcal{S}$, interpreted as 2D directions within π. Moreover, constructing both P from P' and P' from P require $O(N \log N)$ time.*

Before giving the proof, we need to introduce some notation, which will be used in later sections as well. Given a proper *d-n* set $\mathcal{S}$ with directions in one plane and ordered counter-clockwise, define the *ith d-lune* to be $\theta_i = h_{i+1} \backslash h_i$. See Fig. 4(a). As in 2D, $h_i = \bigcup_{j=i+K}^{i+2K-1} \theta_j$. All d-lunes of $\mathcal{S}$ have two common antipodal points which are called the *poles* of $\mathcal{S}$.

Proof. Let c be the great circle of s corresponding to the plane π. Assume first that the polygon P' exists. Each edge of P' then corresponds to a normal point in c. At this stage if we create a polyhedron P with these normal points, then it would be a cylinder with two ends unbounded. To make it bounded, we move two of these normal points towards the two poles of $\mathcal{S}$, respectively, but within their respective d-lunes. This still remains a solution to the *d-n* set $\mathcal{S}$. See Fig. 4(b). Finally, we construct P by taking for each normal point the tangent plane at it, taking the halfspace of the plane that contains s, and then computing the intersection of these halfspaces. This takes $O(N \log N)$ time [7,21].

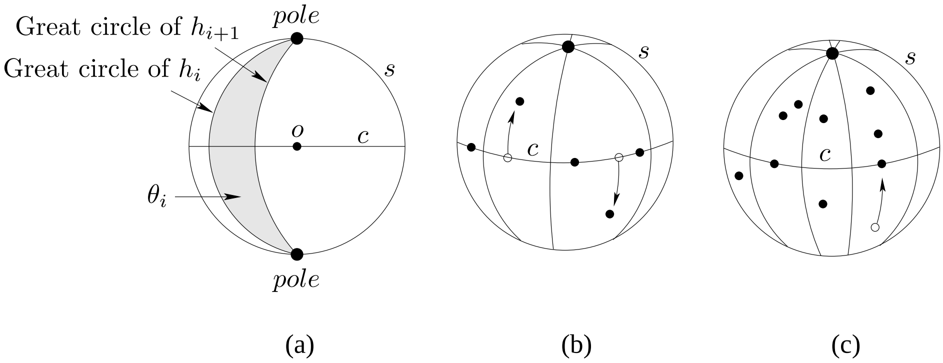


Fig. 4. (a) d-lunes. (b) P from p. (c) p from P.

Now assume a polyhedron P for the 3D problem exists. Each face of P then corresponds to a normal point on s. Move each of these points onto c along the great circle through the point and the poles, using the shorter arc. See Fig. 4(c). If two points overlap on c, then slightly move one of them on c but within their respective d-lunes. Now we have N distinct normal points within a plane. We find their ordering along c by sorting them in $O(N \log N)$ time. Then we construct the convex polygon from these ordered normal points in $O(N)$ time. □

4.2. *Directions covered by two planes*

Now we consider the case when all view directions are covered by two planes $\overline{\pi}$ and $\widetilde{\pi}$. The *d-n* set $\mathcal{S}$ hence gets split into two *d-n* sets $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$, depending on which plane each direction belongs to. (One pair of opposite directions can belong to both planes.) We assume that for each of $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ the *d-n* pairs are numbered counter-clockwise (within their planes). This then also defines d-lunes $\overline{\theta}_i$ and $\widetilde{\theta}_j$ and view differences $\overline{\delta}_i$ and $\widetilde{\delta}_i$ as before. All indices are taken modulo $2\overline{K} := |\overline{\mathcal{S}}|$ and $2\widetilde{K} := |\widetilde{\mathcal{S}}|$. We set $\overline{D} = \sum_{i=0}^{2\overline{K}-1} \overline{\delta}_i$ and $\widetilde{D} = \sum_{j=0}^{2\widetilde{K}-1} \widetilde{\delta}_j$ as before. Moreover, since a feasible polyhedron is also feasible for each of the two planes, by Theorem 4 and Theorem 1 we can assume that $\overline{D}, \widetilde{D} \leq N$ and that Lemma 1 holds for both planes.

We assume the numbering is such that $\overline{d}_0 = \widetilde{d}_0$ if the two sets $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ have a common direction. So, the great circle of $\overline{h}_0$ and $\overline{h}_{\overline{K}}$ passes through the poles of $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$. (See Fig. 5(a).) On the other hand, if $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ have no common direction then the two poles of $\overline{\mathcal{S}}$ (resp. $\widetilde{\mathcal{S}}$) are strictly within two opposite d-lunes $\widetilde{\theta}_0$ and $\widetilde{\theta}_{\overline{K}}$ of $\widetilde{\mathcal{S}}$ (resp. $\overline{\theta}_0$ and $\overline{\theta}_{\widetilde{K}}$ of $\overline{\mathcal{S}}$). (See Fig. 5(b).) Intersecting the two sets of lunes splits the sphere s into a grid-like structure, except near the poles if $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ have no direction in common.

Let $\theta_{a,b} = \overline{\theta}_a \cap \widetilde{\theta}_b$; this is a spherical polygon called *d-polygon*, and the union of the d-polygons covers the sphere s. If $\Delta_{a,b}$ is the number of normal points that will be finally assigned to $\theta_{a,b}$, then the following must hold:

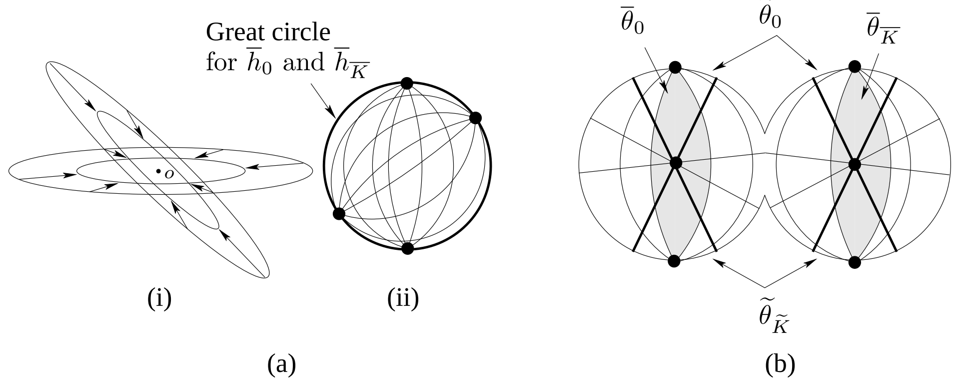


Fig. 5. (a) (i) Two planes of $\mathcal{S}$ with common directions, and (ii) arrangement of the d-lunes for such $\mathcal{S}$. (b) Two opposite views of the sphere s (the back view is rotated to front and is drawn in the right) showing the arrangement of d-lunes if $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ have no direction in common.

- $\sum_j \Delta_{i,j} \geq \bar{\delta}_i$ for all $0 \leq i < 2\overline{K}$,
- $\sum_i \Delta_{i,j} \geq \widetilde{\delta}_j$ for all $0 \leq j < 2\widetilde{K}$,
- $\sum_{\ell=i+\overline{K}}^{i-1} \sum_j \Delta_{\ell,j} = \overline{n}_i$ for all $0 \leq i < 2\overline{K}$,
- $\sum_{\ell=j+\widetilde{K}}^{j-1} \sum_i \Delta_{i,\ell} = \widetilde{n}_j$ for all $0 \leq j < 2\widetilde{K}$,

where the unspecified sums run over all indices for which $\Delta_{a,b}$ exists, i.e., the two respective d-lunes intersect. (Observe that the inequality in the first two conditions is possible, because the view difference δ of a d-lune θ can be much smaller than the number of normal points that θ can finally have in it.) Satisfying these four conditions will be called the *valid assignment problem*. It is quite similar to the Edmond's transportation problem studied in many linear programming textbooks, see for example Bazaraa et.al. [1], and it is not difficult to develop an algorithm to find a valid assignment if one exists.

But before that we need some more preliminaries: A crucial ingredient for finding a valid assignment is to use a *matrix assignment problem* defined as follows: Let M be a matrix of m rows and n columns. Let $R_1, \ldots, R_m$ and $C_1, \ldots, C_n$ be non-negative integers with $\sum_{i=1}^{m} R_i \leq \sum_{j=1}^{n} C_j$. R_i (similarly C_j) indicates the ith *row sum*, i.e., the sum of the elements of the ith row, (similarly the jth *column sum*) of M. We want to assign the entries $(M_{i,j})_{1\leq i\leq m, 1\leq j\leq n}$ with non-negative integers such that $\sum_{j=1}^{n} M_{i,j} = R_i$ and $\sum_{i=1}^{m} M_{i,j} \leq C_j$. With a simple greedy-algorithm we can show the following:

Lemma 2. *Any matrix assignment problem has a solution, and we can find the positive integers of such a solution in $O(m+n)$ time.*

Proof. The following pseudo code can do the job.

Procedure **matrixFilling**$([R_1, \ldots, R_m,\ C_1, \ldots, C_n])$

Input: Non-negative R_i's and C_j's with $\sum_{i=1}^{m} R_i \leq \sum_{j=1}^{n} C_j$.
Output: A matrix $M_{i,j}$ with $\sum_{j=1}^{n} M_{i,j} = R_i$ and $\sum_{i=1}^{m} M_{i,j} \leq C_j$.

```
  // Assume M_{i,j} is initialized to all-zero.
  if m > 0 and n > 0 then
    set M_{m,n} = min{R_m, C_n}
    if M_{m,n} = R_m then
      matrixFilling([R_1, ..., R_{m-1}], [C_1, ..., C_{n-1}, C_n - R_m])
    else
      matrixFilling([R_1, ..., R_{m-1}, R_m - C_n], [C_1, ..., C_{n-1}])
    end if
  end if
```

Clearly, in every iteration one row or column is removed, so the run-time (apart from initializing $M_{i,j}$) is $O(m+n)$ and there are at most $m+n$ non-zero matrix entries.

To see the correctness, assume first that $m = 1$ and $n = 1$. Then $\min\{R_m, C_n\} = R_m$ by $\sum_{i=1}^{m} R_i \leq \sum_{j=1}^{n} C_j$, and so the matrix with single entry $M_{m,n} = R_m$ satisfies the condition, and our algorithm computes exactly that. Now if $m > 1$ or $n > 1$, then $M_{m,n} = \min\{R_m, C_n\}$ satisfies the sum for the row/column that is removed for the recursive call. Moreover, all integers passed into the recursive call are non-negative, and the row-sum is no more than the column-sum since we removed $M_{m,n}$ from both sums. So by induction the recursively obtained solution satisfies the conditions for all other rows/columns. □

We now turn to finding a valid assignment. In 2D, we had a simple condition (Theorem 1) that was necessary and sufficient for this. We have not been able to find a similar condition for 3D when the directions are in two planes. Obviously, a necessary condition is that within each plane the *d*-*n* set can be realized, but as the following example will show, this is not always sufficient. Consider a proper *d*-*n* set $\{\langle d_0, 1\rangle, \langle d_1, 2\rangle, \langle d_2, 99\rangle, \langle d_3, 98\rangle\}$ with two copies as $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ having their two planes perpendicular to each other as shown in Fig. 6(a) and (b) respectively. For both of them, we have $\delta_0 = 1$, $\delta_1 = 97$, $\delta_2 = 0$ and $\delta_3 = 0$. With $N = 100$, since $\sum \delta_i \leq 100$, both $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ have a feasible polyhedron. However, all together, their two planes are so arranged that $\overline{\theta}_1$ and $\widetilde{\theta}_1$ are opposite to each other. Since each of these two disjoint d-lunes must get at least 97 normal points, there is no valid assignment.

We now develop an algorithm to find a valid assignment if one exists. We can even add extra conditions that will be useful later:

Lemma 3. *We can find a valid assignment, if one exists, in* $O(\overline{K} + \widetilde{K})$ *time. Moreover, if* $\max\{\overline{D}, \widetilde{D}\} < N$*, then* $\Delta_{0,0} > 0$ *and* $\Delta_{\overline{K},\widetilde{K}} > 0$.

Proof. We have two cases:

Two planes have a pair of common directions. In this case a valid assignment

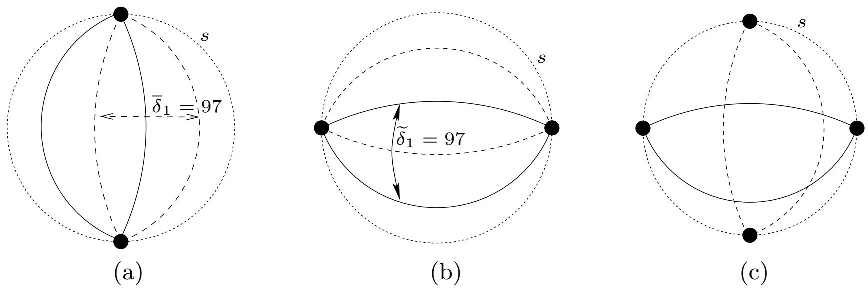


Fig. 6. (a) $\overline{\mathcal{S}}$. (b) $\widetilde{\mathcal{S}}$. (c) $\mathcal{S} = \overline{\mathcal{S}} \cup \widetilde{\mathcal{S}}$ has no valid assignment.

always exists and we will find one such valid assignment. Recall that the great circle of $\overline{h}_0$ and $\overline{h}_{\overline{K}}$ divides the d-polygons into two $\overline{K} \times \widetilde{K}$ "matrices"—one in $\overline{h}_0$ and the other one in $\overline{h}_{\overline{K}}$ (see Fig. 5(a)(ii)). After possible renaming, assume that $\overline{D} = \min\{\overline{D}, \widetilde{D}\}$. Assign normal points to the d-polygons of $\overline{h}_0$ by using Lemma 2 with the rows of M having row-sums $\overline{\delta}_0 + \frac{1}{2}(\widetilde{D} - \overline{D})$, $\overline{\delta}_1, \ldots, \overline{\delta}_{\overline{K}-1}$ and the columns having column-sums $\widetilde{\delta}_0, \ldots, \widetilde{\delta}_{\widetilde{K}-1}$. Similarly, assign normal points to the d-polygons of $\overline{h}_{\overline{K}}$ with the rows of M having row-sums $\overline{\delta}_{\overline{K}} + \frac{1}{2}(\widetilde{D} - \overline{D})$, $\overline{\delta}_{\overline{K}+1}, \ldots, \overline{\delta}_{2\overline{K}-1}$ and the columns having column-sums $\widetilde{\delta}_{\widetilde{K}}, \ldots, \widetilde{\delta}_{2\widetilde{K}-1}$. Finally, if $\widetilde{D} < N$, then assign $\frac{1}{2}(N - \widetilde{D})$ normal points to each of $\theta_{0,0}$ and $\theta_{\overline{K},\widetilde{K}}$.

For the correctness of the above algorithm, since $\overline{h}_i = \bigcup_{l=i+\overline{K}}^{i+2\overline{K}-1} \overline{\theta}_l$ and $\overline{h}_i$ includes either $\overline{\theta}_0$ or $\overline{\theta}_{\overline{K}}$, the total number of normal points in $\overline{h}_i$ is $\sum_{l=i+\overline{K}}^{i+2\overline{K}-1} \overline{\delta}_l + \frac{1}{2}(\widetilde{D} - \overline{D}) + \frac{1}{2}(N - \widetilde{D})$, which by Lemma 1 is $\overline{n}_i$. Similarly, for $\widetilde{h}_j$ the total number of normal points is $\sum_{l=j+\widetilde{K}}^{j+2\widetilde{K}-1} \widetilde{\delta}_l + \frac{1}{2}(N - \widetilde{D}) = \widetilde{n}_j$. So, the assignment is valid and if $\max\{\overline{D}, \widetilde{D}\} < N$, then the two opposite d-polygons $\theta_{0,0}$ and $\theta_{\overline{K},\widetilde{K}}$ have at least one normal point each. Finally, using Lemma 2 twice takes $O(\overline{K} + \widetilde{K})$ time.

Two planes have no common direction. Let $\overline{W}_1$ and $\overline{W}_2$ be the two sets of d-lunes that are in between $\overline{\theta}_0$ and $\overline{\theta}_{\overline{K}}$ and let $\overline{X}_1$ and $\overline{X}_2$ be the sum of the view differences of the d-lunes of $\overline{W}_1$ and $\overline{W}_2$ respectively. More formally,

$$\overline{W}_1 = \{\overline{\theta}_1, \ldots, \overline{\theta}_{\overline{K}-1}\}, \overline{W}_2 = \{\overline{\theta}_{\overline{K}+1}, \ldots, \overline{\theta}_{2\overline{K}-1}\}, \overline{X}_1 = \sum_{i=1}^{\overline{K}-1} \overline{\delta}_i, \text{ and } \overline{X}_2 = \sum_{i=\overline{K}+1}^{2\overline{K}-1} \overline{\delta}_i$$

Similarly,

$$\widetilde{W}_1 = \{\widetilde{\theta}_1, \ldots, \widetilde{\theta}_{\widetilde{K}-1}\}, \widetilde{W}_2 = \{\widetilde{\theta}_{\widetilde{K}+1}, \ldots, \widetilde{\theta}_{2\widetilde{K}-1}\}, \widetilde{X}_1 = \sum_{j=1}^{\widetilde{K}-1} \widetilde{\delta}_j, \text{ and } \widetilde{X}_2 = \sum_{j=\widetilde{K}+1}^{2\widetilde{K}-1} \widetilde{\delta}_j$$

W.l.o.g assume that all d-lunes of $\overline{W}_1$ (similarly $\overline{W}_2$) intersect all d-lunes of $\widetilde{W}_1$

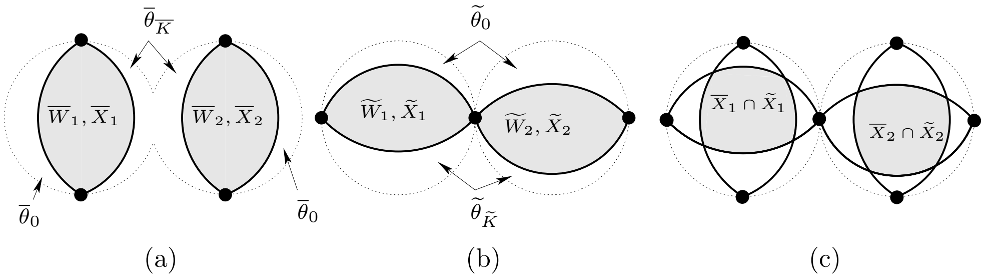


Fig. 7. (a) $\overline{W}_1, \overline{W}_2, \overline{X}_1$ and $\overline{X}_2$. (b) $\widetilde{W}_1, \widetilde{W}_2, \widetilde{X}_1$ and $\widetilde{X}_2$. (c) $\overline{W}_1 \cap \widetilde{W}_1$ and $\overline{X}_2 \cap \widetilde{X}_2$.

(respectively $\widetilde{W}_2$), which form two $(\overline{K}-1) \times (\widetilde{K}-1)$-matrices of d-polygons (see Fig. 7). We assign the normal points by the following four steps:

- We place as many normal points as possible, namely $\min\{\overline{X}_1, \widetilde{X}_1\}$ and $\min\{\overline{X}_2, \widetilde{X}_2\}$, into these two matrices by using Lemma 2 twice. The remaining normal points must all be within $\overline{\theta}_0, \widetilde{\theta}_0, \overline{\theta}_{\overline{K}}, \widetilde{\theta}_{\widetilde{K}}$, which yields lower bounds (and hence necessary conditions) onto $\overline{\delta}_0, \widetilde{\delta}_0, \overline{\delta}_{\overline{K}}, \widetilde{\delta}_{\widetilde{K}}$. These conditions turn out to be sufficient as well, and we find the assignment by using Lemma 2 twice more by the next two steps.
- Put as many normal points as possible, namely $\max\{\overline{X}_1, \widetilde{X}_1\} - \min\{\overline{X}_1, \widetilde{X}_1\} + \max\{\overline{X}_2, \widetilde{X}_2\} - \min\{\overline{X}_2, \widetilde{X}_2\}$, in the remaining d-polygons except for $\theta_{0,0}$, $\overline{\theta}_{\overline{K},0}$, $\widetilde{\theta}_{0,\widetilde{K}}$, $\theta_{\overline{K},\widetilde{K}}$ (the four d-polygons incident to both poles).
- Put the remaining normal points, namely $\max\{\overline{D}, \widetilde{D}\} - \max\{\overline{X}_1, \widetilde{X}_1\} - \max\{\overline{X}_2, \widetilde{X}_2\}$, into these four polygons.

  Until now the d-lunes of $\overline{W}_1, \overline{W}_2, \widetilde{W}_1$ and $\widetilde{W}_2$ get an exact total of $\overline{X}_1, \overline{X}_2, \widetilde{X}_1$ and $\widetilde{X}_2$ normal points, respectively, with each d-lune $\overline{d}_i$ in $\overline{W}_1$ or $\overline{W}_2$ (similarly, each d-lune $\widetilde{d}_i$ in $\widetilde{W}_1$ or $\widetilde{W}_2$) getting $\overline{\delta}_i$ (similarly, $\widetilde{\delta}_i$) normal points. The d-lunes $\overline{d}_0$ and $\overline{d}_{\overline{K}}$ get $\overline{\delta}_0$ and $\overline{\delta}_{\overline{K}}$ normal points. W.l.o.g if we assume that $\widetilde{D} \leq \overline{D}$, then the d-lunes $\widetilde{d}_0$ and $\widetilde{d}_{\widetilde{K}}$ get $\widetilde{\delta}_0 + \frac{1}{2}(\overline{D} - \widetilde{D})$ and $\widetilde{\delta}_{\widetilde{K}} + \frac{1}{2}(\overline{D} - \widetilde{D})$ normal points.
- Finally, if $\max\{\overline{D}, \widetilde{D}\} < N$, then we assign $\frac{1}{2}(N - \overline{D})$ normal points to each of $\theta_{0,0}$ and $\theta_{\overline{K},\widetilde{K}}$, which by Lemma 1 implies that $\theta_{0,0}$ and $\theta_{\overline{K},\widetilde{K}}$ are positive.

To see the validity of the assignment, the total number of normal points in $\overline{h}_i$ is $\sum_{j=i}^{i+\overline{K}-1} \overline{\delta}_i + \frac{1}{2}(N - \overline{D})$, which by Lemma 1 is $\overline{n}_i$. For $\widetilde{h}_j$, the total number of normal points is $\sum_{j=i}^{i+\widetilde{K}-1} \widetilde{\delta}_i + \frac{1}{2}(\overline{D} - \widetilde{D}) + \frac{1}{2}(N - \overline{D})$, which by Lemma 1 is $\widetilde{n}_i$. Using Lemma 2 four times takes $O(\overline{K} + \widetilde{K})$ time, and as mentioned in the first step above, checking whether Lemma 2 can be applied for $\overline{\theta}_0, \widetilde{\theta}_0, \overline{\theta}_{\overline{K}}$ and $\widetilde{\theta}_{\widetilde{K}}$ can be done in constant time. □

The valid assignment by the above lemma yields how many normal points should be placed in each d-polygon, but not the actual locations. To find their actual location, we need to solve what we call the *valid selection problem*: Assign normal points such that no hemisphere contains all normal points. (If one hemisphere contains all normal points, then the resulting polyhedron is unbounded. If this is allowed, then the existence of a valid assignment is necessary and also sufficient for the existence of a feasible polyhedron.)

4.2.1. *Insufficiency of a valid assignment*

Before we study how to find a valid selection, we first show that this is a non-trivial problem, by describing an instance which has a valid assignment, but no valid selection. Consider the 2D proper *d-n* set $\mathcal{S}'$ of Fig. 8(a). It has twelve *d-n* pairs with only positive view differences $\delta_0 = 2$, $\delta_4 = 1$, and $\delta_8 = 1$. We use $N = 4$, so $N = D$. A key property of $\mathcal{S}'$ is that this defines very "thin" d-lunes. Moreover, the directions can be adjusted such that the gap between these thin d-lunes are increased or decreased. With such flexibility, we use $\mathcal{S}'$ twice as $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ in two different planes as shown in Fig. 8(b) and take $\mathcal{S} = \overline{\mathcal{S}} \cup \widetilde{\mathcal{S}}$. There are only two possible valid assignments for $\mathcal{S}$ which are shown in Fig. 8(c). But in either case all positive d-polygons are strictly within a single hemisphere. So, no valid selection exists.

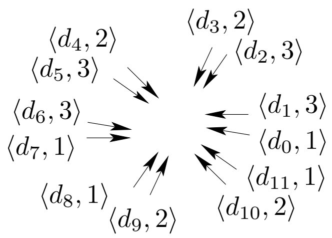
4.2.2. *Finding a valid selection*

Despite this negative example, we can find a valid selection in two cases: (i) $\max\{\overline{D}, \widetilde{D}\} < N$, and (ii) $\overline{D} = N = \widetilde{D}$ and all directions see at least four faces. Note that neither case covers the above example.

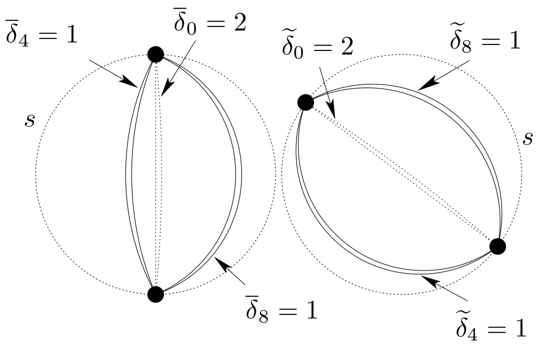
In the first case, by Lemma 3 we can find a valid assignment where $\theta_{0,0}$ contains a normal point, say x_3, and $\theta_{\overline{K},\widetilde{K}}$ contains a normal point, say x_4. Let x_1, x_2 be two other normal points. These four normal points exist by $N \geq 4$. W.l.o.g we assume that x_1, x_2 and x_3 are all within one hemisphere; they then span a spherical triangle t, which intersects $\theta_{0,0}$. See also Fig. 9(a). The antipodal triangle t' to t, i.e., the reflection of t about the origin o, hence intersects $\theta_{\overline{K},\widetilde{K}}$, and we can move x_4 so that it is strictly inside $t' \cap \theta_{\overline{K},\widetilde{K}}$. This will ensure that no hemisphere contains all of x_1, x_2, x_3, x_4.

Now consider the case when $\overline{D} = N = \widetilde{D}$ and each direction sees at least four faces. This case is significantly more complicated. In fact, we are not able to find a valid selection for any given assignment, but we can find a valid selection if we are allowed to change the given assignment slightly.

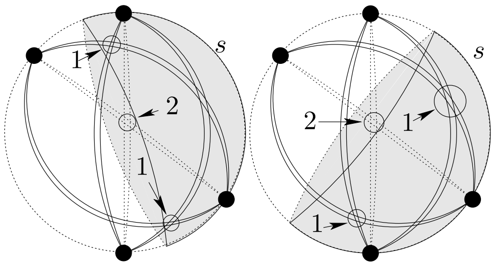
We first define octants of the sphere by choosing three great circles as follows. The first one is the great circle g^* that contains the four poles of $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$. The second great circle $\overline{g}$ is obtained by rotating a great circle, starting at g^*, through the poles of $\overline{\mathcal{S}}$ until the four lunes defined by g^* and $\overline{g}$ contain at least two normal



(a)



(b)



(c)

Fig. 8. An example of insufficiency of a valid assignment. (a) A 2D d-n set $\mathcal{S}'$. (b) Two copies $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$ of $\mathcal{S}'$ in two different planes in 3D. In $\overline{\mathcal{S}}$, all directions are in xz-plane and from the positive z-axis, the directions d_0, d_1, d_4, d_5, d_8 and d_9 are roughly $80°, 80° + \epsilon, 250°, 250° + \epsilon, 330°, 330° + \epsilon$ away, respectively, for some small value of ϵ. Other directions in $\overline{\mathcal{S}}$ are opposite to these directions. In $\widetilde{\mathcal{S}}$, the direction plane is rotated around the positive z-axis roughly by $45°$ towards the positive x-axis, and the directions d_4 and d_5 are adjusted so that they are now roughly $200°$ and $200° + \epsilon$ away from the positive z-axis, respectively. (c) Only two possible valid assignments for $\mathcal{S}$, and for both of them all three positive d-polygons are strictly within a hemisphere of s (shown shaded).

points each. That this is possible is non-trivial and the following lemma shows how to do that.

Lemma 4. *Given a proper d-n set $\mathcal{S}$ of size $2K$, where the directions are in one plane, $n_i \geq 4$ for any $\langle d_i, n_i\rangle$, and $D = N$. Then for any great circle g^* passing through the poles of $\mathcal{S}$, we can find in $O(K)$ time another great circle g also passing through the poles of $\mathcal{S}$ such that the four lunes created by g^* and g contain at least two normal points each, after a suitable distribution of normal points in the d-lunes*

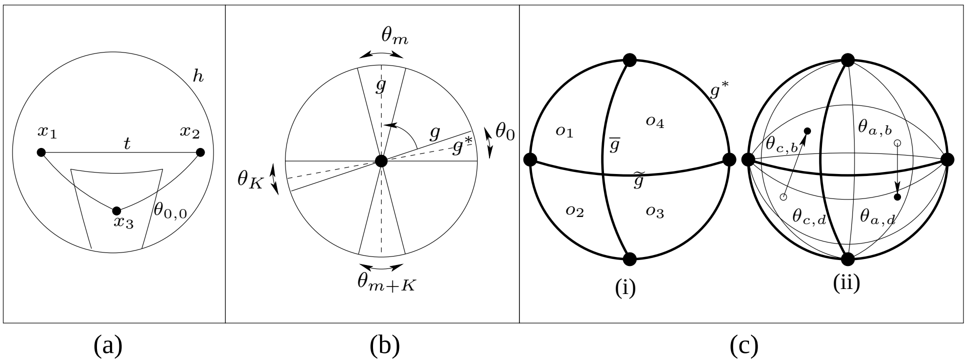


Fig. 9. (a) Arranging normal points to avoid an empty half-sphere. (b) Four lunes created by g^* and g (shown as broken lines). (c) Shifting normal points within octants.

intersected by g^* and g.

Proof. We have two cases based on the position of g^*. If g^* is the great circle of a visible hemisphere of $\mathcal{S}$, then w.l.o.g we assume that g^* is the great circle of h_1 (and h_{K+1}). Otherwise, w.l.o.g we assume that g^* properly intersects θ_0 and θ_K. We initialize g to be the great circle of h_1. Note that g is the first great circle, in counter-clockwise direction from g^*, that is the boundary of a visible hemisphere. Then we rotate g counter-clockwise until the lunes satisfy the conditions. To be precise, let m be minimal such that $\sum_{i=1}^{m} \delta_i \geq 2$ and $\sum_{i=K+1}^{m+K} \delta_i \geq 2$, and choose g to be strictly inside θ_m (and θ_{m+K}). See also Fig. 9(b) for the later case.

We claim that the four lunes of g^* and g contain at least 2 normal points each, provided we suitably redistribute normal points in $\theta_0, \theta_m, \theta_K$ and θ_{m+K}. Since $N = D$, there is only one valid assignment: each θ_i contains exactly δ_i normal points. Also, we know that $\min\{\delta_i, \delta_{i+K}\} = 0$ for any i. We may therefore (after renaming, if needed) assume that $\delta_{m+K} = 0$. On the other hand, we must have $\delta_m > 0$ and $\sum_{i=1}^{m-1} \delta_i \leq 1$ by minimality of m. Now we consider the four lunes of g^* and g, which we describe by the d-lunes that they strictly contain:

- The lune containing $\theta_{K+1}, \ldots, \theta_{m+K-1}$: This lune contains at least $\sum_{i=K+1}^{m+K-1} \delta_i$ normal points, which is at least 2 by the choice of m and $\delta_{m+K} = 0$.
- The lune containing $\theta_1, \ldots, \theta_{m-1}$: This lune contains at least 2 normal points by the choice of m and if we include all normal points from θ_m. However, since some normal points from θ_m may be needed elsewhere, we will only use $2 - \sum_{i=1}^{m-1} \delta_i$ normal points from θ_m for it, which gives exactly 2 normal points for this lune.
- The lune containing $\theta_{m+K+1}, \ldots, \theta_{2K-1}$: Note that no normal points from θ_0 have been used for the previous lunes, so we will include all of them

(if any) here. Hence the number of normal points is $\sum_{i=m+K+1}^{2K-1} \delta_i + \delta_0 = \sum_{i=m+K}^{m-1} \delta_i - \sum_{i=1}^{m-1} \delta_i \geq n_m - 1 \geq 3$, since $\sum_{i=1}^{m-1} \delta_i \leq 1$, $\sum_{i=m+K}^{m-1} \delta_i = n_m$ by Lemma 1 and $D = N$, and $n_m \geq 4$.

- The lune containing $\theta_{m+1}, \ldots, \theta_{K-1}$: Note that no normal points from θ_K have been used for the previous lunes, so we will include all of them (if any) here. Also, this lune gets $\delta_m - (2 - \sum_{i=1}^{m-1} \delta_i)$ normal points from θ_m. So the number of normal points is $\sum_{i=m+1}^{K-1} \delta_i + \delta_K + \delta_m + \sum_{i=1}^{m-1} \delta_i - 2 = \sum_{i=1}^{K} \delta_i - 2 = n_1 - 2 \geq 2$, since $\sum_{i=1}^{K} \delta_i = n_1$ by Lemma 1 and $D = N$, and $n_1 \geq 4$.

Clearly, the minimal m satisfying the condition, and the point distribution, can be found in $O(K)$ time. □

We use the above lemma twice, once for $\overline{\mathcal{S}}$ and once for $\widetilde{\mathcal{S}}$, with g^* in both cases as the great circle passing through the poles of $\overline{\mathcal{S}}$ and $\widetilde{\mathcal{S}}$. Let the two great circles that come out of the lemma be $\overline{g}$ and $\widetilde{g}$ respectively.

Now we have eight octants defined by three great circles $g^*, \overline{g}$ and $\widetilde{g}$. A fairly straightforward proof shows that if each octant contains a normal point, then no hemisphere can be empty. However, our given valid assignment need not have a normal point in all octants. But, since the great circles were chosen such that each lune has at least two normal points, we can change the valid assignment to a different valid assignment by shifting points from octants having two normal points to empty octants. The following lemma shows how to do that. See also Fig. 9(c).

Lemma 5. *If there are empty octants, then we can change the valid assignment to another one without empty octants.*

Proof. Let o_1, o_2, o_3 and o_4 be the four octants of a hemisphere of g^*, and assume that o_1 and o_2 are in one hemisphere of $\overline{g}$ and o_2 and o_3 are in one hemisphere of $\widetilde{g}$. (See Fig. 9(c)(i)). Each two consecutive octants together (i.e., $o_1 \cup o_2$, $o_2 \cup o_3$, $o_3 \cup o_4$, and $o_4 \cup o_1$) contain at least two normal points by the choice of $\overline{g}$ and $\widetilde{g}$.

Assume that o_1 is empty. Then o_2 and o_4 contain at least two normal points each. So let $\theta_{a,b}$ and $\theta_{c,d}$ be d-polygons in o_2 and o_4 that contains at least one normal point each. Recall that $\theta_{a,b} = \overline{\theta}_a \cap \widetilde{\theta}_b$ and $\theta_{c,d} = \overline{\theta}_c \cap \widetilde{\theta}_d$. Observe that the d-polygon $\theta_{c,b}$ intersects o_1. Similarly, the d-polygon $\theta_{a,d}$ intersects o_3. Now change the valid assignment by moving one normal point from each of $\theta_{a,b}$ and $\theta_{c,d}$ to $\theta_{a,d}$ and $\theta_{c,b}$ respectively. See Fig. 9(c)(ii). Then the number of normal points in any d-lune has not been changed, and we still have a valid assignment. Also, o_1 and o_3 now have one normal point each, and o_2 and o_4 have lost one each. Since o_2 and o_4 had at least two normal points before, now no octant is empty. □

After doing so, we can choose arbitrary normal points within the d-polygons and obtain a valid selection.

None of our steps is computationally expensive, and the time complexity is dominated by the time to compute the intersection of the tangent half-planes of the computed normal points. In summary, we obtain:

Theorem 5. *Given a proper d-n set $\mathcal{S}$ and an integer $N \geq 4$, where the directions of $\mathcal{S}$ are covered by two planes. We can construct a feasible convex polyhedron P, if it exists, in $O(N \log N + |\mathcal{S}|)$ time, in each of the following cases: (i) $\max\{\overline{D}, \widetilde{D}\} < N$, or (ii) $\overline{D} = N = \widetilde{D}$ and $n \geq 4$ for each d-n pair $\langle d, n\rangle$ in $\mathcal{S}$.*

5. NP-hardness for arbitrary directions

Theorem 6. *Given a proper d-n set $\mathcal{S}$ of size $2K$ with three planes of directions, it is NP-hard to decide the existence of a feasible convex polyhedron for $\mathcal{S}$.*

Proof. We will prove that when the directions span three planes, it is NP-hard to find a set of points such that the hemisphere of every view direction of a *d-n* set contains exactly the prescribed number of points. Note that finding such a set of points is a necessary step in reconstructing a feasible convex polyhedron.

To prove that the problem of finding such a set of points is NP-hard, we apply a reduction from the problem of testing whether a 2-edge connected cubic planar graph G has an independent set of size k, which is NP-hard [3]. An *independent set* I of G is a set of vertices such that no two vertices of I are connected by an edge.

Since G is a 2-edge connected cubic planar graph, it is 3-edge colorable. Moreover, the 3-edge coloring can be found in polynomial time: Compute a 4-vertex coloring of the dual graph [22] and convert it into a 3-edge coloring of G with standard techniques [25].

We draw G as follows: First place all vertices in a vertical line. Let $\mathcal{L}$ be the set of all lines of slope $i\pi/3$, $i = 0, 1, 2$, through the set of vertices of G. We place the vertices in such a way that no three lines of $\mathcal{L}$ intersect at one point, except at a vertex. Note that such placement of vertices can be done easily in polynomial time. Next draw each edge e of color j with 3 segments: one segment of slope $j\pi/3$ at each end and one segment of slope $(j+1)\pi/3$ connecting them. Add to $\mathcal{L}$ three new lines (of slope $i\pi/3$, $i = 0, 1, 2$) through each of the two bends of e. We choose the segment lengths such that these newly added lines through the bends do not cross an intersection point of the existing lines of $\mathcal{L}$. This can always be done by drawing the middle segment sufficiently far out, and suitable lengths can be computed in polynomial time. See Fig. 10.

Now we have a (not necessarily planar) drawing of G and a system of lines $\mathcal{L}$ with three slopes such that any trivalent point (a point that belongs to three lines of $\mathcal{L}$) corresponds to a vertex of G or a bend of an edge of G and that no other three lines of $\mathcal{L}$ cross in one point. Since G is cubic and 3-edge-colored, one easily verifies that there are $n + m$ lines in each direction in $\mathcal{L}$, where n and m are the number of vertices and edges of G. Also $m = \frac{3}{2}n$, so $|\mathcal{L}| = \frac{15}{2}n$.

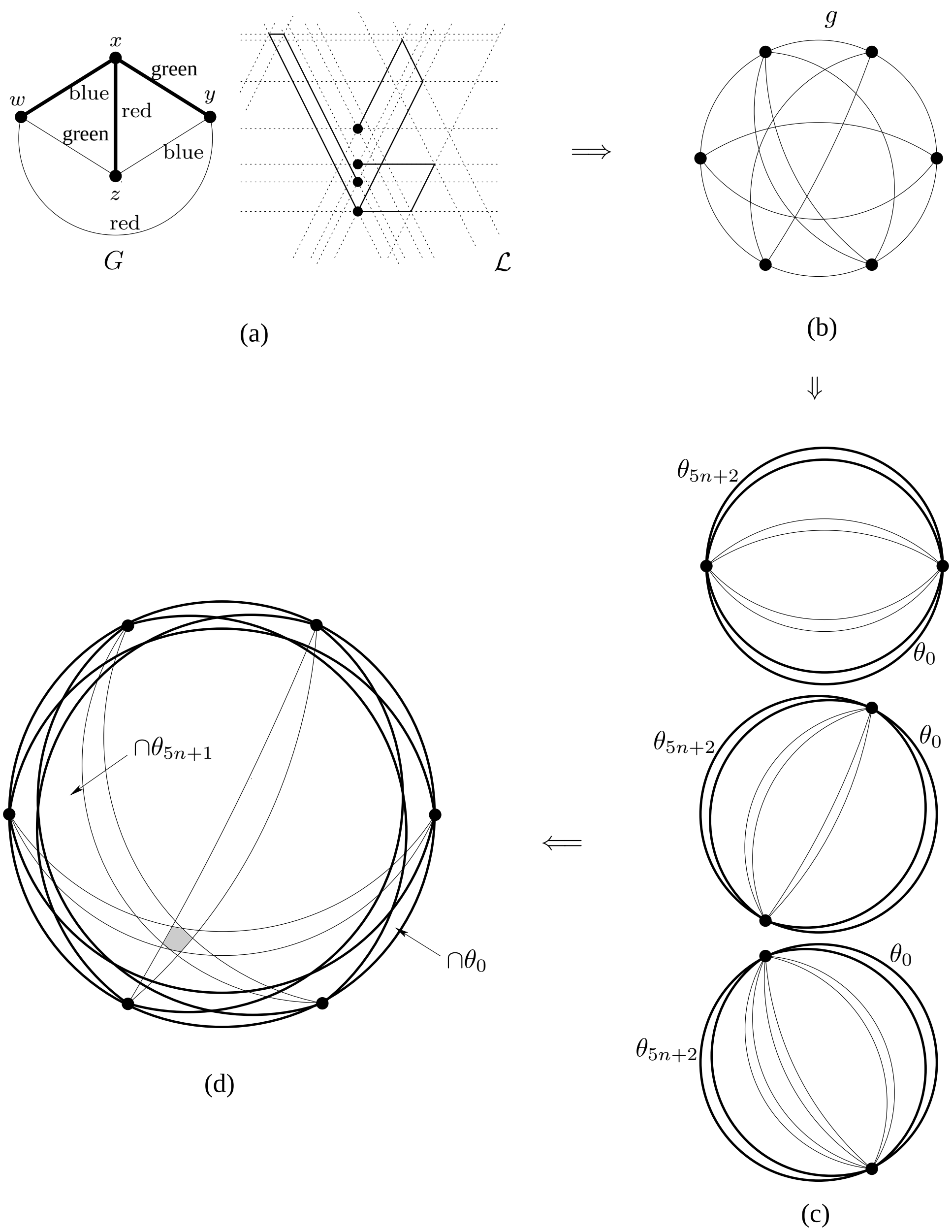


Fig. 10. (a) A 2-edge connected cubic planar graph G with its 3-edge coloring, and creating $\mathcal{L}$ from G. For simplicity, only three edges of G (shown as bold) are converted to $\mathcal{L}$. (b) Projecting lines onto a sphere. (c) Then converting them to thin d-lunes. The boundary thin d-lunes are shown bold. (d) Arrangements of the d-lunes, partially shown in one hemisphere for clarity's sake; the d-polygon of a degree-3 vertex of G is shown shaded. The three copies of θ_{5n+1+K} intersect, since this is the reflection of the intersection of the three copies of θ_{5n+1}, which we mark by $\cap\theta_{5n+1}$ in the figure.

We will eventually project $\mathcal{L}$ onto the sphere and then create a *d-n* set such that any solution to it can be converted to a set of points of $\mathcal{L}$ with certain properties. This will be helpful, since there is a correspondence between independent sets of G and points placed on $\mathcal{L}$ as follows:

Lemma 6. *G has an independent set of size k if and only if there exists a set $\mathcal{T}$ of $\frac{9}{2}n - 2k$ points such that each line of $\mathcal{L}$ intersects exactly one point of $\mathcal{T}$ and each point of $\mathcal{T}$ intersects either one or three (but not two) lines of $\mathcal{L}$.*

Proof. Given an independent set I of size k of G, we construct $\mathcal{T}$ by the following three steps:

(1) Add the point of every vertex of G in I. This adds k trivalent points.
(2) For every edge (v, w) of G, at least one endpoint (say v) is not in I. Add the point of the bend that is adjacent to v. This adds $m = \frac{3}{2}n$ trivalent points.
(3) By the construction, no line of $\mathcal{L}$ is intersected twice by the points chosen into $\mathcal{T}$ thus far by the above two steps. For every line in $\mathcal{L}$ not intersected by a point of $\mathcal{T}$, add one more point that intersects this line only. Since $3k$ and $\frac{9}{2}n$ lines are already intersected by a point of $\mathcal{T}$ in steps (1) and (2), respectively, and since $|\mathcal{L}| = \frac{15}{2}n$, this third step adds $\frac{15}{2}n - 3k - \frac{9}{2}n = 3n - 3k$ points.

Therefore, the total number of points added into $\mathcal{T}$ by the above three steps is $k + \frac{3}{2}n + 3n - 3k = \frac{9}{2}n - 2k$ and the other properties are easily verified.

For the other direction, assume that we are given such a point set $\mathcal{T}$, and assume it contains ℓ trivalent points. Then $|\mathcal{L}| - 3\ell = \frac{15}{2}n - 3\ell$ lines are covered by points that are on one line only, so $|\mathcal{T}| = \frac{15}{2}n - 2\ell$, which with $|\mathcal{T}| = \frac{9}{2}n - 2k$ implies $\ell = \frac{3}{2}n + k$. Let H be the graph obtained from G by subdividing each edge twice at its two bends. Each of the $\ell = \frac{3}{2}n + k$ trivalent points belongs to a vertex or a bend of G, hence a vertex of H. These trivalent vertices are an independent set I' of H, since every line of $\mathcal{L}$ contains only one point of $\mathcal{T}$ implying that every edge of H intersects only one vertex of I'. I' contains at most one bend per edge (v, w) of G, and if both v and w are in I', then neither bend of edge (v, w) is in I'. So by removing one vertex per edge of G we can convert I' into an independent set of size k in G. □

Now we create an instance of our reconstruction problem, i.e., a proper *d-n* set $\mathcal{S}$ and N, from the set $\mathcal{L}$ as follows. (Also see Fig. 10.) First do a stereographic projection, i.e., consider $\mathcal{L}$ as lines in an xy-plane, place a sphere s outside this plane, and map each line l of $\mathcal{L}$ to the great circle defined by the intersection of s with the plane through the center of s and l. All lines of the same slope hence get mapped to great circles with common poles, and the three pole-sets for the three directions all lie in one xy-plane, which for ease of description we assume to be the

($z = 0$)-plane. Note that the arrangement of line appears twice on the sphere, once on each side of the ($z = 0$)-plane.

We now set up the directions of $\mathcal{S}$ such that each great circle of a line gets replaced by a pair of opposite lunes through the same poles. We call these thin lunes *non-boundary* thin lunes. These lunes are thin enough such that there is no common intersection of more than 3 of them. We also replace the great circle of the ($z = 0$)-plane by 12 lunes as follows. For each pair of poles, each half circle between them gets replaced by two adjacent thin lunes, divided at the ($z = 0$)-plane. Moreover, the four lunes for each pair of poles are in two opposites pairs. We call these thin lunes *boundary* thin lunes.

Now we set up N and the integers of the d-n sets of $\mathcal{S}$. We do this in such a way that the following four conditions hold:

- The sum of view differences is exactly N, so the total view difference is exactly the number of normal points in any solution.

  The next three conditions are for the half-space above the ($z = 0$)-plane.
- The non-boundary thin lunes replacing lines all have view-difference 1. Hence any assignment of normal points will have to assign exactly one point to this line.
- The spaces between thin lunes all have view-difference 0. Hence we can only place normal points at the intersection of three thin lunes, which correspond to trivalent points, or at the thin lunes replacing the ($z = 0$)-plane.
- The total number of points in this half-space is $\frac{9}{2}n - 2k$.

An instance of our reconstruction problem satisfying the above condition will imply a set of points with properties as in Lemma 6, and hence will yield an independent set of size k in G.

We now explain how exactly to choose the integers for the d-n sets in $\mathcal{S}$ and the integer N. Remember that, for each slope of $\mathcal{L}$, we get one plane of directions in $\mathcal{S}$, and thus we have three planes of directions in $\mathcal{S}$. We find the integers for d-n sets in one plane only; they are the same for the other two planes.

First we see how many d-n pairs do we have in a particular plane of $\mathcal{S}$. Let g be the great circle that corresponds to the ($z = 0$)-plane. Recall that each slope of $\mathcal{L}$ has $\frac{5}{2}n$ lines, which give $5n + 4$ thin lunes. Moreover, every pair of "adjacent" thin lunes has a *thick* lune, i.e., a *non-thin* lune, between them, except for the pair of thin lunes that divides at g. Thus we have $5n+2$ thick lunes, giving a total of $10n+6$ lunes. It implies that we have a total $K = 5n + 3$ pairs of opposite directions in a plane of $\mathcal{S}$, which we denote as follows. Consider the lunes above the ($z = 0$)-plane. We assume that d_0 is the direction that corresponds to g, d_1 is the direction that creates a thin lune near g, d_{2i} and d_{2i+1} (for $1 \le i \le \frac{5}{2}n$) are the directions that create the thin lune for the ith line, and d_{5n+2} is the direction for the other thin lune near g. The remaining directions are the opposite of these directions. Also see Fig. 10(b, c).

We now assign the corresponding integers as follows: $n_0 = 1$, $n_1 = 1$, $n_{2i} = i$, $n_{2i+1} = i+1$ (for $1 \le i \le \frac{5}{2}n$), $n_{5n+2} = \frac{5}{2}n$ and $n_K = n_{5n+3} = \frac{9}{2}n - 2k$. Since d_0 and d_K are opposite, $N = n_0 + n_K = \frac{9}{2}n - 2k + 1$. All other opposite view directions have integers as computed, i.e., $n_{K+i} = N - n_i$, for $i = 1, \ldots, K-1$.

Note that $\delta_0 = \delta_1 = 0$. For $1 \le i \le \frac{5}{2}n$, $\delta_{2i} = 1$ and $\delta_{2i+1} = 0$, as desired. $\delta_{5n+2} = (\frac{9}{2} - 2k) - \frac{5}{2}n = 2n - 2k$. On the other side of the $(z = 0)$-plane, all δ_i's are 0, with the sole exception of δ_{5n+1+K}, which is $n_{5n+2+K} - n_{5n+1+K} = (N - \frac{5}{2}n) - (N - \frac{5}{2}n - 1) = 1$. Hence the sum of all view-differences is $\frac{5}{2}n + 2n - 2k + 1 = N$ as desired.

Clearly, this satisfies all conditions, and so a solution to the problem of finding a set of points such that the hemisphere of every view direction of a *d*-*n* set contains exactly the prescribed number of points implies an independent set of size k in G.

Conversely, if we have an independent set of size k, then we can find by Lemma 6 a set of $\frac{9}{2}n - 2k$ points to cover all lines with single or trivalent points. Note that for a particular plane of directions, the non-boundary thin lunes intersect boundary thin lunes θ_0 and θ_{5n+2} of the other two planes. Moreover, we can assume a numbering such that the three boundary thin lunes θ_0 from three planes intersect and similarly, the three boundary thin lunes θ_{5n+2} intersect. Also see Fig. 10(d).

Now, for each trivalent point, place a normal point in the d-polygon of the intersection of the three corresponding non-boundary thin lunes. For each single point, place a normal point such that it intersects the non-boundary thin lune of this line and the two boundary lunes θ_{5n+2} for the other two directions of lines. This places $2n - 2k$ points into each of the θ_{5n+2} as desired, and we hence obtain $\frac{9}{2}n - 2k$ normal points above the $(z = 0)$-plane. We must place one more normal point below the $(z = 0)$-plane, and in such a way that it intersects the three copies of θ_{5n+1+K}. This is possible because θ_{5n+1+K} is a thick lune and quite close to g; hence the three copies of it (in three different directions) intersect. See Fig. 10(d).

This shows that our instance has a solution if and only if G has an independent set of size k, and hence proves the NP-hardness. □

6. Conclusion

In this paper we have studied the problem of constructing convex polygons and convex polyhedra given the number of visible edges and visible faces in some orthogonal projections. In 2D, we have given necessary and sufficient conditions for the existence of a feasible polygon of a given size and have given an algorithm to construct one, if it exists. When the polygon size is unknown, we have given an algorithm to find the maximum and minimum size of a feasible polygon. In 3D, when the directions span a single plane we have shown that a feasible polyhedron can be constructed from a feasible polygon and vice versa. We have also given an algorithm to construct a feasible polyhedron when the directions are covered by two planes. Finally, we have shown that the problem becomes NP-hard when the directions are covered by three or more planes.

References

1. M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, *Linear Programming and Network Flows* (John Wiley, 2005).
2. S. Bereg, 3D realization of two triangulations of a convex polygon, in *Proc. 20th European Workshop on Computational Geometry*, Seville, Spain, (March 2004) pp. 49–52.
3. T. Biedl, G. Kant and M. Kaufmann, On triangulating planar graphs under the four-connectivity constraint, *Algorithmica* **19** (4) (1997) 427–446.
4. A. Bottino, L. Jaulin and A. Laurentini, Reconstructing 3D Objects from Silhouettes with Unknown Viewpoints: The Case of Planar Orthographic Views, in *Proc. 8th Iberoamerican Congress on Pattern Recognition*, Havana, Cuba, (November 2003) pp. 26–29.
5. A. Bottino and A. Laurentini, Introducing a New Problem: Shape-from-Silhouette when the Relative Positions of the Viewpoints is Unknown, *IEEE PAMI* **25**(11) (2003) 1484–1493.
6. E. D. Demaine and J. Erickson, Open Problems on Polytope Reconstruction, Manuscript (1999), Available at: http://erikdemaine.org/papers/PolytopeReconstruction/
7. H. Edelsbrunner, *Algorithms in Combinatorial Geometry* (Springer-Verlag, 1986).
8. R. J. Gardner, *Geometric Tomography* (Cambridge University Press, 1995).
9. M. Hasan, Reconstruction and visualization of polyhedra using projections, PhD Thesis, University of Waterloo, Canada (2005).
10. *Discrete Tomography: Foundations, Algorithms, and Applications (Applied and Numerical Harmonic Analysis)*, eds. G. T. Herman and A. Kuba (Birkhäuser, Boston, 1999).
11. *Advances in Discrete Tomography and Its Applications (Applied and Numerical Harmonic Analysis)*, eds. G. T. Herman and A. Kuba (Birkhäuser, Boston, 2007).
12. C. H. Hoffman, *Geometric and Solid Modelling* (Morgan Kaufmann 1989).
13. A. Laurentini, How many 2D silhouettes does it take to reconstruct a 3D object?, *Computer Vision & Image Understanding* **67**(1) (1997).
14. H. Lipson and M. Shpitalni, Optimization-based reconstruction of a 3D object from a single freehand line drawing, *Computer Aided Design* **28**(8) (1996) 651–663.
15. G. Markowsky and M. Wesley, Fleshing Out Wire Frames, *IBM Journal of Research and Development* **24**(5) (1980) 582–597.
16. G. Markowsky and M. Wesley, Fleshing out projections, *IBM Journal of Research and Development* **25**(6) (1981) 934–954.
17. B. Marlin and G. Toussaint, Constructing convex 3-polytopes from two triangulations of a polygon, *Computational Geometry: Theory and Applications* **28**(1) (2004) 41–47.
18. W. Matusik, C. Buehler, R. Raskar, S. J. Gortler and L. McMillan, Image-Based Visual Hulls, in *Proc. SIGGRAPH 2000*, New Orleans, Louisiana (July 2000) pp. 369–374.
19. I. V. Nagendra and U. G. Gujar, 3-D objects from 2-D orthographic views– a survey, *Computer and Graphics* **12**(1) (1988) 111–114.
20. M. Penna, A shape from shading analysis for a single perspective image of a polyhedron, *IEEE PAMI* **11**(6) (1989) 545–554.
21. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction* (Springer-Verlag, 1985).
22. N. Robertson, D. Sanders, P. Seymour and R. Thomas, The four color theorem, *J. Comb. Theory, Ser. B* **70** (1997) 2–44.
23. K. Sugihara, A necessary and sufficient condition for a picture to represent a polyhedral scene, *IEEE PAMI* **6**(5) (1984) 578–586.
24. K. Sugihara, *Machine Interpretation of Line Drawing* (MIT Press, 1986).

25. R. Thomas, An update on four-color theorem, *Notices of American Mathematical Society* **45**(7) (1998) 848–859.
26. P. A. C. Varley, Automatic creation of boundary-representation models from single line drawings, PhD Thesis, University of Wales (2002).
27. W. Wang and G. G. Grinstein, Survey of 3D Solid Reconstruction from 2D Projection Line Drawings, *Computer Graphics Forum* **12**(2) (1993) 137–158.
28. Q.-W. Yan, C. L. Philip Chen and Z. Tang, Efficient Algorithm for the Reconstruction of 3D Objects from Orthographic Projections, *Computer Aided Design* **26**(9) (1994) 699–717.