# Fast enumeration of point-hyperplane incidences.

Peter Braß[*,†] and Christian Knauer[*]

22. August 2000

**Abstract**

In this paper we study the complexity of enumerating all incidences between a set of $n$ points $P$ and a set of $m$ hyperplanes $H$ in $d$-dimensional euclidean space $\mathbb{R}^d$. We describe a deterministic algorithm that computes an encoding of all point-hyperplane incidences between $P$ and $H$ in $\mathcal{O}\big((m+n)\log(m+n) + (mn)^{\frac{d}{d+1}}(\log(mn))^{\gamma_d}\big)$ time, where $\gamma_d$ is an appropriate constant of order $\mathcal{O}(d \log d)$. The encoding we use is a covering of the incidence graph with complete bipartite subgraphs. We complement our algorithm with a construction that yields $\Omega\big(m + n + m^{\frac{d}{2d-1}} n^{\frac{2d-2}{2d-1}} + m^{\frac{2d-2}{2d-1}} n^{\frac{d}{2d-1}}\big)$ incidences that can not be encoded efficiently.

**Keywords:** Computational geometry, Cuttings, Hopcroft's problem, Point-hyperplane incidences.

## 1 Introduction

Determining the incidences between a set of points and a set of lines, or curves, or hyperplanes, is an important step in some algorithms. For $n$ points and $m$ lines it is well known that there are at most $\mathcal{O}\big(m+n+m^{\frac{2}{3}}n^{\frac{2}{3}}\big)$ such incidences, as reached e.g. by a lattice section and some lattice lines, and these incident pairs can be found very efficiently, in $\mathcal{O}\big((m+n+m^{\frac{2}{3}}n^{\frac{2}{3}})\log(m+n)\big)$ time. Similar results exist for point-curve incidences for some types of curves. For dimension $d \geq 3$ there are $mn$ incidences possible between $m$ lines and $n$ hyperplanes, e.g. if all the points lie on a line, and all hyperplanes contain that line. So we cannot hope for anything better than $\mathcal{O}(mn)$ time for determining all incidences, and $\mathcal{O}(mn)$ can be trivially achieved by testing all pairs.

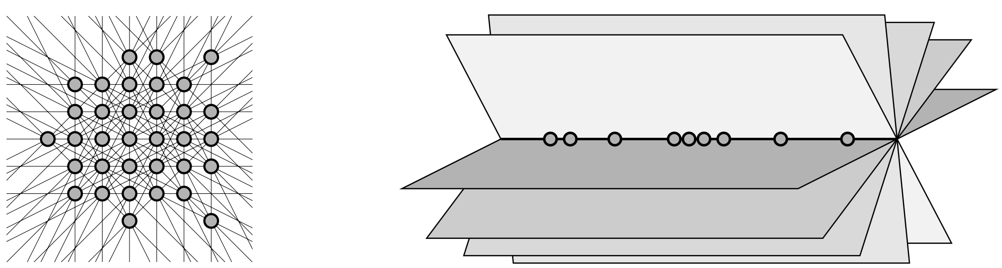

Figure 1: Configurations with many incidences.

This, however, is unsatisfactory, since these $mn$ incidences can occur only in a very special situation, which could be described much more efficiently than just listing all incident pairs. It is the aim of this note to show that indeed we can be faster, beating the output complexity by a better coding of the output. For if we accept an output consisting of a sequence of *set pairs*

[*]Institut für Informatik, Freie Universität Berlin, Takustraße 9, D–14195 Berlin, Germany. {brass, knauer}@inf.fu-berlin.de

[†]This research was supported by the Deutsche Forschungsgemeinschaft under Grant No. BR 1465/5-1.

$(P_i, H_i)_{i=1}^k$ such that each $p, h$ with $p \in P_i, h \in H_i$ for some $i$ is an incident point-hyperplane pair, and each incident pair occurs in exactly one set pair, then we can determine all incident pairs in $\mathcal{O}\big((m+n)\log(m+n) + (mn)^{\frac{d}{d+1}}(\log(mn))^{\gamma_d}\big)$ time, where $\gamma_d$ is an appropriate constant of order $\mathcal{O}(d \log d)$. And we get even some additional structure information, since all our set pairs $(P_i, H_i)$ are of the form that there is an affine subspace $A_i$ with $P_i \subset A_i \subset \bigcap_{h \in H_i} h$.

So we obtain a compressed presentation for the incidence structure, which can be interpreted as a cover of the incidence graph by a set of complete bipartite subgraphs. Such compressions by biclique covers were already studied for graphs by Feder and Motwani [8], who showed that some graph algorithms can work efficiently also on such compressed representations, and were used by Agarwal and Varadarajan [2] in an intermediate step of a polygon approximation algorithm and by Agarwal et al. [1] as a compact representation of visibility graphs. We believe that such compressed presentations should also be possible for a number of other geometrically defined graphs with a bipartite extremal situation, e.g. the set of almost-unit-distances among $n$ points.

The problem is also very similar to Hopcroft's problem, which asks to determine the existence of at least one point-hyperplane incidence, whereas we wish to find all. Especially interesting in this connection is the model of Erickson [7] for lower bounds in a class of algorithms for Hopcroft's problem. There he also constructs systems of points and lines with a 'complicated' incidence structure which cannot be coded efficiently: his function $\zeta_d(n, m)$ is the maximum over all sets of $n$ points, $m$ hyperplanes of the size of the minimum coding of their incidence structure. He shows that for the 'counting' version of Hopcroft's problem that function $\zeta_d(n, m)$ is a lower bound for the complexity of any 'partitioning' algorithm. Our construction below improves also that bound on $\zeta_d$.

## 2 Results

**Definition 1 (Incidence encoding).** *A sequence of set pairs $\mathcal{E} = (P_i, H_i)_{i=1}^k$ such that each $p, h$ with $p \in P_i, h \in H_i$ for some $i$ is an incident point-hyperplane pair, and each incident pair occurs in exactly one set pair is called an* encoding *of of all point-hyperplane incidences between $P$ and $H$. The size of the encoding $\mathcal{E}$ is $\sum_{i=1}^k |P_i| + |H_i|$.*

An encoding is a covering of the incidence graph $\mathcal{G}(P, H)$ with the complete bipartite subgraphs $P_i \times H_i$.

**Theorem 1.** *For each dimension $d$ there is a constant $\gamma_d = \mathcal{O}(d \log d)$ such that the encoding of the incidence structure of any set $H$ of $m$ hyperplanes and any set $P$ of $n$ points in $d$-dimensional euclidean space $\mathbb{R}^d$ can be computed deterministically in $\mathcal{O}\big((m+n)\log(m+n) + (mn)^{\frac{d}{d+1}}(\log(mn))^{\gamma_d}\big)$ time.*

A lower bound is provided by the theorem

**Theorem 2.** *For all $m, n, d$ there is a set $H$ of $m$ hyperplanes and a set $P$ of $n$ points in $d$-dimensional euclidean space $\mathbb{R}^d$ for which the incidence graph $\mathcal{G}(P, H)$ cannot be encoded smaller than $\Omega\big(m + n + m^{\frac{d}{2d-1}} n^{\frac{2d-2}{2d-1}} + m^{\frac{2d-2}{2d-1}} n^{\frac{d}{2d-1}}\big)$.*

For $n > m$ this reduces to $\Omega\big(n + n^{\frac{2d-2}{2d-1}} m^{\frac{d}{2d-1}}\big) = \Omega\big(n + n^{1-\frac{1}{2d-1}} m^{\frac{1}{2}+\frac{1}{4d-2}}\big)$. This should be compared to Erickson's bound $\Omega\big(n + n^{\frac{2}{3}} m^{\frac{2}{3}} + n^{\frac{5}{6}} m^{\frac{1}{2}} + n^{\frac{9}{10}} m^{\frac{2}{5}} + n^{\frac{14}{15}} m^{\frac{1}{3}} + \cdots + n^{1-\frac{2}{d(d+1)}} m^{\frac{2}{d+1}}\big)$ (for $n > m$); the higher-dimensional cases of Erickson's construction become effective only if $n$ is much larger than $m$. In the diagonal case $m = n$ our construction gives $\Omega\big(n^{\frac{3}{2}-\frac{1}{4d-2}}\big)$, compared to Erickson's $\Omega\big(n^{\frac{4}{3}}\big)$.

Our construction gives a set of $n$ points, $m$ hyperplanes such that any $d$ hyperplanes intersect in at most one point, with $\Omega\big(n + m + n^{\frac{2d-2}{2d-1}} m^{\frac{d}{2d-1}}\big)$ incidences. With this intersection condition, the graph of point-hyperplane incidences is $K_{2,d}$-free, so there are at most $\mathcal{O}(n + m + n\sqrt{m})$ incidences. With the randomized partition argument of Clarkson et al. [6] this upper bound can be slightly improved to $\mathcal{O}\big(n + m + n^{\frac{2d-2}{2d-1}} m^{\frac{d}{2d-1}}\big)$, matching our construction. So the construction of theorem 2 is best possible under that intersection condition.

## 2.1 The Algorithm

The algorithm is based on a divide-and-conquer approach using cuttings, and a direct solution in very unbalanced cases. The relevant results on cuttings that we use are

**Definition (Cuttings).** *An $\epsilon$-cutting $\Xi$ for a set of $m$ hyperplanes $H$ in $\mathbb{R}^d$ and a parameter $\epsilon > 0$ is a finite set of* closed simplices *covering $\mathbb{R}^d$ with the property that any simplex is intersected in its* interior *by at most $\epsilon \cdot m$ hyperplanes from $H$.*

**Theorem (Chazelle, [5]).** *For* any *collection $H$ of $m$ hyperplanes in $\mathbb{R}^d$ and a parameter $r$, $1 < r \leq m$, a $(1/r)$-cutting $\Xi(H)$ consisting of $\mathcal{O}(r^d)$ simplices with* disjoint interiors *exists, and it can be computed* deterministically *in $\mathcal{O}(r^{d-1}m)$ time, together with an optimal data structure for point location in the cutting.*

**Definition (Point location data structure).** *A data structure $\mathcal{L}(H)$ supports* point location *queries to a set of $m$ hyperplanes $H$ in $d$-dimensional euclidean space $\mathbb{R}^d$ if we can determine the face $F(p)$ of the arrangement $\mathcal{A}(H)$, formed by the hyperplanes in $H$, that contains a query point $p \in \mathbb{R}^d$ in its relative interior*[1].

**Theorem (Chazelle, [5]).** *For* any *collection $H$ of $m$ hyperplanes in $\mathbb{R}^d$ a point location data structure $\mathcal{L}(H)$ of size $\mathcal{O}(m^d)$ can be computed* deterministically *in $\mathcal{O}(m^d)$ time, so that any point can be located in $\mathcal{O}(\log m)$ time.*

Let $\chi_d > 0$ be a constant such that the total number of simplices of all dimensions in a $1/r$-cutting in $d$-dimensional space is at most $\chi_d r^d$. The cutting computed by Chazelle's algorithm consists of at most $\chi_d r^d$ simplices of all dimensions where $\chi_d = C_0 \cdot (d \log d)^{2d^2} d^{2d}$ for an appropriate constant $C_0$ that is independent of $d$ (see [4], theorem 5.7.1 on page 75).

The algorithm now works as follows

**0.** Removal of trivial cases:

- **0.1** If $n, m < 10$ solve the problem directly by testing all pairwise incidences.
- **0.2** If $d = 1$, sort $H$ and $P$ and step through the sorted list to find the incidences.

**1.** If $m > n$, dualize the problem.

**2.** If $n > m^{d+1}$, solve the problem directly:

- **2.1** Construct the complete arrangement of the $m$ hyperplanes in $\mathcal{O}(m^d \log m) = \mathcal{O}(n^{\frac{d}{d+1}} \log n)$ time, and a point location structure for this arrangement.
- **2.2** Find for each of the $n$ points, using the point location structure, the face of the arrangement that contains this point in its relative interior, and collect for each face the list of all points in that face.
- **2.3** Go through the list of faces, reporting for each face the list of hyperplanes containing that face, and the list of points contained in that face.

**3.** Let $r := n^{\frac{1}{d+1}} m^{-\frac{1}{d(d+1)}}$. Compute a $(1/r)$-cutting $\Xi(H)$ for $H$ of size $\mathcal{O}(r^d)$, in $\mathcal{O}(r^{d-1}m)$ time, together with a point-location data structure $\mathcal{L}(\Xi)$ for the cutting. Let $(F_i)_{i=1}^k$ be the list of all $k \leq \chi_d r^d$ simplices (of all dimensions) of that cutting.

**4.** Determine for each point from $P$ one simplex $F_i$ from the cutting $\Xi$ which contains that point in its relative interior (if $F_i$ is not fuldimensional, there may be several possible simplices, since the cutting is not necessarily a simplicial complex: then any one will do).
This can be done by performing $n$ point-location queries to $\mathcal{L}(\Xi)$, which takes $\mathcal{O}(n \log r)$ time, and results in a partition of $P = \biguplus_{F_i \in \Xi} \mathrm{P}(F_i)$ ($\mathrm{P}(F_i)$ is the set of points in the relative interior of $F_i \in \Xi$).

[1]The face $F(p)$ is supposed to be specified by the set of hyperplanes that contain it.

**5.** For each simplex $F_i \in \Xi$ and each hyperplane $h \in H$, check in $\mathcal{O}(1)$ time if $h$ contains $F_i$, or if $h$ intersects $F_i$ in its relative interior (but does not contain it). Determine thus for each simplex $F_i$ the list $\mathrm{I}(F_i)$ of hyperplanes intersecting that simplex and the list $\mathrm{C}(F_i)$ of hyperplanes containing that simplex.
This results in $k \leq \chi_d r^d$ such lists of size at most $m$ each, and needs $\mathcal{O}(r^d m)$ time.

**6.** For each simplex $F_i \in \Xi$ perform the following steps:

**6.1** Report the set pair $(\mathrm{P}(F_i), \mathrm{C}(F_i))$.

**6.2** Apply the algorithm recursively to $\mathrm{P}(F_i)$ and $\mathrm{I}(F_i)$.

## 2.2 Analysis of the Algorithm

The correctness of our algorithm is obvious. So let $T(n, m, d)$ be the maximum running time of this algorithm on any set of $n$ points and $m$ hyperplanes. We wish to prove that $T(n, m, d) \leq \log(m+n) S(n, m, d)$ with $S(n, m, d) := \alpha_d (m+n) + \beta_d (mn)^{\frac{d}{d+1}} (\log(mn))^{\gamma_d}$ for some properly chosen $\alpha_d$, $\beta_d$, $\gamma_d$ and all sufficiently big $n$, $m$.

Let $P$, $H$ be a set of $n$ points and $m$ hyperplanes, respectively, that maximize the running time of the algorithm, and consider the time needed by the algorithm. Steps 0, and 1 can obviously be done in $\mathcal{O}(n \log n + m \log m)$ time. The same holds for step 2, since the construction of the complete hyperplane arrangement with at most $m^d$ cells takes at most $\mathcal{O}(m^d \log m) = \mathcal{O}(n^{\frac{d}{d+1}} \log n)$ time, including the construction of a point location structure for that arrangement. Then the distribution of the $n$ points on the faces of the arrangement (step 2.2) can be done in time $\mathcal{O}(\log m) = \mathcal{O}(\log n)$ per point, and the collection of all set pairs (step 2.3) can be done in $\mathcal{O}(m^{d+1}) = \mathcal{O}(n)$ time (just testing for each face all $m$ hyperplanes whether they contain that face). This gives a total of $\mathcal{O}(n \log n)$.

Now after the possible dualization in step 1. we can assume that $n > m$, and if $n > m^{d+1}$, then the algorithm stops after step 2., and satisfies our claimed complexity bound. So in the following we can assume that $m < n < m^{d+1}$.

Step 3 needs $\mathcal{O}(r^{d-1} m)$ time (construction of the cutting), step 4 needs $\mathcal{O}(n \log r)$ (distribution of the points on the simplices), and step 5 needs $\mathcal{O}(r^d m)$ (distribution of the hyperplanes on the simplices). Let $n_i$ denote the number of points in the relative interior of the simplex $F_i$ and $m_i$ be the number of hyperplanes intersecting $F_i$ ($n_i = |\mathrm{P}(F_i)|$, $m_i = |\mathrm{I}(F_i)|$), then the recursive calls in step 6 need at most $\sum_{i=1}^{k} T(n_i, m_i, d)$ time. Thus we have the recursive bound for $T$

$$T(n, m, d) \leq A(n \log n + m \log m) + B r^d m + \sum_{i=1}^{k} T(n_i, m_i, d)$$

with $k \leq \chi_d r^d$ for some constants $A, B$.

Since $(\mathrm{P}(F_i))_{i=1}^{k}$ is a partition of $P$, we have $\sum_{i=1}^{k} n_i = n$, and by the cutting property we have $m_i \leq \frac{m}{r}$ for each full-dimensional simplex $F_i$. But if a hyperplane intersects a simplex $F$ in its relative interior, and $F$ is a face of the full-dimensional simplex $F'$, then it intersects also $F'$ in it interior (since it separates some vertices of $F$, which are also vertices of $F'$). Thus we have $m_i \leq \frac{m}{r}$ for all $i$, even if the simplex $F_i$ is not full-dimensional.

Now we proceed by induction, using monotonicity and concavity of the function $S(n, m, d)$ (Jensen's inequality) and monotonicity of $kS(\frac{n}{k}, m, d)$ (in $k$) as well as $n \geq m$ and $\chi_d \geq 1$: then the sum is bounded by

$$\begin{aligned}
\sum_{i=1}^{k} T(n_i, m_i, d) &\le \sum_{i=1}^{k} \log(m_i + n_i) S(n_i, m_i, d) \le \log(m+n) \sum_{i=1}^{k} S\left(n_i, \frac{m}{r}, d\right) \\
&\le \log(m+n) k S\left(\frac{n}{k}, \frac{m}{r}, d\right) \le \log(m+n) \chi_d r^d S\left(\frac{n}{\chi_d r^d}, \frac{m}{r}, d\right) \\
&\le \log(m+n) \chi_d r^d \alpha_d \left(\frac{n}{\chi_d r^d} + \frac{m}{r}\right) \\
&\quad + \log(m+n) \chi_d r^d \beta_d \left(\frac{mn}{\chi_d r^{d+1}}\right)^{\frac{d}{d+1}} \left(\log\left(\frac{mn}{\chi_d r^{d+1}}\right)\right)^{\gamma_d} \\
&\le \alpha_d n \log(m+n) + \alpha_d \chi_d (mn)^{\frac{d}{d+1}} \frac{\log(m+n)}{r} \\
&\quad + \beta_d \chi_d^{\frac{1}{d+1}} (mn)^{\frac{d}{d+1}} \left(\log\left(\frac{m^{\frac{d+1}{d}}}{\chi_d}\right)\right)^{\gamma_d} \log(m+n)
\end{aligned}$$

Now we insert this in the recursive bound and use that for $m < n < m^{d+1}$ we have $n < (mn)^{\frac{d}{d+1}}$.

$$\begin{aligned}
T(n,m,d) &\le A(n \log n + m \log m) + B r^d m + \sum_{i=1}^{k} T(n_i, m_i, d) \\
&\le 2An \log n + B(mn)^{\frac{d}{d+1}} + \alpha_d n \log(m+n) \\
&\quad + \alpha_d \chi_d (mn)^{\frac{d}{d+1}} \frac{\log(m+n)}{r} + \beta_d \chi_d^{\frac{1}{d+1}} (mn)^{\frac{d}{d+1}} \left(\log\left(\frac{m^{\frac{d+1}{d}}}{\chi_d}\right)\right)^{\gamma_d} \log(m+n) \\
&\le \log(m+n) \Big( 2A(mn)^{\frac{d}{d+1}} + \frac{B}{\log(m+n)} (mn)^{\frac{d}{d+1}} + \alpha_d (mn)^{\frac{d}{d+1}} + \alpha_d \chi_d (nm)^{\frac{d}{d+1}} \frac{1}{r} \\
&\quad + \beta_d \chi_d^{\frac{1}{d+1}} (mn)^{\frac{d}{d+1}} \left(\log\left((mn)^{\frac{d+1}{2d}}\right)\right)^{\gamma_d} \Big) \\
&\le \log(m+n) (mn)^{\frac{d}{d+1}} \\
&\quad \left(2A + \frac{B}{\log(m+n)} + \alpha_d + \frac{\alpha_d \chi_d}{r} + \beta_d \chi_d^{\frac{1}{d+1}} \left(\frac{d+1}{2d}\right)^{\gamma_d} (\log(mn))^{\gamma_d}\right) \\
&\le \log(m+n) (mn)^{\frac{d}{d+1}} (\log(mn))^{\gamma_d} \\
&\quad \left( (2A + B + \alpha_d + \alpha_d \chi_d)(\log(mn))^{-\gamma_d} + \beta_d \chi_d^{\frac{1}{d+1}} \left(\frac{d+1}{2d}\right)^{\gamma_d} \right).
\end{aligned}$$

Using $mn > 10$ (or any arbitrary lower bound) gives

$$\begin{aligned}
T(n,m,d) &\le \log(m+n) (mn)^{\frac{d}{d+1}} (\log(mn))^{\gamma_d} \\
&\quad \left( (2A + B + \alpha_d + \alpha_d \chi_d)(\log(10))^{-\gamma_d} + \beta_d \chi_d^{\frac{1}{d+1}} \left(\frac{d+1}{2d}\right)^{\gamma_d} \right) \\
&\le \log(m+n) \beta_d (mn)^{\frac{d}{d+1}} (\log(mn))^{\gamma_d} < \log(m+n) S(n,m,d)
\end{aligned}$$

if $\beta_d, \gamma_d$ are chosen sufficiently big such that

$$(2A + B + \alpha_d + \alpha_d \chi_d)(\log(10))^{-\gamma_d} + \beta_d \chi_d^{\frac{1}{d+1}} \left(\frac{d+1}{2d}\right)^{\gamma_d} \le \beta_d.$$

Note that for $\gamma_d > \frac{\log\left(\chi_d^{\frac{1}{d+1}}\right)}{\log\left(\frac{2d}{d+1}\right)} = \mathcal{O}(d \log d)$ we can chose any $\beta_d > \frac{\left(2A+B+\alpha_d+\alpha_d\chi_d\right)}{(\log(10))^{\gamma_d}\left(1-\chi_d^{\frac{1}{d+1}}\left(\log\left(\frac{d+1}{2d}\right)\right)^{\gamma_d}\right)}$ to satisfy the above inequality. This proves theorem 1.

## 3 The Lower Bound

In the following we construct a set of $n$ points and $m$ hyperplanes with $\Omega\big(m + n + m^{\frac{d}{2d-1}} n^{\frac{2d-2}{2d-1}} + m^{\frac{2d-2}{2d-1}} n^{\frac{d}{2d-1}}\big)$ incidences that does not permit a small encoding. We exclude the possibility of a small encoding by choosing the hyperplanes in such a way that any $d$ hyperplanes intersect at most in a single point; thus any bipartite subconfiguration $(P_i, H_i)$ in our encoding has either $|P_i| \leq 1$ or $|H_i| \leq d-1$, so the size of the encoding is at least $\frac{1}{d-1}$ of the number of incidences.

As points we choose the $d$-dimensional integer lattice cube $\{1, \ldots, \nu\}^d$ for $\nu = n^{\frac{1}{d}}$.

As hyperplanes we use all hyperplanes that have a nonempty intersection with this lattice cube, and that have integer normal vectors from a set we are going to construct. We select these normal vectors from the integer lattice cube $\{1, \ldots, \mu\}^d$ in such a way that any $d$ of them are linearly independent. Bárány, Harcos, Pach and Tardos proved

**Theorem (Bárány et al., [3]).** *The maximum number of lattice vectors that can be chosen from $\{1, \ldots, \mu\}^d$ in such a way that any $d$ of them are linearly independent ist $\Theta(\mu^{\frac{d}{d-1}})$*

We choose such a maximum-cardinality set of $\rho_d \mu^{\frac{d}{d-1}}$ vectors. Then for each vector there are at most $d\mu\nu$ hyperplanes with that normal vector that have nonempty intersection with the lattice cube $\{1, \ldots, \nu\}^d$, since for each point the inner product is a positive integer of at most $d\mu\nu$. Thus we have a total of at most $\rho_d \mu^{\frac{d}{d-1}} d\mu\nu$ hyperplanes. And for each normal vector, there is exactly one incidence between each point and a hyperplane with that normal vector; this gives a total $\rho_d \mu^{\frac{d}{d-1}} n$ incidences. Choosing $\mu = \left(\frac{1}{\rho_d d} m n^{-\frac{1}{d}}\right)^{\frac{d-1}{2d-1}}$ we obtain sets of $n$ points, $m$ hyperplanes, with $\rho_d \big(\frac{1}{d\rho_d} m\big)^{\frac{d}{2d-1}} n^{\frac{2d-2}{2d-1}}$ incidences. Dualizing this construction, and adding points incident to a single hyperplane or hyperplanes through a single point, gives the remaining terms of the lower bound claimed in theorem 2.

It should be noted that this construction is just one extreme case of a whole family: if we choose the points from the lattice cube in such a way that each *affine* $i$-dimensional subspace contains at most $\mathcal{O}(1)$ points, and the hyperplane normals their lattice cube in such a way that any *linear* $d-i$-dimensional subspace contains at most $\mathcal{O}(1)$ normal vectors, then again we find a set with many incidences that does not admit a small encoding. But the maximum size of such subsets is known only in the $i = 0$ case used here.

## References


[1] P.K. Agarwal, N. Alon, B. Aronov and S. Suri Can visibility graphs be represented compactly? *Discrete Comput. Geom.*, 12(3):347–365, 1994.

[2] P.K. Agarwal and K.R. Varadarajan. Efficient algorithms for approximating polygonal chains. *Discrete Comput. Geom.*, 23(2):273–291, 2000.

[3] I. Bárány, G. Harcos, J. Pach, and G. Tardos. Covering lattice points by subspaces. submitted to Period. Math. Hung.

[4] H. Brönnimann. *Derandomization of Geometric Algorithms.* Ph.D. thesis, Dept. Comput. Sci., Princeton University, Princeton, NJ, May 1995.

[5] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, 1993.

[6] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5(2):99–160, 1990.

[7] J. Erickson. New lower bounds for Hopcroft's problem. *Discrete Comput. Geom.*, 16:389–418, 1996.

[8] T. Feder and R. Motwani. Clique partitions, graph compression and speeding-up algorithms. *J. Comput. Syst. Sci.*, 51(2):261–272, Oct. 1995.