Subquadratic Encodings for Point Configurations*

Jean Cardinal†1**, Timothy M. Chan**‡2**, John Iacono**§1**,**
Stefan Langerman¶1**, and Aurélien Ooms**‖1

1 **Département d'Informatique, Université libre de Bruxelles (ULB), Belgium**
2 **Department of Computer Science, University of Illinois at Urbana-Champaign, USA**

Abstract

For many algorithms dealing with sets of points in the plane, the only relevant information carried by the input is the combinatorial configuration of the points: the orientation of each triple of points in the set (clockwise, counterclockwise, or collinear). This information is called the *order type* of the point set. In the dual, realizable order types and abstract order types are combinatorial analogues of line arrangements and pseudoline arrangements. Too often in the literature we analyze algorithms in the real-RAM model for simplicity, putting aside the fact that computers as we know them cannot handle arbitrary real numbers without some sort of encoding. Encoding an order type by the integer coordinates of some realizing point set is known to yield doubly exponential coordinates in some cases. Other known encodings can achieve quadratic space or fast orientation queries, but not both. In this contribution, we give a compact encoding for abstract order types that allows efficient query of the orientation of any triple: the encoding uses $O(n^2)$ bits and an orientation query takes $O(\log n)$ time in the word-RAM model. This encoding is space-optimal for abstract order types. We show how to shorten the encoding to $O(n^2(\log\log n)^2/\log n)$ bits for realizable order types, giving the first subquadratic encoding for those order types with fast orientation queries. We further refine our encoding to attain $O(\log n/\log\log n)$ query time at the expense of a negligibly larger space requirement. In the realizable case, we show that all those encodings can be computed efficiently. Finally, we generalize our results to the encoding of point configurations in higher dimension.

1 Introduction

At SoCG'86, Chazelle asked [16]:

> "How many bits does it take to know an order type?"

This question is of importance in Computational Geometry for the following two reasons: First, in many algorithms dealing with sets of points in the plane, the only relevant information carried by the input is the combinatorial configuration of the points given by the orientation of each triple of points in the set (clockwise, counterclockwise, or collinear) [7]. Second, computers as we know them can only handle numbers with finite description and we cannot

---

* Details, proofs, and improvements in the arXiv version [5].

† `jcardin@ulb.ac.be`. Supported by the "Action de Recherche Concertée" (ARC) COPHYMA, convention number 4.110.H.000023.

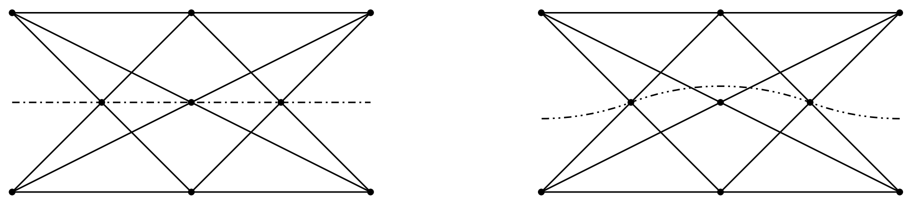
‡ `tmc@illinois.edu`.

§ `eurocg18@johniacono.com`. Supported by NSF grants CCF-1319648, CCF-1533564, CCF-0430849 and MRI-1229185, a Fulbright Fellowship and by the Fonds de la Recherche Scientifique-FNRS under Grant n° MISU F 6001 1 and two Missions Scientifiques.

¶ `slanger@ulb.ac.be`. Directeur de recherches du Fonds de la Recherche Scientifique-FNRS.

‖ `aureooms@ulb.ac.be`. Supported by the Fund for Research Training in Industry and Agriculture (FRIA).

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.



(a) Realizable order type. **(b)** Abstract order type which is not realizable.

Figure 1 Pappus's configuration.

assume that they can handle arbitrary real numbers without some sort of encoding. The study of *robust* algorithms is focused on ensuring the correct solution of problems on finite precision machines. Chapter 41 of The Handbook of Discrete and Computational Geometry is dedicated to this issue [23].

The (counterclockwise) orientation $\nabla(p, q, r)$ of a triple of points p, q, and r with coordinates (x_p, y_p), (x_q, y_q), and (x_r, y_r) is the sign of the determinant

$$\begin{vmatrix} 1 & x_p & y_p \\ 1 & x_q & y_q \\ 1 & x_r & y_r \end{vmatrix}.$$

Given a set of n labeled points $P = \{ p_1, p_2, \ldots, p_n \}$, we define the *order type* of P to be the function $\chi\colon [n]^3 \to \{ -, 0, + \}\colon (a, b, c) \mapsto \nabla(p_a, p_b, p_c)$ that maps each triple of point labels to the orientation of the corresponding points, up to isomorphism. The order type of a point set has been further abstracted into combinatorial objects known as (rank-three) *oriented matroids* [10]. The *chirotope axioms* define consistent systems of signs of triples [3]. From the topological representation theorem [4], all such *abstract* order types correspond to pseudoline arrangements, while, from the standard projective duality, order types of point sets correspond to straight line arrangements. See Chapter 6 of The Handbook for more details [21].

When the order type of a pseudoline arrangement can be realized by an arrangement of straight lines, we call the pseudoline arrangement *stretchable*. As an example of a nonstretchable arrangement, Levi gives Pappus's configuration where eight triples of concurrent straight lines force a ninth, whereas the ninth triple cannot be enforced by pseudolines [19] (see Figure 1). Ringel shows how to convert the so-called "non-Pappus" arrangement of Figure 1 (b) to a simple arrangement while preserving nonstretchability [22]. All arrangements of eight or fewer pseudolines are stretchable [13], and the only nonstretchable simple arrangement of nine pseudolines is the one given by Ringel [20]. More information on pseudoline arrangements is available in Chapter 5 of The Handbook [11].

Figure 1 shows that not all pseudoline arrangements are stretchable. Indeed, most are not: there are $2^{\Theta(n^2)}$ abstract order types [8] and only $2^{\Theta(n \log n)}$ realizable order types [1, 15].

Information theory implies that we need quadratic space for abstract order types whereas we only need linearithmic space for realizable order types. Hence, storing all $\binom{n}{3}$ orientations in a lookup table seems wasteful. Another obvious idea for storing the order type of a point set is to store the coordinates of the points, and answer orientation queries by computing the corresponding determinant. While this should work in many practical settings, it cannot work for all point sets. Perles's configuration shows that some configuration of points, containing collinear triples, forces at least one coordinate to be irrational [18]. Order types of points in general position can always be represented by rational coordinates. It is well known, however, that some configurations require doubly exponential coordinates, hence coordinates with exponential bitsizes if represented in the normal way [17].

Goodman and Pollack defined λ-matrices which can encode abstract order types using $O(n^2 \log n)$ bits [14]. They asked if the space requirements could be moved closer to the information-theoretic lower bounds. Felsner and Valtr showed how to encode abstract order types optimally in $O(n^2)$ bits via the wiring diagram of their corresponding allowable sequence [8, 9] (as defined in [12]). Aloupis et al. gave an encoding of size $O(n^2)$ that can be computed in $O(n^2)$ time and that can be used to test for the isomorphism of two distinct point sets in the same amount of time [2]. However, it is not known how to decode the orientation of one triple from any of those encodings in, say, sublinear time. Moreover, since the information-theoretic lower bound for realizable order types is only $\Omega(n \log n)$, we must ask if this space bound is approachable for those order types while keeping orientation queries reasonably efficient.

Our Results

In this contribution, we are interested in *compact* encodings for order types: we wish to design data structures using as few bits as possible that can be used to quickly answer orientation queries of a given abstract or realizable order type.

▶ **Definition 1.1.** For fixed k and given a function $f : [n]^k \to [O(1)]$, we define a $(S(n), Q(n))$-encoding of f to be a string of $S(n)$ bits such that, given this string and any $t \in [n]^k$, we can compute $f(t)$ in $Q(n)$ query time in the word-RAM model.

We give the first optimal encoding for abstract order types that allows efficient query of the orientation of any triple: the encoding is a data structure that uses $O(n^2)$ bits of space with queries taking $O(\log n)$ time in the word-RAM model.

▶ **Theorem 1.2.** *All abstract order types have an $(O(n^2), O(\log n))$-encoding.*

Our encoding is far from being space-optimal for realizable order types. We show that its construction can be easily tuned to only require $O(n^2 (\log \log n)^2 / \log n)$ bits in this case.

▶ **Theorem 1.3.** *All realizable order types have an $(O(\frac{n^2(\log\log n)^2}{\log n}), O(\log n))$-encoding.*

We further refine our encoding to reduce the query time to $O(\log n / \log \log n)$.

▶ **Theorem 1.4.** *All abstract order types have an $(O(n^2), O(\frac{\log n}{\log\log n}))$-encoding.*

▶ **Theorem 1.5.** *All realizable order types have a $(O(\frac{n^2 \log^{\varepsilon} n}{\log n}), O(\frac{\log n}{\log\log n}))$-encoding.*

In the realizable case, we give quadratic upper bounds on the preprocessing time required to compute an encoding in the real-RAM model.

▶ **Theorem 1.6.** *In the real-RAM model and the constant-degree algebraic decision tree model, given n real-coordinate input points in $\mathbb{R}^2$ we can compute the encoding of their order type as in Theorems 1.4 and 1.5 in $O(n^2)$ time.*

We generalize our encodings for chirotopes of point sets in higher dimension.

▶ **Theorem 1.7.** *All realizable chirotopes of rank $k \geq 4$ have a $(O(\frac{n^{k-1}(\log\log n)^2}{\log n}), O(\frac{\log n}{\log\log n}))$-encoding.*

▶ **Theorem 1.8.** *In the real-RAM model and the constant-degree algebraic decision tree model, given n real-coordinate input points in $\mathbb{R}^d$ we can compute the encoding of their chirotope as in Theorem 1.7 in $O(n^d)$ time.*

Our data structure is the first subquadratic encoding for realizable order types that allows efficient query of the orientation of any triple. It is not known whether a subquadratic constant-degree algebraic decision tree exists for the related problem of deciding whether a point set contains a collinear triple. Any such decision tree would yield another subquadratic encoding for realizable order types. We see the design of compact encodings for realizable order types as a subgoal towards subquadratic nonuniform algorithms for this related problem, a major open problem in Computational Geometry. Note that pushing the preprocessing time below quadratic would yield such an algorithm.

2 Encoding Order Types via Hierarchical Cuttings

We assume that we can access some arrangement of lines or pseudolines that realizes the order type we want to encode. We thus exclusively focus on the problem of encoding the order type of a given arrangement. This does not pose a threat against the existence of an encoding. In this extended abstract, we sketch the general idea for a simple subquadratic encoding. For full details, proofs, and improvements, we refer to the arXiv version [5].

Hierarchical Cuttings

We encode the order type of an arrangement via hierarchical cuttings as defined in [6]. A cutting in $\mathbb{R}^d$ is a set of (possibly unbounded and/or non-full dimensional) constant-complexity cells that together partition $\mathbb{R}^d$. A $\frac{1}{r}$-cutting of a set of n hyperplanes is a cutting with the constraint that each of its cells is intersected by at most $\frac{n}{r}$ hyperplanes. There exist various ways of constructing $\frac{1}{r}$-cuttings of size $O(r^d)$. In the plane, hierarchical cuttings can be constructed for arrangement of pseudolines with the same properties.

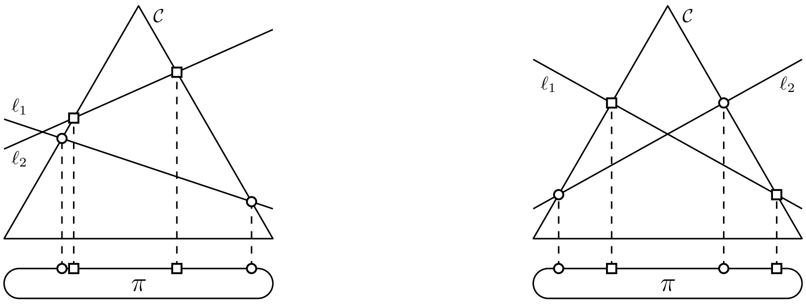
Idea

We want to preprocess n pseudolines $\{ \ell_1, \ell_2, \dots, \ell_n \}$ in the plane so that, given three indices a, b, and c, we can compute their orientation, that is, whether the intersection $\ell_a \cap \ell_b$ lies above, below or on ℓ_c. Our data structure builds on cuttings as follows: Given a cutting Ξ and the three indices, we can locate the intersection of ℓ_a and ℓ_b inside Ξ. The location of this intersection is a cell of Ξ. The next step is to decide whether ℓ_c lies above, lies below, contains or intersects that cell. In the first three cases, we are done. Otherwise, we can answer the query by recursing on the subset of pseudolines intersecting the cell containing the intersection. We build on hierarchical cuttings to solve all subproblems efficiently.

Intersection Location

When the ℓ_a are straight lines, locating the intersection $\ell_a \cap \ell_b$ in Ξ is trivial if we know the real parameters of ℓ_a and ℓ_b and of the descriptions of the subcells of Ξ. However, in our model we are not allowed to store real numbers. To circumvent this annoyance, and to handle arrangements of pseudolines, we make a simple observation illustrated by Figure 2.

▶ Observation 1. Two pseudolines ℓ_a and ℓ_b intersect in the interior of a full-dimensional cell $\mathcal{C}$ if and only if each pseudoline properly intersects the boundary of $\mathcal{C}$ exactly twice and their intersections with its boundary alternate.



(a) $\ell_a \cap \ell_b \cap \mathcal{C} = \emptyset$. **(b)** $\ell_a \cap \ell_b \cap \mathcal{C} \neq \emptyset$.

Figure 2 Cyclic permutations (π).

This gives us a way to encode the location of the intersection of ℓ_a and ℓ_b in Ξ using only bits. We define the *cyclic permutation* of a full-dimensional cell $\mathcal{C}$ and a finite set of pseudolines $\mathcal{L}$ to be the finite sequence of properly intersecting pseudolines from $\mathcal{L}$ encountered when walking along the boundary of $\mathcal{C}$ in clockwise or counterclockwise order, up to rotation and reversal. Location in a non-full-dimensional cell can be encoded similarily.

Encoding

Given n pseudolines in the plane and some fixed parameter r, compute a hierarchical $\frac{1}{r}$-cutting of those pseudolines. This hierarchical cutting consists of ℓ levels labeled $0, 1, \ldots, \ell - 1$. Level i has $O(r^{2i})$ cells. Each of those cells is further partitioned into $O(r^2)$ subcells. The $O(r^{2(i+1)})$ subcells of level i are the cells of level $i+1$. Each cell of level i is intersected by at most $\frac{n}{r^i}$ pseudolines, and hence each subcell is intersected by at most $\frac{n}{r^{i+1}}$ pseudolines.

We compute and store a combinatorial representation of the hierarchical cutting as follows: For each level of the hierarchy, for each cell in that level, for each pseudoline intersecting that cell, for each subcell of that cell, we store two bits to indicate the location of the pseudoline with respect to that subcell. When a pseudoline intersects the interior of a 2-dimensional subcell, we also store the two indices of the intersections of that pseudoline with the subcell in the cyclic permutation associated with that subcell, beginning at an arbitrary location in, say, clockwise order. Location in a non-full-dimensional subcell can be encoded similarily.

The hierarchy is such that each subcell of the last level is intersected by no more than $t = \frac{n}{r^\ell}$ pseudolines. For those subcells, we answer the queries by table lookup. The use of hierarchical cuttings essentially guarantees we get quadratic preprocessing time, quadratic space, and logarithmic query time in the abstract case. In the realizable case, we know there can only be $2^{O(t \log t)}$ distinct lookup tables. Choosing the right superconstant t leads to subquadratic space in that case.

References

1 Noga Alon. The number of polytopes configurations and real matroids. *Mathematika*, 33(1):62–71, 1986.

2 Greg Aloupis, John Iacono, Stefan Langerman, Özgür Özkan, and Stefanie Wuhrer. The complexity of order type isomorphism. In *SODA*, pages 405–415. SIAM, 2014.

3 Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M Ziegler. Oriented matroids. In *Encyclopedia of Mathematics*, volume 46. Cambridge University Press, 1993.

4 Jürgen Bokowski, Susanne Mock, and Ileana Streinu. On the Folkman-Lawrence topological representation theorem for oriented matroids of rank 3. *European Journal of Combinatorics*, 22(5):601–615, 2001.

5 Jean Cardinal, Timothy M. Chan, John Iacono, Stefan Langerman, and Aurélien Ooms. Subquadratic encodings for point configurations. *ArXiv e-prints*, 2018. arXiv:1801.01767 [cs.CG].

6 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete & Computational Geometry*, 9:145–158.

7 Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10. Springer Science & Business Media, 2012.

8 Stefan Felsner. On the number of arrangements of pseudolines. In *SOCG*, pages 30–37. ACM, 1996.

9 Stefan Felsner and Pavel Valtr. Coding and counting arrangements of pseudolines. *Discrete & Computational Geometry*, 46(3):405–416, 2011.

10 Jon Folkman and Jim Lawrence. Oriented matroids. *Journal of Combinatorial Theory, Series B*, 25(2):199–236, 1978.

11 Jacob E. Goodman. Pseudoline arrangements. In *Handbook of Discrete and Computational Geometry, 2nd Ed.*, pages 97–128. Chapman and Hall/CRC.

12 Jacob E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Mathematics*, 32(1):27–35, 1980.

13 Jacob E. Goodman and Richard Pollack. Proof of Grünbaum's conjecture on the stretchability of certain arrangements of pseudolines. *Journal of Combinatorial Theory, Series A*, 29(3):385–390, 1980.

14 Jacob E. Goodman and Richard Pollack. Multidimensional sorting. *SIAM Journal on Computing*, 12(3):484–507, 1983.

15 Jacob E. Goodman and Richard Pollack. Upper bounds for configurations and polytopes in $\mathbb{R}^d$. *Discrete & Computational Geometry*, 1:219–227, 1986.

16 Jacob E. Goodman and Richard Pollack. Allowable sequences and order types in discrete and computational geometry. In *New Trends in Discrete and Computational Geometry*, pages 103–134. Springer, 1993.

17 Jacob E. Goodman, Richard Pollack, and Bernd Sturmfels. Coordinate representation of order types requires exponential storage. In *STOC*, pages 405–410. ACM, 1989.

18 Branko Grünbaum. *Convex Polytopes*. Springer, 2005.

19 Friedrich Levi. Die teilung der projektiven ebene durch gerade oder pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss*, 78:256–267, 1926.

20 Jürgen Richter. Kombinatorische realisierbarkeitskriterien für orientierte matroide. *Mitt. Math. Sem. Univ. Giessen*, 194:1–112.

21 Jürgen Richter-Gebert and Günter M. Ziegler. Oriented matroids. In *Handbook of Discrete and Computational Geometry, 2nd Ed.*, pages 129–151. Chapman and Hall/CRC, 2004.

22 Gerhard Ringel. Teilungen der ebene durch geraden oder topologische geraden. *Mathematische Zeitschrift*, 64(1):79–102, 1956.

23 Chee K. Yap. Robust geometric computation. In *Handbook of Discrete and Computational Geometry, 2nd Ed.*, pages 927–952. Chapman and Hall/CRC.