

- [GP92] L.J. Guibas and M. Pellegrini. New algorithmic results for lines-in-3-space problems. Technical Report TR-92-005, International Computer Science Institute, January 1992.
- [Mat90] J. Matoušek. Cutting hyperplane arrangements. In *Proceedings of the 6th ACM Symposium on Computational Geometry*, pages 1–9, 1990.
- [Mit70] D.S. Mitrinovic. *Analytic inequalities*. Springer Verlag, 1970.
- [Mul88] K. Mulmuley. A fast planar partition algorithm. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, 1988.
- [Pel91] M. Pellegrini. Ray shooting and isotopy classes of lines in 3-dimensional space. In *Proceedings of the 1991 Workshop on Algorithms and Data Structures*, number 519 in Lecture Notes in Computer Science, pages 20–31. Springer Verlag, 1991.

- [AS91a] P. K. Agarwal and M. Sharir. Applications of a new space partitioning technique. In *Proceedings of the 1991 Workshop on Algorithms and Data Structures*, number 519 in Lecture Notes in Computer Science, pages 379–391. Springer Verlag, 1991.
- [AS91b] P. K. Agarwal and M. Sharir. Counting circular arc intersections. In *Proceedings of the 7th ACM Symposium on Computational Geometry*, pages 10–20, 1991.
- [BO79] J.L. Bentley and T. Ottman. Algorithms for reporting and counting geometric intersections. *IEEE Trans. on Computers*, C-28:643–647, 1979.
- [CE88] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. Technical Report CS-TR-148-88, University of Illinois, April 1988.
- [CEG⁺90] K.L. Clarkson, H. Edelsbrunner, L.J. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete & Computational Geometry*, 5:99–160, 1990.
- [CEGS89] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. A singly exponential stratification scheme for real semi-algebraic varieties and its applications. In *Proceedings of the 16th International Colloquium on Automata, languages and Programming*, number 372 in Lecture Notes in Computer Science, pages 179–193. Springer Verlag, 1989.
- [CEGS90] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Algorithms for bichromatic line segments problems and polyhedral terrains. (UIUCDCS-R-90-1578), March 1990.
- [Cla87] K.L. Clarkson. New applications of random sampling in computational geometry. *Discrete & Computational Geometry*, 2:195–222, 1987.
- [CS89] K. Clarkson and P. Shor. Applications of random sampling in computational geometry II. *Discrete & Computational Geometry*, 4:387–422, 1989.
- [CSW90] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. In *Proceedings of the 6th ACM Symposium on Computational Geometry*, pages 23–33, 1990.
- [EGS88] H. Edelsbrunner, L. Guibas, and M. Sharir. The complexity of many faces in arrangements of lines and segments. In *Proceedings of the 4th ACM Symposium on Computational Geometry*, pages 44–55, 1988.
- [EMP⁺82] H. Edelsbrunner, H. Mauer, F. Preparata, E. Welzl, and D. Wood. Stabbing line segments. *BIT*, (22):274–281, 1982.
- [GOS89] L. Guibas, M. Overmars, and M. Sharir. Counting and reporting intersections in arrangements of line segments. Technical Report TR434, Courant Institute, 1989.

Lemma 5 ([GP92]) *Given a sphere S and a segment e , the following holds:*

- A) *$lhp(e) \in S$ and $rhp(e) \notin S$ imply $e \cap S \neq \emptyset$ and $|e \cap S| = 1$.*
- B) *$rhp(e) \in S$ and $lhp(e) \notin S$ imply $e \cap S \neq \emptyset$ and $|e \cap S| = 1$.*
- C) *$lhp(e) \notin S$ and $rhp(e) \notin S$ and $center(S) \in W(e)$ and $S \cap l(e) \neq \emptyset$ imply $e \cap S \neq \emptyset$ and $|e \cap S| = 2$.*
- D) *$e \cap S \neq \emptyset$ implies that one and only one of A, B and C is true.*

Proof. Follows from elementary geometry. ■

The analysis in [GP92] gives a characteristic dimension $d = 5$. Using Lemma 5 and the nested batching technique it is easy to derive the following theorem:

Theorem 4 *Given n segments and m spheres in R^3 it is possible to count all the segment-sphere intersections in expected time $O(n^{5/6+\epsilon}m^{5/6} + m^{1+\epsilon} + n \log^6 m)$.*

8 Conclusions

We have shown a variation on the batching technique of [EGS88] called *nested batching technique*. This technique can be used under quite general assumption to count the pairs of geometric objects satisfying a logical predicate. We have given an application of this technique to counting the number of intersections of a collection of circular arcs on the plane. The resulting algorithm is asymptotically faster than the previously known algorithm. For the special case of counting intersections of arcs of unit discs the algorithm almost matches the best known algorithm for counting intersections of segments. The technique is generalizable to deal with other intersections counting problems.

9 Acknowledgments

Thanks to Pankaj Agarwal for his comments on an early version.

References

- [Aga89] P.K. Agarwal. A deterministic algorithm for partitioning arrangements of lines and its applications. In *Proceedings of the 5th ACM Symposium on Computational Geometry*, pages 11–22, 1989.
- [Aga90] P.K. Agarwal. Partitioning arrangements of lines II: Applications. *Discrete & Computational Geometry*, 5:533–573, 1990.
- [Aga91] P.K. Agarwal. Geometric partitioning and its applications. Technical Report CS-1991-27, Dept. of Computer Sci. Duke University, July 1991.

$d''(\text{points}, 3) = 3$. Condition (4) is similar to (3). Conditions (5) and (6) are similar to condition (1).

A circle can be mapped to a point $\varphi(C) = (a, b, r)$ in 3-dimensional parametric space. Also, as shown in [AS91b], a circle C' can be mapped into a region $\Psi(C')$ such that C meets C' if and only if $\varphi(C) \in \Psi(C')$ (Lemma 2.2 in [AS91b]). The boundary of $\Psi(C')$ consists of two surfaces ψ_I and ψ_O it is easy to show that the region $\Phi(C')$ corresponds to the points on the positive side of one surface and on the negative side of the other surface. Therefore we define as our polynomial the product of the polynomials defining ψ_I and ψ_O taking a the “positive” side the region where this polynomial has a negative valuation. Using this trick¹ and the decomposition method in [CEGS89] we obtain $d''(\text{circles}, 7) = d''(\text{circles}, 7) = 3$.

Since counting circular arc intersections has been reduced to a nested batched problem of characteristic dimension 3 we obtain:

Theorem 2 *Given n circular arcs, it is possible to count all intersections in expected time $O(n^{3/2+\epsilon})$.*

It is possible to tailor the proof of Theorem 1 in order to prove directly Theorem 2 by using an algorithm to test condition (7) in time $O(m^{3/4}n^{3/4+\epsilon} + m^{1+\epsilon} + n \log^2 m)$ (Theorem (2.6) in [AS91b]), as the base of the inductive proof. Having stated Theorem 1 in a more general context we can easily deal with the case of counting intersections of arcs of unit circles.

6 Arcs of unit circle

If the arcs are all from circles of unit (or constant) radius we can improve the algorithm for counting incidences. When we compare points and circles the characteristic dimension becomes $d_j = 2$, since we can use a simple vertical decomposition of [CEG+90] instead of the method in [CEGS89]. Testing whether two unit circles intersect becomes easily a test of comparing points and circles of radius two. The characteristic dimension of this problem is $d = 2$. Using the nested batching technique of Section 4 we have the following theorem:

Theorem 3 *Given n arcs of unit radius it is possible to count all intersections in expected time $O(n^{4/3+\epsilon})$.*

7 Sphere-segment intersections in 3-space

The nested batched technique is easily applicable to other counting problems. As an example we consider the problem of counting the number of intersections between segments and spheres in 3-space.

Given a sphere S , let $center(S)$ be its center. Given a segment e let $lhp(e)$ be its left end-point, $rhp(e)$ its right end-point and $W(e)$ the region of R^3 containing e and bounded by two planes both perpendicular to e and passing through its left end-point and its right end-point.

¹An alternative method is to split condition (7) into two conditions testing the relative position of $\varphi(C)$ with respect to surfaces ψ_O on one level and with respect to surfaces ψ_I on the next level.

But $\sum_{i=1, M(r)} m_i^{1+\delta} \leq [M(r)c \log^{1+\delta} r/r^{1+\delta}] m^{1+\delta}$. We can sum up terms in m and $m^{1+\delta}$, put $E = B_j M(r)c \log^{1+\delta} r/r^{1+\delta} + M(r) + M(r)B_{j-1} + O(r^{O(1)})$, and use the inequality 3, obtain

$$T_j(m, n) \leq [D_j \sum_{i=1}^M m_i^{d/(d+1)} n_i^{d/(d+1)+\epsilon} + E m^{d/(d+1)} n^{d/(d+1)+d\delta}]$$

Now we eliminate the summation using first the bound on m_i and then the Hölder-Minkowsky inequality [Mit70].

$$\begin{aligned} \sum_{i=1}^{M(r)} m_i^{d/(d+1)} n_i^{d/(d+1)+\epsilon} &\leq \\ [(c' \log r/r)m]^{d/(d+1)} \sum_{i=1}^{M(r)} n_i^{d/(d+1)+\epsilon} &\leq \\ [c' \log r/r]^{d/(d+1)} m^{d/(d+1)} M(r)^{1/(d+1)-\epsilon} n^{d/(d+1)+\epsilon} \end{aligned}$$

Assuming $\epsilon \geq d\delta$, we obtain, for some constant c''

$$T_j(m, n) \leq [D_j c'' \log^{d/(d+1)} r/r^{d/(d+1)} M(r)^{1/(d+1)-\epsilon} + E] m^{d/(d+1)} n^{d/(d+1)+\epsilon} \quad (5)$$

Recalling that $M(r) = r^d \log^{O(1)} r$ the coefficient becomes:

$$D_j c'' \log^{d/(d+1)} r/r^{d/(d+1)} M(r)^{1/(d+1)-\epsilon} + E = D_j c'' \log^{d/(d+1)} r \log^{O(1)/(d+1)-\epsilon} r(1/r)^{d\epsilon} + E$$

We choose r sufficiently large so that the constant in the equation (5) is less than $[D_j/2 + E]$, and we choose $D_j = 2E$, we obtain

$$T_j(m, n) \leq D_j m^{d/(d+1)} n^{d/(d+1)+\epsilon}$$

which completes the proof. \blacksquare

5 Counting circular arcs intersections

In this section we show how the general nested batching technique is used to solve the problem of counting intersections of circular arcs. From the discussion in Section 3 we just have to count the number of pairs of arcs satisfying conditions π_1 , π_2 and π_3 . We concentrate our discussion on condition π_1 , similar arguments hold for π_2 and π_3 .

Condition (1) involve testing a point against a line in R^2 , so clearly we have $d(\text{lines}, 1) = 2$ since we can use randomized or deterministic algorithms to find a $1/r$ -cutting of size $O(r^2)$ [Cla87, Mat90, Aga89]. We can dualize points to lines and lines to points in R^2 [EMP+82], therefore $d(\text{points}, 1) = 2$. Condition (2) is similar to condition (1).

Condition (3) involves testing a point with a circle. Using a standard decomposition of an arrangement of circles [CEG+90] we obtain $d(\text{circles}, 3) = 2$. Points and circles in R^2 have a natural duality in 3-space: (i) circle C becomes the point $\varphi(C) = (a, b, r)$, (ii) point $p = (a', b')$ dualize into the set of circles passing through it $\Phi(p) = \{(a, b, r) | \text{Dist}((a, b), (a', b')) = r\}$. The surface $\Phi(p)$ is a second degree surface in 3-dimensional parametric space described in [AS91b]. We can use the technique in [CEGS89] obtaining an $1/r$ -cutting with

$$T_j(m, n) \leq \sum_{\tau} T_j(m_{\tau}, n_{\tau}) + M(r)(m + n) + M(r)T_{j-1}(m, n_{\tau}) + O(mr^{O(1)}) \quad (1)$$

The following is the main theorem of this paper:

Theorem 1 For any ϵ and $\delta \leq \epsilon/d$:

$$T_j(m, n) = D_j n^{d/(d+1)+\epsilon} m^{d/(d+1)} + B_j m^{1+\delta} + A_j n \log^{(1+j)} m$$

where D_j, B_j, A_j depend on ϵ and δ .

Proof. The proof is an induction on j and n inspired by similar argument is in [EGS88].

For $j = 0$, $C_0 = TRUE$ is satisfied by nm pairs. The product is computed in constant time. We consider $j > 0$ and we use the inductive hypothesis on T_{j-1} . Equation (1) becomes:

$$\begin{aligned} T_j(m, n) &\leq \sum_{\tau} T_j(m_{\tau}, n_{\tau}) + O(mr^{O(1)}) + \\ &M(r)(D_{j-1} n^{d/(d+1)+\epsilon} m^{d/(d+1)} + B_{j-1} m^{1+\delta} + A_{j-1} n \log^{1+(j-1)} m) + \\ &M(r)(m + n) \end{aligned} \quad (2)$$

Fix δ, ϵ and choose $r = r(\delta, \epsilon)$ to be sufficiently large (how large will be determined later in the proof). If $m \geq n^d$ then, from Lemma 4, $T_j(m, n) \leq cm^{1+\delta}$ and the theorem is proved assuming $B_i \geq c$. Suppose $m \leq n^d$. In this case

$$m^{1+\delta} = m^{d/(d+1)} m^{1/(d+1)+\delta} \leq m^{d/(d+1)} n^{d/(d+1)+\delta} \quad (3)$$

First notice that at each level of the recursion the term $M(r)n + M(r)A_{j-1}n \log^j m$ in (2) contributes $O(n \log^j m)$ on one level of the recursion; and there are at most $O(\log m)$ levels. This is a consequence of the fact that the objects in A (whose counter is n) are represented as points. The objects in B (which are counted in m) are represented as surfaces. The overall contribution of this term is $O(n \log^{j+1} m)$. It is sufficient to drop this term from the recursion (2) and prove that the modified inequality

$$T_j(m, n) \leq \sum_{\tau} T_j(m_{\tau}, n_{\tau}) + M(r)m + M(r)(D_{j-1} m^{d/(d+1)} n^{d/(d+1)+\epsilon} + B_{j-1} m^{1+\delta}) + O(mr^{O(1)}) \quad (4)$$

satisfies the bound

$$T_j(m, n) = D_j m^{d/(d+1)} n^{d/(d+1)+\epsilon} + B_j m^{1+\delta}.$$

Choosing B_j and D_j large enough so that the bound holds for small values of m and n , we apply the inductive argument on $T_j(m_{\tau}, n_{\tau})$.

$$\begin{aligned} T_j(m, n) &\leq \sum_{i=1}^{M(r)} [D_j m_i^{d/(d+1)} n_i^{d/(d+1)+\epsilon} + B_j m_i^{1+\delta}] \\ &+ M(r)m + M(r)D_{j-1} m^{d/(d+1)} n^{d/(d+1)+\epsilon} + M(r)B_{j-1} m^{1+\delta} + O(mr^{O(1)}) \end{aligned}$$

$\mathcal{C}_j(A, B) = |\{(a, b) \in A \times B \mid \wedge_{i=1, j} C_i\}|$. We set $M(r) = \max_j\{M_j(r), M'_j(r)\}$. For lines and points on the plane we obtain $M_j(r) = r^2$ simple regions. For circles and points on the plane we obtain $M'_j(r) = r^3\beta(r)$ simple regions, using the technique in [CEGS89], where $\beta(r)$ is a sub-polylog function.

Let $|A| = m$ and $|B| = n$.

Lemma 3 *Given a set A and B as above, for every $0 \leq j \leq k$, we can build in time $O(n^{d+\epsilon})$ a data structure such that, for every $b \in B$, $\mathcal{C}_j(\{b\}, A)$ can be computed in time $O(\log^j n)$.*

Proof. We prove the claim by induction on j . For $j = 0$, $T_0(m, n) = O(1)$ therefore the lemma is satisfied. Assume $j > 0$. Applying property (I) we compute a decomposition of the space into $M'_j(r)$ elementary cells each one intersected by no more than $n/r \log r$ surfaces in $\{S_a^j = 0 \mid a \in A\}$. For each elementary cell we select a point in its interior and we compute the set S_τ^+ of surfaces not intersecting the region in positive position. We associate the cardinality of S_τ^+ with the region and we recurse this construction. Also, for each region τ and for all the elements $a \in A$ whose corresponding surface is in S_τ^+ we build the data structure to compute \mathcal{C}_{j-1} , which can be built in time $T_{j-1}(n)$. The total time and space for this construction satisfies the recurrence:

$$T_j(n) \leq M'_j(r)T_j(n/r \log r) + M'_j(r)T_{j-1}(n) + O(nr^{O(1)} + nM'_j(r))$$

$$T_j(O(1)) = O(1)$$

Since $M'(r) = r^d \log^{O(1)} r$ the solution to this recurrence is $T_j(n) = O(n^{d+\epsilon})$.

Given a query point p we locate by exhaustive search the region τ containing it than we recurse on the two data structures associated with τ . At the last level we collect the counters of the positive regions and we sum them up to form the final result. The query time $Q_j(n)$ satisfies this recurrence:

$$Q_j(n) \leq Q_j(n/r \log r) + Q_{j-1}(n) + O(M'(r))$$

whose solution is $O(\log^j(n))$. ■

From Lemma 3 easily follows:

Lemma 4 *If $m \geq n^d$ then, for every $j \leq k$, $T_j(m, n) = O(m^{1+\epsilon})$.*

Let us suppose now that $m \leq n^d$. We dualize points and surfaces at level j , and we compute the partition (II). For each cell τ of the partition we have a set of n_τ points contained it and a set of m_τ surfaces cutting through. By the property (II), $m_\tau \leq cm/r \log r$ for some constant c . For each cell τ the relative sign of the points in τ and the surfaces outside τ is similar for each point in τ . This implies that in time $O(m)$ we can determine, for all points in τ , and for most arcs in B_ν , the truth value of the predicate associated with the j -th level. For the surfaces intersecting τ we proceed recursively.

From the above discussion $T_j(m, n)$ satisfies this recurrence:

1. $rep(\gamma')$ above $line(\gamma)$.
2. $rep(\gamma')$ outside $circle(\gamma)$.
3. $lep(\gamma')$ inside $circle(\gamma)$.
4. $rep(\gamma')$ above $line(\gamma)$.

We can set up a multi-layered batching algorithm with these new predicates and obtain a similar result as for π_1 .

4 The nested batching technique

We set our problem in a more general context. Let \mathcal{A} and \mathcal{B} be two classes of geometric objects and $A \subset \mathcal{A}$ and $B \subset \mathcal{B}$ two finite subsets. Our goal is to compute efficiently $\mathcal{C}_k(A, B) = |\{(a, b) \in A \times B | F(a, b)\}|$, where $F(a, b) = \wedge_{j=1, k} C_j$ is a conjunction of a constant number k of elementary conditions. Each elementary condition C_j is of the form $S_a^j(p_b^j) > 0$, where $S_a^j(\cdot)$ is a polynomial whose structure depends on j and \mathcal{A} and whose coefficients depend only on $a \in A$. Also, p_b^j is a tuple of real numbers depending only on $b \in B$. Moreover, we require that we can rewrite C_j in the dual form $S_b^j(p_a^j) > 0$, where $S_b^j(\cdot)$ is a polynomial whose structure depends on j and \mathcal{B} , whose coefficients depend only on $b \in B$. Also, p_a^j is a tuple of real numbers depending only on $a \in A$. For technical reasons we assume that $C_0 = TRUE$. If $S_a^j(p_b^j) > 0$ we say that the *point* p_b^j is on the *positive side* of the *surface* $S_b^j(\cdot) = 0$. From now on we will use the terms *point* and *surface* in this context. Note that, for a given j , an element $a \in A$ (resp. $b \in B$) is mapped to a point or to a surface. We call the point and the surface *dual* one to the other.

The following conditions must be also satisfied:

- (I) Given a family of surfaces $\{S_a^j(x) = 0 | a \in A\}$ there is a constant $d'(\mathcal{A}, j)$ such that the space underlying the family of surfaces can be partitioned into $M_j'(r) = O(r^{d'(\mathcal{A}, j)} \log^{O(1)} r)$ cells of constant descriptive complexity such that each cell meets at most $|A|/r \log r$ surfaces in the family. Moreover, we assume that this decomposition can be computed in expected time $O(|A|r^{O(1)})$.
- (II) Given a family of surfaces $\{S_b^j(x) = 0 | b \in B\}$ there is a constant $d(\mathcal{B}, j)$ such that the space underlying the family of surfaces can be partitioned into $M_j(r) = O(r^{d(\mathcal{B}, j)} \log^{O(1)} r)$ cells of constant descriptive complexity such that each cell meets at most $|B|/r \log r$ surfaces in the family. Moreover, we assume that this decomposition can be computed in expected time $O(|B|r^{O(1)})$.

Conditions (I) and (II) can be rephrased requiring that there is a $1/r$ -cutting of a certain size (see [Aga91] for a survey on cuttings) for the arrangement of surfaces.

Let $d_j = \max\{d'(\mathcal{A}, j), d(\mathcal{B}, j)\}$ and let $d = \max_j d_j$. We will refer to d as the *characteristic dimension of the formula F* , while d_j is the *characteristic dimension of the elementary conjunct C_j* .

Let $T_j(m, n)$ be the time needed to determine the number of pairs (a, b) satisfying the conditions associated with the elementary conjuncts from 1 to j , that is the time to compute

Given an arc γ we denote its end-points by α and β . Let $center(\gamma)$ be the center of $circle(\gamma)$, and $disk(C)$ the disk bounded by the circle C . Let $l(\gamma, \alpha)$ be the line through $center(\gamma)$ and α and $l^+(\gamma, \alpha)$ is the halfplane containing γ . Similarly we define $l(\beta, \gamma')$ and $l^+(\beta, \gamma')$.

Lemma 1 *Given a circle C and an arc γ' with endpoints α and β , such that $\alpha \notin disk(C)$ and $\beta \notin disk(C)$. $|C \cap \gamma'| = 2$ if and only if $|C \cap circle(\gamma')| = 2$ and $center(C) \in l^+(\beta, \gamma') \cap l^+(\alpha, \gamma')$.*

Proof. It follows from elementary geometry. ■

From Lemma 1 it follows that condition (5) can be replaced by:

- (5) $center(\gamma) \in l^+(lep(\gamma'), \gamma')$
- (6) $center(\gamma) \in l^+(rep(\gamma'), \gamma')$
- (7) $circle(\gamma)$ meet $circle(\gamma')$

There are seven conditions to be satisfied. Conditions (1), (2), (3), (4), (5) and (6) require testing a point in R^2 with a line or a circle in R^2 . Condition (7) involves two circles. Each condition involves only simple objects. This property of “simplicity” is important for the batching technique we will use in Section 4.

3.1 Necessary and sufficient conditions for π_2 and π_3

Before describing the algorithm we show that for the counters π_2 and π_3 we can similarly write a list of necessary and sufficient conditions involving only simple geometric objects.

Lemma 2 *Given a circle C , such that $\alpha \in disk(C)$ and $\beta \in disk(C)$. $|C \cap \gamma'| = 2$ if and only if $|C \cap circle(\gamma')| = 2$ and $center(C) \in l^-(\beta, \gamma') \cap l^-(\alpha, \gamma')$.*

Proof. Follows from elementary geometry. ■

Given $\gamma \in X_\nu$ and $\gamma' \in B_\nu$ the conditions for which an upper arc γ' is counted in $\pi_2(\gamma)$ are the following:

1. $lep(\gamma')$ inside $circle(\gamma)$.
2. $rep(\gamma')$ inside $circle(\gamma)$.
3. $center(\gamma) \in l^-(lep(\gamma'), \gamma')$.
4. $center(\gamma) \in l^-(rep(\gamma'), \gamma')$
5. $circle(\gamma')$ meet $circle(\gamma)$.

For π_3 , we need to split the count into four different counters. $\pi_3 = \pi_{31} + \pi_{32} + \pi_{33} + \pi_{34}$. The first counter is for the case when the left end-point of γ' is in U_γ and the right end-point is inside $circle(\gamma)$ and above $line(\gamma)$. The second counter is for the case when the left end-point of γ' is in U_γ and the right end-point is below $line(\gamma)$. The third and fourth counters are obtained from the first and second by reversing the role of the left and right end-points. We give explicitly the condition for π_{31} . Arc $\gamma' \in B_\nu$ is counted in $\pi_{31}(\gamma)$ if:

2. Arcs whose both endpoints lie in W_γ . The number of arcs in this category which meet γ is counted in $\pi_2(\gamma)$. For each arc counted in $\pi_2(\gamma)$ we have exactly two intersections.
3. Arcs for which one endpoint lies in U_γ and the other does not lie in U_γ . The number of arcs in this category which meet γ is $\pi_3(\gamma)$. For each arc contributing to $\pi_3(\gamma)$ we have exactly one intersection.

Clearly $I(\gamma, B_\nu) = 2\pi_1(\gamma) + 2\pi_2(\gamma) + \pi_3(\gamma)$. Let $|A_\nu| = a_\nu$ and $|B_\nu| = b_\nu$. For each category of arcs in B_ν , Agarwal and Sharir [AS91b] build a data structure of size $O(b_\nu^{3+\epsilon})$. For any query arc γ' , satisfying the conditions of the arcs in X_ν , the value of $\pi_1(\gamma')$, $\pi_2(\gamma')$ and $\pi_3(\gamma')$ can be computed in time $O(\log^5 b_\nu)$ using those data structures (Lemma 4.4 in [AS91b]).

The time spent to set up the data structures and perform the queries at node ν is $O(a_\nu \log^5 b_\nu + b_\nu^{3+\epsilon})$. Using a simple batching technique the query time and the preprocessing can be balanced to $O(b_\nu a_\nu^{2/3+\epsilon} + a_\nu^{1+\epsilon}) = O(b_\nu^{5/3+\epsilon})$. Using properties (a) and (b) we obtain an overall $O(n^{5/3+\epsilon})$ expected time method (Theorem 4.5 in [AS91b]).

In the next section we show that a more refined taxonomy and a different batching schema can yield a better time bound.

3 A refined decomposition schema

Given an arc γ we define:

- $lep(\gamma)$ is the left end point of the arc γ .
- $rep(\gamma)$ is the right end point of the arc γ
- $line(\gamma)$ is the line connecting the two endpoints of γ .
- $circle(\gamma)$ is the circle containing the arc γ .

Given $\gamma \in X_\nu$ and $\gamma' \in B_\nu$ the following conditions are necessary and sufficient for the pair (γ, γ') to be counted in $\pi_1(\gamma)$.

- (1) $lep(\gamma')$ above $line(\gamma)$.
- (2) $rep(\gamma')$ above $line(\gamma)$.
- (3) $lep(\gamma')$ outside $circle(\gamma)$.
- (4) $rep(\gamma')$ outside $circle(\gamma)$.
- (5) γ' meet $circle(\gamma)$.

We can notice that the contribution of γ to the list of necessary and sufficient conditions is given by a circle and a line, while this is not the case for γ' . Condition (5) involves a “composite” object (an arc) instead of a “simple” object (point, line, circle). Next, we show how to refine condition (5) so that it involves only simple objects.

2 Decomposition strategy

Given a collection Γ of n circular arcs in R^2 , let $I(\Gamma)$ be the number of pairs of arcs of Γ that intersect, where each pair is counted with multiplicity one or two according to the number of intersections generated by the pair. We do not consider the special case of tangent arcs, but the algorithm can be modified easily to take this into account. In linear time we can split each arc in Γ into a constant number of x -monotone arcs, obtaining a collection Γ' . Clearly $I(\Gamma) = I(\Gamma')$. Without loss of generality we will deal with a collection of x -monotone arcs.

Let Γ_1, Γ_2 be two sets of circular arcs. Let $I(\Gamma_1, \Gamma_2)$ be the number of *bichromatic* intersections generated by an arc in Γ_1 and an arc in Γ_2 . We will show that it is sufficient to solve the bichromatic case to solve the general problem.

Since the arcs of Γ are x -monotone and we can build an *hereditary segment tree* for Γ (see [CEGS90]). We start generating a *segment tree* \mathcal{T} on the interval decomposition of the x -axis induced by the projection on the x -axis of the $2n$ end-points of arcs in Γ . We associate each elementary interval on the x -axis with a leaf of \mathcal{T} in a left to right order, and then we build \mathcal{T} as a minimum height binary tree over its leaves. Each node ν is associated with an interval δ_ν on the x -axis, which is the union of intervals associated with the descendants of ν . We also have a vertical strip $S_\nu = \delta_\nu \times [-\infty, +\infty]$. We store at ν the set A_ν of arcs of Γ whose x -projection contains δ_ν but not δ_ω where ω is the parent of ν . We store at ν the set $B_\nu = \bigcup_z A_z$, where z is a descendent of ν , included ν itself. We also assume that the arcs in A_ν and B_ν are truncated within S_ν . It is easy to check the following properties:

(a)

$$\sum a_\nu \leq \sum b_\nu \leq 4n \log n$$

(b) Lemma 4.1 in [AS91b]:

$$I(\Gamma) = \sum_{\nu \in \mathcal{T}} I(A_\nu, B_\nu)$$

From (b) it follows that to compute $I(\Gamma)$ we just compute $I(A_\nu, B_\nu)$ for each node ν and sum these values over all nodes. Property (a) ensures that, by building the hereditary segment tree, the sum of the input sizes of the subproblems has increased only of a logarithmic factor. From now on in the discussion, we consider one specific node ν and we bound the time needed to compute $I(A_\nu, B_\nu)$.

Let X_ν be the subset of *upper arcs* in A_ν , where an upper arc is a portion of an upper semi-circle. We will show how to compute $I(X_\nu, B_\nu)$. A similar argument is used to count $I(A_\nu/X_\nu, B_\nu)$. For every $\gamma \in X_\nu$ we define the region U_γ as the part of the vertical strip S_ν above γ . We define W_γ as the part of the disk whose boundary contains γ which falls within the vertical strip S_ν . Using the auxiliary definitions U_γ and W_γ , we divide the arcs in B_ν into three categories:

1. Arcs of B_ν whose both endpoints lie in U_γ . We denote with $\pi_1(\gamma)$ the number of arcs of B_ν in this category which meet γ . For each arc counted in $\pi_1(\gamma)$ we have exactly two intersections.

1 Introduction

Computing intersections of geometric objects is a fundamental topic in computational geometry. Given n simple objects (segments, circular arcs, Jordan arcs, etc.) it is possible to count and report all intersection points in $O(n^2)$ time, assuming that in $O(1)$ time we can compare any two objects. Since the number of intersection K can be quadratic in n the naive approach is worst case optimal for *reporting* all intersections. It is thus of interest to find output sensitive algorithms for which the time complexity depends on the actual number of intersections to report. The first such algorithm for Jordan arcs due to Bentley and Ottmann [BO79] runs in $O((n + K) \log n)$ time and it is based on a sweeping line approach. For a collection of line segments Chazelle and Edelsbrunner [CE88] presented an optimal $O(n \log n + K)$ algorithm. Clarkson and Shor [CS89] and Mulmuley [Mul88] give randomized algorithms whose expected running time is $O(n \log n + K)$ for a collection of Jordan arcs.

If we are interested in *counting* the number of intersections without reporting them explicitly, the number of intersection K itself is not anymore a lower bound for the problem. Since K can be as large as n^2 , we are interested in finding subquadratic algorithms which do not depend on K . For counting line segments intersections Guibas et al. [GOS89] give a randomized algorithm whose expected running time is $O(n^{4/3+\epsilon})$ for any $\epsilon > 0$. A deterministic algorithm with a $O(n^{4/3} \log n)$ time bound is in [Aga90, Mat90].

For the problem of counting intersections of circular arcs the first substantially subquadratic algorithms have been presented in [AS91b] (see [AS91b] for a review of some previous partial results). Agarwal and Sharir [AS91b] give an algorithm to compute intersections of circular arcs whose expected running time is $O(n^{5/3+\epsilon})$, for any $\epsilon > 0$. For the special case of circular arcs with the same radius their time bound reduces to $O(n^{3/2+\epsilon})$.

In this paper we present a modification of the general approach in [AS91b] for counting intersection of circular arcs. The main results are the following:

1. A randomized algorithm for counting intersections of circular arcs whose expected running time is $O(n^{3/2+\epsilon})$ (Theorem 2).
2. A randomized algorithm for counting intersections of *unit* circular arcs whose expected running time is $O(n^{4/3+\epsilon})$ (Theorem 3).

The asymptotic time bound of Theorem 3 is similar to the time bound obtained for line segments in [GOS89], thus settling an open question posed in [AS91b]. Line segments can be seen as arcs of circles of infinite radius. The bounds of Theorems 2 and 3 are achieved by using the *nested batching technique*. This technique is a generalization of the batching method in [EGS88] for batching the computation of queries over a multi-level tree similar to those discussed in [CSW90] and [AS91a]. For a special case this technique had been used in [Pel91], here we give a complete description of the method in a more general setting.

The paper is organized as follows, in Section 2 we briefly survey the approach in [AS91b], in Section 3 we give an improved problem decomposition. In Section 4 we describe the nested batching technique. In Section 5 we derive the result for general circular arcs. In Section 6 we discuss the case of unit circular arcs. In Section 7 we apply the nested batching technique to counting the intersections of segments and spheres in 3-space.

A New Algorithm for Counting Circular Arc Intersections

Marco Pellegrini*

TR-92-010

February 1992

Abstract

We discuss the following problem: given a collection Γ of n circular arcs in the plane, count all intersections between arcs of Γ . We present an algorithm whose expected running time is $O(n^{3/2+\epsilon})$, for every $\epsilon > 0$. If the arcs have all the same radius the expected time bound is $O(n^{4/3+\epsilon})$, for every $\epsilon > 0$. Both results improve on the time bounds of previously known asymptotically fastest algorithms. The technique we use is quite general and it is applicable to other counting problems.

*International Computer Science Institute, 1947 Center St., Berkeley, CA 94704 U.S.A. and Dept. of Computer Science, King's College, Strand, London WC2R 2LS U.K. e-mail: m.pellegrini@oak.cc.kcl.ac.uk