On the Union of Fat Wedges and Separating a Collection of Segments by a Line

Alon Efrat* Günter Rote† Micha Sharir‡

Abstract

We call a line ℓ a *separator* for a set S of objects in the plane if ℓ avoids all the objects and partitions S into two nonempty subsets, one consisting of objects lying above ℓ and the other of objects lying below ℓ. In this paper we present an $O(n \log n)$-time algorithm for finding a separator line for a set of n segments, provided the ratio between the diameter of the set of segments and the length of the smallest segment is bounded. No subquadratic algorithms are known for the general case. Our algorithm is based on the recent results of [13], concerning the union of 'fat' triangles, but we also include an analysis which improves the bounds obtained in [13].

1 Introduction

Given a set S of n objects in the plane, we call a line ℓ a *separator* of S if ℓ does not meet any object of S, and partitions S into two non-empty subsets, one consisting of all objects lying fully above ℓ and the other consisting of all objects lying fully below ℓ.

This and some related problems have been studied in several recent papers. For instance, Freimer et al. [8] present an algorithm for *shattering* a set of objects, i.e. finding a set of separator lines that form an arrangement such that none of its cells contains more than a single object.

Let us assume that the objects in S are all line segments. If ℓ is a separator of S, then by tilting and moving ℓ about, we can make it pass through two endpoints of segments in S while still avoiding the (interiors of the) other segments. In this extreme position, ℓ defines an edge of the *visibility graph* E of S (see [10, 17]). We can thus compute the visibility graph and select those edges whose extensions in both directions do not meet S. Using the algorithm of Ghosh and Mount [10], we can compute the visibility graph and select these special edges in time $O(n \log n + |E|)$. Since in the worst case $|E|$ can be $\Theta(n^2)$, the resulting algorithm is worst-case quadratic.

A simpler solution, that also requires quadratic time, is obtained by dualizing the problem. Using a standard duality transform [5], the segments of S become n double wedges, and a separator line becomes a point lying in the complement of the union of these double

*School of Mathematical Sciences, Tel Aviv University, alone@math.tau.ac.il

†Institut für Mathematik, Technische Universität Graz

‡School of Mathematical Sciences, Tel Aviv University, and Courant Institute of Mathematical Sciences, New York University

wedges, strictly between the upper and lower envelopes of the double wedges. Hence, to determine the existence of a separator, or actually to find the set of all separators, we can simply compute the union of these dual double wedges and collect all components of its complement that lie between the envelopes. This can be done by computing the arrangement induced by the $2n$ lines that are dual to the endpoints of the segments in S, and then by determining for each face of the arrangement whether it lies in some double wedge. All this can be done in time $O(n^2)$; with topological sweeping [6], $O(n)$ space is sufficient.

In this dual setting, the problem of finding a separator for a set of segments is more or less equivalent to the problem of determining whether the union of n double wedges has (bounded) holes or whether it is simply connected. This is closely related to problems raised by Overmars, Guibas and Sharir, and others, which ask to test whether the union of n given triangles fully contains another given triangle. Quadratic-time solutions to these problems are easy, following the technique just outlined, and the goal is to obtain subquadratic solutions. No such solution is known in general as yet. As a matter of fact, Seidel [18] recently showed that, under a fairly reasonable computational model, all these problems require time $\Omega(n^2)$.

There is, however, a special case that admits much faster solutions. This is when all wedges are *fat*, meaning that their angles are all at least δ, for some fixed parameter $\delta > 0$. In this case, it was shown in [1, 13] that the overall complexity of the union of n fat double wedges (i. e., the total number of straight-line pieces of the boundary) is linear in n, with the constant of proportionality depending on δ, and that the union can be computed in close-to-linear time. (A slightly improved algorithm is given in [14].) The paper [13] actually generalizes these results to the case of fat triangles, but we will be concerned here only with double wedges. The latest progress is due to van Kreveld [9], who improves the dependence of the constant of proportionality on δ to $O(1/\delta)$ (but only for fat triangles and polygons, not for double wedges)

In this paper we study the problem of determining whether there exists at least one separator line for certain special collections of segments. Specifically, we assume that the given segments all lie in some bounded disk, say the unit disk, and their lengths are all bounded from below by some constant $c > 0$. In other words, we assume a bounded ratio ρ between the diameter of the union of the segments in S and the length of the smallest segment in S. The segments are allowed to intersect. We present a solution to this restricted problem, whose running time is $O(n \log n)$, with the constant of proportionality depending on the ratio ρ. To be more precise, our algorithm runs in time

$$O(\Big(\min\{n\rho^2 \log\rho \log\rho n, \quad n\rho \log^2 n \log\log n\}\Big) ; \tag{1}$$

thus, even if ρ is not fixed but is only $o(n/(\log^2 n \log\log n))$, we still obtain a subquadratic solution.

The key idea for obtaining such an efficient solution is to partition S into a subset S_1 of "flat" segments and a subset S_2 of "steep" segments. For each subset, the space of all lines avoiding all segments in the subset can be computed in close to linear time, because the segments in each subset can be dualized to 'fat' double wedges (see below for more details). Then we test whether the intersection of the duals of the two resulting sets of avoiding lines contains any point between the upper and lower envelopes of the wedges; this can also be easily accomplished in subquadratic time by a line-sweeping technique in the dual plane.

Thus our paper can be regarded as an application of the analysis of the union of fat wedges, as given in [1, 13, 14, 9]. The interesting feature of our application is that its time complexity depends on the behavior of the constants of proportionality in the bounds given in these papers. We first derive improved bounds on those constants, which are better than those implicitly given in [13]. Specifically, we show that the union of n wedges, each having an angle at least δ, has boundary complexity $O(\min\{n\delta^{-2}\log(1/\delta),\ n\log\log n/\delta\})$, improving the bound $O(n\delta^{-3})$ which is implied in [13]. The second term in this bound follows from the recent result of van Kreveld [9] mentioned above. However, van Kreveld's bound does not seem to apply to wedges, so it becomes slightly super-linear in n. Thus, for constant δ and large n, the first term, which is a result of the analysis given in this paper, is the best bound known so far.

2 Geometric Preliminaries

We begin with a few notations. Let $S = \{s_1, \ldots, s_n\}$ be the given collection of segments. The segments in S can intersect, but for simplicity of exposition we will assume that no two segments have a common endpoint.

We split S into two subsets S_1, S_2, so that the segments of S_1 (resp. of S_2) have slopes with absolute value ≤ 1 (resp. > 1). We use two duality transforms, τ_1, τ_2, applying τ_i to the segments of S_i, for $i = 1, 2$.

The first duality transform τ_1 maps a point (a, b) to the line $y = ax + b$, and a line $y = cx + d$ to the point $(-c, d)$. It is well known that this duality preserves incidence between points and lines, and maps a point lying above (resp. below) a line ℓ to a line lying above (resp. below) the dual point of ℓ. Under this transform, a segment $s = pq$ is mapped to a double wedge $s^\star$ consisting of points that lie between the lines $p^\star$, $q^\star$, dual to p, q, respectively. A (non-vertical) line ℓ meets s if and only if its dual point $\ell^\star$ lies inside the double wedge $s^\star$. Thus a line ℓ avoids all segments in S if $\ell^\star$ lies outside the union of their dual double wedges. It is also easy to show that ℓ is a separator of S if and only if $\ell^\star$ also lies between the upper and lower envelopes of the dual double wedges (and in the complement of their union).

We call a wedge or a double wedge *δ-fat* if its angle is at least δ.

Lemma 2.1 *Suppose that a segment $s = pq$ of length at least c is contained in the unit disk and that the angle formed between s and the y-axis is at least $\pi/4$. Then the double wedge dual (by the transform τ_1) to s is $(c\sqrt{2}/6)$-fat.*

Proof: Note first that we must have $c \leq 2$. The assumptions on s imply that the slopes k_p, k_q of the lines dual to p and q respectively (these are the x-coordinates of p and q) are between -1 and 1 and differ by at least $\frac{c\sqrt{2}}{2}$. The angle θ between them satisfies

$$\tan\theta = \frac{|k_q - k_p|}{1 + k_q k_p} \geq \frac{c\sqrt{2}}{4} \equiv \tan\theta_0 \ .$$

Hence

$$\theta \geq \theta_0 \geq \frac{1}{2}\sin 2\theta_0 = \frac{\tan\theta_0}{1+\tan^2\theta_0} = \frac{\frac{c\sqrt{2}}{4}}{1+\frac{c^2}{8}} \geq \frac{c\sqrt{2}}{6} \ ,$$

using, in the last inequality, the fact that $c \leq 2$. □

This lemma implies that all double wedges obtained by applying τ_1 to the segments of S_1 are $(c\sqrt{2}/6)$-fat.

Consider next the segments of S_2. We apply to them another duality transform, τ_2, that maps a point (a, b) to the line $y = bx + a$ and a line $y = cx + d$ to the point $(-\frac{1}{c}, -\frac{d}{c})$. Note that $\tau_2 = \tau_1 \circ \sigma$, where σ is the transformation $(x, y) \mapsto (y, x)$. It is easily verified that Lemma 2.1 implies that the double wedges obtained by applying τ_2 to the segments of S_2 are $(c\sqrt{2}/6)$-fat.

In the algorithm, we will need to merge the unions of the double wedges of $\tau_1(S_1)$ and those of $\tau_2(S_2)$. To do so, we will need to place these two unions in a common dual plane. This is achieved by applying the transformation $\tilde{\tau} = \tau_1 \circ \tau_2^{-1} = \tau_1 \circ \sigma \circ \tau_1^{-1}$ to the union of the double wedges in $\tau_2(S_2)$. As is easily checked, $\tilde{\tau}$ is a projective transformation which maps a point (a, b) to the point $(\frac{1}{a}, \frac{b}{a})$, and a line $y = cx + d$ to the line $y = dx + c$. Note that points on the y-axis are mapped to points at infinity, that the right half-plane is mapped onto itself and the left half-plane is also mapped onto itself. Moreover, within each half-plane, $\tilde{\tau}$ consistently preserves sidedness of points and lines. That is, in the right half-plane a point p lies above a line ℓ if and only if $\tilde{\tau}(p)$ lies above $\tilde{\tau}(\ell)$; in the left half-plane p lies above ℓ if and only if $\tilde{\tau}(p)$ lies below $\tilde{\tau}(\ell)$.

3 On the Union of Fat Wedges

The problem of computing the union of fat wedges (or, more generally, of fat triangles) has been recently studied in [1, 13, 14]. We recall the results of these papers, and look somewhat closer at the dependence of the bounds that they provide on the 'fatness' δ.

Theorem 3.1 *Let F be a set of m δ-fat wedges in the plane. The boundary of the union of F consists of $O(m\delta^{-2}\log(1/\delta))$ straight segments and rays, and the union can be computed in time $O(m\delta^{-2}\log(1/\delta) + m\log m)$.*

the proof appears in the appendix

4 The Algorithm and its Analysis

Having all this technical machinery, we can now present the algorithm for computing a separator of S.

The Algorithm

1. Translate and scale the set S of segments so that it fits into the unit disk. This could for example be done very easily by computing the smallest enclosing axis-parallel rectangle and placing the corners of this rectangle on the unit disk. This ensures that the smallest segment has length at least $c = \Omega(1/\rho)$, if the ratio between the diameter of S and the length of the shortest segment of S is ρ. For simplicity we again call the resulting set S.

2. Partition the set S of segments into two subsets S_1, S_2, such that all segments in S_1 (resp. in S_2) have slopes with absolute value ≤ 1 (resp. > 1). Let $n_1 = |S_1|$, $n_2 = |S_2|$.

3. Apply the first duality transform τ_1 to the segments in S_1, obtaining a collection of n_1 $(c\sqrt{2}/6)$-fat double wedges (see Lemma 2.1). Similarly, apply the second transform τ_2 to the segments in S_2, obtaining a collection of n_2 $(c\sqrt{2}/6)$-fat double wedges.

4. Compute the complement W_1 of the union of the double wedges of $\tau_1(S_1)$, and the complement W_2 of the union of the double wedges of $\tau_2(S_2)$, applying the algorithm in Theorem 3.1.

5. Compute the image $\widetilde{W}_2$ of W_2 under the transformation $\tilde{\tau} = \tau_1 \circ \tau_2^{-1} = \tau_1 \circ \sigma \circ \tau_1^{-1}$ discussed in Section 2.

6. Compute the upper and lower envelopes of all the double wedges in $\tau_1(S)$.

7. Apply a standard line-sweeping algorithm to compute the intersection of W_1 and $\widetilde{W}_2$. Run the sweeping algorithm and stop it as soon as it finds an intersection point x between the boundaries of W_1 and $\widetilde{W}_2$ or a vertex x of one of these sets which is contained in the other set, such that x does not lie on either envelope. Such a point x is the dual (under τ_1) of an extreme separator for S. If no such point is found, S does not admit a separator.

The Analysis

The correctness of the algorithm follows from the discussion in the introduction and in the preceding section. Indeed, if a separator of S exists, then there exists an extreme separator ℓ such that (i) its dual point $x = \tau_1(\ell)$ lies strictly between the upper and lower envelopes of the dual double wedges of S, (ii) ℓ avoids all the segments in S_1, (iii) ℓ avoids all segments in S_2, and (iv) ℓ passes through the endpoints of two segments in S. It follows easily that x must lie in W_1 and in $\widetilde{W}_2$. If the two segments through whose endpoints the separator passes are both in S_1 or both in S_2 then x must be a vertex of the boundary of W_1 or $\widetilde{W}_2$, respectively, that also lies in the other region. If one of these segments belongs to S_1 and one to S_2, then x must be an intersection point of the boundaries of W_1 and $\widetilde{W}_2$. In any case, such a point x will be identified in Step 7, and therefore the algorithm is correct.

Next consider the running time of the algorithm.

Steps 1–3 take linear time. Step 4 can be performed, By use to be Lemma 2.1 and Theorem 3.1, in time

$$O(n(\tfrac{\sqrt{2}}{6}c)^{-2}\log(1/(\tfrac{\sqrt{2}}{6}c)) + n\log n)$$

and hence, combined with van Kreveld's result, in time

$$O(\min\{n\rho^2\log\rho n,\ n\rho\log^2 n\log\log n\}),$$

where ρ is the ratio between the diameter of S and the length of the shortest segment of S.

Step 5 can be done in time linear in the complexity of W_2, that is, in time $O(\min\{n\rho^2\log\rho n,\ n\rho\ \log^2 n\ \log\log n)$. The calculation of the upper and lower envelopes in Step 6 can be done in time $O(n \log n)$ [2, 12]. Note that there are only $O(n)$ vertices on both envelopes.

Finally, the line-sweeping algorithm of Step 7 takes time $O((N+k)\log N)$, where N is the total number of segments forming the boundaries of W_1 and of $\widetilde{W}_2$, and k is the number of 'events' that the algorithm processes. The number of events initially put on the priority queue is proportional to N, each event that is being processed generates only a constant number of new events, and the number k of events being processed, until the first extreme separator (if any) is detected, is $O(N+n)$ — the algorithm will process only new events that correspond to vertices of the envelopes; any other intersection point must correspond to an extreme separator, as argued above, which will then terminate the algorithm. Thus the total cost of Step 7 is

$$O\left((N+n)\log(N+n)\right) \quad = \quad O\left(\min\{n\rho^2 \log\rho \log\rho n, \quad n\rho \log n \log\log n\right) .$$

The space requirement is dominated by the need to store the boundaries of W_1 and $\widetilde{W}_2$ and is thus $O(\min\{n\rho^2 \log\rho,\ n\rho \log n\ \log\log n\})$. In summary, we have shown:

Theorem 4.1 *Given a set S of n line segments in the plane, such that the ratio between the diameter of S and the length of the smallest segment in S is ρ, one can determine whether S admits a separator line (and find such a line if it exists) in $O(n\rho^2 \log\rho \log\rho n)$ time and $O(n\rho^2 \log\rho)$ space.*

Remark. As mentioned, that ρ can be more than a constant. In particular, if ρ is $o(n/(\log^2 n \log\log n))$ the algorithm still runs in subquadratic time.

5 Open Problems

Of course, the most challenging open problem is to find less restrictive conditions in which a subquadratic algorithm for solving the separation problem can be found. Another interesting open problem is to improve the shattering algorithm of [8] to run in subquadratic time, for the restricted case studied in this paper.

References

[1] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig, Approximate motion planning and the complexity of the boundary of the union of simple geometric figures, *Proc. 6th Ann. Symp. on Computational Geometry,* 1990, pp. 281–289.

[2] P. Alevizos, J. D. Boissonnat, and F. P. Preparata, An optimal algorithm for the boundary of a cell in a union of rays, *Algorithmica* 5 (1990), 573–590.

[3] B. Chazelle and H. Edelsbrunner, An optimal algorithm for intersecting line segments in the plane, *J. Assoc. Comput. Mach.* 38 (1992), 1–54.

[4] K. Clarkson and P. Shor, Applications of random sampling in computational geometry, *Discrete Comput. Geom.* 4 (1989), 387–421.

[5] H. Edelsbrunner, *Algorithms in Combinatorial Geometry,* Springer Verlag, Heidelberg, 1987.

[6] H. Edelsbrunner and L. Guibas, Topologically sweeping an arrangement, *J. Computer System Sciences,* 38 (1989), 165–194.

[7] H. Edelsbrunner, L. Guibas, and M. Sharir, The complexity and construction of many faces in arrangements of lines and of segments, *Discrete Comput. Geom.* **5** (1990), 161–196.

[8] R. Freimer, J. S. B. Mitchell, and C. D. Piatko, On the complexity of shattering using arrangements, *Proc. 2nd Canadian Conf. on Computational Geometry,* 1990, pp. 218–222.

[9] M. van Krevedl, On Fat Partitioning, Fat Covering and the Union Size of Polygons Technical Report SOCS 93.2, Scholl of computer Science, McGill Univ., Montreal, PQ, 1987.

Computing the volume of the union of spheres.

[10] S. Ghosh and D. Mount, An output-sensitive algorithm for computing visibility graphs, *SIAM J. Computing* 20 (1991), 888–910.

[11] S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes, *Combinatorica* **6** (1986), 151–177.

[12] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, *Information Proc. Letters* 33 (1989), 169–174.

[13] J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl, Fat triangles determine linearly many holes, to appear in *SIAM J. Computing.*

[14] N. Miller and M. Sharir, Efficient randomized incremental construction of the union of fat triangles, manuscript, 1991.

[15] K. Mulmuley, A fast planar partition algorithm I, *J. Symbolic Computation* 10 (1990), 253–280.

[16] M. Overmars, personal communication.

[17] M. Overmars and E. Welzl, New methods for computing visibility graphs, *Proc. 4th Ann. Symp. on Computational Geometry,* 1988, pp. 164–171.

[18] R. Seidel. *Private commonication*

A

Proof of theorem 3.1: We re-examine the proofs given in [13], and refer the reader to that paper for more details. In the bounds that we state below, the constants of proportionality are assumed to be independent of both m and δ. We first choose $\lceil \frac{2\pi}{\delta/3} \rceil = O(1/\delta)$

equally spaced orientations θ_i from the full circle of possible orientations, and we partition F into $\lceil \frac{2\pi}{\delta/3} \rceil$ subfamilies $F_1, F_2, \ldots$ with the property that each wedge in F_i contains the ray at orientation θ_i emanating from its apex, and the angles between that ray and each of the sides of the wedge are at least $\delta/3$. Let us denote the cardinalities of the subfamilies by $m_i = |F_i|$. Since the boundary of the union K_i of F_i can be regarded as the upper envelope of the $2m_i$ rays bounding the wedges of F_i (after rotating the coordinate system so that θ_i points in the negative y direction), this boundary has complexity $O(m_i)$. (This is a special case of a result proved in [2]; it can also be easily shown using standard Davenport-Schinzel theory (cf. [11]).)

We now take each pair, F_i, F_j, of subfamilies, and bound the boundary complexity of the union $K_i \cup K_j$. The analysis given in [13] implies that this complexity is $O((m_i + m_j)/\delta^2)$. We will improve this bound to $O((m_i + m_j)/\delta \cdot \log(1/\delta))$ as follows: We will first bound the number of holes in the union $K_i \cup K_j$; by the combination lemma of [7], our bound will carry over from the number of holes to the boundary complexity.

Let us shrink each wedge in F_i by rotating its sides inwards until they form angles $\delta/3$ with θ_i, and similarly for F_j. If we imagine this shrinking as a continuous process, we see that the number of holes decreases only when an apex of some wedge becomes uncovered and two holes grow together (see [13]). It follows that the shrinking process may eliminate at most $m_i + m_j$ holes. Denote by K_i^* (resp. K_j^*) the union of F_i (resp. F_j) after the shrinking process.

Following the analysis of [13], the only holes that we need to consider now are quadrangular holes bounded by two wedges W, W' of F_i and two wedges V, V' of F_j, so that each of W, W' fully penetrates through both V, V', and vice versa (see Figure 1). The number of holes of all other types, as argued in [13], is only $O(m_i + m_j)$, because they can be associated with ('charged' to) the vertices of K_i^* and K_j^*. Let us rotate the coordinate system so that the orientation θ_i becomes $0 < \alpha < \frac{\pi}{2}$ and θ_j becomes $\pi - \alpha$, and suppose that W' lies lower than W (i.e. to the south-east of W) and V' lies lower than V (i.e. to the south-west of V). We charge each such hole either to the pair (W, W') or to the pair (V, V'); the pair (W, W') is charged if the apex of W' has a higher y coordinate than the apex of V'; otherwise the pair (V, V') is charged; see Figure 1.

Consider now the collection of holes (of the above special structure) that have been charged to a pair (W, W') of wedges of F_i. To each such hole there corresponds an interval along the lower ray of W, which is its intersection with the corresponding wedge V'; this interval is marked in Figure 1. By the fact that the holes are charged to (W, W'), these intervals must be disjoint. Figure 2 shows the densest possible packing of such intervals along the lower boundary of W, yielding the maximum number of holes that can be charged to (W, W'). Now, referring to the notations in Figure 2, since the triangles $A_i M_i M_{i+1}$ are all similar to the triangle $B M_0 A_0$, with the angle $2\delta/3$ at A_i and at B, respectively, we have

$$\frac{M_i M_{i+1}}{A_i M_i} = \frac{M_0 A_0}{M_0 B} \equiv p \geq \sin \frac{2\delta}{3} .$$

The lengths of the intervals on the lower edge of W form a geometric progression:

$$\frac{AM_{i+1}}{AM_i} = \frac{AM_1}{AM_0} = \frac{AM_0 + M_0 M_1}{AM_0} = 1 + \frac{M_0 M_1}{A_0 M_0} = 1 + p,$$

since $AM_0 = A_0 M_0$ by construction; hence, if k holes are charged to the pair (W, W'), we

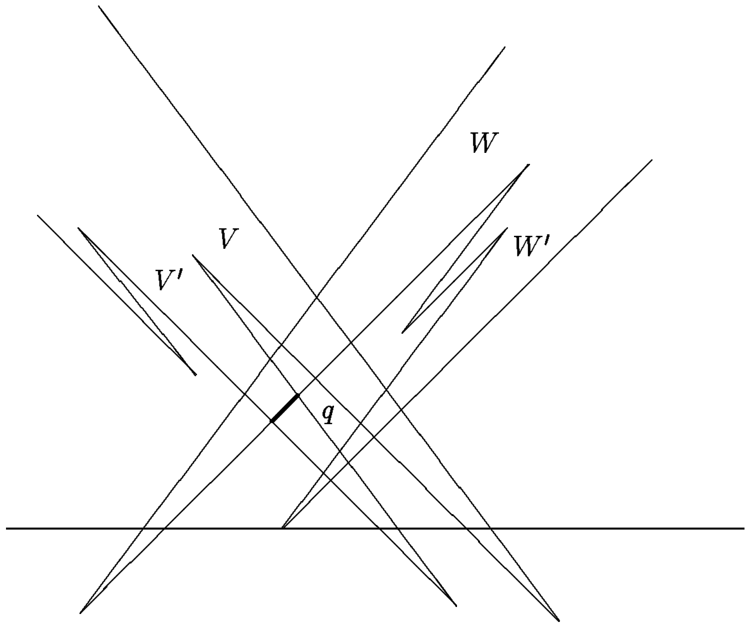

Figure 1: A quadrangular hole q charged to (W, W')

have

$$\frac{AM_k}{AM_0} = (1+p)^k \, .$$

On the other hand,

$$\frac{AM_k}{AM_0} < \frac{AB}{AM_0} = \frac{AM_0 + M_0 B}{AM_0} = 1 + 1/p \, ,$$

and thus

$$(1+p)^k < 1 + \frac{1}{p} \, .$$

Using the inequality $\ln(1+p) \geq p/(p+1)$, we obtain

$$k \leq \Big(1 + \frac{1}{p}\Big) \cdot \ln\Big(1 + \frac{1}{p}\Big) = O\Big(\frac{1}{p} \log \frac{1}{p}\Big) = O\Big(\frac{1}{\sin(2\delta/3)} \log \frac{1}{\sin(2\delta/3)}\Big) = O\Big(\frac{1}{\delta} \log \frac{1}{\delta}\Big) \, .$$

It has been shown in [13], by a simple visibility argument, that the number of pairs (W, W') that can yield holes of this type in the union $K_i^* \cup K_j^*$ is $O(m_i)$. Hence the total number of holes in $K_i^* \cup K_j^*$, and thus also in $K_i \cup K_j$, is $O((m_i + m_j)\delta^{-1} \log(1/\delta))$. Returning now to the union $K_i \cup K_j$ of the *original* families F_i, F_j (before the shrinking process), we have seen above that they form at most $O(m_i + m_j)$ more holes than $K_i^* \cup K_j^*$; thus, the same asymptotic bound holds for them.

To bound the boundary complexity of $K_i \cup K_j$, we apply the combination lemma of [7], as in [13]:

Lemma A.1 *Let c_1 and c_2 denote the boundary complexity of K_1 and K_2, respectively. If $K_1 \cup K_2$ has q holes, the boundary complexity of $K_1 \cup K_2$ is at most $O(c_1 + c_2 + q)$.*

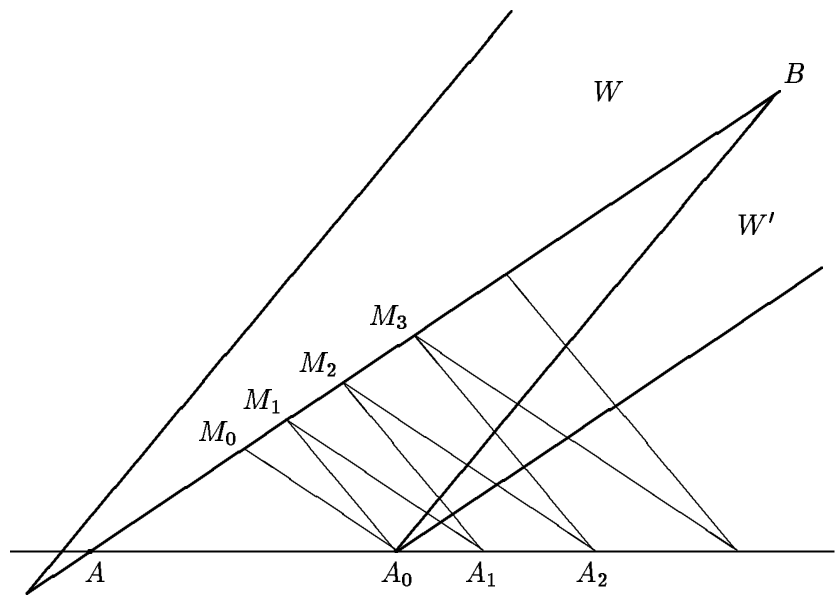

Figure 2: The maximum possible number of holes charged to the pair (W, W')

This lemma implies that the bound $O((m_i + m_j)\delta^{-1}\log(1/\delta))$ is also valid for the overall complexity of the union $K_i \cup K_j$.

Since each boundary vertex of the union of all double wedges must appear as a vertex in the union of two families F_i and F_j, we may just sum this bound over all pairs of families. Noting that $\sum_{i,j}(m_i + m_j) = O(m \cdot (1/\delta))$, we obtain that the overall boundary complexity of the union of F is $O(m\delta^{-2}\log(1/\delta))$, as asserted.

To compute the union, we adapt an idea from [14]. We first compute the union K_i of each F_i, using known algorithms for computing upper envelopes [2, 12], in time $O(m_i \log m_i)$, for $i = 1, \ldots, t$. Next we apply one of the recent optimal algorithms for line segment intersections [3, 4, 15], to compute the union of all the K_i's. The algorithm runs in time $O(N \log N + k)$, where N is the total size of all the K_i's, and k is the number of intersections between their boundaries. By what has just been observed, we have $N = O(m)$ and $k = O(m\delta^{-2}\log(1/\delta))$. Thus the algorithm takes time $O(m\delta^{-2}\log(1/\delta) + m\log m)$. $\square$