Computing the smallest enclosing ball of balls

Mike Bostock and Robin Houston

May 17, 2017

Abstract

We consider the problem of computing the smallest ball enclosing a set of balls. There is an elegant randomised algorithm due to Welzl (1991) that computes the smallest ball enclosing a set of points: it was shown by Fischer and Gärtner (2003); Fischer (2005) that a natural extension of Welzl's algorithm to sets of balls does not work in general. We observe that a trivial modification to this algorithm makes it work correctly.

1 Welzl's algorithm

Welzl's algorithm computes the smallest ball that contains all the points in a finite set $S \subseteq \mathbb{R}^d$. We shall denote this smallest ball as $\mathrm{mb}(S)$.

It uses an algorithm BallBoundedBy(T) that takes a finite set of points in $\mathbb{R}^d$ and computes the smallest d-ball whose boundary contains all the points in T. This amounts to solving a system of $|T|$ quadratic equations. [Explain how this reduces to one quadratic equation and $|T| - 1$ linear equations, etc.] There may be no such ball, in which case the algorithm returns **null**.

Using BallBoundedBy as a subroutine, the following recursive algorithm due to Welzl (1991) computes the smallest ball containing a finite set of points, with some of them constrained to be on the ball's boundary. Again there may be no such ball, in which case the algorithm will return **null**.

```
function WelzlP($S, T$)
    if $S = \emptyset$ then
        return BallBoundedBy($T$)
    end if
    Choose $p \in S$ uniformly at random
    $D \leftarrow$ WelzlP($S - \{p\}, T$)
    if $D =$ null or $p \in D$ then
        return $D$
    else
        return WelzlP($S - \{p\}, T \cup \{p\}$)
    end if
end function
```

Now we may compute $\mathrm{mb}(S)$ simply by calling WelzlP($S, \emptyset$).

2 Enclosing ball of balls

Let's write $\mathrm{mb}(S)$ to mean the smallest ball containing all the balls in S, where S is a finite set of balls in $\mathbb{R}^d$. [Show this exists and is unique.

Introduce convex combinations of balls here, or in the previous section?] Let's also write $\mathrm{mb}(S, T)$ to mean the smallest ball containing all the balls in $S \cup T$ that has all the balls in T tangent to its boundary. In general $\mathrm{mb}(S, T)$ may not exist, and if it does exist it may not be unique. [Give examples.]

In this section we exhibit a variant of Welzl's algorithm that is designed to be easy to analyse. Below we'll show how it can be made simpler and more efficient to implement, at the expense of being harder to analyse.

There is an algorithm MbTangent(T) that computes $\mathrm{mb}(T)$ provided that all the balls in T are tangent to $\mathrm{mb}(T)$, and returns **null** otherwise. This can be implemented in various ways. One possibility is first to compute the smallest ball to which all the balls in T are internally tangent, if there is one. This is similar to the algorithm BallBoundedBy described above. [Explain further.] If there is no such ball, return **null**. If there is, check whether it is equal to $\mathrm{mb}(T)$ by checking whether its centre is contained in the convex hull of the centres of the balls in T. (Fischer and Gärtner, 2003, Lemma 2.2, Fischer, 2005, Lemma 3.3)

Here is our version of Welzl's algorithm adapted to computing the bounding ball of balls.

```
{ Compute $\mathrm{mb}(S \cup T)$ if $\mathrm{mb}(S \cup T) = \mathrm{mb}(S, T)$ }
function WelzlB($S, T$)
    if $S = \emptyset$ then
        return MbTangent($T$)
    end if
    Choose $B \in S$ uniformly at random
    $D \leftarrow$ WelzlB($S - \{B\}, T$)
    if $D =$ null or $B \not\subseteq D$ then
        return WelzlB($S - \{B\}, T \cup \{B\}$)
    else
        return $D$
    end if
end function
```

The only change to the actual algorithm is that a **null** return value from a recursive call is handled in the opposite way: we treat **null** as a ball that contains nothing, rather than as a ball that contains everything. More important is the way we have weakened the description of the algorithm: it only computes $\mathrm{mb}(S, T)$ when it is equal to $\mathrm{mb}(S \cup T)$. Of course this makes no difference in the case we principally care about, where the algorithm is initially called with $T = \emptyset$.

3 Proof of correctness

Theorem 3.1. *For any finite sets S, T of d-balls,* WelzlB(S, T) *returns* $\mathrm{mb}(S, T)$ *if* $\mathrm{mb}(S, T) = \mathrm{mb}(S \cup T)$*, and* **null** *otherwise.*

Proof. We proceed by induction on $|S|$. For the base case: if $S = \emptyset$ then the claim is true by the correctness of MbTangent. So assume we have chosen $B \in S$, and that the claim holds for WelzlB($S - \{B\}, T$) and WelzlB($S - \{B\}, T \cup \{B\}$).

Let M equal $\mathrm{mb}((S - \{B\}) \cup T)$. Suppose first that $B \subseteq M$. In this case $M = \mathrm{mb}(S \cup T)$. We know WelzlB($S - \{B\}, T$) returns M if all the balls in T are tangent to M and **null** otherwise, so in either case it is correct to return WelzlB($S - \{B\}, T$).

Next suppose that $B \not\subseteq M$. Let N equal $\mathrm{mb}(S \cup T)$. We know B is tangent to N. [This should probably be an explicit lemma. It follows from the standard argument using convex combinations of balls.] If all the balls in T are also tangent to N then $\textsc{WelzlB}(S - \{B\}, T \cup \{B\})$ returns N, correctly. If not then it returns **null**, also correctly. □

4 Simplifying the base case

[TODO. Weaken the requirements on $\textsc{MbTangent}(T)$, so it's allowed – but not required – to return a minimal ball to which all the balls in T are internally tangent even when that is not equal to $\mathrm{mb}(T)$. Requires more sophisticated analysis of the geometry. This is the next part I want to sketch out.]

[TODO. NEEDS MORE THOUGHT. If we make some assumptions about the ordering – which should be satisfied both when keeping a fixed order and when using move-to-front – I suspect we can argue that the balls passed to MbTangent have affinely independent centres, so one can get away with not handling that case. I certainly hope so, since the D3 implementation of `enclose3` doesnt work when the centres of the three disks are in a line!]

5 Practical considerations

[Maybe we should remove this section, since we'll probably have to discuss this stuff above anyway.] [Just like Welzls original algorithm. We can just shuffle the input balls once at the outset, rather than making a random choice at each step. This is more efficient and still runs in expected linear time. Also the move-to-front heuristic seems to make it faster in practice: explain why this is intuitively plausible, though not supported by rigorous theory. (Also do a more thorough literature search to see if anyone has done any theoretical analysis of the move-to-front heuristic.)]

6 Experimental results

[Compare our algorithm to the MSW algorithm, and report how much faster it is.]

7 Move-to-front

[Can we say anything about whether the move-to-front heuristic has the effect that $\textsc{MbTangent}(T)$ is only ever called when all the disks in T are tangent to $\mathrm{mb}(T)$? This sounds vaguely plausible, and seems to be true in the small number of two-dimensional examples I've tried.]

References

Kaspar Fischer. *Smallest enclosing balls of balls*. PhD thesis, Swiss Federal Institute of Technology, ETH Zürich, 2005.

Kaspar Fischer and Bernd Gärtner. The smallest enclosing ball of balls: Combinatorial structure and algorithms. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, SCG '03, pages 292–301, New York, NY, USA, 2003. ACM. ISBN 1-58113-663-3. doi: 10.1145/777792.777836. URL `http://doi.acm.org/10.1145/777792.777836`.

Emo Welzl. Smallest enclosing disks (balls and ellipsoids). *New Results and New Trends in Computer Science*, pages 359–370, 1991.