Point Location in Well-Shaped Meshes Using Jump-and-Walk*

Jean-Lou De Carufel† Craig Dillabaugh‡ Anil Maheshwari§

Abstract

We present results on executing point location queries in well-shaped meshes in $\mathbb{R}^2$ and $\mathbb{R}^3$ using the *Jump-and-Walk* paradigm. If the jump step is performed on a nearest-neighbour search structure built on the vertices of the mesh, we demonstrate that the walk step can be performed in guaranteed constant time. Constant time for the walk step holds even if the jump step starts with an approximate nearest neighbour.

1 Introduction

Point location is a topic that has been extensively studied since the origin of computational geometry. In $\mathbb{R}^2$, the point location problem, typically referred to as planar point location, can be defined as follows. Preprocess a planar subdivision, specified as the union of n triangles, so that given a query point q, the triangle containing q can be reported efficiently. There are several well known results showing that a data structure with $O(n)$ space can report such queries in $O(\log n)$ time. In $\mathbb{R}^3$, we have a subdivision of the three dimensional space into tetrahedra, and again given a query point q we wish to return the tetrahedron containing q. From a theoretical standpoint, the problem of general spatial point location in $\mathbb{R}^3$ is still open [1].

In this paper, we consider a more specialized problem. We wish to answer point location queries in two and three dimensions for well-shaped triangular and tetrahedral meshes. A well-shaped mesh, denoted by $\mathcal{M}$, is one in which all its simplices have bounded aspect ratio (see Definition 1). This assumption is valid for mesh generation algorithms that enforce the well-shaped property on their output meshes [6]. Our motivation came from an external memory setting, where we have examined data structures for representations that permit efficient path traversals in meshes that typically do not fit in the main memory [3].

Let P be the set of vertices defining $\mathcal{M}$. In this paper we show that, given a query point q, and its (exact or approximate) nearest neighbor $p \in P$, the number of triangles (or tetrahedra) intersected by the line segment pq is bounded by a constant. On the basis of this result, we develop a point location method following the *Jump-and-Walk* paradigm, which typically works as follows (Devroye *et al.* [2], Mücke *et al.* [8]):

Jump Step: Select a set of possible start (jump) points and store them in a data structure that can efficiently answer proximity queries. Given a query point q, locate a nearest neighbor of q (say p) and then Jump to p.

Walk Step: Walk through the sequence of simplices, starting at p, in a straight line, towards q, until the simplex containing q is located.

While jump-and-walk gives expected search times in most instances, and is often slightly less efficient theoretically than other techniques, it has the advantage of being simple and is often very efficient in practice.

1.1 Our Results

We present results for the "Walk"-step in the Jump-and-Walk strategy in $\mathbb{R}^2$ and $\mathbb{R}^3$ for well shaped triangular and tetrahedral meshes. In particular, we show that

1. Given a well-shaped mesh $\mathcal{M}$ in $\mathbb{R}^2$ or $\mathbb{R}^3$, jump-and-walk search can be performed in the time required to perform an *exact* nearest neighbour search on the vertices of $\mathcal{M}$ plus *constant time* for the walk-step to find the triangle/tetrahedron containing the query point.

2. Given a well-shaped mesh $\mathcal{M}$ in $\mathbb{R}^2$ or $\mathbb{R}^3$, jump-and-walk search can be performed in the time required to perform an *approximate* nearest neighbour (see Definition 2) search on the vertices of $\mathcal{M}$, plus *constant time* for the walk-step to find the triangle/tetrahedron containing the query point.

While we present results in both $\mathbb{R}^2$ and $\mathbb{R}^3$, we feel that our most interesting contribution is the constant time walk-step in $\mathbb{R}^3$ using approximate nearest neighbour for the jump-step. In $\mathbb{R}^3$, there are no efficient structures for answering exact nearest neighbor queries; in spite of that we are able to show that the walk-step can be performed in constant time using only the knowledge of an approximate nearest neighbor. The major advantage of our approach, in addition to being theoretically optimal, as it matches the query time for this setting as presented in [10], is that it is also practical. The practicality of approximate nearest neighbor

*Research supported by funding from NSERC.

†Computational Geometry Lab, School of Computer Science, Carleton University

‡School of Computer Science, Carleton University, cdillaba@cg.scs.carleton.ca

§School of Computer Science, Carleton University, anil@scs.carleton.ca

searching has already been demonstrated, see ANN Library [7], and the implementation of the walk-step is fairly trivial and straightforward.

2 Background

In this paper we consider point location in triangular and tetrahedral meshes in two and three dimensions respectively. We use $\mathcal{M}$ to denote a mesh in either $\mathbb{R}^2$ or $\mathbb{R}^3$, and if we want to be specific we use $\mathcal{M}_2$ to denote a triangular mesh in $\mathbb{R}^2$, and $\mathcal{M}_3$ to denote a tetrahedral mesh in $\mathbb{R}^3$. We assume that the triangles and tetrahedra, which are collectively referred as simplices, are *well-shaped*, a term which will be defined shortly. If all simplices of a mesh $\mathcal{M}$ are well-shaped, then $\mathcal{M}$ is said to be a *well-shaped mesh*. Triangles in $\mathcal{M}_2$ are considered adjacent if and only if they share an edge. Similarly, tetrahedra in $\mathcal{M}_3$ are considered adjacent if and only if they share a face.

Well-Shaped Meshes: We begin by stating the *well-shaped* property (refer to [6]).

Definition 1 *We say that a mesh $\mathcal{M}_2$ ($\mathcal{M}_3$) is* well-shaped *if for any triangle (tetrahedron) $t \in \mathcal{M}_2$ ($t \in \mathcal{M}_3$), the ratio formed by the radius $r(t)$ of the incircle (insphere) of t and the radius $R(t)$ of the circumcircle (circumpshere) of t is bounded by a constant ρ, i.e. $\frac{R(t)}{r(t)} < \rho$.*

In this paper, all meshes and simplicies (triangles and tetrahedra) are assumed to be well-shaped. We make the following observations related to Definition 1.

Observation 1 *Let t be a triangle (tetrahedron).*

1. *Let v be any vertex of t. Denote by e_v (f_v) the opposite edge (face) of v in t. Let*

   $$\texttt{mdist}(v,t) = \min_{x \in e_v} |xv| \qquad (\texttt{mdist}(v,t) = \min_{x \in f_v} |xv|),$$

   where the minimum is taken over all points x on e_v (f_v). Therefore, $\texttt{mdist}(v,t)$ is an upper-bound on the diameter of the incircle (insphere of) t. Formally, $2r(t) \leq \texttt{mdist}(v,t)$.

2. *Let e be the longest edge of t. The diameter of the circumcircle (circumsphere) of t is at least as long as e. Formally, $2R(t) \geq |e|$.*

Observation 2 *There is a lower bound of α for each of the angles in any triangle of $\mathcal{M}_2$. There is a lower bound of Ω for each of the solid angles in any tetrahedron of $\mathcal{M}_3$. In particular, we have $\alpha \leq \frac{\pi}{3}$ and $\cos(\alpha) = \frac{1+\sqrt{\rho(\rho-2)}}{\rho}$ for the two dimensional case. For the three dimensional case, $\Omega \leq 3\arccos\left(\frac{1}{3}\right) - \pi$ and $\sin\left(\frac{\Omega}{2}\right) = \frac{3\sqrt{3}}{8\rho^2}$ (see [5]).*

Jump-and-Walk for Point Location: In [8], point location queries using the jump-and-walk are addressed for Delaunay triangulations of a random set of points in $\mathbb{R}^2$ and $\mathbb{R}^3$. Devroye *et al.* [2] showed that the expected search times for the jump-and-walk in Delaunay triangulations range from $\Omega(\sqrt{n})$ to $\Omega(\log n)$, depending on the distribution and the specific data structure employed for the jump step.

Nearest Neighbour Queries: The nearest neighbour query works as follows. Given a point set P and a query point q, return the point $p \in P$ nearest to q, i.e. for all $v \in P$, $|pq| \leq |vq|$. A closely related query is the approximate nearest neighbour (ANN) query defined as follows.

Definition 2 *Let P be a set of points in $\mathbb{R}^d$, q be a query point and $p \in P$ be an exact nearest neighbour of q. Given an $\epsilon \geq 0$, we say that a point $\hat{p} \in P$ is an $(1+\epsilon)$-*approximate nearest neighbour *of q if $|\hat{p}q| \leq (1+\epsilon)|pq|$.*

3 Planar Point Location in $\mathcal{M}_2$

Let P be the set of vertices of a well-shaped mesh $\mathcal{M}_2$ and q be a query point lying in a triangle of $\mathcal{M}_2$. Let p be a nearest neighbour of q. Consider the set of triangles encountered in a straight-line walk from p to q in $\mathcal{M}_2$.

Lemma 1 *The walk-step along pq visits at most $\left\lfloor \frac{\pi}{\alpha} \right\rfloor$ triangles.*

Proof. Without loss of generality, suppose $|pq| = 1$. Let $\mathcal{C}(q,|pq|)$ be the circle with centre q and radius $|pq|$. Since p is a nearest neighbour of q, there is no vertex of $\mathcal{M}_2$ in the interior of $\mathcal{C}(q,|pq|)$. Denote by ℓ the line through pq and let $p' \neq p$ be the intersection of ℓ with $\mathcal{C}(q,|pq|)$ (see Fig. 1). Since $\mathcal{C}(q,|pq|)$ is a unit circle, the arc $\widehat{pp'}$ has length π. Any triangle intersecting pq intersects ℓ in the interior of $\mathcal{C}(q,|pq|)$. All such triangles have one vertex to the left of ℓ and two vertices to the right of ℓ or vice-versa. (If a vertex is on ℓ, consider it to be on the right.) We separate the triangles intersecting ℓ into two sets, L and R containing the triangles with exactly one vertex to the left, and right, of ℓ, respectively. Consider an arbitrary triangle $t \in L$ and let the vertices of t be a, b and c. The edge ab (respectively ac) intersects $\mathcal{C}(q,|pq|)$ at b' and b'' (respectively at c' and c'') (see Fig. 1). Let $\theta = \angle bac$. Since ab and ac are two secants which intersect $\mathcal{C}(q,|pq|)$, we know that $\theta = \frac{1}{2}(\widehat{b'c'} - \widehat{c''b''})$, from which we conclude $\widehat{b'c'} \geq 2\theta$. Hence $\widehat{b'c'} \geq 2\theta \geq 2\alpha$ by Observation 2. Therefore, we can conclude that the set L contains at most $\left\lfloor \frac{\pi}{2\alpha} \right\rfloor$ triangles. The same bound holds for triangles in R. Thus the number of triangles intersecting pq is at most $\left\lfloor \frac{\pi}{\alpha} \right\rfloor$. $\square$

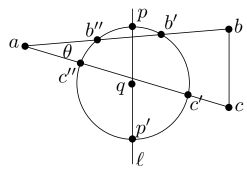

Figure 1: A triangle with fixed minimum angle covers an arc bounded by a minimum fixed length on $\mathcal{C}$.

Next consider the scenario where $\hat{p} \in P$ is an approximate nearest neighbor of the query point q. Note that $\mathcal{C}(q, |\hat{p}q|)$ may contain vertices of $\mathcal{M}_2$. Therefore, the proof of Lemma 1 does not apply for the walk-step along $\hat{p}q$. Next, we prove that for an ANN search structure (see Definition 2), we can find an ϵ such that the number of triangles encountered in a straight line walk from $\hat{p}$ to q, is bounded by a constant. We begin with the following lemma.

Lemma 2 *Let t be a well-shaped triangle and $\mathcal{C}$ be a circle of radius $r(\mathcal{C})$ such that none of the vertices of t are in the interior of $\mathcal{C}$. If (i) at least two edges of t intersect $\mathcal{C}$ or if (ii) t contains the centre of $\mathcal{C}$, then t has at least one edge of length at least $2r(\mathcal{C}) \sin\alpha$.*

Proof. Let $t = \triangle abc$. From Observation 2, we know that $\alpha \leq \angle bac$.

(i) Suppose that all the edges of t are strictly smaller than $2r(\mathcal{C}) \sin\alpha$ for a contradiction. Let $\mathcal{C}'$ be the biggest circle that can be constructed such that none of the vertices of t are in the interior of $\mathcal{C}$ and at least two edges of t intersect $\mathcal{C}'$. Thus, $\mathcal{C}'$ is strictly smaller than the circumcircle of the equilateral triangle of side length $2r(\mathcal{C}) \sin\alpha$. Therefore, by elementary geometry,

$$\begin{aligned} r(\mathcal{C}') &< \frac{2\sqrt{3}r(\mathcal{C})}{3} \sin\alpha \\ &\leq \frac{2\sqrt{3}r(\mathcal{C})}{3} \sin\left(\frac{\pi}{3}\right) \qquad \text{by Observation 2,} \\ &= r(\mathcal{C}), \end{aligned}$$

which is a contradiction.

(ii) Suppose that less than two edges of t intersect $\mathcal{C}$ and t contains the centre of $\mathcal{C}$. If t contains $\mathcal{C}$, then all the edges of t are longer than $2r(\mathcal{C}) \geq 2r(\mathcal{C}) \sin\alpha$. Suppose exactly one edge of t intersects $\mathcal{C}$. Let ab be this edge. We form a new triangle t' by translating ac and bc inward until one of ac or ab intersects $\mathcal{C}$. Now t' satisfies the hypothesis of Case (i).

□

Observation 3 *Let $t_i = \triangle abc$ be a well-shaped triangle.*

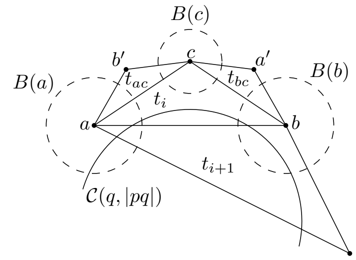

Figure 2: Neighbourhood of the triangle t_i from which the path pq leaves $\mathcal{C}$.

1. *Let t_{i+1} be the well-shaped triangle adjacent to t_i at edge ab. The edges of t_{i+1} have length at least $|ab| \sin\alpha$.*

2. *Let $a \in \mathcal{M}_2$ be a vertex and ab be an edge incident to a. The edges of any triangle incident to a have length at least $|ab| \sin^{\lfloor \frac{\pi}{\alpha} \rfloor} \alpha$.*

Proof.

1. It follows from the well-shaped property.

2. From Observation 3-1, if a triangle t in $\mathcal{M}_2$ has an edge of length L, then no triangle that can be reached by walking from t through at most d edge adjacent triangles has an edge shorter than $L \sin^{d+1}\alpha$. Then the result follows from Observation 2.

□

Consider the walk from $\hat{p}$ to q in $\mathcal{M}_2$. It intersects the boundary of $\mathcal{C}(q, |pq|)$ at a point x. Let t_i be the first triangle we encounter in the walk from $\hat{p}$ to q that contains x.

Observation 4 *t_i has an edge of length at least $2|pq| \sin^2\alpha$.*

Proof. If t_i contains q, then by Lemma 2, t_i has an edge of length at least $2|pq| \sin\alpha \geq 2|pq| \sin^2\alpha$. If t_i does not contain q, then there is an edge of t_i intersecting $\hat{p}q$ in the interior of $\mathcal{C}(q, |pq|)$. Let a and b be the two vertices of this edge. Consider the triangle t_{i+1} adjacent to t_i across ab. If $q \in t_{i+1}$, then $|ab| \geq 2|pq| \sin^2\alpha$ by Lemma 2 and Observation 3-1, otherwise t_{i+1} has two edges intersecting $\mathcal{C}(q, |pq|)$. Again, by Lemma 2 and Observation 3-1, $|ab| \geq 2|pq| \sin^2\alpha$. □

Denote the vertices of t_i by a, b and c. Let $\mathcal{G}$ be the union of all the triangles incident to a, b, and c (see Fig. 2).

Lemma 3 *Let $x \in t_i$ be the intersection of $\hat{p}q$ with the boundary of $\mathcal{C}(q, |pq|)$. Let $y \in \mathcal{G}$ be the intersection of the line through $\hat{p}q$ with the boundary of $\mathcal{G}$ such that x is between q and y. Then $|xy| \geq 2|pq| \sin^{\lfloor \frac{\pi}{\alpha} \rfloor + 4}\alpha$.*

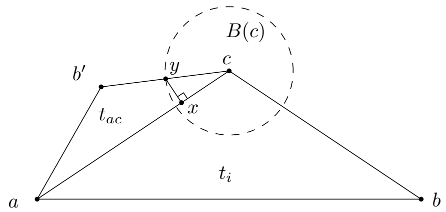

Figure 3: Illustration of the proof of Lemma 3.

Proof. Denote by $t_{ac} = \triangle ab'c$ (respectively by $t_{bc} = \triangle a'bc$) the triangle adjacent to t_i at ac (respectively at bc) (see Fig. 2(a)). Note that t_{ac} and t_{bc} are in $\mathcal{G}$.

By Observations 3-2 and 4, the length of all edges incident to a (respectively to b and to c) is at least $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+2}\alpha$ (respectively $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+2}\alpha$ and $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+3}\alpha$ by Observation 3-1). Therefore $\mathcal{G}$ contains a ball $B(a)$ (respectively $B(b)$ and $B(c)$) with centre a (respectively b and c) and radius $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+2}\alpha$ (respectively $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+2}\alpha$ and $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+3}\alpha$), which does not contain any vertices of $\mathcal{M}_2$ in its interior.

To minimize $|xy|$, we take x on the boundary of t_i. We will find a lower bound for $|xy|$ by supposing, without loss of generality, that $y \in b'c$. Since y is supposed to be on the boundary of $\mathcal{G}$, it cannot be inside $B(c)$. With $x \in ac$ and $b \in b'c \setminus B(c)$, the smallest possible value for $|xy|$ is $2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+4}\alpha$ (see Fig. 3). □

We can now state our main result.

Theorem 4 *Let $\mathcal{M}_2$ be a well-shaped triangular mesh in $\mathbb{R}^2$. Given $\hat{p}$, an $(1+\epsilon)$-approximate nearest neighbour of a query point q from among the vertices of $\mathcal{M}_2$, the straight line walk from $\hat{p}$ to q visits at most $2\left\lfloor\frac{\pi}{\alpha}\right\rfloor$ triangles.*

Proof. Following the notation of Lemma 3, if $\hat{p} \in \mathcal{G}$, then the straight line walk from $\hat{p}$ to q visits at most $2\left\lfloor\frac{\pi}{\alpha}\right\rfloor$ triangles. There are $\left\lfloor\frac{\pi}{\alpha}\right\rfloor$ triangles for the part of the walk inside $\mathcal{C}(q,|pq|)$ (see Lemma 1) and $\left\lfloor\frac{\pi}{\alpha}\right\rfloor$ triangles for the part of the walk inside $\mathcal{G}$. Indeed, in the worst case, the walk inside $\mathcal{G}$ will either cross ab', $b'c$, ca' or $a'b$. So this walk will either cross the triangles incident to a, the triangles incident to b or the triangles incident to c.

We can ensure that $\hat{p} \in \mathcal{G}$ by building an ANN search structure with $\epsilon \le 2\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+4}\alpha$ on the vertices of $\mathcal{M}_2$. Indeed, in this case

$$\begin{aligned} |\hat{p}q| &\le \left(1+2\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+4}\alpha\right)|pq| \\ &= |pq| + 2|pq|\sin^{\lfloor\frac{\pi}{\alpha}\rfloor+4}\alpha \\ &\le |pq| + |xy| \qquad \text{by Lemma 3,} \\ &= |qx| + |xy| \\ &= |qy| \end{aligned}$$

because q, x and y are aligned in this order. So $\hat{p}$ must be in $\mathcal{G}$. □

4 Spatial Point Location in $\mathcal{M}_3$

Searching in a well-shaped three dimensional mesh $\mathcal{M}_3$ can be performed using essentially the same technique as outlined for $\mathcal{M}_2$ in Section 3. Let P denote the set of vertices of $\mathcal{M}_3$. For a query point q, let $p \in P$ be its nearest neighbour. We will perform the walk-step starting at p and walk towards q in a straight line, and we will show that we visit only a constant number of tetrahedra. Let $\mathcal{S}(q,|pq|)$ denote a ball of radius $|pq|$ centred at q.

Theorem 5 *Let $\mathcal{M}_3$ be a well-shaped triangular mesh in $\mathbb{R}^3$. Given p, a nearest neighbour of a query point q from among the vertices of $\mathcal{M}_3$, the walk from p to q visits at most $\frac{1}{64}\rho^3(\rho^2+4)^3$ tetrahedra.*

Proof. We do not prove Theorem 5 due to lack of space. Refer to the extended version of the paper. □

Next, we assume that $\hat{p}$ is an approximate nearest neighbor of q. First, we establish a geometric lemma. Consider an arbitrary ball $\mathcal{S}$. We say $\mathcal{S}$ is an empty ball if it contains no vertex of $\mathcal{M}_3$. Note that edges and faces of $\mathcal{M}_3$ may intersect $\mathcal{S}$. Let f be a face in $\mathcal{M}_3$ that intersects $\mathcal{S}$. We have the following Lemma.

Lemma 6 *Let $\mathcal{T} = abcd \in \mathcal{M}_3$ be a tetrahedron and $\mathcal{S}$ be a sphere of radius $r(\mathcal{S})$ such that none of the vertices of $\mathcal{T}$ are in the interior of $\mathcal{S}$. If (i) $f = \triangle abc$ is tangent to $\mathcal{S}$ or if (ii) f crosses $\mathcal{S}$ in a way that $f \cap \mathcal{S}$ is a disk, then $\frac{2}{\rho}r(\mathcal{S})$ is a lower bound on the length of edges ad, bd and cd.*

Proof. (i) Let the tangent point be x. Let $\mathcal{H}$ be the supporting plane of f. Without loss of generality, assume that $\mathcal{H}$ is horizontal, and $\mathcal{S}$ is below $\mathcal{H}$. There are two tetrahedra of $\mathcal{M}_3$ that are adjacent to f. We will focus on the tetrahedron that is below $\mathcal{H}$, and denote it by t_{i+1}. Let d be the fourth vertex of t_{i+1}. If we place x at the pole of $\mathcal{S}$ (we are free to rotate $\mathcal{S}$) and take the equator of $\mathcal{S}$ and project it onto $\mathcal{H}$, we obtain a cylinder, say $\mathcal{C}$. The complement of $\mathcal{S}$ with respect to $\mathcal{C}$ defines the region in which d can be placed (see Figure 4). If d is outside this region then $|xd|$ is greater than the radius of $\mathcal{S}$, and we have a nice lower bound on $|xd|$. Let the point d' be the projection of d onto $\mathcal{H}$.

Now consider some placement of the point d, and assume that d touches the surface of $\mathcal{S}$ (which is the worst case in this setting). Consider the line segments xd and dd' and observe that

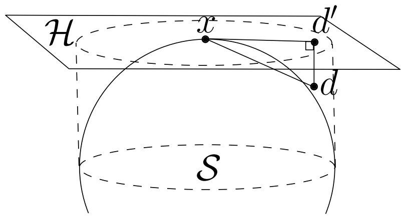

Figure 4: Illustration of proof of Lemma 6.

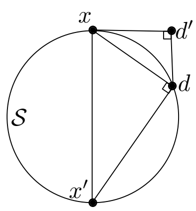

Figure 5: Determining the bound for $|xd|$.

(a) dd' is at at least twice the radius of the insphere $r(t_{i+1})$ of t_{i+1} by Observation 1-1. Formally, $|dd'| \geq 2r(t_{i+1})$.

(b) xd lies completely within t_{i+1}. Then $|xd| \leq 2R(t_{i+1})$ by Observation 1-2.

Without loss of generality we assume that $\mathcal{S}$ is centred at the origin of our coordinate system. Consider the situation on the plane through the parallel lines Ox and dd' (both lines are normal to $\mathcal{H}$) as depicted in Fig. 5. By the definition of a well-shaped tetrahedron we know that $\frac{R(t_{i+1})}{r(t_{i+1})} \leq \rho$, and by the observations above, we have $\frac{|xd|}{|dd'|} \leq \rho$. Let $x' \neq x$ be the intersection of the line through Ox with $\mathcal{S}$. By elementary geometry, the triangles $\triangle xdx'$ and $\triangle dd'x$ are similar. Therefore, $\frac{2r(\mathcal{S})}{|xd|} = \frac{|xx'|}{|xd|} = \frac{|xd|}{|dd'|} \leq \rho$, so $|xd| \geq \frac{2}{\rho} r(\mathcal{S})$ (see Fig. 5).

(ii) If f crosses $\mathcal{S}$, then the intersection of f with $\mathcal{S}$ forms a circle on $\mathcal{S}$ (because $\mathcal{S}$ is empty). Let x' be the center of this circle. If we translate $\mathcal{T}$ so that f is tangent to $\mathcal{S}$ at x' then $\mathcal{T}$ satisfies the hypothesis of Case (i).

□

Observation 5 *Let $t_i = abcd$ be a well-shaped tetrahedron.*

1. *Let t_{i+1} be the well-shaped tetrahedron adjacent to t_i at edge ab. There exists a constant k_Ω that depends only on Ω such that the edges of t_{i+1} have length at least $|ab|k_\Omega$.*
2. *Let $a \in \mathcal{M}_3$ be a vertex and ab be an edge incident to a. The edges of any tetrahedron incident to a have length at least $|ab|k_\Omega^{\lfloor \frac{2\pi}{\Omega} \rfloor}$.*

Proof. 1. Let $v_0 = a$, $v_1 = b$, $v_2 = c$ and $v_3 = d$. Denote the volume of t_i by V and the solid angle at vertex v_i by θ_i. We have (see [5])

$$\sin\left(\frac{\theta_0}{2}\right) = \frac{12V}{\sqrt{\prod_{1 \leq i < j \leq 3} ((|v_0v_i| + |v_0v_j|)^2 - |v_iv_j|^2)}} \quad (1)$$

Let $l = |v_0v_1|$ and suppose without loss of generality that the edges of t_{i+1} have length at least 1 (hence $l > \frac{\sqrt{3}}{3}$). We first explain how to find the biggest possible value $l_{\max}$ for l such that $\theta_0 \geq \Omega$. The worst case is when the edges v_1v_2, v_2v_3 and v_1v_3 all have minimum length 1. Therefore, suppose that $\triangle v_1v_2v_3$ is an equilateral triangle. We are looking for the position of v_0 that maximizes l and such that $\theta_0 \geq \Omega$. Let Δ be the line perpendicular to $\triangle v_1v_2v_3$ that contains the centroid of $\triangle v_1v_2v_3$. To maximize l, we need to take v_0 on Δ.

Therefore, the height of t_i with respect to $\triangle v_1v_2v_3$ is equal to $\sqrt{l^2 - \frac{1}{3}}$ and $V = \frac{\sqrt{3l^2-1}}{12}$. As we move v_0 up, the solid angle θ_0 decreases. Therefore, by (1), we need to find the biggest $l > \frac{\sqrt{3}}{3}$ such that

$$\sin\left(\frac{\Omega}{2}\right) = \frac{\sqrt{3l^2-1}}{(4l^2-1)\sqrt{4l^2-1}}. \quad (2)$$

Let $l_{\max}$ be the biggest $l > \frac{\sqrt{3}}{3}$ that satisfies (2). Since (2) reduces to a cubic equation in l^2, $l_{\max}$ exists, it is unique and it can be computed exactly. We have $k_\Omega = \frac{1}{l_{\max}}$.

2. The proof is similar to the one of Observation 3-2. It uses Observation 5-1 and the fact that the full solid angle is 4π.

□

Consider the walk from $\hat{p}$ to q in $\mathcal{M}_3$. It intersects the boundary of $\mathcal{S}(q, |pq|)$ at a point x. Let t_i be the first tetrahedron we encounter in the walk from $\hat{p}$ to q that contains x.

Observation 6 *t_i has an edge of length at least $\frac{2}{\rho}|pq|$.*

Proof. Similar to the proof of Observation 4. □

We can now apply the same approach as we used in $\mathcal{M}_2$ to show that the number of tetrahedron visited along $\hat{p}q$ is a constant. Denote the vertices of t_i by a, b, c and d. Let $\mathcal{G}$ be the union of all the tetrahedra incident to a, b, c and d.

Lemma 7 *Let $x \in t_i$ be the intersection of $\hat{p}q$ with the boundary of $\mathcal{S}(q, |pq|)$. Let $y \in \mathcal{G}$ be the intersection of the line through $\hat{p}q$ with the boundary of $\mathcal{G}$ such that x is between q and y. Then $|xy| \geq \frac{2}{\rho}|pq|k_\Omega^{\lfloor \frac{2\pi}{\Omega} \rfloor + 1} \sin\left(\frac{\Omega}{2}\right)$.*

Proof. We follow the proof of Lemma 3. In two dimensions, the lower bound on $|xy|$ was computed by calculating the shortest exit out of a well-shaped triangle t. This shortest exit is perpendicular to an edge of t and constrained by the radius of the ball $B(c)$. In three dimensions, we calculate the shortest exit out of a well-shaped tetrahedron t. This shortest exit is perpendicular to a face of t, it goes through an edge of t and it is constrained by the radius of a ball in three dimensions. This leads to $|xy| \geq \frac{2}{\rho}|pq|k_\Omega^{\lfloor \frac{2\pi}{\Omega} \rfloor + 1} \sin\left(\frac{\Omega}{2}\right)$. □

Theorem 8 *Let $\mathcal{M}_3$ be a well-shaped tetrahedral mesh in $\mathbb{R}^3$. Given $\hat{p}$, an $(1+\epsilon)$-approximate nearest neighbour of a query point q from among the vertices of $\mathcal{M}_3$, the straight-line walk from $\hat{p}$ to q visits at most $\frac{1}{64}\rho^3(\rho^2+4)^3 + \lfloor \frac{2\pi}{\Omega} \rfloor$ tetrahedra.*

Proof. This proof is similar to the proof of Theorem 4 with $\epsilon \leq \frac{2}{\rho}k_\Omega^{\lfloor \frac{2\pi}{\Omega} \rfloor + 1} \sin\left(\frac{\Omega}{2}\right)$. □

5 Discussion

Our interest in this problem as such grew out of our research into efficient path traversals of large size well-shaped meshes in external memory settings (see [3]). However, it was assumed that the starting tetrahedron on such a path was given as part of the query. Adding the jump-and-walk point location step, results in efficiently answering a number of queries, without this assumption. Such queries include reporting the intersection of a box with the mesh (analogous to a window query in $\mathbb{R}^2$) or any other convex shape, and reporting streamlines.

To this point in the paper we have omitted any discussion of the data structures employed in the point location step. An ideal option in many ways is to employ a kd-tree, which is simple, can answer nearest neighbour queries in both $\mathbb{R}^2$ and $\mathbb{R}^3$ (and has I/O-efficient variants) [9]. Unfortunately, nearest neighbour queries in kd-trees, while good in the expected case [4], can in the worst-case require linear time. However, we are not aware of any work, which analyzes the worst-case query times for kd-trees with respect to vertices of a well-shaped mesh. An interesting follow on research topic to this paper would be to examine if query times for exact nearest neighbours in kd-trees, for points drawn from a well-shaped mesh, are in fact optimal.

In essence, what we have shown in this paper is that jump-and-walk strategy for point location in well-shaped meshes in $\mathbb{R}^2$ and $\mathbb{R}^3$, is practical, simple, and efficient, and requires only the knowledge of an approximate nearest neighbor. It will be worthwhile to explore other geometric configurations where the jump-and-walk can lead to efficient ways to perform point location queries.

Acknowledgements: We thank the referees for their helpful comments.

References

[1] M. de Berg, M. J. van Kreveld, M. H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer.

[2] L. Devroye, C. Lemaire, and J.-M. Moreau. Expected time analysis for Delaunay point location. *Computational Geometry: Theory and Applications*, 29(2):61–89, 2004.

[3] C. Dillabaugh. I/O efficient path traversal in well-shaped tetrahedral meshes. In *CCCG*, pages 121–124, 2010.

[4] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.

[5] A. Liu and B. Joe. Relationship between tetrahedron shape measures. *BIT Numerical Mathematics*, 34:268–287, 1994.

[6] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Geometric separators for finite-element meshes. *SIAM J. Sci. Comput*, 19(2):364–386, 1998.

[7] D. Mount and S. Ayra. ANN: A library for approximate nearest neighbor searching. `http://www.cs.umd.edu/~mount/ANN/`, Jan. 2010.

[8] E. P. Mücke, I. Saias, and B. Zhu. Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations. *Computational Geometry: Theory and Applications*, 12(1-2):63–83, 1999.

[9] J. T. Robinson. The k-d-b-tree: A search structure for large multidimensional dynamic indexes. In *SIGMOD*, pages 10–18, 1981.

[10] S.-H. Teng. Fast nested dissection for finite element meshes. *SIAM Journal on Matrix Analysis and Applications*, 18(3):552–565, 1997.