Convex Hull for Probabilistic Points

F. Betul Atalay
Dept. of Computer Engineering
TOBB University of Economics and Technology
Sogutozu, Ankara, Turkey
fatalay@etu.edu.tr

Sorelle A. Friedler
Dept. of Computer Science
Haverford College
370 W. Lancaster Avenue
Haverford, PA 19041, USA
sorelle@cs.haverford.edu

Dianna Xu
Dept. of Computer Science
Bryn Mawr College
101 N. Merion Avenue
Bryn Mawr, PA 19010, USA
dxu@cs.brynmawr.edu

***Abstract*—We analyze the correctness of an $O(n \log n)$ time divide-and-conquer algorithm for the convex hull problem when each input point is a location determined by a normal distribution. We show that the algorithm finds the convex hull of such probabilistic points to precision within some expected correctness determined by a user-given confidence value ϕ. In order to precisely explain how correct the resulting structure is, we introduce a new certificate error model for calculating and understanding approximate geometric error based on the fundamental properties of a geometric structure. We show that this new error model implies correctness under a robust statistical error model, in which each point lies within the hull with probability at least ϕ, for the convex hull problem.**

I. Introduction

The *Convex Hull Problem* is the problem of determining a minimum convex bounding polygon that covers n points in the Euclidean plane.

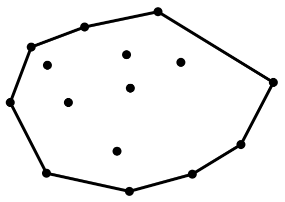
Fig. 1. A point set and its convex hull

This is a classic problem in computational geometry, with well known solutions including Graham's scan and divide-and-conquer (both take $O(n \log n)$ time) [8], [15]. The convex hull is a fundamental primitive for many graphics problems, such as calculation of basic shape representations (e.g., bounding boxes) [17] and collision detection [13]. In application domains, point locations are often the result of a machine learning algorithm that outputs a probability distribution for each point's location (for a survey, see [11]). For example, in augmented reality, the markerless tracking problem that aims to track the position and orientation of a camera in a scene without using markers may take a hybrid approach that relies on both computer vision techniques and probabilistic GPS location information of the type generated by such machine learning algorithms [18]. In this paper, we are interested in examining what happens when the expected values of such probabilistic points are given as input to the divide-and-conquer convex hull algorithm, with the goal of guaranteeing approximate correctness of the resulting convex hull without requiring extensive modification to existing algorithms.

We will show that the divide-and-conquer convex hull algorithm still produces an approximately correct convex hull even when its input point locations aren't known exactly. This will require some modifications to the algorithm as well as an introduction of a new error model in order to define what we mean by an approximately correct convex hull. We will build this new approximate notion on boolean functions that certify geometric properties necessary to a correct calculation of the convex hull. These functions are borrowed from the study of kinetic data structures [5], and so some of this work will find application to other problems studied within that boolean certification framework (as well as allowing future work to extend these results to hold on moving points). A careful analysis will show how potential errors in these certifications propagate to the overall structure being calculated. The convex hull will be approximate in the sense that only a given percent of the points will be expected to lie within it. This matches the desires of some applications - for example, when determining the home range of an animal from (noisy) location observations, the goal is to compute a boundary containing some percentage of such observations [6].

A. Related Work

Approximate correctness of a geometric structure has been considered under a number of different models, including interpretations where the structure is considered to be fully correct some percentage of time or where it is considered to be partially correct every time the algorithm is run. We are most interested in this second interpretation, within which partial correctness has been considered under the absolute error model [7], the relative error model [3], and the robust error model [16]. Within the *absolute error model* a structure is considered to be correct up to some given fixed error bound ε that is constant for any set of points [7]. Under the *relative error model* a structure is considered to be correct up to some percentage based on the geometric structure [3]. The *robust error model* is a per-point error model under which a structure is correct based on the percentage of points which are correct [16]. We will compare the error model we introduce to the robust error model.

While classical computational geometry assumes exact knowledge of point location, goals of relaxing such assumptions have spurred several recent papers. Loeffler and Kreveld [14] have considered approximate convex hulls under an imprecise point setting, where exact point location is unknown within a region but guaranteed not to be outside of it. They consider the convex hull under multiple variants of the relative error model and achieve running times that range from $O(n \log n)$ to $O(n^{13})$. When considering approximate nearest neighbor searching, a model where points are described as probability distributions over their possible locations has also been considered [1]. This latter model of point location, commonly used in application domains, is the same as the one we use here (and is described in more detail in Section II-A). The convex hull problem has been considered within the discrete version of this point location model (where the distributions are discrete) by Agarwal *et al.* [2]. Their results give a running time of $O(m \log^3 m)$, where m is the number of possible point locations in their discrete distributions. The robust error model in Argarwal *et al* implies the one we compute. While we solve a weaker version of the problem, we improve the running time to $O(n \log n)$, where n is the number of points. Additionally, ours is the first solution to hold on continuous distributions.

We use an $O(n \log n)$ divide-and-conquer algorithm to compute the convex hull on a set of probabilistic points under normal distributions. Our solution is approximately correct under a robust error model with the correctness taken in expectation over all possible point locations, so that each point has at least ϕ probability of being in the hull, for a parameter ϕ. Ours is the first solution to hold for probabilistic points with a continuous location probability distribution.

We achieve these results not by introducing a new algorithm, but by introducing a new error model and associated analysis of the standard divide-and-conquer algorithm for calculating the convex hull via its upper envelope in the dual space [15]. We introduce a *certificate error model* in which a structure is considered ϕ-correct if each Boolean certificate used to calculate the structure is correct with probability at least ϕ. We will show that approximate correctness under the certificate error model implies approximate correctness under the robust error model for the convex hull.

B. Contributions

The rest of this paper shows the following results:

1) We introduce a certificate error model guaranteeing that each certificate is correct with probability ϕ, and a proof that this new error model implies the robust error model for the convex hull problem. (See Sections IV and V.)
2) We adapt an $O(n \log n)$ algorithm to compute the convex hull for probabilistic points. We show that this algorithm is approximately correct in expectation over all possible point locations, under a robust error model guaranteeing that each point is within the hull with probability at least ϕ. (See Section III.)

II. Preliminaries

A. Probabilistic Points

We define a *probabilistic point* $p_j = (N_j, v_j)$ where N_j is a normal probability distribution over its possible locations

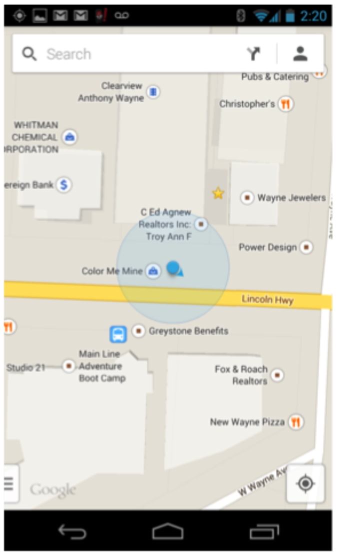

Fig. 2. A Google Maps screenshot showing a probabilistic point p_j under a normal distribution D_j where the central blue dot is v_j and the lighter blue circle is its associated $\beta_j(\phi, D_j)$.

and $v_j \in \mathbb{R}^d$ is an *expected value* for the point p_j given distribution N_j. We are given a set P of n probabilistic points. $D_j = \{x \in \mathbb{R}^d | N_j^{pdf}(x) > 0\}$ is the positive region of the probability density function $N_j^{pdf} : \mathbb{R}^d \rightarrow \{y \in \mathbb{R} | y \geq 0\}$. We assume that the region D_j is bounded. Let $\beta_j(\phi, D_j) \subset \mathbb{R}^d$ be the *boundary region* of point p_j defined as the minimum-area convex set such that p_j is within the region with probability ϕ, i.e., $\int_{x \in \beta_j(\phi, D_j)} N_j^{pdf}(x)dx = \phi$. $\phi \in [0,1]$ is a user-given confidence value, and $\Phi = 100 \cdot \phi$ is ϕ in percent form. We assume that $\beta_j(\phi, D_j)$ can be calculated in $O(1)$ time. For example, Figure 2 shows $\beta_j(\phi, D_j)$ as the truncated Gaussian.

For the remainder of this paper we will refer to these probabilistic points $p_j = (D_j, v_j)$ simply as *points*. Within a machine learning context, these points would be generated by a *model* $M(j, E) \rightarrow p_j$ that, when given a point identity j and environmental data E, would return the probabilistic point p_j. More details about such models can be found in a survey of location models [11]. We will assume that this model is good enough that the point locations can generally be distinguished from each other, i.e., that for $p_i, p_j \in P$, drawn from the distribution created by the model,

$$Pr[\beta_i(\phi, D_i) \cap \beta_j(\phi, D_j) = \emptyset] \geq \phi \ .$$

B. Certificates

Given a set of probabilistic points, we develop a framework that approximately maintains a geometric structure G up to some expected correctness. We define a set of certificates C that guarantee local geometric relationships crucial to the correctness of the entire structure. For example, a single certificate might guarantee that three points are oriented in a counter-clockwise relationship. C can be considered a *proof* of the correctness of G. These certificates are the same as those maintained in classic kinetic data structure (KDS) settings [5] (we will extend them later). The set C consists of pairs containing a Boolean function c which operates on a set of points $P_i \subset P$ and the set of points P_i on which that function evaluates to `True`. Such a pair (c, P_i) is called a *certificate*. Within a single set C, there can be multiple types of such functions c, certifying different geometric properties. For notational ease we will abuse notation below and refer to all such functions as c. A set of certificates C must satisfy the following local geometric properties as given in [5].

Property II.1 (Locality). *For all points $p_j \in P$, $|\{P_i \mid p_j \in P_i \text{ and } (c, P_i) \in C\}|$ is $O(\mathtt{polylog}(n))$ or $O(n^\epsilon)$ for arbitrarily small values of ϵ.*

Property II.2 (Compactness). *$|C|$ is $O(n\ \mathtt{polylog}(n))$ or $O(n^{1+\epsilon})$ for arbitrarily small ϵ.*

Property II.3 (Exclusivity). *$|P_i| \leq k$ for all $(c, P_i) \in C$, $P_i \subset P$, and small constant k.*

Locality and compactness are both required within the KDS framework and exclusivity is also generally assumed [5]. Thus, we can draw on a large body of existing work defining certificates for a wide variety of problems. (See [10]). Notably, these certificates certify the steps of certain locally constrained algorithms and incrementally constructed problem solutions. Divide and conquer algorithms often make good candidates for such problem certification mechanisms; Each decision in the merge process constitutes a certificate.

We add to the KDS understanding of certificates to take into account the probabilistic nature of the points.

Definition II.4 (ϕ-correct certificate). *A certificate (c, P_i) for which*

$$Pr[c(P_i) = \mathtt{True}] \geq \phi$$

with the probability taken over the distribution of possible point locations for points $p_j = (N_j, v_j)$ for $p_j \in P_i$.

For example, a simple certificate $(above_\phi, P_i)$ with $P_i = \{p_1, p_2\}$ certifies that p_1 is above p_2 with probability at least ϕ. (See Section IV for a more extensive example of a problem using such certificates.) It will be useful to note that $v_j \in \beta_j(\phi, D_j)$ for all $p_j \in P_i$ since N_j is a normal distribution. If all certificates are ϕ-correct for $\phi = 1$, then the geometric structure G has been correctly calculated. The main motivation of this paper is to consider the correctness for values of $\phi < 1$.

Given knowledge of $\beta(P_i) = \{\beta_j(\phi, D_j) | p_j \in P_i\}$, we now determine the correctness of certificate (c, P_i). ϕ^k-correctness can be achieved by creating certificates (c', P_i) with new function c' such that $c'(P_i) = \mathtt{True}$ if and only if for all possible point locations $P_i = \{_{j=1}^{k}\ p_j \in \beta_j(\phi, D_j) \mid \beta_j(\phi, D_j) \in \beta(P_i)\}$ we have $c'(P_i) = \mathtt{True}$. This can be easily improved to ϕ-correctness by determining $\beta_j(\phi^{1/k}, D_j)$ instead. However, this is a conservative lower bound on the correctness of the certificate. In the example certificate $above_\phi(P_i)$, we would be guaranteeing that p_1 is above p_2 and that $\beta_1(\phi, D_1)$ does not intersect $\beta_2(\phi, D_2)$. Instead, we could calculate directly the probability that p_1 is above p_2 and set $above'_\phi(P_i) = \mathtt{True}$ as long as that probability is at least ϕ. This guarantees that $above'_\phi(P_i)$ is ϕ-correct.

III. Convex Hull Algorithm

Recall that the convex hull is defined as the smallest convex region containing a set of points. In order to determine certificates that guarantee a solution to this problem, we turn to the KDS definition of convex hull certificates [5] that we will review in this section. The KDS solution for this problem makes use of a divide and conquer algorithm to find the convex hull via finding the upper and lower envelopes in the dual setting, where a point (a, b) is represented by the line $y = ax + b$ [1] [5], [15].

Given a set of n lines $L = \{l_1, l_2, \ldots, l_n\}$ where l_i is of the form $y = a_i x + b_i$, if we think of these lines as defining n halfplanes, $y \geq a_i x + b_i$, each lying *above* one of the lines, then the *upper envelope* of L is the boundary of the intersection of these half planes (see Figure 3). The *lower envelope* is defined symmetrically.

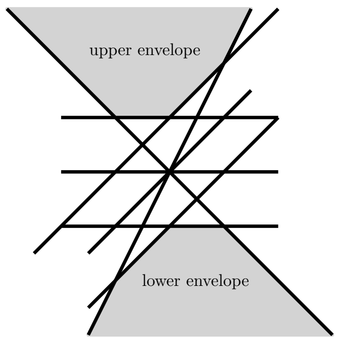

Fig. 3. Upper and lower envelopes

Classic computational geometry has a well-established equivalency of the convex hull of points and the upper/lower envelopes of a collection of lines under the point-line duality transformation [5], [15], in that the clockwise order of the points along the upper (lower) convex hull of a set of points P is equal to the left-to-right order of the sequence of the lines on the upper (lower) envelope of the dual P^* (see Figure 4).

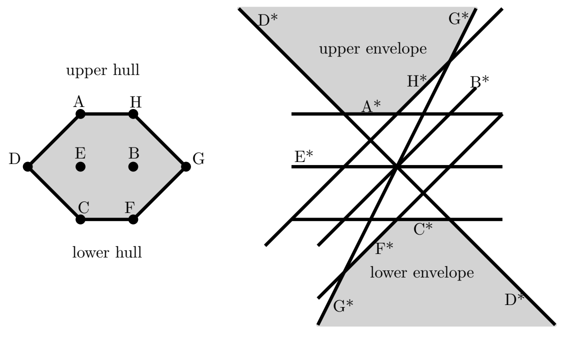

Fig. 4. Equivalence of convex hulls and envelopes

[1]Note that standard notation dualizes (a, b) to $y = ax - b$, however, KDS [5] uses $ax + b$, which we follow to avoid confusion when discussing certificates.

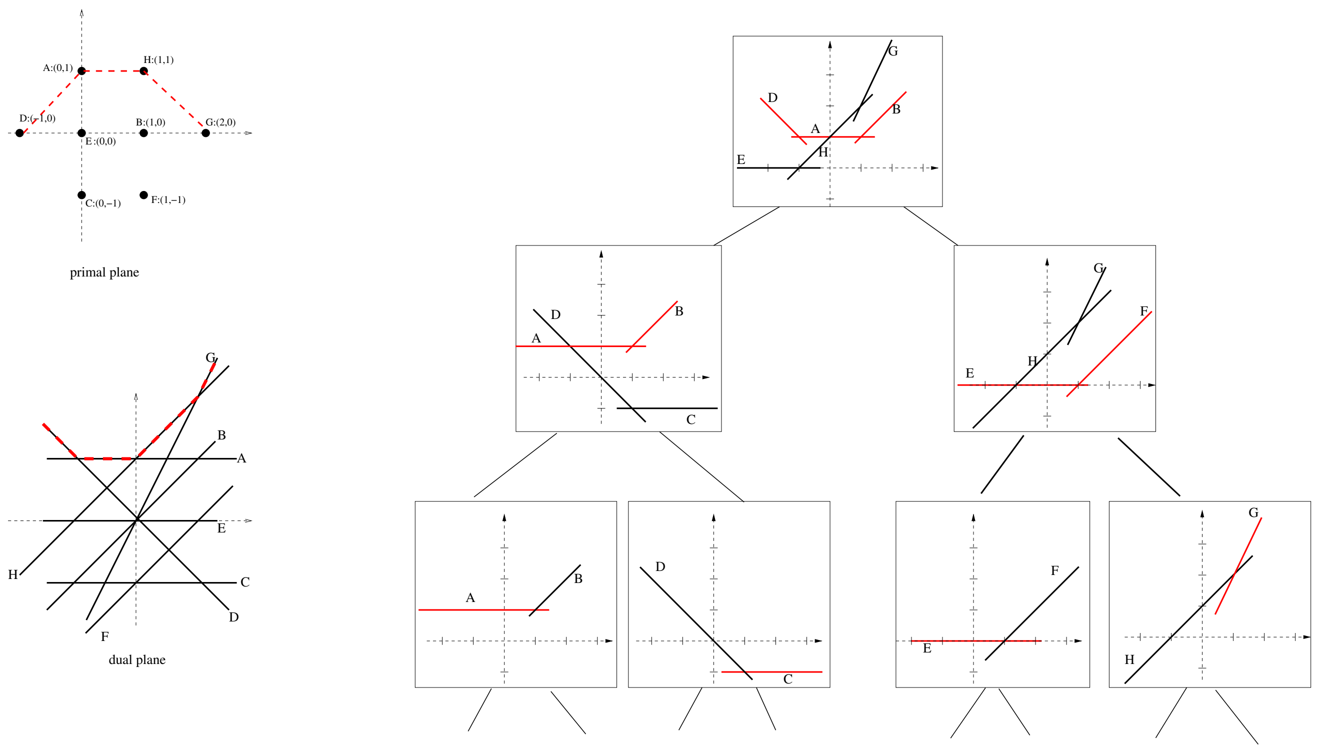

Fig. 5. Top-left: points in primal plane. Bottom-left: dual plane, where a point (a, b) is represented by the line $y = ax + b$. Right: the merge tree corresponding to the upper envelope computation in dual space. Leaf nodes are single lines and are omitted in the figure. The certificates proving the top-most merge are as follows: (i) a chain intersection certificate guaranteeing that $EH <_x AB$ and $EH <_y A$ and $AB <_y H$, that is, EH is to the left of AB, EH is below line A, AB is below line H (ii) a diverging certificate guaranteeing that $B \leq_s G$ and $AB <_y G$, that is, B's slope is less than or equal to G's slope and vertex AB is below line G, (iii) and, another diverging certificate guaranteeing that $E \leq_s D$ and $EH <_y D$. The certificates are explained in more detail in the paper introducing them [5].

The KDS algorithm computes the convex hull in the dual because it allows easier management of certificates. Since our analysis will rely on certificates, we will make use of this algorithm as well. Please refer to Figure 5 as we describe it briefly below.

We focus only on the upper envelope of dual lines: the lower envelope computation is symmetric. To find the upper envelope of a set of lines, the lines are partitioned into two subsets of roughly equal size, their upper envelopes are computed recursively, and then the two resulting upper envelopes—let them be called red and black— are merged. The merge step is performed by sweeping a vertical line through all of the vertices of the red and black upper envelopes, from left to right. As the line sweeps, the most recently encountered red and black vertices are maintained along with the information whether the red or the black chain is above. As the sweep encounters the next red (black) vertex, the algorithm determines if it is above or below the corresponding black (red) edge. If it is above, the current vertex is added to the merged envelope. If the above/below ordering of the envelopes has changed, that means the red and black envelopes have crossed, and the intersection point is also added to the merged upper envelope. This algorithm takes time $O(n \log n)$ [15]. We use the same algorithm and return the same points as the hull, so our algorithm also takes time $O(n \log n)$. However, our certificates are ϕ-correct, so we will need to more carefully consider the correctness of the resulting hull (Section V). First, we will describe these certificates in more detail.

As originally presented in [5], the comparisons done during the merge step lead to the following certificates: (i) *x-certificates* ($<_x$) are used to certify the x-ordering of the red and black vertices, (ii) *y-certificates* ($<_y$) are used to certify the y-ordering of a vertex with respect to an edge of the opposite color. (iii) *slope-certificates* ($\leq_s$) involve comparisons between line slopes. Slope-certificates are not required in the above sweep algorithm, however they are needed in order to make the KDS local—avoiding linearly many y-certificates per edge.

In a *merge tree*, we keep track of all levels of the divide and conquer algorithm's merge step and certify the properties that determine each choice in the merge of two recursively determined upper envelope chains. Recall from [5] and [4] that the leaf nodes of the merge tree are single lines and the root node contains all line segments determining the resulting upper envelope. The certificates, as presented in [5], come in dependent groups. That is, the invalidation of any one certificate invalidates all other certificates in that group, and the existence of any one certificate implies the existence of all other certificates in the group. These certificates are *tangent certificates*, *chain intersection certificates*, and *diverging certificates* (see Figure 6). Diverging and tangent

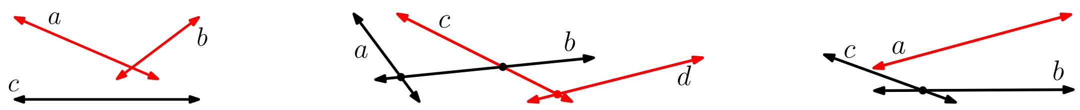

Fig. 6. Convex hull certificates [5] in the dual setting showing the two chains involved in red and black. Left: Tangent certificates guarantee that line c is below the vertex ab and that the slope of line c is greater than the slope of a and less than the slope of b. Center: Intersection certificates guarantee that the vertex ab is to the left of vertex cd and below line c and that vertex cd is to the right of vertex ab and below line b. Right: Diverging certificates guarantee that b's slope is less than or equal to the slope of line a and that the vertex cb is below line a.

certificates represent the two ways in which an edge will not be involved in the resulting upper envelope while intersection certificates represent a merge point of the two chains. For more precise details defining these certificates see [5]. One assumption implicit in the presentation of these certificates and in the statement of the original divide and conquer algorithm is that the points are in general position, i.e., no two lines in the dual setting are parallel. We make the same assumption. See Figure 5 for an example involving eight points (shown both in primal space and the dual space), and the merge tree corresponding to the upper envelope computation.

Convex Hull Algorithm: Using these certificates, we now have the following algorithm. Create the certificates based on the expected values of the points. We will show in Lemma IV.1 this gives a set of ϕ-correct certificates. Find the convex hull using these certificates and the $O(n \log n)$ divide and conquer algorithm [5]. Then find the boundary of the convex hull of those points. Specifically, given points H determined by the certification process as the hull, we report $\mathcal{CH}(\{\beta_i(\phi, D_i) \mid p_i \in H\})$ where $\mathcal{CH}$ takes the convex hull of the convex bounding regions of the points on the hull (e.g., using the algorithm of [9] which takes time $O(n \log n)$). The resulting algorithm takes time $O(n \log n)$.

IV. Certificate Error Model

In order to reason about the correctness of this convex hull algorithm when expected values are used instead of precise points, we introduce a new error model that evaluates the correctness of a geometric structure based on the correctness of its component certificates.

Definition IV.1 (ϕ-correct within the certificate error model)**.** *Given a set of certificates C guaranteeing a geometric structure, for all $(c, P_i) \in C$, (c, P_i) is ϕ-correct.*

When we originally construct the set of certificates for a problem we will do so based on the set of expected values of the points, $V(P_i) = \{v_j | p_j \in P_i\}$. The question then becomes what is the relationship between certificates involving the expected values and those involving the set of points at their full distribution of locations. Certificates that are 1-correct based on the expected values are easily achieved by making the direct comparisons based on the known expected values, the question is how these relate to the certificates involving the points. Recall that $v_i \in \beta_i(\phi, D_i)$ for normal distributions.

Lemma IV.1. *Given that*

$$Pr[\beta_j(\phi, D_j) \cap \beta_k(\phi, D_k) = \emptyset] \geq \phi \textit{ for all } p_j, p_k \in P_i$$

where $P_i \subset P$, then

$$\big(Pr[c(V(P_i)) = \texttt{True}] = 1\big) \implies \big(Pr[c(P_i) = \texttt{True}] \geq \phi\big) .$$

Proof: Consider the cases when $c(P_i) = \texttt{False}$: either $\beta_j(\phi, D_j) \cap \beta_k(\phi, D_k) \neq \emptyset$ or at least one of the true locations p'_j associated with $p_j \in P_i$ is outside of $\beta_j(\phi, D_j)$. Both of these cases occur with probability at most $1 - \phi$, so the certificate probability guarantee of ϕ has been verified. ■

With this lemma, we have the following basic procedure for translating any existing set of certificates into a ϕ-correct set:

1) Create a set of 1-correct certificates based on the expected values of the points. These certificates are a set of ϕ-correct certificates involving points, by Lemma IV.1.
2) Solve the problem using these certificates as usual.
3) Return a worst case result based on the boundaries of the points in the solution.

Step 3 will be explained in more detail for each specific problem solution.

Here, we begin with an example problem that demonstrates the value and structure of the certificate error model without the complexity of the convex hull certificates that will be described in Section III.

A. 1D Maximum Problem

The *1D Maximum Problem* determines the maximum point among a set of n one-dimensional points. We will assume that these points are in general position. We will use the same certificates as those in the KDS solution to this problem, which relies on a max heap that maintains the full ordering of the points. The $n - 1$ certificates guarantee the parent-child relationships in the heap [5].

Following the general procedure outlined above, parent-child certificates are created based on their expected values. We will say that $above_\phi(p, q)$ certifies that $Pr[p > q] \geq \phi$. The original certificates created are $above_1(v_p, v_q)$. By Lemma IV.1, this gives us certificates $above_\phi(p, q)$. Since $v_p \in \beta_p(\phi, D_p)$ and $v_q \in \beta_q(\phi, D_q)$, we also have that $above_\phi(p, q) \implies \big(above_\phi(p, v_q) \wedge above_\phi(v_p, q)\big)$. We find the point m that is the maximum based on these certificates and report the value $\texttt{max} = max\{p \in \beta_m(\phi, D_m)\}$ as the result of the 1D maximum problem.

For example, consider the max heap shown in Figure 7 maintaining probabilistic points m, p, q, r and s. There are four certificates: $above_\phi(m, p)$, $above_\phi(p, q)$, $above_\phi(p, r)$ and $above_\phi(m, s)$ associated with this heap. Recall that $above_\phi(p, q)$ certifies that $p > q$ with probability at least ϕ. The expected values v_p at which these points are believed to be are shown within each node. In the case where the boundary

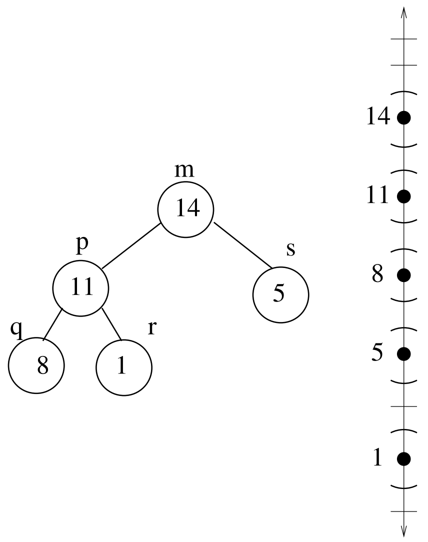

Fig. 7. Max-heap for the 1D Maximum problem.

extends 1 unit in each direction centered at v_m, the maximum returned is $15 = max\{\beta_m(\phi, D_\phi)\}$.

However, the true value of a point may differ from what is believed, which may cause a certificate to be incorrect. For example, if point q has a true value of 16, it makes the certificate $above_\phi(p, q)$ false. Moreover, as in this case, an incorrect certificate may mean that q's real value (16) is higher than the maximum given by the heap. Our goal is to show that this bad case does not happen too often. We will use the assumption that the points' boundary regions may not intersect with high probability.

Specifically, we will show that ϕ-correctness under the certificate error model for the 1D maximum problem implies correctness under a previously studied error model, the robust error model. Under the *robust error model*, more commonly known as a robust statistical estimator, a structure is robust to outliers up to some breakdown point [16]. Defining the error model more specifically is problem-dependent.

Definition IV.2 (1D Maximum Problem: ϕ-correct within the robust error model)**.** *For a point set P with n points, the returned maximum point* `max` *is such that*

$$|\{p \in P \mid p \leq max\}| \geq \phi \cdot n \ .$$

Since we are working with probabilistic points, we are interested in an *expected* ϕ-correct robust error model.

Definition IV.3 (1D Maximum Problem: expected ϕ-correct within the robust error model)**.** *For a point set P with n points, the returned maximum point* `max` *is such that*

$$\mathbb{E}\left[\ |\{p \in P \mid p \leq max\}|\ \right] \geq \phi \cdot n \ .$$

If, for all points $p \in P$, $Pr[p \leq \texttt{max}] \geq \phi$ then by linearity of expectations, expected ϕ-correctness is implied, with the expectation taken over the point location distributions. We will thus proceed with the expected version of the definition.

Theorem IV.4 (1D Maximum Problem: Certificate error model implies robust error model)**.** *Given a max heap with certificate set C, points P, and returned maximum point* `max`*, if for all $(c, P_i) \in C$, (c, P_i) is ϕ-correct then* `max` *is expected ϕ-correct within the robust error model.*

Proof: Since the heap is ϕ-correct under the certificate error model, we expect $100 - \Phi$ percent of the certificates to be incorrect. There are $n - 1$ certificates in the heap. We will examine how many points can be greater than the maximum for each incorrect certificate. If a certificate $above_\phi(p, q)$ is incorrect then $p \notin \beta_p$ or $q \notin \beta_q$. We will associate each point outside of its boundary with the certificate it participates in as a child. Recall from Definition II.4 that the location of q' being drastically different from v_q does not invalidate the child certificates of q - those were still made with respect to $q = (v_q, D_q)$. Thus, each incorrect certificate (associated with some point that is outside of its boundary) will cause at most its child point to be above the maximum. The only remaining point to consider is the maximum point, which doesn't participate in any certificate as a child. By the definition of $\texttt{max} = max\{p \in \beta_m(\phi, D_m)\}$, the true value of m is greater than `max` with probability at most $1 - \phi$.

So n points will be above the maximum reported value each with probability at most $1 - \phi$. Thus, we expect that the reported maximum point will be within the top $100 - \Phi$ percent of the points and so the result is expected to be ϕ-correct under the robust error model. ■

V. Convex Hull Error Model Correctness

We will consider correctness of the hull under the robust error model which states that the convex hull is ϕ-correct if at least Φ percent of the points are contained within the hull.

Definition V.1 (Convex Hull: expected ϕ-correct within the robust error model)**.** *Given a set of n points P and its set of points H on the convex hull, where $\mathcal{H}$ is the associated hull region,*

$$\mathbb{E}\left[|\{p \in P \mid p \in \mathcal{H}\}|\right] \geq \phi \cdot n \ .$$

This is a restatement of an idea Tukey referred to as "peeling" in which some number or percentage of outlying points are iteratively deleted and the convex hull is computed on the remaining points [12]. A conservative interpretation of this definition states the goal as computing the minimum convex area containing Φ percent of the points. Given the probabilistic nature of our points, we will instead compute the minimum convex area containing the entirety of the boundary (the truncated distribution of point locations) for the points on the hull based on their expected values. We will show that this convex hull satisfies the weaker Definition V.1 above.

The certificate error model remains the same: a convex hull is ϕ-correct if each certificate is ϕ-correct.

Theorem V.2 (Convex Hull: Certificate error model implies robust error model)**.** *Given a set of certificates C certifying* *a convex hull solution for points P, with returned set of probabilistic points H that make up the hull, where $\mathcal{H}$ is the associated hull region,*

$$\forall_{(c,P_i)\in C} \ , (c, P_i) \ \textit{is} \ \phi - \textit{correct} \longrightarrow$$

$$\mathbb{E}\left[|\{p \in P \mid p \in \mathcal{H}\}|\right] \geq \phi \cdot n \ .$$

In order to reason about the certificate correctness of the convex hull, we will first need to understand the properties of these certificates in more detail. We will say that a point p has been *excluded* from the convex hull by some certificate when believing that certificate's false assertion causes p to be outside of the resulting convex hull. See Figure 8 for an example where point D is excluded from the convex hull due to an incorrect tangent certificate. Note that if a certificate incorrectly causes p to be *on* the convex hull, p has not been excluded.

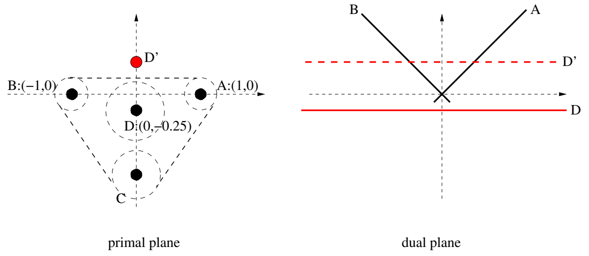

Fig. 8. Left: probabilistic points in primal plane. Right: Consider the merge of the black envelope consisting of lines A and B, and the red envelope consisting of D only. This merge is certified by a tangent certificate guaranteeing that line D is below vertex AB ($D <_y AB$), the slope of line D is greater than the slope of B ($B <_s D$), and less than the slope of A ($D <_s A$). If point D's real position is D', notice that the line corresponding to D' in dual space is above point AB, and this makes the tangent certificate incorrect. Point D is excluded from the convex hull by this incorrect tangent certificate.

Recall the tree of certificates that show the choices in the divide and conquer convex hull algorithm, which we called the merge tree (an example was shown in Figure 5). Here, we consider the levels of the merge tree in which a single point (a line in the dual setting) participates.

Lemma V.1. *Each point $p \in P$ with associated dual line ℓ can only be excluded from the convex hull by an incorrect certificate in the highest level L of the merge tree in which ℓ participates.*

Proof: Suppose there is an incorrect certificate that involves ℓ in some level L' below level L. Despite this incorrect certificate, ℓ advanced to level $L' + 1 \leq L$, so ℓ was found to be on the upper envelope for all lines in its subtree at level L'. Points reported as on the convex hull (lines in the upper envelope in the dual setting) have not been excluded from the hull. So as long as ℓ advances to some level above L', incorrect certificates in level L' can not exclude ℓ. So ℓ can only be excluded from the convex hull by an incorrect certificate that keeps it from being reported on the convex hull. This level is, by construction of the merge tree, the highest level at which ℓ participates in the merge tree. ■

Next, we examine the properties of certificates from a single line's point of view.

Lemma V.2. *Each point $p \in P$ with associated dual line ℓ can be excluded from the convex hull by at most one certificate when considering a single level L of the merge tree.*

Proof: First, we consider which lines have the potential to be excluded from the hull for each type of certificate. Referring to the line labels of Figure 6, note that only c may be excluded by an incorrect tangent certificate, a or d may be excluded due to an incorrect intersection certificate, and only b may be excluded by an incorrect diverging certificate, since these are the lines not believed to be on the hull. Note also that some of the lines in these certificates (a and d in the intersection certificates and c in the diverging certificates) only contribute a vertex to the geometric relationship being guaranteed between the lines. We will say that such lines *participate as a vertex* in the certificate, while the other lines will be said to *participate as a line*.

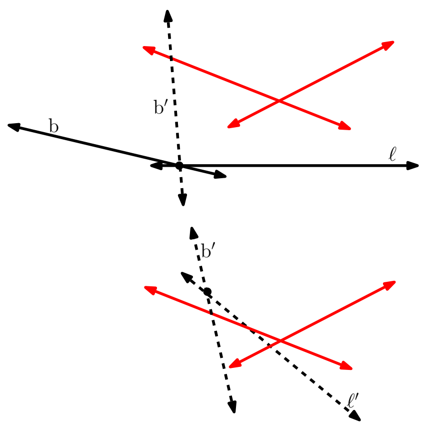

Fig. 9. The solid lines show the expected positions of ℓ and b while the dashed lines show their possible true positions. ℓ participates as a line in a tangent certificate with the red upper envelope and as a vertex in a diverging certificate. There is no way for ℓ to be on the upper envelope without violating the tangent certificate it participates in. In the top figure, the diverging certificate fails, but ℓ is not on the upper envelope, while in the bottom figure, both certificates fail and ℓ is on the resulting upper envelope.

Suppose that ℓ takes the role of c in the tangent certificate (see Figure 9): we will consider what other certificates ℓ could participate in. Since ℓ already participates in one certificate as a line, it can not participate in any others as a line. This is because ℓ can only have one type of slope relationship with the chain that is above it - either tangent, intersecting, or diverging. That leaves participation as a vertex. Depending on the number of lines participating in the merge at this level, ℓ may have zero, one, or two endpoints contributing to its upper envelope segment. At any existing endpoint, ℓ could participate as a vertex in a diverging certificate or an intersection certificate, but may participate in at most one certificate per vertex. For either a failing diverging or intersection certificate to exclude ℓ, the vertex of ℓ that is participating in the certificate must be above the line from the other (red) chain. But if one of the endpoints of ℓ is above the other chain, then the tangent certificate is incorrect. So for ℓ to be excluded, the tangent certificate it participates in must be incorrect.

A similar analysis when ℓ takes the role of b in the diverging certificate argues that for ℓ to be excluded in that case, the diverging certificate it participates in as b must be incorrect. Finally, remember that if ℓ participates as a line in an intersection certificate it can not be excluded from the hull as it is already found to be on the hull (for this level). We have considered all possible cases when ℓ participates as a line and in each of them there is a single certificate that must be wrong for ℓ to be excluded from the hull and no other certificate may exclude ℓ from the hull on its own. ■

Now we can put these lemmas together for the proof of Theorem V.2.

Proof: By Lemmas V.1 and V.2 we know that each point can be excluded from the hull only by an incorrect certificate in its highest level of the merge tree and that each point has at most one certificate per level that can exclude it from the hull, so each point has at most one certificate in the whole merge tree that can exclude it. If a convex hull is ϕ-correct under the certificate error model, then each certificate has $1-\phi$ probability of being incorrect and thus each point not on the hull has at most $1-\phi$ probability of being excluded. By linearity of expectation, the expected number of points outside the reported hull is at most $(1-\phi)n$. So the convex hull is expected to be ϕ-correct under the robust error model. ■

VI. Conclusion

In this paper, we introduced a new way of understanding and quantifying approximate geometric correctness via our certificate error model. Accompanying this, we showed how this error model could be applied to probabilistic points of the type generated by machine learning models in the context of two problems - the 1D maximum and the convex hull problems. We gave an $O(n \log n)$ time algorithm for the convex hull on probabilistic points, with approximate correctness guaranteed under the robust error model and under this certificate error model. The strength of the certificate error model lies in its generalizability to any problem that can be stated in terms of its component Boolean properties. These results could also be generalized to probabilistic points in a motion setting, since certificates are also the basis for kinetic data structure (KDS) results.

References

[1] Pankaj K. Agarwal, Boris Aronov, Sariel Har-Peled, Jeff M. Phillips, Ke Yi, and Wuzhou Zhang. Nearest neighbor searching under uncertainty II. In *PODS*, 2013.

[2] Pankaj K. Agarwal, Sariel Har-Peled, Subhash Suri, Hakan Yldz, and Wuzhou Zhang. Convex hulls under uncertainty. In *Proc. of the 22nd Annual European Symposium (ESA)*, pages 37–48, 2014.

[3] Sunil Arya and David M. Mount. Approximate range searching. *Computational Geometry: Theory and Applications*, 17(3-4):135–152, 2000.

[4] Mikhail J. Atallah. Some dynamic computational geometry problems. In *Computers & Mathematics with Applications*, volume 11(12), pages 1171–1181, 1985.

[5] Julien Basch and Leonidas J. Guibas. Data structures for mobile data. *J. of Algorithms*, 31(1):1–28, April 1999.

[6] S. Benhamou and D. Cornélis. Incorporating movement behavior and barriers to improve kernel home range space use estimates. *The Journal of Wildlife Management*, 74:1353–1360, 2010.

[7] Guilherme D. da Fonseca and David M Mount. Approximate range searching: The absolute model. *Computational Geometry: Theory and Applications*, 43(4):434–444, 2010.

[8] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, 2008.

[9] Olivier Devillers and Mordecai Golin. Incremental algorithms for finding the convex hulls of circles and the lower envelopes of parabolas. *Information Processing Letters*, 56(3):157 – 164, 1995.

[10] L. Guibas. Kinetic data structures. In D. Mehta and S. Sahni, editors, *Handbook of Data Structures and Applications*, pages 23–1–23–18. Chapman and Hall/CRC, 2004.

[11] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[12] Peter J. Huber. The 1972 wald lecture robust statistics: A review. *The Annals of Mathematical Statistics*, 43:1041–1067, 1972.

[13] Pablo Jiménez, Federico Thomas, and Carme Torras. 3d collision detection: a survey. *Computers & Graphic*, 25(2):269–285, 2001.

[14] Maarten Löffler and Marc van Kreveld. Largest and smallest convex hulls for imprecise points. *Algorithmica*, 56:235–269, 2010.

[15] Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1998.

[16] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. Wiley & Sons, NY, 2005.

[17] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision.* Cengage Learning, 2014.

[18] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202, 2008.