# A Computational Geometry Method for DTOA Triangulation

Nageswara S. V. Rao
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831
Email: raons@ornl.gov

Xiaochun Xu and Sartaj Sahni
Computer and Information Science and Engineering Department
University of Florida
Gainesville, FL 32611
Email: {xxu,sahni}@cise.ufl.edu

***Abstract*— We present a computational geometry method for the problem of triangulation in the plane using measurements of distance-differences. Compared to existing solutions to this well-studied problem, this method is: (a) computationally more efficient and adaptive in that its precision can be controlled as a function of the number of computational operations, making it suitable to low power devices, and (b) robust with respect to measurement and computational errors, and is not susceptible to numerical instabilities typical of existing linear algebraic or quadratic methods. This method employs a binary search on a distance-difference curve in the plane using a second distance-difference as the objective function. We establish the unimodality of the directional derivative of the objective function within each of a small number of suitably decomposed regions of the plane to support the binary search. The computational complexity of this method is $O(\log^2 1/\gamma)$, where the computed solution is guaranteed to be within a $\gamma$-precision region centered at the actual solution. We present simulation results to compare this method with existing DTOA triangulation methods.**

**Keywords: Triangulation, difference in time of arrival, computational geometry, computational complexity.**

## I. Introduction

The problem of computing the location of an object from measurements of distance-differences from three known locations is well-studied (for at least three decades) under the title of Difference of Time-of-Arrival (DTOA) localization. This problem arises in a number of established areas such as tracking in aerospace systems [1], [2]. Recently, it has received renewed attention due to the increasing proliferation of wireless sensor networks [3], [4] and embedded networked systems [5]. In several of these applications, the wireless nodes are limited in power and yet the localization computations may have to be repeated quite frequently. Consequently, it has become important to trade-off the number and type of computations needed for localization to save power by gracefully degrading the quality of solution. In addition, the computational precision of arithmetic operations may be limited in some sensor nodes, but its impact on the precision of localization is not well understood. These factors motivate a closer examination of the computational aspects of DTOA triangulation methods; however, our results could be of more general interest as well.

There are two basic formulations of the DTOA localization problem: (a) distance-differences to an object, such as the origin of a plume described by spatial diffusions [6], are measured from known locations, and the problem is to estimate the location of the object; and (b) a device, such as a sensor node, receives distance-differences from beacons with known locations, and the problem is to estimate the location of the sensor node, that is self-localization. The classical source localization problem using DTOA measurements has been solved using two general approaches: (i) linear algebraic solution which typically involves matrix inversion and solving a quadratic equation [2], [7], and (ii) intersection of hyperbolic curves [8]. A recent overview of network-based localization methods may be found in [3]–[5], [9]. In general, the quality of the location estimate is a complex function of the precision with which the underlying numerical operations are implemented, and consequently, there is no apparent and simple way of relating the computations to the "quality" of location estimate. In particular, it is unclear if devoting more computational operations would increase the accuracy of these methods, or conversely if it is possible to reduce the computations without drastically affecting the quality of location estimate. In addition, sensor errors can have drastic effects on DTOA localization methods. For example, as will be shown in Section IV under simple random noise conditions, the quadratic equation of [2], [7] may have imaginary roots thereby rendering the method incomplete. Also, numerical instabilities may arise in the computations implemented with low precision operations wherein matrix inversions needed for linear algebraic methods may become ill-conditioned resulting in large estimation errors.

The underlying geometric nature of this problem has been well-known [1] although we are unaware of methods that exploit it to fine tune computations as done in several computational geometry methods [10], [11][1]. We present a computational geometric method for DTOA localization based on a binary search on an algebraic curve defined by a distance-difference function. We exploit the monotonicity of the directional derivative of the other distance-difference on it to support the binary search. The computational complexity of this method is $O(\log^2 1/\gamma)$, where the computed solution is

[1]The term triangulation has also been used in the context of computational geometry and kinetic data structures [12] to refer to the decomposition of planar regions into triangles which is quite different from its usage in this paper.

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **JUL 2007** | | **00-00-2007 to 00-00-2007** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **A Computational Geometry Method for DTOA Triangulation** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Florida,Computer and Information Science and Engineering Department,Gainesville,FL,32611** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**10th International Conference on Information Fusion, 9-12 July 2007, Quebec, Canada. U.S. Government or Federal Rights License**

14. ABSTRACT
**see report**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT **unclassified** | b. ABSTRACT **unclassified** | c. THIS PAGE **unclassified** | **Same as Report (SAR)** | **7** | |

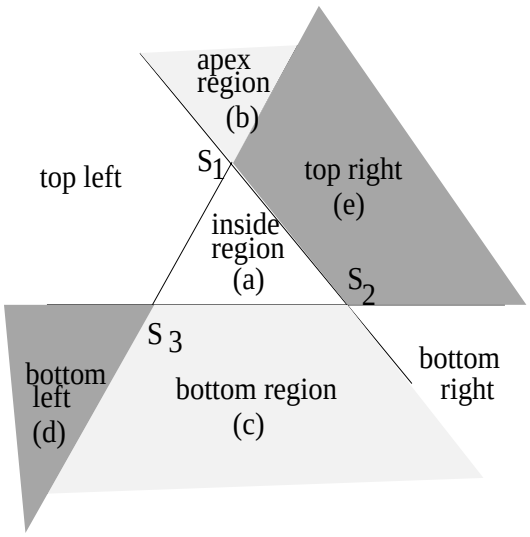



Fig. 1. Regions of monitoring area: (a) inside region, (b) apex region, (c) bottom region, (d) top left and right, (e) bottom left and right.

guaranteed to be within a $\gamma$-precision region centered at the actual solution; by a minor modification to the termination condition this region can be changed to $[-\gamma,\gamma]\times[-\gamma,\gamma]$-box centered at the actual solution. Alternatively, by fixing the number of operations to $k$, one can achieve the precision $\gamma = O\left(2^{-\sqrt{k}}\right)$. This method is robust with respect to distance measurement errors: (i) $\gamma$ is of the same order of magnitude as errors in distance measurements; in methods that involve division operations such guarantees cannot be made; and (ii) it is complete in that it will always return an answer, even under random measurement errors. This method is a generalization of the DTOA localization method in [6] proposed as a part of plume identification when the source is inside the acute triangle formed by sensors. In our case, the object can be located anywhere in the monitoring region. In addition, we also provide a detailed analysis of the underlying computation and the proof of the required monotonicity property of the underlying directional derivative.

This paper is organized as follows. We describe our geometric DTOA triangulation method in Section II. We prove the correctness of the method by establishing the monotonicity properties of the underlying directional derivative in Section III. We present simulation results in Section IV.

## II. Geometric DTOA Method

We are given three sensors $S_i$, $i=1,2,3$ located at $(x_i,y_i)$, $i=1,2,3$. For any point $P=(x,y)$ in the plane, we define $d(P,S_i)=\sqrt{(x-x_i)^2+(y-y_i)^2}$ and

$$\Delta_{ij}(P) = d(P,S_i) - d(P,S_j),$$

for $i,j=1,2,3$. We consider the *DTOA localization* problem of estimating the location of a source $S$ from the measurements of $\Delta_{12}(S)$ and $\Delta_{13}(S)$, given by $\delta_{12}$ and $\delta_{13}$, respectively. As we move $P$ from $S_i$ to $S_j$ along the line segment $\overline{S_iS_j}$, $\Delta_{ij}(P)$ varies monotonically and linearly from $-d(S_1,S_2)$ to $d(S_1,S_2)$, and equals 0 at the bisector point. We consider the locus of points defined by

$$L_{i,j}(\delta) = \{P | \Delta_{ij}(P) = \delta\}$$

which is described by the algebraic equation $d(P,S_i) - d(P,S_j) = \delta$.

We consider the generic configuration shown in Figure 1 such that $\delta_{12}\le 0$ and $\delta_{13}\le 0$. In the next section, we show that any configuration can be transformed into the generic one shown in Figure 1 that consists of seven regions. In each of the seven regions $\Delta_{13}(.)$ monotonically varies on $L_{12}(.)$; in particular, it monotonically decreases inside the apex and bottom regions and monotonically increases in all other regions as will be proven in the next section. The overall strategy to estimate $S=(x,y)$ is to perform a binary search on $L_{12}(.)$ to locate $\hat{S}=(\hat{x},\hat{y})$ such that $|\Delta_{13}(\hat{S})-\delta_{13}|\le\gamma$. For any point $P$ in the plane, let $R_{P,\gamma}$ be the *$\gamma$-precision region*, which corresponds to a "distorted box" centered at $P$, whose sides are formed by displaced hyperbolic curves as shown in Figure 2. The above condition implies that $\hat{S}\in R_{S,\gamma}$.

As we move $P$ along $L_{12}(\delta)$ in one direction, $\Delta_{13}(P)$ varies monotonically within each region. The basic idea is to utilize this monotonicity of $\Delta_{13}(P)$ to support a binary search: repeatedly compute a $P$ along $L_{12}(\delta_{12})$ until we reach $\hat{S}\in R_{S,\gamma}$. The details are presented in algorithm geometric_DTOA($\delta_{12},\delta_{13}$).

*algorithm* geometric_DTOA($\delta_{12},\delta_{13}$);
**begin**
1. $(x_{12},y_{12}) \leftarrow$ intersection point of $L_{12}$ with $\overline{S_1S_2}$;
2. $\mathcal{I}_{X1} \leftarrow$ set of X-coordinates of intersections of $L_{12}$ with $\overline{S_1S_3}$;
3. $\mathcal{I}_{X2} \leftarrow$ set of X-coordinates of intersections of $L_{12}$ with $\overline{S_2S_3}$;
4. $\mathcal{I}_X \leftarrow \{D_{X1}, x_{12}, D_{X2}\} \bigcup \mathcal{I}_{X1} \bigcup \mathcal{I}_{X2}$;
5. $\mathcal{I}_S \leftarrow$ sort $(\mathcal{I}_{X1})$;
6. let $\mathcal{I}_S = \{x_{(1)}, x_{(2)}, \ldots, x_{|\mathcal{I}_S|}\}$ and $\{y_{(1)}, y_{(2)}, \ldots, Y_{|\mathcal{I}_S|}\}$
7. be the corresponding $Y$-coordinates;
8. **for** $i = 1, \ldots |\mathcal{I}_S| - 1$ **do**
9. $x_{(i)} \leftarrow \max(x_{(i)}, D_{X1})$; $x_{(i)} \leftarrow \min(x_{(i)}, D_{X2})$;
10. $y_{(i)} \leftarrow \max(y_{(i)}, D_{Y1})$; $y_{(i)} \leftarrow \min(y_{(i)}, D_{Y2})$;
11. $\mathcal{S} \leftarrow \emptyset$;
12. **for** $i = 1, \ldots |\mathcal{I}_S| - 1$ **do**
13. $\square \leftarrow$ region_sign$\left(\frac{x_{(i)}+x_{(i+1)}}{2}, y_i, y_{(i+1)}\right)$;
14. $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{$ locate_$L_{13}\left(x_{(i)}, x_{(i+1)}, y_{(i)}, y_{(i+1)}, \square\right)\}$;
15. return $\mathcal{S}$;
**end**

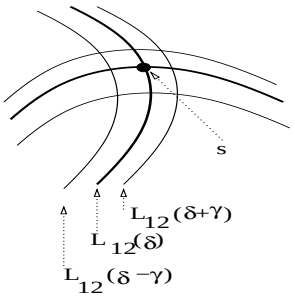



Fig. 2. $R_{S;\gamma}$ is a "distorted" box corresponding to $\gamma$-precision region centered at $S$.

*algorithm* region_sign($x, y_L, y_R$);
**begin**
1. $y \leftarrow$ locate_$L_{12}(y_L, y_R, x)$;
2. $\sigma_1 \leftarrow$ sign$(y - y_1 - (y_2 - y_1)(x - x_1)/(x_2 - x_1))$;
3. $\sigma_2 \leftarrow$ sign$(y)$;
4. $\sigma_3 \leftarrow$ sign$(y - y_1 x/x_1)$;
5. **if** $(\sigma_1, \sigma_2, \sigma_3) = (+,+,+)$ or $(-,-,-)$ **then** return($<$);
6. **else** return($>$);

**end**

Let $[D_{X_1}, D_{X_2}] \times [D_{Y1}, D_{Y2}]$ be the monitoring region within which $S$ is to be localized. The basic idea of algorithm geometric_DTOA is identify individual segments of $L_{12}$ that are entirely contained in single regions shown in Figure 1, and perform a binary search on $L_{12}$ with $L_{13}$ as the objective function within the region. The correct sign ($>$ or $<$) for the search within the region is supplied by the function region_sign($x_R, y_L, y_R$) as follows. This function computes a point $P = (x, y)$ on $L_{12}$ (line 1) and evaluates the triple of signs by substituting $x$ and $y$ into the equations of lines $\overline{S_1S_2}$, $\overline{S_1S_3}$, and $\overline{S_2S_3}$ in lines 2, 3 and 4 respectively. It then returns $<$ if the computed triple matches that of apex or bottom region in line 5, and returns $>$ otherwise in line 6.

In algorithm geometric_DTOA, the individual segments of $L_{12}$ that are entirely contained in single regions of Figure 1 are identified in lines 1-3, and are arranged in ascending order of X-coordinates of the end points of these segments in lines 5-7. The segments that lie outside the monitoring region are constrained to be within $[D_{X_1}, D_{X_2}] \times [D_{Y1}, D_{Y2}]$ in lines 8-10. The $x$ and $y$ coordinates of points on $L_{12}$ within each region are bounded within the intervals $[x_{(i)}, x_{(i+1)}]$ and $[y_{(i)}, y_{(i+1)}]$, respectively in each iteration of lines 13-14; note that each of $x_{(i)}$ corresponds to $L_{12}$ intersecting the lines through two of $S_1$, $S_2$ and $S_3$, or the end points of $[D_{X1}, D_{X2}]$. A binary search on $L_{12}$ with $L_{13}$ as the objective function is carried out within each region by calling locate_$L_{13}$(.) in line 14 and the returned points are accumulated in $\mathcal{S}$. This algorithm returns $\mathcal{S}$ which contains one or two candidates for $S$. For simplicity, $\hat{S} \in \mathcal{S}$ on the boundary of the monitoring region is interpreted as either as a source lying on the boundary or outside the region. Computation of $(x_{12}, y_{12})$ in line 1, $\mathcal{I}_{X1}$ in line 2 and $\mathcal{I}_{X2}$ in line 3 is carried out by a binary search on the line $\overline{S_1S_2}$, $\overline{S_1S_3}$, and $\overline{S_2S_3}$, respectively with $L_{12}$ as objective function.

*algorithm* locate_$L_{13}(x_L, x_R, y_L, y_R, \square)$;
**begin**
1. $x \leftarrow (x_L + x_R)/2$;
2. $y \leftarrow$ locate_$L_{12}(x, y_L, y_R)$;
3. $P = (x, y)$;
4. **if** $|\Delta_{13}(P) - \delta_{13}| < \gamma$ **then** return($P$);
5. **else if** $(\Delta_{13}(P) \square \delta_{13})$ **then**
6. locate_$L_{13}(x, x_R, y_L, y_R, \square)$;
7. **else**
8. locate_$L_{13}(x_L, x, y_L, y_R, \square)$;

**end**

As shown in *algorithm* locate_$L_{13}(x_L, x_R, y_L, y_R, \square)$, within each region the search is two-dimensional; it returns $P$ such that $|\Delta_{13}(P) - \delta_{13}| < \gamma$ that is contained in the box

*algorithm* locate_$L_{12}(x, y_L, y_R)$;
**begin**
1. $y \leftarrow (y_L + y_R)/2$; $P = (x, y)$;
2. **if** $|\Delta_{12}(P) - \delta_{12}| > \gamma$ **then** return($y$);
3. **else if** $\Delta_{12}(P) < \delta_{12}$ **then** locate_$L_{12}(x, y, y_R)$;
4. **else** locate_$L_{12}(x, y_L, y)$;

**end**

$[x_L, x_R] \times [y_L, y_R]$. First, in lines 1-3, a point $P = (x, y)$ on $L_{12}$ is located at mid $x$-value computed in line 1 by performing a binary search for $y$ by *algorithm* locate_$L_{12}(x, y_L, y_R)$ in line 2. Once $\Delta_{13}(P)$ is computed, the $x$-range is suitably halved, and this process is recursively carried out until the required precision $\gamma$ is reached in lines 4-8.

Complexity of computation of each element of $\mathcal{I}_{X1}$ and $\mathcal{I}_{X2}$ is $O(\log(1/\gamma))$, and each has at most 2 points. In computing $\hat{S} = (\hat{x}, \hat{y})$, there are altogether $O(\log(1/\gamma))$ calls to locate_$L_{13}$(.), and each call invokes locate_$L_{12}$(.), which in turn has a complexity of $O(\log(1/\gamma))$. Thus the complexity of *algorithm* geometric DTOA($\delta_{12}, \delta_{13}$) is $O(\log^2(1/\gamma))$, which can be adapted by suitably specifying $\gamma$. We have $|\mathcal{I}_S| \leq 5$, since $|\mathcal{I}_{X1}| \leq 2$ and $|\mathcal{I}_{X2}| \leq 2$, and thus there are at most four invocations of locate_$L_{13}$. If the number of basic computational operations are fixed $k$, then we have $\gamma < O\left(2^{-\sqrt{k/4}}\right)$.

## III. Monotonicity of Directional Derivative

In this section, we establish the correctness of the method described in previous section. First, any given configuration of three sensors can be rotated and relabeled such that vertex $S_1$ with both $\delta_{12}$ and $\delta_{13}$ are negative is above $y$ axis, and $S_3$ and $S_2$ are aligned along $x$-axis; note that $S_1$ is the closest vertex to $S$ hence always exists. Then a translation ensures that $S_1 = (x_1, y_1)$, $S_2 = (x_2, 0)$ and $S_3 = (0, 0)$, and $x_1 > 0; y_1 > 0; x_2 > 0$ which establishes that the configuration in Figure 1 is generic.

We consider five separate regions: (a) inside triangle, (b) top apex, and (c) bottom region, (d) bottom left, and (e) top right as shown in Figure 1. The other two regions, namely top left and bottom right, are "flipped" versions of cases in (e) and (d), respectively, and can be similarly handled. We show that the directional derivative of $\Delta_{13}(.)$ along the curve $L_{12}(.)$ is monotone in each of these regions: it is positive in regions (a), (d) and (e), and is negative in regions (b) and (c). The correct sign of the region for the search is supplied by region_sign(.), which establishes the correctness of *algorithm* geometric_DTOA.

We have for $i = 1, 2, 3$,

$$\frac{\partial d(S, S_i)}{\partial x} = \frac{(x - x_i)}{d(S, S_i)} \quad \text{and} \quad \frac{\partial d(S, S_i)}{\partial y} = \frac{(y - y_i)}{d(S, S_i)}.$$

Then directional derivative of $\Delta_{13}(P)$ at $P = (x, y)$ on the locus $L_{12}(\delta_{12}) = \{P | \Delta_{12}(P) = \delta_{12}\}$, for any $\delta_{12}$, is given

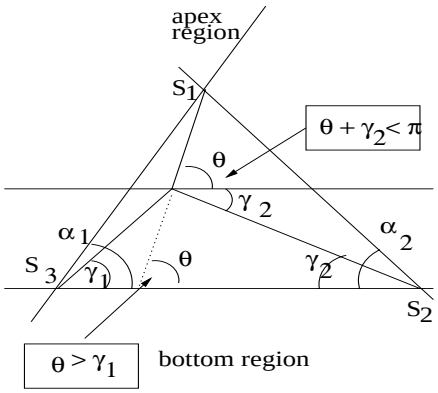



Fig. 3. Source $S = (x, y)$ is located inside the triangle.

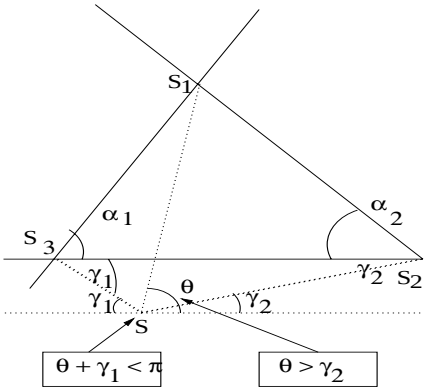



Fig. 5. Source $S = (x, y)$ is located in the bottom region.

by

$$\begin{aligned}
&\nabla_{L_{12}(\delta_{12})}\Delta_{13}(P) \\
&= \begin{bmatrix} \frac{\partial\Delta_{12}(P)}{\partial x} \\ \frac{\partial\Delta_{12}(P)}{\partial y} \end{bmatrix}^T \\
&\quad \circ \frac{1}{\sqrt{\left(\frac{\partial\Delta_{13}(P)}{\partial x}\right)^2 + \left(\frac{\partial\Delta_{13}(P)}{\partial y}\right)^2}} \begin{bmatrix} \frac{\partial\Delta_{13}(P)}{\partial x} \\ \frac{\partial\Delta_{13}(P)}{\partial y} \end{bmatrix} \\
&= \begin{bmatrix} \frac{x-x_1}{d(S,S_1)} - \frac{x-x_3}{d(S,S_3)} \\ \frac{y-y_1}{d(S,S_1)} - \frac{y-y_3}{d(S,S_3)} \end{bmatrix}^T \circ \frac{1}{K} \begin{bmatrix} \frac{x-x_1}{d(S,S_1)} - \frac{x-x_2}{d(S,S_2)} \\ \frac{y-y_1}{d(S,S_1)} - \frac{y-y_2}{d(S,S_2)} \end{bmatrix}
\end{aligned}$$

where $K = \frac{1}{\sqrt{\left(\frac{\partial\Delta(S_1,S_3)}{\partial x}\right)^2 + \left(\frac{\partial\Delta(S_1,S_3)}{\partial y}\right)^2}}$. We use the two following basic identities extensively in our derivations:

$$1 - \cos\alpha = 2\sin^2\alpha/2$$

$$\cos\alpha - \cos\beta = -2\sin\left(\frac{\alpha+\beta}{2}\right)\sin\left(\frac{\alpha-\beta}{2}\right).$$

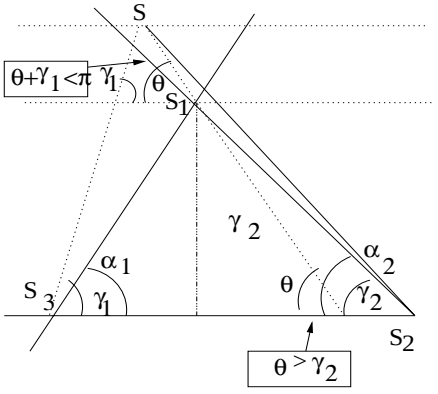



Fig. 4. Source $S = (x, y)$ is located in the apex region.

### A. Inside Triangle

In this case, we have $0 < \theta + \gamma_2 < \pi$ and $\theta > \gamma_1$ as shown in Figure 3. The directional derivative is given by

$$\begin{aligned}
&\nabla_{L_{12}(\delta_{12})}\Delta_{13}(P) \\
&= \begin{bmatrix} \frac{x-x_1}{d(S,S_1)} - \frac{x}{d(S,S_3)} \\ \frac{y-y_1}{d(S,S_1)} - \frac{y}{d(S,S_3)} \end{bmatrix}^T \circ \frac{1}{K} \begin{bmatrix} \frac{x-x_1}{d(S,S_1)} - \frac{x-x_2}{d(S,S_2)} \\ \frac{y-y_1}{d(S,S_1)} - \frac{y}{d(S,S_2)} \end{bmatrix} \\
&= (-\cos\theta - \cos\gamma_1)(-\cos\theta + \cos\gamma_2) \\
&\quad +(-\sin\theta - \sin\gamma_1)(-\sin\theta - \sin\gamma_2) \\
&= 1 + \cos(\theta - \gamma_1) - \cos(\theta + \gamma_2) - \cos(\gamma_1 + \gamma_2) \\
&= 2\sin^2\left(\frac{\gamma_1+\gamma_2}{2}\right) \\
&\quad +2\sin\left(\frac{\gamma_1+\gamma_2}{2}\right)\sin\left(\theta + \frac{\gamma_2-\gamma_1}{2}\right) \\
&= 2\sin\left(\frac{\gamma_1+\gamma_2}{2}\right) \\
&\quad \left[\sin\left(\frac{\gamma_1+\gamma_2}{2}\right) + \sin\left(\theta + \frac{\gamma_2-\gamma_1}{2}\right)\right] \\
&= 4\sin\left(\frac{\gamma_1+\gamma_2}{2}\right)\sin\left(\frac{\theta+\gamma_2}{2}\right)\cos\left(\frac{-(\theta-\gamma_1)}{2}\right)
\end{aligned}$$

We have $0 < \gamma_1 + \gamma_2 < \pi$, $0 < \theta + \gamma_2 < \pi$ which makes the first two $\sin$ terms to be positive. We have $\theta > \gamma_1$ and $0 < \theta < \pi$. Thus $-\pi/2 < \frac{-(\theta-\gamma_1)}{2} < 0$, which makes the third $\cos$ term to be positive. Hence the directional derivative is positive.

### B. Top Apex

In this case, we have $0 < \theta + \gamma_1 < \pi$ and $\theta > \gamma_2$ as shown in Figure 4. The directional derivative of $\Delta(S_1, S_3)$ on the locus $\{(x, y)|\Delta(S_1, S_2) = \delta_{12}\}$, for any $\delta_{12}$, is given by

$$\begin{aligned}
&\nabla_{\Delta(S_1,S_2)}\Delta(S_1, S_3) \\
&= (-\cos\theta - \cos\gamma_1)(-\cos\theta + \cos\gamma_2) \\
&\quad +(\sin\theta - \sin\gamma_1)(\sin\theta - \sin\gamma_2) \\
&= 1 + \cos(\theta + \gamma_1) - \cos(\theta - \gamma_2) - \cos(\gamma_1 + \gamma_2) \\
&= 2\sin\left(\frac{\gamma_1+\gamma_2}{2}\right)
\end{aligned}$$

$$\left[\sin\left(\frac{\gamma_1+\gamma_2}{2}\right)-\sin\left(\theta+\frac{\gamma_1-\gamma_2}{2}\right)\right]$$
$$= \quad 4\sin\left(\frac{\gamma_1+\gamma_2}{2}\right)\cos\left(\frac{\theta+\gamma_1}{2}\right)\sin\left(\frac{-(\theta-\gamma_2)}{2}\right)$$

We have $0 < \gamma_1 + \gamma_2 < \pi$, which makes the first term sin positive. We have $0 < \theta + \gamma_1 < \pi$, which makes the second term cos positive. Since $\theta > \gamma_2$ and $0 < \theta < \pi$, we have $-\pi/2 < \frac{\gamma_2-\theta}{2} < 0$, which makes the third term sin negative. Hence the directional derivative is negative.

### C. Bottom, Bottom Left, and Top Right Regions

(i) **Bottom Region:** For bottom region, the derivation is identical to the case of apex region: the conditions $0 < \theta + \gamma_1 < \pi$ and $\theta > \gamma_2$ are valid based on the geometric conditions specific to this case as shown in Figure 5.

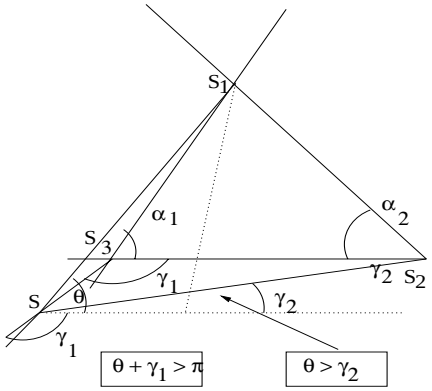
(ii) **Bottom Left:** The case of bottom left is identical to the bottom region except that $\theta + \gamma_1 > \pi$ as shown in Figure III-A, which makes the cos term negative, and hence the directional derivative is positive.

(iii) **Top Right:** The case of top right region shown in Figure 6 is identical to top apex region except that $2\pi > \theta+\gamma_1 > \theta+\alpha_1 > \pi$, which makes the second (cos) term negative, and hence the directional derivative is positive.

Computational results indicating the signs of the directional derivative of randomly generated sources are shown in Figure 7.

## IV. Simulation Results

We simulated a network of three sensors on a $[0, 100000] \times [0, 100000]$ grid such that $S_3$ and $S_2$ are located at $(0, 0)$ and $(100000, 0)$ respectively. Location of $S_1$ is randomly generated on the line segment between $(0, 100000)$ and $(100000, 100000)$. Each sensor measurement corresponds to $(1 + f)r$ where $r$ is the actual distance from sensor to source, and $f$ is uniformly randomly generated in the interval $[0, F]$ for a fixed *multiplicative factor* $F$. While $f$ values are generated independently, sensor error magnitude is proportional to the distance from the sensor to the source. The sensor errors are correlated due to the spatial relationships between the sensor locations; a source close to $S_3$ generates a small error there and larger errors at both $S_1$ and $S_2$, which are




caption Source $S = (x, y)$ is located in the bottom left region.

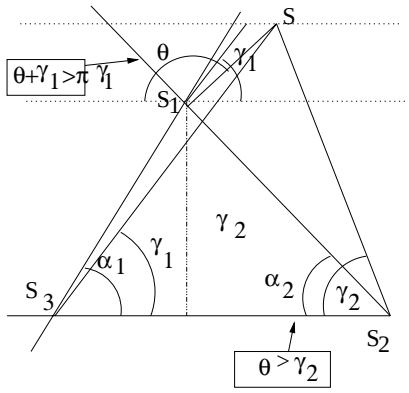



Fig. 6. Source $S = (x, y)$ is located in the top right region.

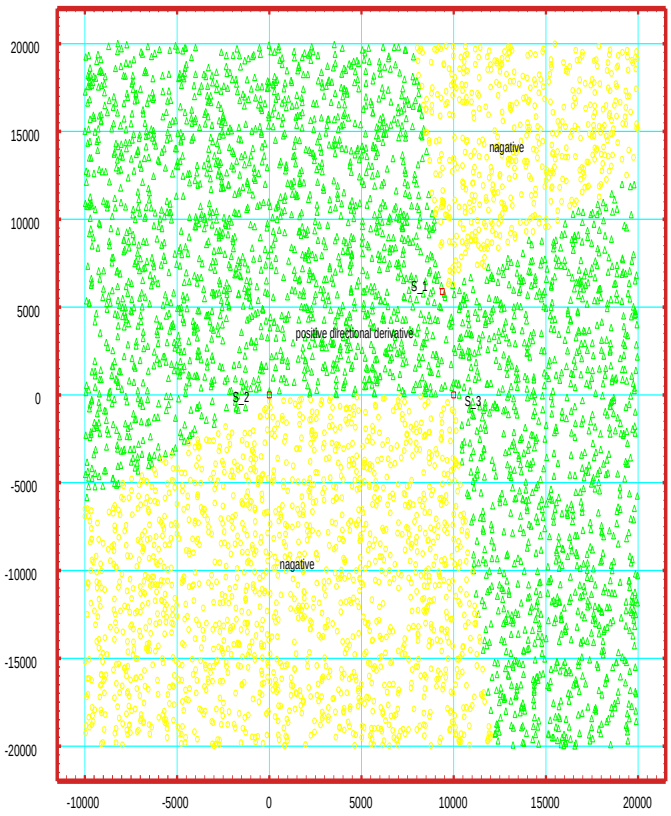



Fig. 7. Source $S = (x, y)$ is randomly selected, and the sign of $\nabla_{\Delta(S_1,S_2)}\Delta(S_1, S_3)$ is computed.

| $F$ | imaginary roots percentage |
|---|---|
| 0.01 | 0.01 |
| 0.02 | 0.047 |
| 0.03 | 0.13 |
| 0.05 | 0.32 |
| 0.10 | 0.963 |

TABLE I

LISTING OF PERCENTAGE OF IMAGINARY SOLUTIONS TO THE DTOA QUADRATIC EQUATION AS A FUNCTION OF THE MULTIPLICATIVE FACTOR $F$.

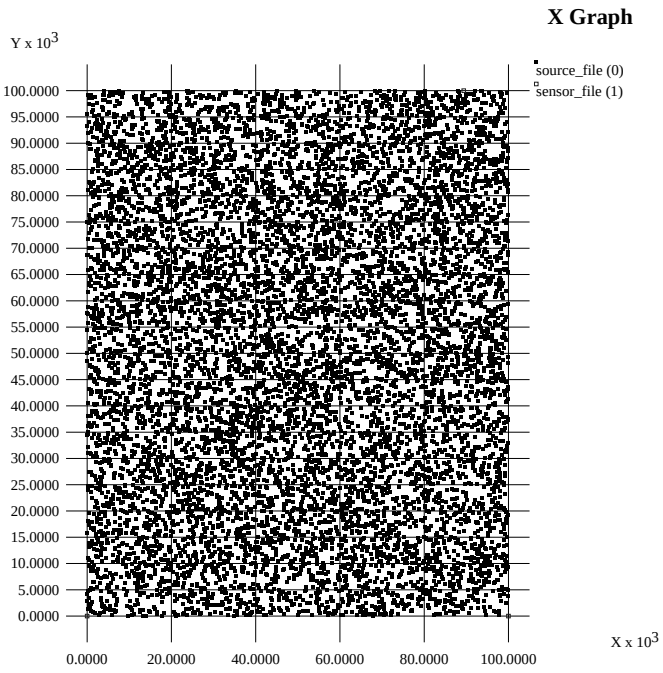
located farther away. From these measurements, we computed distance-differences and tested DTOA localization methods.

We implemented a linear algebra based method of [2], [7] which required a solution to a quadratic equation. When sensor errors are zero ($F = 0$) this method accurately estimated the source location. However, when $F > 0$, this method became incomplete as the quadratic equation had imaginary roots for a small percentage of sources as shown in Table 1 based on simulation of 100,000 sources. The generated sources are shown in Figure 8(a) which are uniformly distributed across $[0, 100000] \times [0, 100000]$ grid. For the case $F = 0.05$ in Table 1, about 0.32% of the sources yielded imaginary solutions to the quadratic equation, and these sources themselves are are concentrated around the sensors. On a related note, the method of [13] accounts for random errors that are independent Gaussian, and hence is not directly applicable to this case.

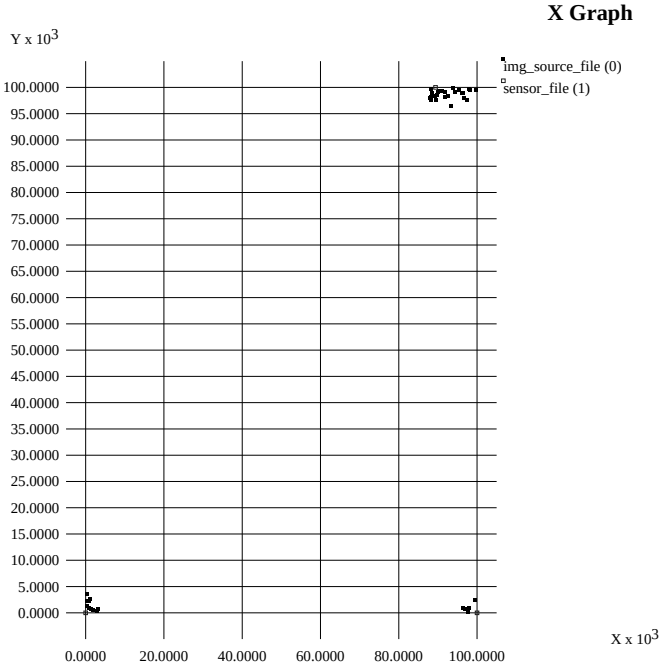
Results of our method are shown in Figure 9 for 10 different values of $\gamma$ with $\gamma_{\min} = 7.856613$ and $\gamma_{\max} = 29.011267$. This method is complete in that always returned the precision region. When sensor errors are zero, this region always included the source. If sensor errors are non-zero, this region did not always include the source depending on the value of $\gamma$. For each value of $\gamma$, we list $f_S$ which is the fraction of the sources that were not included in the precision region. As expected smaller value of $\gamma$ resulted in more missed sources and a large enough value always included the source. Also, for each value of $\gamma$, we are list the maximum number of steps needed in computing $\hat{S}$ in Figure 9 over the 100,000 sources. This method is implemented in C on Linux workstation (Opteron processor), and the typical execution times of algorithm geometric_DTOA for datasets are under a second.

## V. CONCLUSIONS

We presented a computational geometric method for the problem of triangulation in plane using measurements of distance-differences. This problem has been extensively studied in the past and several solutions have been deployed, and our re-examination is motivated in part by the requirements of low power sensor nodes. Our method is computationally efficient and adaptive as well as robust with respect to measurement and computational errors. This method is particularly suited for deployment in nodes that adapt their computations




(a) sources distributed uniformly across monitoring area




(b) sources that yielded imaginary roots

Fig. 8. Sources that yielded imaginary roots in DTOA localization methods based on quadratic equations are concentrated around the sensors.

| $\gamma_i$ | $f_S$ |
|---|---|
| 29.011267 | 0 |
| 22.744946 | 0.6 |
| 18.004549 | 0.7 |
| 25.958393 | 0.3 |
| 7.856613 | 0.9 |
| 14.310971 | 0.8 |
| 27.373035 | 0.2 |
| 26.435724 | 0.1 |
| 25.848469 | 0.4 |
| 23.103365 | 0.5 |

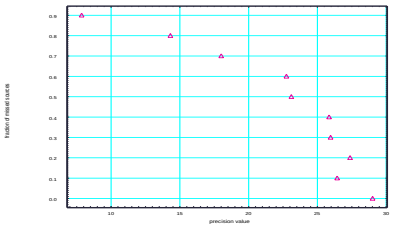


Fig. 9. Fraction of sources lying outside $R_{\hat{S},\gamma}$ corresponding to different $\gamma$ values.

| $\gamma_i$ | number of steps |
|---|---|
| 29.011267 | 67 |
| 22.744946 | 71 |
| 18.004549 | 75 |
| 25.958393 | 69 |
| 7.856613 | 90 |
| 14.310971 | 79 |
| 27.373035 | 68 |
| 26.435724 | 68 |
| 25.848469 | 69 |
| 23.103365 | 71 |

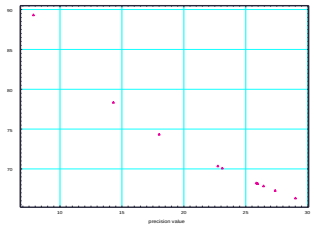


Fig. 10. Number of search steps corresponding to $\gamma$ values.

in response to power budgets. This method can also be applied when distance measurements are available, and can offer similar advantages over the linear algebraic methods that are often used for triangulation based on distances. Furthermore, by computing distance-differences from distance measurements, this method would be less susceptible to one-sided bias errors in distance measurements. This is particularly useful in certain self-localization tasks, where a single sensor is employed to measure distances to reference beacons.

This paper is only a step towards utilizing computational geometric methods for solving localization problems. It would be of future interest to consider extensions of this method for cases where more than three sensors are deployed and multiple measurement sets are provided. It would also be interesting to see if the proposed method can be extended under random noise models. Also, multiple path effects are not considered in this paper, and it would be of interest to explore such extensions. For the special case when $S_1$, $S_2$ and $S_3$ form an acute triangle, a training method was proposed in [6] wherein the localization method can be trained in-situ to account for sensor correlations. The current method can be similarly employed but the training procedure is likely to be more involved. It would be of future interest to explore the "tracking" ability of this method by repeatedly executing it on a stream of distance-difference measurements corresponding to a moving object.

## ACKNOWLEDGMENTS

This work is funded by the SensorNet program at Oak Ridge National Laboratory (ORNL) through Office of Naval Research. ORNL is managed by UT-Battelle, LLC for U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

## REFERENCES

[1] R. Schmidt, "A new approach to geometry of range difference location," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 8, no. 3, 1972.
[2] G. Mellen, M. Pachter, and J. Raquet, "Closed-form solution for determining emitter location using time difference of arrival measurements," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 1056–1058, 2003.
[3] F. Zhao and L. Guibas, *Wireless Sensor Networks*. Elsevier, 2004.
[4] B. Krishnamachari, Ed., *Networking Wireless Sensors*. Cambridge University Press, 2005.
[5] G. Pottie and W. Kaiser, *Principles of Embedded Networked System Design*. Cambridge University Press, 2005.
[6] N. S. V. Rao, "Identification of simple product-form plumes using networks of sensors with rand om errors," in *International Conference on Information Fusion*, 2006.
[7] H. C. Schau and A. Z. Robinson, "Passive source localization employing intersecting sperical surfaces from time-of-arrival differences," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 35, no. 8, pp. 1223–1225, 1987.
[8] B. T. Fang, "Simple solutions for hyperbolic and related position fixes," *IEEE Transactions on Systems, Man and Cybernetics-B*, vol. 26, no. 9, pp. 748–753, 2005.
[9] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location," *IEEE Signal Processing Magazine*, pp. 24–40, July 2005.
[10] F. P. Preparata and I. A. Shamos, *Computational Geometry: An Introduction*. New York: Springer-Verlag, 1985.
[11] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.
[12] L. Guibas, "Kinetic data structures: A state of the art report," in *Proc. 3rd Workshop on Algorithmic Foundations of Robotics*, P. K. Agarwal, L. Kavraki, and M. Mason, Eds. A. K. Peters, 1998.
[13] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. on Image Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.