Chapter 6
Error Control

Many Localization algorithms are range based and adopt distance ranging techniques, in which measuring errors are inevitable. Generally speaking, errors fall into two categories: *extrinsic* and *intrinsic*. The extrinsic error is attributed to the physical effects on the measurement channel, such as the presence of obstacles, multipath and shadowing effects, and the variability of the signal propagation speed, due to changes in the surrounding environment. On the other hand, the intrinsic error is caused by limitations of hardware and software. While extrinsic error is more unpredictable and challenging to handle in realistic deployments, the intrinsic one can also cause many complications when using multihop measurement information to estimate node location. Results from field experiments demonstrate that even relatively small measurement errors can significantly amplify the error in location estimates [62]; thus, for high-accuracy localization algorithms, error control is essential.

6.1 Measurement Errors

6.1.1 Errors in Distance Measurements

Table 6.1 lists the typical measuring (intrinsic) error of a range of nowadays ranging techniques: TDoA, RSS in AHLoS [31], ultra-wideband system [74], RF time-of-flight (ToF) ranging systems [75], and elapsed time between the two time of arrival (EToA) in BeepBeep [34]. In general, the accuracy of RF-based ranging techniques, e.g., RSS, UWB, and RF ToF, can achieve the meter-level accuracy in a range of tens of meters. In contrast, ToA-based methods have more accurate results in the order of centimeters but require extra hardware and energy consumption.

On the other hand, extrinsic errors are caused by environmental factors or unexpected hardware malfunction, leaving difficulties on characterizing them. We will review the state-of-the-art works on controlling the intrinsic and extrinsic errors in the following sections of location refinement and outlier-resistant localization, respectively.

6.1.2 Negative Impact of Noisy Ranging Results

Errors in distance ranging make localization more challenging in the following four aspects [76]:

1. ***Uncertainty***. Figure 6.1 illustrates an example of trilateration under noisy ranging measurements. Trilateration often meets the situation that the three circles do not intersect at a common point. In other words, there does not exist any position satisfying all distance constraints.
2. ***Nonconsistency***. In many cases, one node has many reference neighbors. Any subgroup of them (on less than three) can locate this node by multilateration. The computed results, however, is varying if different groups of references are chosen, resulting in nonconsistency. Thus, when alterative references are available, it is a problem to determine which combination of references provides the best result.
3. ***Ambiguity***. The existence of flip and flex [62, 77] may lead to large localization errors. Although localizability theory presents methods of detecting possible flip and flex ambiguities, these methods do not work when distance measurements are noisy.

Table 6.1 Measurement accuracy of different ranging techniques

Technology	System	Measurement accuracy	Range
TDoA	AHLoS	2 cm	3-10 m
RSS	AHLoS	2-4 m	30-100 m
UWB	PAL UWB	1.5 m	N/A
RF ToF	RF ToF ranging system	1-3 m	100 m
EToA	BeepBeep	1-2 cm	10 m

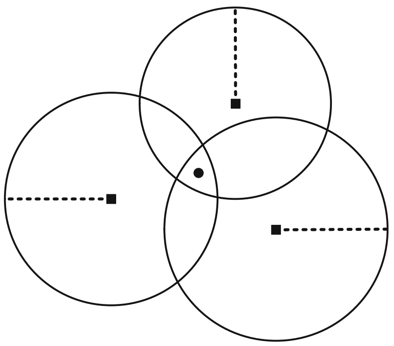


Fig. 6.1 Trilateration under noisy ranging measurements

4. ***Error propagation***. The results of a multihop localization process are based on a series of single hop multilaterations in an iterative manner [31]. In such a process, errors, coming from each step of multilateration, propagate and accumulate [60, 61].

6.2 Error Characteristics

Localization error is a function of a wide range of network configuration parameters, including the numbers of beacons and to-be-localized nodes, node geometry, network connectivity, etc., which constitute a complicated system. Understanding the error characteristics is an essential step toward controlling error. The Cramer Rao lower bound (CRLB) provides a means for computing a lower bound on the covariance of any unbiased location estimate that uses RSS, TdoA, and other ranging techniques. In addition, CRLB can serve as a benchmark for a particular localization algorithm. If the bound is closely achieved, there is little gain to continue improving the algorithm's accuracy. Furthermore, the dependence of CRLB on network parameters helps to understand the error characteristics of network localization.

6.2.1 What is CRLB

The Cramer Rao Lower Bound (CRLB) is a classic result from statistics that gives a lower bound on the error covariance for an unbiased estimate of parameter [78]. This bound provides a useful guideline to evaluate various estimators. One important and surprising advantage of CRLB is that we can calculate the lower bound without even considering any particular estimation method. The only thing needed is the statistical model of the random observations, i.e., $f(X|\theta)$, where X is the random observation and θ is the parameter to be estimated. Any unbiased estimator $\hat{\theta}$ must satisfy

$$\mathrm{Cov}(\hat{\theta}) \geq \{-E[\nabla_\theta(\nabla_\theta \ln f(X|\theta))^{\mathrm{T}}]\}^{-1}, \tag{6.1}$$

where $Cov(\hat{\theta})$ is the error covariance of the estimator, $E[\cdot]$ indicates expected value, and ∇_θ is the gradient operator with respect to θ.

The CRLB is limited to unbiased estimators, which provides estimates that are equal to the ground truth if averaged over enough realizations. In some cases, however, a biased estimation approach can produce both a variance and a mean-squared error that are below the CRLB.

6.2.2 CRLB for Multihop Localization

In network localization, the parameter vector θ of interest consists of the coordinates of nodes to be localized, given by $\theta = [x_1, y_1, x_2, y_2, x_L, y_L]^{\mathrm{T}}$, where L is the number of nodes to be localized. The observation vector X is formed by stacking the distance measurements $\hat{d}_{ij}$. Let M denote the size of X. We assume the distance measurement are Gaussian [62, 79], so the pdf of X is vector Gaussian. According to (6.1), we find that $CRLB = \{(1/\sigma^2)[G'(\theta)]^{\mathrm{T}}[G'(\theta)]\}^{-1}$, where σ^2 is the variance of each distance measurement error and $G'(\theta)$ is the $M \times 2L$ matrix whose mnth element is

$$G'(\theta)_{mn} = \begin{cases} \dfrac{x_i - x_j}{d_{ij}}, & \text{if}\theta_n = x_i; \\ \dfrac{x_j - x_i}{d_{ij}}, & \text{if}\theta_n = x_j; \\ \dfrac{y_i - y_j}{d_{ij}}, & \text{if}\theta_n = y_i; \\ \dfrac{y_j - y_i}{d_{ij}}, & \text{if}\theta_n = y_j; \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

The above result on CRLB is with the assumption that the location information of beacons is exact. When beacon nodes have location uncertainty, we can also characterize localization accuracy using a covariance bound that is similar to CRLB. Both these two bounds are tight in the sense that localization algorithms achieve these bounds for highly accurate measurements. In addition, according to (6.2), CRLB can be computed analytically and efficiently and avoid the need for expensive Monte Carlo simulations. The computational efficiency of CRLB facilitates to study localization performance of large-scale networks.

6.2.3 CRLB for One-Hop Localization

One-hop multilateration is the source of the location error that could be amplified by the iterative fashion of network localization. CRLB for multilateration exactly demonstrates how distance measurement errors and node geometry affect location accuracy.

Consider the one-hop localization problem: there are m reference nodes $v_1, v_2, \ldots, v_m$ and one node v_0 to be localized. From (6.1) and (6.2), we obtain

$$\sigma_0^2 = \sigma^2 m \left[\sum_{i=1}^{m-1} \sum_{j>i}^{m} \sin^2 \alpha_{ij} \right]^{-1}, \tag{6.3}$$

where σ_0^2 is the variance of the estimate location of v_0, α_{ij} is the angle between each pair of reference nodes (i, j). According to Eq. (6.3), the uncertainty of location estimate consists of two parts: the ranging error (σ_0^2) and the geometric relationship of references and the to-be-localized node (α_{ij}). Eliminating the impact of ranging errors, the error amplification effect caused by the node geometry has been demonstrated as the *geographic dilution of precision* (GDoP), which is defined as σ_0/σ.

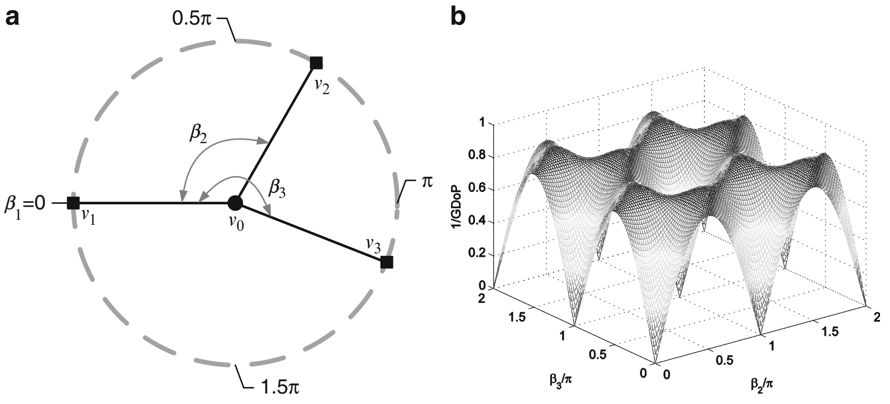


Fig. 6.2 The impact of node geometry on the accuracy multilateration

To gain more insights of GDoP, we consider a simplified case of multilateration, where the to-be-localized node v_0 is put at the center of a circle and $m = 3$ reference nodes v_1, v_2, v_3 lie on the circumference of that circle, setting all references the same distance to v_0. Fixing v_1 at $\beta_1 = 0$, according to the definition of GDoP, it becomes a function of the locations of v_2 and v_3, denoted by $\beta_2, \beta_3 \in [0, 2\pi]$, respectively. We plot the GDoP in Fig. 6.2 and conclude that different geometric forms of multilateration provide different levels of localization accuracy. In particular, in this circular trilateration, the highest location accuracy would be achieved if reference nodes are evenly separated, namely, $\beta_1 = 0, \beta_2 = \frac{2}{3}\pi$ and $\beta_3 = \frac{4}{3}\pi$.

6.3 Localization

AmbiguitiesIn the literature of graph realization problem, graph rigidity theory distinguishes between *flexible* and *rigid* graphs. Flexible graphs can be continuously deformed to produce an infinite number of different realizations preserving distance constraints, while rigid graphs have a finite number of discrete realizations. For rigid graphs, however, two types of discontinuous ambiguities exist, preventing a realization from being unique [62, 77]:

- **Flip**. Figure 6.3 shows an example of flip, where the two nodes in the middle create a mirror through which the position of v can be reflected without any change of inter-node distance.
- **Flex**. Discontinuous flex ambiguities occur when the removal of one edge allows the graph to be continuously deformed to a different realization and the removed edge can be reinserted with the same length. An example of flex ambiguity is illustrated in Fig. 6.4.

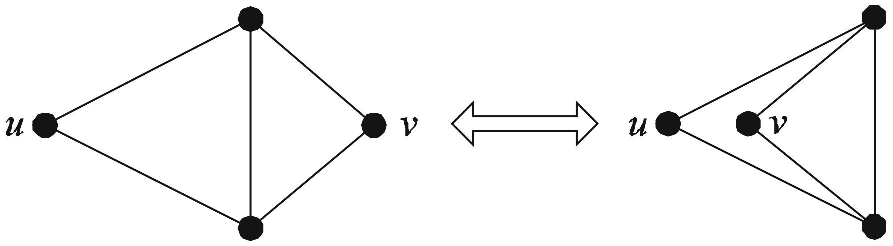


Fig. 6.3 An example of flip ambiguity

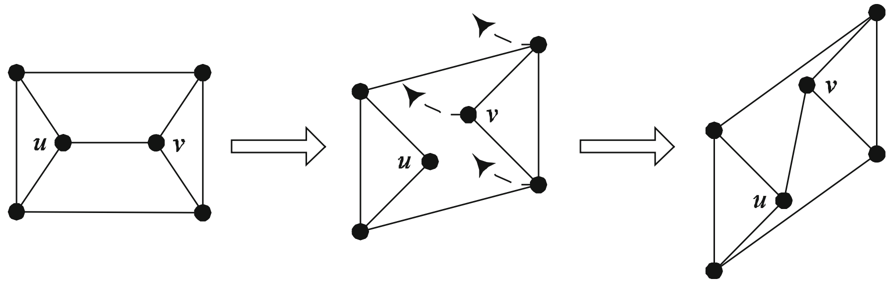
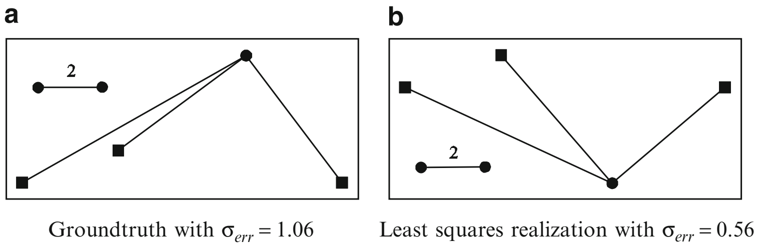


Fig. 6.4 An example of flex ambiguity



a
b
2
2
Groundtruth with $\sigma_{err} = 1.06$
Least squares realization with $\sigma_{err} = 0.56$

Fig. 6.5 An example of flip due to noisy distance measurements, where *black boxes* denote beacon nodes, and the *black circle* is the to-be-localized node

Graph rigidity theory [77] suggests ways of determining whether flip or flex ambiguities exist in a graph by checking global rigidity. However, this kind of criterion fails when distance measurements are noisy. Even when the underlying graph is globally rigid in the graph theoretic sense, realizations of the graph rarely satisfy the distance constraints exactly; furthermore, alterative realizations can exist and satisfy the constraints better than the correct one (the ground truth). An example of trilateration, which is a globally rigid structure, is illustrated in Fig. 6.5, where internode distance measurements are generated from a Gaussian distribution with a mean of the true distance and standard deviation $\sigma = 0.5$. Figure 6.5a is the ground truth realization with error metric $\sigma_{\mathrm{err}} = 1.06$ which is defined as the average difference between the computed distances and the measured distances. Figure 6.5b is the least-squares realization which actually localizes the node at its mirror location with respect to the three beacon nodes, but with a much better error metric $\sigma_{\mathrm{err}} = 0.56$.

Compared to flex ambiguities, flip ambiguities are more likely to occur in practical localization procedures and have attracted a lot of research efforts. In this section, we focus on the strategies of flip avoidance.

Moore et al. [62] outline certain criteria to select subgraphs to be used in localization against flip ambiguities due to noisy distance measurements. Rather than arbitrary quadrilaterals, they use "robust quadrilaterals" (robust quads) to localize nodes. As shown in Fig. 6.6, a robust quad consists of four subtriangles (ΔABC, ΔADC, ΔABD and ΔBCD) that satisfy

$$b \sin^2(\theta) > d_{\min} \tag{6.4}$$

where b is the length of the shortest side, θ is the smallest angle, and $d_{\min}$ is a predetermined constant according to the average measurement error. The idea is that the vertices of a quad can be placed correctly with respect to each other, i.e., without flip ambiguity. Moore et al. demonstrate that the probability of a robust quadrilateral experiencing internal flips given zero mean Gaussian measurement error can be bounded by setting $d_{\min}$ appropriately. In effect, $d_{\min}$ filters out quads that have too many positional ambiguities. The approximate level of filtering is based on the distance measurements. For instance, let $d_{\min} = 3\sigma$, then for Gaussian noise, we can bound the probability of flip for a given robust quadrilateral to be less than 1%, which poses minimal threat to the stability of the localization algorithm. Furthermore, these robustness conditions have a tendency to orphan nodes, either because they could not be localized by a robust quad or because their local map fail to overlap sufficiently with the global map. This tendency is acceptable because the orphaned nodes are likely to display large error. The drawback of this strategy is that under conditions of sparse networks or high measurement noisy, the algorithm may be unable to localize a useful number of nodes. Suggested in [27], there are other criteria that can better characterize the robustness of a given subnetwork against noisy ranging measurements.

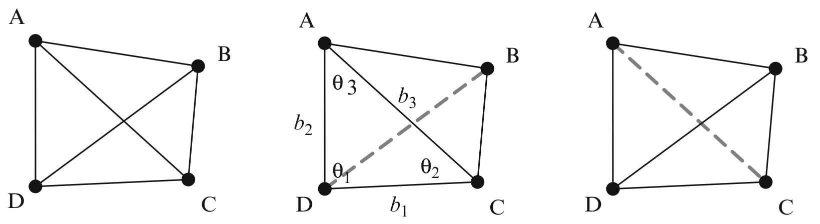


Fig. 6.6 Robust quadrilateral

Kannan et al. [80] propose another formal geometric analysis of flip ambiguity similar to robust quads [62]. Flip ambiguities are classified into two categories: *substantial flip ambiguity* and *negligible flip ambiguity*, based on the distance Δd between the two possible positions of the to-be-localized node. A flip ambiguity is substantial if $\Delta d \geq \delta_S$, some given bound. Otherwise, it is a negligible flip ambiguity. Instead of filtering out possible flip ambiguities as suggested in [62], they consider the identification of the substantial flip ambiguities only, because the location error introduced by negligible flip ambiguities is comparable to the uncertainty demonstrated by GDoP discussed previously. This strategy enables more nodes to be localized compared to robust quads. For a quadrilateral ABCD with known node positions A and B, Kannan et al. outline an algorithm to determine the region for the position of D such that node C can be uniquely localized using the measurements of the distances |AC|, |BC|, and |DC|.

6.4 Location Refinement

Since localization is often conducted in a distributed and iterative manner, error propagation is considered as a serious problem, in which nodes with inaccurate location estimates contaminate the localization process based on them. Existing studies [76, 79, 81] have demonstrated that location refinement is an effective technique to tackle this issue.

The basic location refinement requires nodes update their locations in a number of rounds. At the beginning of each round a node broadcasts its location estimate, receives the location information from its neighbors, and computes an LS-based multilateration solution to estimate its new location. In general, the constraints imposed by the distance to the neighbors will force the new location toward the ground truth location of the node. After a specified number of iterations or when the location update becomes small, the refinement stops and reports the final localization result.

The basic refinement algorithm is fully distributed, easy to implement, and efficient in communication and computation. An essential drawback of the basic refinement algorithm is that it is inherently unclear under what conditions the iteration would converge and how accurate the final solution would be, because in each round a node will update its location unconditionally, and there is no guarantee to make the new location better than the old one. We often call this basic refinement algorithm *refinement without error control*. In contrast, in this section we discuss *refinement with error control*, in which a node updates its location only when the new location is better than the old one. For simplicity, in the rest of the section, without special statements, when referring to location refinement, we mean refinement with error control.

6.4.1 A Framework of Location Refinement

To deal with error propagation, a number of location refinement algorithms have been proposed. In general, they are composed of three major components [79]:

1. ***Node registry***. Each node maintains a registry that contains the node location estimate and the corresponding estimate confidence (uncertainty).
2. ***Reference selection***. When redundant references are available, based on an algorithm-specified strategy, each node selects the reference combination achieving the highest estimate confidence (lowest uncertainty) to localize itself.
3. ***Registry update***. In each round, if higher estimate confidence (lower uncertainty) is achieved, a node updates its registry and broadcasts this information to its neighbors.

Algorithm 6.1 outlines the framework of location refinement, in which how to select appropriate reference combinations is the key step. Different strategies of addressing this issue lead to different location refinement algorithms.

6.4.2 Metrics for Location Refinement

Although GDoP characterizes the effects of node geometry on location estimate, it cannot be directly applied to the localization procedure due to the need of the ground truth location of each node. This is a challenging problem and has attracted a lot of research efforts.

Savarese et al. [81] propose a method that gives a confidence value to each node and weights one-hop multilateration results based on such confidence values. The estimate confidence is defined as follows. Beacons immediately start off with confidence 1; to-be-localized nodes begin with a low confidence (0.1) and raise their confidences at subsequent refinement iterations. In each round, a node chooses those reference nodes that will raise its confidence to localize itself, and sets its

Algorithm 6.1 A framework of location refinement

```
1: Each node holds the tuple ($p$, $e$), where $p$ is the node location estimate, $e$ is the corresponding
   estimate confidence (uncertainty).
2: Initialization step (optional):
   Each node computes an initialized location estimate.
3: In each round, nodes update their registries.
   do
      for all to-be-localized node $t$ do
      examine local neighborhood $N(t)$
      select the best reference combination and compute the estimate location $\hat{p}_t$ and confidence $\hat{e}_t$
      decide whether to update the registry of $t$ with the new tuple
         while the termination condition is not met.
```

confidence to the average of reference confidences after a successful multilateration. Nodes close to beacons will raise their confidence at the first iteration, raising in turn the confidences of node two hops away from beacons in the next iteration, etc. This strategy is based on the intuition that the estimated locations of nodes close to beacons are more reliable but puts little emphasis on node geometry.

Besides introducing the estimate confidence, Savarese et al. also consider the issue of ill-connected nodes, e.g., a cluster of n nodes with no beacons and connected to the main network by a single link, which are inherently hard or even impossible to locate. To detect non-ill-connected nodes, they adopt a heuristic criterion: a non-ill-connected node must have three edge disjoint paths to three distinct beacons. None of ill-connected nodes participate in the location refinement, which would make the algorithm convergence much faster.

By analyzing the effects of ranging errors and reference location errors on the estimated locations, Liu et al. [79] design a location refinement scheme with error management. Each node maintains information (p, e), where p is the estimated location, and e is the corresponding estimate error, a metric reflecting the level of uncertainty. At the beginning, each beacon is initialized with a registry (*beacon_loc*, 0), and the to-be-localized nodes are initialized as (*unknown_loc*, ∞). To handle errors, a robust LS (RLS) solution is adopted instead of the traditional LS solution $(A^TA)^{-1}A^Tb$ (discussed in Section 3.1) that gives

$$\hat{x}_t = \arg\min_x |Ax - b|^2$$

Let ΔA and Δb denote the perturbations of A and b, respectively. The RLS solution aims at

$$\hat{x}_t = \arg\min_x |(A + \Delta A)x - (b + \Delta b)|^2$$

With the assumption that ΔA and Δb are zero mean, the cost to minimize is

$$\begin{aligned}\varepsilon &= E|(Ax - b) + (-\Delta b)|^2 \\ &= (x^TA^TAx - 2x^TA^Tb + b^Tb) + (x^TE[\Delta A^T\Delta A]x - 2x^TE[\Delta A^T] + E[\Delta b^T\Delta b]) \\ &= x^T(A^TA + E[\Delta A^T])x - 2x^T[A^Tb + E[\Delta A^T\Delta b]] + (b^Tb + E[\Delta b^T\Delta b])\end{aligned}$$

Accordingly, the RLS solution is given by

$$\hat{x}_t = (A^TA + C_A)^{-1}[A^Tb + r_{Ab}]$$

where $C_A = E[\Delta A^T \cdot \Delta A]$ is the covariance matrix of perturbation of ΔA, corresponding to the uncertainties of reference locations, and $r_{Ab} = E[\Delta A^T\Delta b]$ is the correlation between ΔA and Δb. If ΔA and Δb are uncorrelated, the value of this term is 0. The analysis from [79] suggest that r_{Ab} is often negligible compared to the term A^Tb. Thus, the RLS solution becomes

$$\hat{x}_t = (A^T A + C_A)^{-1} A^T b. \tag{6.5}$$

Compared to the LS solution, the RLS solution uses the error statistics C_A as regularization, which would improve stability significantly when A is nearly singular or ill-conditioned. Based on the RLS solution, the location estimate error caused by noisy distance measurements can be expressed by

$$\begin{aligned} E|e_{\Delta b}|^2 &= E|(A^T A + C_A)^{-1} A^T \Delta b|^2 \\ &= E[\Delta b^T A (A^T A + C_A)^{-T} (A^T A + C_A)^{-1} A^T \Delta b] \\ &= \operatorname{trace}[A (A^T A + C_A)^{-T} (A^T A + C_A)^{-1} A^T \operatorname{Cov}(\Delta b)] \end{aligned}$$

Similarly, the error due to reference location uncertainty is

$$\begin{aligned} E|e_{\Delta a}|^2 &= E|(A^T A + C_A)^{-1} B \Delta a|^2 \\ &= \operatorname{trace}[B^T (A^T A + C_A)^{-T} (A^T A + C_A)^{-1} B \operatorname{Cov}(\Delta a)] \end{aligned}$$

where $a = (a_{11}, a_{21}, a_{n1}, a_{12}, a_{22}, a_{n2})^T$, a vector rearranging elements in matrix A, Δa is the perturbation of a because of location uncertainty, and B is a matrix satisfying $A^T b = Ba$, i.e.,

$$B \triangleq \begin{pmatrix} b_1 & b_2 & \ldots & b_n & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & b_1 & b_2 & \ldots & b_n \end{pmatrix},$$

in which b_1, b_2, b_n are elements in b. The total location error is the summation of these two terms, as they are assumed to be uncorrelated, i.e., $\hat{e} = E|e_{\Delta b}|^2 + \beta E|e_{\Delta a}|^2$, where β is a parameter to compensate for the over-estimation of the error due to a. A small value of β works well in practice [79].

By defining Quality of Trilateration (QoT) [76], the accuracy of trilateration can be characterized, enabling the comparison and selection among various geometric forms of trilateration. Assuming some probability distribution of ranging errors, probability tools are accordingly applied to quantify trilateration. The large value of QoT indicates the estimate location is, with high probability, close to the real location.

Let $t = \mathrm{Tri}(s, \{s_i, i = 1,2,3\})$ denote a trilateration for a target node s based on three reference nodes s_i. The quality of trilateration t is defined as

$$Q(t) = \int_p \prod_{i=1}^{3} f_{s,s_i}(d(p, p(s_i)))dp, \; p \in \mathrm{Disk}(p_t(s), R), \tag{6.6}$$

where $f_{s,s_i}(x)$ is the pdf of the distance measurement between s and s_i, and $p_t(s)$ is the estimated location based on trilateration t, and Disk(p, R) is a disk area centered at p with radius R. The parameter R is application specific for different accuracy requirements. To gain more insight of QoT, Yang et al. [76] provide some instances

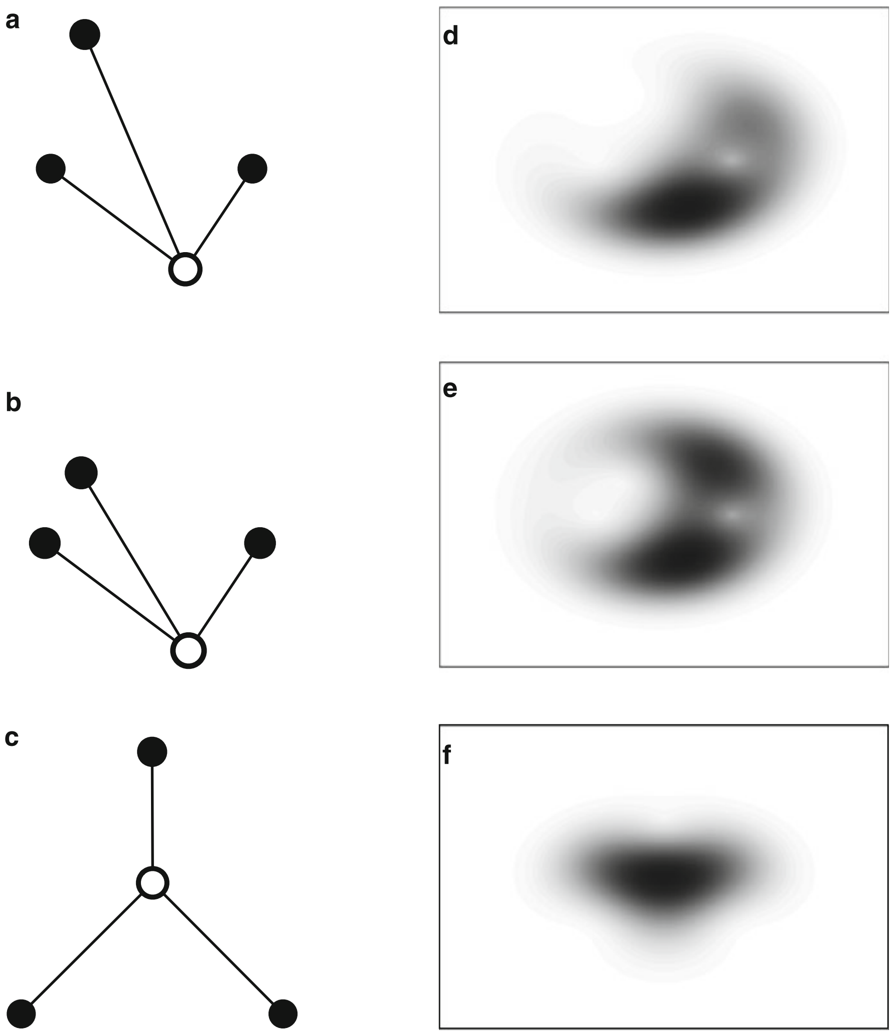


Fig. 6.7 The impact of geometric relations on QoT

to illustrate the impact of geometric relationship on QoT, shown in Fig. 6.7. Figure 6.7(a), (b), and (c) show the ground truths of three examples of trilateration. The black circles are the references and the white ones are the nodes to be localized. Based on the assumption that ranging measurements are with normal noises, the corresponding probability distributions are shown in Fig. 6.7. For the first instance, Fig. 6.7 displays the probability distribution of a general case. And for the second instance, Fig. 6.7(e) indicates a high probability of flip ambiguity as three references nodes are almost collinear. In the third instance, Fig. 6.7(f) plots a concentrated probability distribution which is accord with the fact that three references in Fig. 6.7(c) are well separated around the node to be localized.

Similar to [81], each node maintains a confidence associated with its location estimate. The confidence of s (based on t) is computed according to the confidences of references $C(s_i)$:

$$C_t(s) = Q(t)\prod_{i=1}^{3} C(s_i). \tag{6.7}$$

In each iteration, a to-be-localized node selects the trilateration that achieves the highest confidence to localize itself. Different from [81] that only takes the reference nodes reliability into account, QoT also considers the effects of geometry when computing confidence. Compared to conventional LS-based approaches, QoT provides additional information that indicates how accurate a particular trilateration is. Such difference enables QoT the ability of distinguishing and avoiding poor trilaterations that are of much location uncertainty.

6.5 Outlier-Resistant Localization

Compared with intrinsic errors, extrinsic errors are more unpredictable and caused by non-systematic factors. Especially in some cases, the errors can be extremely large due to the following factors:

- **Hardware malfunction or failure**. Distance measurements will be meaningless when encountering ranging hardware malfunction. Besides, incorrect hardware calibration and configuration also deteriorate ranging accuracy, which is not much emphasized by previous studies. For example, RSS suffers from transmitter, receiver, and antenna variability, and the inaccuracy of clock synchronization results in ranging errors for TDoA.
- **Environment factors**. RSS is sensitive to channel noise, interference, and reflection, all of which have significant impact on signal amplitude. The irregularity of signal attenuation remarkably increases, especially in complex indoor environments. In addition, for the propagation time based ranging measurements, e.g., TDoA, the signal propagation speed often exhibits variability as a function of temperature and humidity, so we cannot assume that the propagation speed is a constant across a large field.
- **Adversary attacks**. As location-based services are getting prevalent, the localization infrastructure is becoming the target of adversary attacks. By reporting fake location or ranging results, an attacker, e.g., a compromise (malicious) node, can completely distort the coordinate system. Different from the previous cases, the large errors here are intentionally generated by adversaries.

These severe errors can be seen as outliers of measurements. We classify the outlier-resistant approaches into two major categories: explicitly sifting and implicitly de-emphasizing. The explicitly sifting methods are usually based on the intuition that normal ranging measurements are compatible while an outlier is likely to be inconsistent with other normal and outlier rangings. By examining the inconsistency, we can identify and reject outlier measurements. In contrast, the implicitly de-emphasizing methods do not accept or reject a localization result by fixing a threshold, but employ robust statistics methods, for example, high breakdown point estimators and influence functions, to mitigate the negative effects of outliers.

6.5.1 Explicitly Sifting

The basic idea of outlier sifting is that the redundancy of geometric constraints can, to some extent, reveal the inconsistency of outlier ranging and normal ranging. Suppose m location references locating at p_i, $i = 1,2,\ldots,m$, are used to locate a target node by multilateration. Liu et al. [82] uses the mean square error ς^2 of the distance measurements as an indicator of inconsistency, i.e.,

$$\varsigma^2 = \frac{1}{m}\sum_{i=1}^{m}(\delta_i - |\bar{p}_0 - p_i|)^2, \tag{6.8}$$

where δ_i is the measured distance to the i^{th} reference and $\bar{p}_0$ is the estimated location of the target node. A threshold-based approach is proposed to determine whether a set of location references is consistent. Formally, a set of location references $L = \{(p_i,\ \delta_i),\ i = 1,2,\ldots,m\}$ obtained at a sensor node is τ-consistent if the corresponding mean square error ς^2 satisfies $\varsigma^2 \le \tau^2$.

Apparently, the threshold τ has significant impact on localization performance. The determination of τ depends on the measurement error model, which is assumed to be available. Based on the measurement error model, an appropriate τ is determined by performing simulation off-line. This threshold is stored at each senor node. In general, when the error model changes frequently and significantly, the fixed value of τ would degrade the performance. For simplicity, Liu et al. [82] assume the measurement error model will not change.

Given a set L of n location references and a threshold τ, it is desirable to compute the largest set of τ-consistent location references, because LS-based methods can deal with measurement errors better if there are more normal ranging results. The naive approach is to check all subsets of L with i location references about τ-consistency, where i starts from n and decreases until a subset of L is found to be τ-consistent or it is not possible to find such a set. Suppose the largest set of consistent location references consists of m elements. Then the sensor node has to perform LS-based localization at least $1 + \binom{n}{m+1} + \binom{n}{m+2} + \cdots + \binom{n}{n}$

times to figure out the right one. Although such an approach can provide the optimal result, it requires a large amount of computation when n and m are large numbers, which sometimes is unacceptable for resource constrained sensor nodes. To address this issue, Liu et al. [82] adopt a greedy algorithm, which is efficient but suboptimal. The greedy algorithm works iteratively. It starts with the set of all available location references. In each iteration, it checks whether the current set of location references is τ-consistent. If positive, the algorithm outputs the estimated location and stops. Otherwise, it considers all subsets of location references with one fewer location reference, and chooses the subset with the minimum mean square error as the input to the next iteration. Similar to the brute-force algorithm aforementioned, the greedy algorithm continues until it finds a set of τ-consistent location references or when it is not possible to find such a set. In general, through the greedy algorithm, the sensor node needs to perform LS-based localization for at most $1+n+(n-1)+\cdots+4$ times, which is much better than the brute-force algorithm.

Another way is to handle phantom nodes that claim fake locations. Hwang et al. [83] propose a speculative procedure, which can effectively and efficiently filter out phantom nodes. The filtering procedure is illustrated in Algorithm 6.2, where $Nbr(v)$ is the node set consisting of v and its neighbors, and E is used to keep consistent edges. G is initially empty. After computing the locations of all neighbors in the local coordinate system L by trilateration, for any two neighboring nodes j and k, if the difference between the measured distance and the computed distance is less than a threshold ε, the edge $e(j, k)$ is inserted into E. The threshold value ε depends on the noise in the ranging measurement. The largest connected cluster is regards as the largest consistent subset in the speculative plane L. This filter is done

Algorithm 6.2 Speculative filtering

```
for $i = 0$ to $iter$ do
    node $v$ randomly selects two neighbors $u$ and $w$
    create local coordinate system $L$ using $v$, $u$, $w$ and their inter-distances $\hat{d}_{vu}$, $\hat{d}_{vw}$ and $\hat{d}_{uw}$
    initialize undirected graph $G(V, E)$
    create nodes $v$, $u$, $w$ with locations $p_v$, $p_u$, $p_w$ in $V$, respectively
    for each node $k \in Nbr(v)$ do
        calculate the location of $k$, $p_k$, in $L$ by trilateration using $p_v$, $p_u$, $p_w$ and $\hat{d}_{kv}$, $\hat{d}_{ku}$, $\hat{d}_{kw}$
        create node $k$ with location $p_k$ in $V$
    end for
    for each pair of nodes $j, k \in V$ and the distance $\hat{d}_{jk}$ do
        if $\left|\hat{d}_{jk} - |p_j - p_k|\right| < \varepsilon$ then
            create edge $e(j, k)$ in $E$
        end if
    end for
    find the largest connected cluster $C$ and save it
end for
choose the one with the largest size among all saved $C$
```

iter times, where *iter* is determined by the application requirement, and the cluster with the largest size is chosen as the final result. The theoretical foundations of this strategy are the following two arguments:

- If the three pivots are honest nodes, the cluster output by Algorithm 6.2 contains no phantom nodes.
- If one pivot is a phantom node, the size of largest cluster is smaller than the one when none of pivots is a phantom node.

Departing from the two works previously discussed, which focus on security scenarios, in a recent work [84], Jian et al. propose a more general framework of sifting noisy and outlier distance measurements for localization. They formally define the problem of outlier detection for localization, and build the theoretical foundations based on graph embeddability and rigidity. Accordingly, an outlier detection algorithm is designed based on bilateration and generic cycles. Their results suggest the algorithm significantly improves the localization accuracy by wisely rejecting outliers. We discuss this work more detailed here.

Based on the grounded graphs associated with network instances, Jian et al. formulate normal ranging results and outlier ranging results as normal edges and outlier edges, respectively. In their error model of distance ranging, normal edges contain no ranging noise, while the measured distance of an outlier edge is an arbitrary continuous random variable. They argue that this assumption and abstraction is a good starting point to address the outlier detection problem. Through introducing an error threshold, their proposed algorithm can handle a more practical error model, where normal edges are with moderate ranging errors. Based on the normal edge and outlier edge model, the definition of outlier detection is straightforward: given a weighted graph $G = <V, E, W>$ consisting of normal and outlier edges, identify those outlier edges in G.

The theoretical foundations are built based on graph embeddability and rigidity. The first result provided by Jian et al. is:

Theorem 6.1. *Given a weighted grounded graph $G =<V, E, W>$, if G is unembeddable, the E contains at least one outlier edge.*

This result is intuitive: if G contains no outlier, then the ground truth is an embedding, and G cannot be unembeddable. Nevertheless, this is the best we can do for detecting outliers only based on ranging information. Formally, if G is embeddable, even G actually contains some outliers, we have no way to detect them.

Theorem 6.1 only provides a fine-granularity way to detect outliers, but cannot tell which are outliers and which are not. To address this issue, a concept of outlier disprovable is proposed:

Definition 6.1. *Given a weighted graph $G = <V, E, W>$, G is outlier disprovable if and only if the embeddability of G implies that it contains no outlier edge.*

Jian et al. prove the second result:

Algorithm 6.3 Outlier Detection Algorithm

```
for all redundantly rigid component $H$ in $G$ do
    if $H$ is embeddable then
        mark every edge $e \in H$ a normal edge
    else
        for all edge $e \in H$ not marked a normal edge do
            mark $e$ an outlier edge
        end for
    end if
end for
```

Theorem 6.2. *Given a weighted graph G, G is outlier disprovable if and only if G is redundantly rigid.*

A graph is rigid if it has no continuous deformation other than global rotation, translation and reflection while preserving distance constraints; otherwise, it is flexible. A graph is called redundantly rigid if it remains rigid after removing any single edge. Based on these two results, an outlier detection algorithm is designed as Algorithm 6.3. Different from [82] and [83], which are based on quadrilateral structures and require dense networks, Algorithm 6.3 pays more attention to exploring and utilizing the redundantly rigid topological structures, and thus, works properly in networks with moderate connectivity.

Algorithm 6.3 addresses the problem of outlier detection and identification theoretically. However, in practice, it suffers from the computational prohibitiveness, termed as combinational explosion, which is implied by the following result.

Theorem 6.3. *Let $G_1 = <V_1, E_1>$ and $G_2 = <V_2, E_2>$ be two redundantly rigid graphs with $|V_1 \cap V_2| \geq 2$. Then $G_1 \cup G_2$ is redundantly rigid.*

Suppose we have checked the embeddability of G_1 and G_2, both of which are redundantly rigid. According to Algorithm 6.3, we still need to check the embeddability of $G_1 \cup G_2$, which is actually implied by the checking results of G_1 and G_2 if $|V_1 \cap V_2| \geq 2$. To tackle this issue, Jian et al. introduce the concept of *generic cycle*, which is the minimally redundant rigidity.

Definition 6.2. *A graph $G = <V, E>$ with $|V| \geq 4$ is called a generic cycle if $|E| = 2|V| - 2$ and G satisfies*

$$i(X) \leq 2|X| - 3 \ \ \textit{for all} \ \ X \subset V \ \ \textit{with} \ \ 2 \leq |X| \leq |V| - 1$$

where $i(X)$ denotes the number of edges induced by X in G.

Based on generic cycles, another outlier detection algorithm is proposed, outlined by Algorithm 6.4, which avoids the computational prohibitiveness.

Algorithm 6.4 Edget-based Outlier Detection Algorithm.

```
for all $e$ not marked in $G$ do
    if there is a generic cycle $H$ containing $e$ is embeddable then
        mark every edge $\in H$, including e, a normal edge
    else
        mark $e$ an outlier edge
    end if
end for
```

6.5.2 *Implicitly De-emphasizing*

(a) **What is Robust Statistics?**

In the literature of statistics, classical methods, e.g., mean and least squares, rely heavily on some idealized assumptions about input data sets, which are often not met in practice. Particularly, it is often assumed that the data residuals, i.e., the difference between the computed value and the input value, are normally distributed, or at least approximately. However, when there are outliers in the input data set, these methods often show very poor performance. Robust statistics is a theoretical framework concerning the outlier rejection problem, which provides alterative approaches to classical statistical methods in order to produce estimators that are not unduly affected by outliers. Two of the most common measures of robustness are *breakdown point* and *influence function*.

The breakdown point of an estimator is the fraction of data that can be given arbitrarily large values without giving an arbitrarily large result. For instance, it is obvious from the formula of the mean estimator, $\frac{1}{n}(x_1 + x_2 + +x_n)$, that if we hold x_1, x_2, x_{n-1} fixed and let x_n approach infinity, the statistic result also goes to infinity. In short, even one gross outlier can ruin the result of the mean estimator. Thus, such an estimator has a breakdown point of 0. In contrast, the median estimator can still give out a reasonable result when half of data goes to infinity. Accordingly, the breakdown point of the median estimator is 50%. The higher the breakdown point of an estimator, the more robust it is. Clearly, 50% is the theoretically highest breakdown value that can be achieved by an estimator, because if more than half of data is contaminated, it is impossible to distinguish the underlying distribution from the contaminating distribution.

Other than the breakdown point, the influence function is used to characterize the importance of individual data samples. A smaller absolute value of the influence function means the data item receives less weight in the estimation. The influence function is proportional to the derivative of the estimator. A robust estimator should have a bounded influence function, which does not go to infinity when the data value becomes arbitrarily large.

(b) **Robust Statistics Based Localization**

According to robust statistics [85], the least squares algorithm is sensitive to outliers, since its breakdown point is zero. One of the most commonly used robust fitting algorithms is the method of least median of squares (LMS), introduced by Rousseeuw et al. [85], which is adopted in [86] to design a robust localization algorithm. Instead of minimizing the summation of the residue squares, LMS minimizes the median of the residue squares, i.e., it estimates the location using

$$\bar{p}_0 = \arg\min_{p_0} med_i(\delta_i - ||p_0 - p_i||)^2, \tag{6.9}$$

where the parameters are defined in Section 6.5.1. In contrast to the least squares method, in which a single influential outlier may destroy the estimation, a single outlier has little effect on the objective function of LMS, and will not bias the estimate significantly. Results from [85] show that LMS has a breakdown point of 50%; in other words, LMS tolerates up to 50% outliers among all measurements and still outputs the correct estimate.

It is computationally prohibitive to get the exact solution of LMS. Rousseeuw et al. proposed an efficient and statistically robust alterative. First, using LS solution, we compute several candidate estimations according to random subsets of samples. The median of the residue squares for each candidate is then computed, and the one with the least median of residue squares is chosen as a tentative estimate. However, this tentative estimate is computed based on a small subset of data samples. As discussed previously, a better estimation can be achieved if more normal ranging results are included. To address this issue, the samples are weighted based on their residue for the tentative estimate, followed by a weighted least square fitting to get the final estimate. A simple threshold-based weighting strategy is as follows:

$$w_i = \begin{cases} 1, & \left|\frac{r_i}{s_0}\right| \le \gamma; \\ 0, & \text{otherwise.} \end{cases} \tag{6.10}$$

where γ is a predetermined threshold, r_i is the residue of the i-th sample for the least median subset estimate $\bar{p}_0$, and s_o is the scale estimate given by [85] for the two dimensional estimated variable $\bar{p}_0$,

$$s_0 = 1.4826(1 + \tfrac{5}{n-2})\sqrt{\text{med}_i r_i^2(\bar{p}_0)}, \tag{6.11}$$

where n is the number of available samples. The term $(1 + \frac{5}{n-2})$ is used to compensate the tendency for a small scale estimate when there are few samples.

In summary, the LMS-based robust localization algorithm has the following steps:

- **Parameters selection**. Suppose n references are available. Choose an appropriate subset size m, the total numbers M of subsets, and a threshold γ.
- **Subsets generation**. Randomly draw M subsets of size m from the data set. Find the estimate $\bar{p}_0$ (using LS solution) for each subset. For each $\bar{p}_0$, calculate the median of residues r_i of each reference, $i = 1, 2, \ldots, n$.

- **Least median calculation**. Calculate $\bar{p}_0 = \arg\min_{p_0} med_i(\delta_i - |p_0 - p_i|)^2$ and residues with respect to $\bar{p}_0$, $r_i(\bar{p}_0)$.
- **Weights assignment**. Calculate s_0 based on Eq. (6.11) and assign weight w_i to each sample using Eq. (6.10).
- **Location estimation**. Do a weighted least squares fitting to all data to get the final location estimate.

The basic idea of LMS implementation is that, at least one subset among all randomly drawn subsets does not contain any outlier, and the estimate from this good subset will fit the inliers well. The chosen values of m and M have significant impact on the probability of such solution. The probability P of getting at least one good subset without outlier is calculated as follows. Assuming the contamination rate is ε, then

$$P = 1 - (1 - (1 - \varepsilon)^m)^M. \tag{6.12}$$

Given $m = 4$ and $M = 20$, the LMS algorithm is resistant up to 30% contamination with $P \geq 0.99$.

Inspired by robust statistics, the recent work SISR [87], from the perspective of influence function, analyzes the non-robustness of LS-based methods to outliers, and uses a residual shaping influence function to de-emphasize the “bad nodes” and “bad links” during the localization procedure.

The motivation of SISR is illustrated in Fig. 6.8, where different localization schemes would lead to different solutions. It is more desirable to get the uneven solution rather than the even solution, because E cannot be localized accurately in any case, given that it has large measurement error. Furthermore, since A, B, C and D could potentially be localized with great accuracy, a localization method that returns

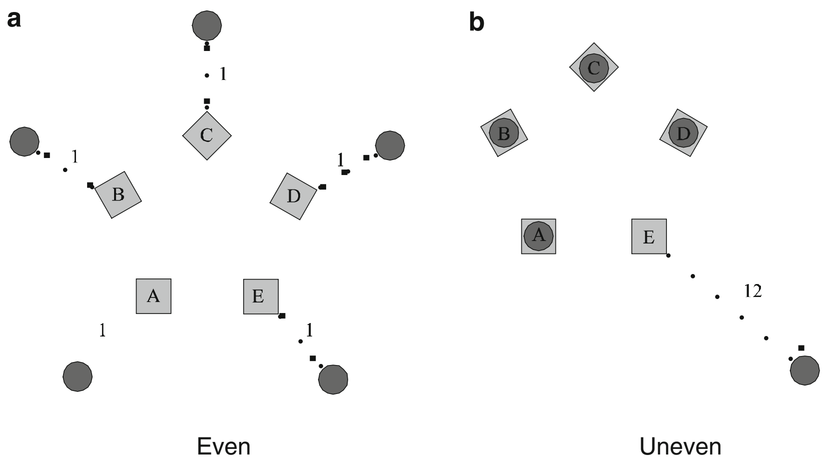


Fig. 6.8 Two possible solutions of nodes A, B, C, D and E, where E has large measurement errors. Squares indicate the ground truth locations; and circles the computed localization solutions. **(a)** The measurement errors from E is amortized over A, B, C and D. **(b)** Solutions for A, B, C and D are accurate, but that for E is very inaccurate

the uneven solution ought to de-emphasize the measurements of E to avoid contaminating the localization results of A, B, C and D. The conventional least squares method would find the even solution, as it does not distinguish between normal and outlier ranging measurements. To overcome this issue, SISR makes a key modification to the conventional least squares method: the residual function is shaped. As discussed previously, the influence function is proportional to the derivative of the estimator; in particular, here the estimator is the residual function. Thus, by shaping the residual function, the influence function is accordingly shaped in order to dampen the impact of outlier measurements and emphasize normal measurements. This modification make SISR find the uneven solution. The implementation of SISR is as follows.

Instead of optimizing the sum of squared residues, i.e., $F = \sum_{i,j} r^2(i,j)$, where $r(i,j)$ is the residue corresponding to edge (i, j), SISR solves the optimization problem of $F = \sum_{i,j} s(i,j)$, where

$$s(i,j) = \begin{cases} \alpha r(i,j)^2, & \text{if } |r(i,j)| < \tau \\ \ln(|r(i,j)| - u) - v, & \text{otherwise} \end{cases}, \tag{6.13}$$

where α, τ, u and v are parameters to be configured.

Figure 6.9 sketches the residual function of SISR, which has the following two properties:

- The shaping function increases with a smaller slope when the residual is large; in other words, the influence function of a measurement with large residual is smaller. In particular, the function dampens the impact of residuals larger than a threshold τ. This is called the wing-shaped section.
- The shaping function has a narrow and deep well for residuals close to 0. The normal measurements can therefore be emphasized by growing the shaped residuals more rapidly. This is called the U-shaped section.

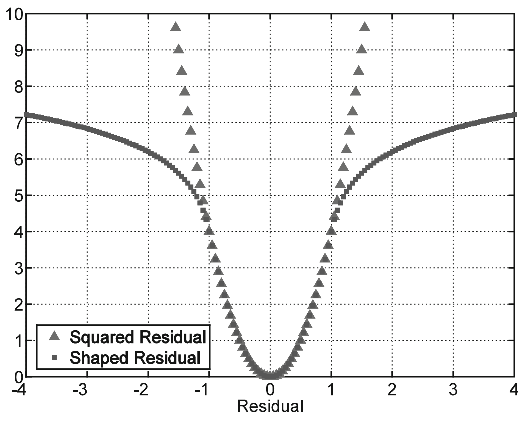


Fig. 6.9 Comparison between the standard squared residual used in conventional least squares and the shaped residuals used in SISR with $\alpha = 4$ and $\tau = 1$

The α and τ are two parameters that tune the shape of the SISR function in order to control its sensitivity to errors. The α is used to control the height of the U-shaped section, while τ controls its width. On the other hand, minimizing $F = \sum_{i,j} s(i,j)$ is a nonlinear optimization problem, and usually involves some iterative searching techniques, such as gradient descent or Newton method, to get the solution. Thus, it is necessary to make the SISR function piecewise-continuous and piecewise-differential at τ (apparently, it is piecewise-continuous and piecewise-differential at any other point). Accordingly, the other parameters u and v can be solved by

$$u = \tau - \frac{1}{2\alpha\tau}, \tag{6.14}$$

$$v = \ln\left(\frac{1}{2\alpha\tau}\right) - \alpha\tau^2. \tag{6.15}$$

Kung et al. [87] suggest that the value of τ has significant impact on the performance of the SISR estimator. On one hand, a small τ leads to more accurate localization results, while increases the probability of falling into an incorrect local minimum. On the other hand, a more permissive τ reduces this probability at the expense of localization accuracy. When τ approaches to infinity, SISR is actually reduced to the conventional least squares method. They design an iterative refinement scheme to exploit the above trade-off, and the proposed scheme works well in practice [87].

6.6 Summary

Although more ranging techniques are developed, noises and outliers are inevitable in distance measurements. Numerous simulations and experimental studies have suggested that ranging error can degrade the performances of many localization algorithms drastically. How to handle noises and outliers is essential for a wide range of location-based services. In this chapter, we review the measurement accuracies of different ranging techniques and how errors in distance measurements affect the localization results from four aspects: uncertainty, non-consistency, ambiguity and error propagation. We discuss the state-of-the-art works on characteristics of localization error, elimination of location ambiguities, location refinement schemes and outlier-resistant localization.

With no noise at all, the localization issue with distance information in dense networks, e.g., quadrilateral networks or trilateration networks, is trivial. However, in the presence of even a small amount of noise, for complete networks (graphs), localization is hard [88]. One promising research direction in this area is to adopt multimodal measurements, distance and angle information. Besides reducing the computational complexity of localization, multimodal measurements can provide more robustness to noises.