The Distributed Construction of a Global Coordinate System in a Network of Static Computational Nodes from Inter-Node Distances [*]

Lambert Meertens & Stephen Fitzpatrick

March 2004

Kestrel Institute Technical Report KES.U.04.04

Kestrel Institute, 3260 Hillview Avenue,
Palo Alto, CA 94304, USA
meertens@kestrel.edu & fitzpatrick@kestrel.edu

Abstract

This report details how each node in a network of computational devices can construct a spatial map of its own position and those of other, nearby nodes based on inter-node distance information; and how each node can align and reconcile its own map with those of nearby nodes; with the result that the collection of maps forms a consistent, global coordinate system.

[*]This work is sponsored in part by DARPA through the 'Network Embedded Systems Technology' program under contract #F30602-01-C-0123, monitored by the Air Force Research Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

1 Introduction

The problem that is addressed in this report is:

- given a collection of communicating, uniquely identified, computational nodes located at fixed positions in the (Euclidean) plane
- in which each node has estimates of some of the distances between itself and its neighbours (i.e., nearby nodes) and between some pairs of its neighbours,
- each node is to construct a spatial map showing its own position and the positions of some of its neighbours
- such that the maps are approximately consistent (i.e., all of the positions for a given node are to be approximately the same).

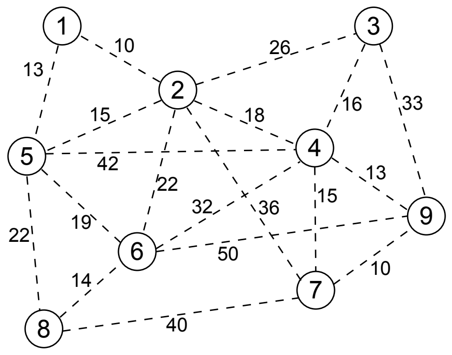


Figure 1: Distances between nodes

For example, Figure 1 shows a collection of nodes and estimates of distances between some pairs of those nodes. An individual node may acquire some subset of the distance estimates, as illustrated in Figure 2 for node ②.

Each node determines a map of its own neighbourhood and where maps overlap, they are required to agree. For example, Figure 3 shows the areas covered by maps for nodes ② and ④; they overlap on nodes ②, ③ and ⑥ and so are required to (approximately) have the same positions for each of these nodes.

The chosen approach to solving this problem has two main elements.

	①	②	③	⑤
②	10			
③	—	26		
⑤	13	15	—	
⑥	—	22	—	19

Figure 2: Distance matrix for node ②

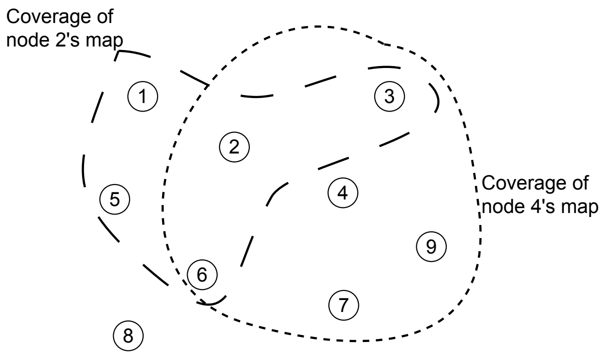


Figure 3: Overlapping local maps

- Local map formation: Each node uses its distance information to construct a map of its neighbourhood. For any given node, its map can only be determined up to a translation, rotation and reflection (unless other sources of information are used to fix the locations of some nodes).
- Alignment of local maps: For any given pair of maps that have at least three non-collinear nodes in common, the positions of the nodes that appear in both maps can be used to calculate an isometric coordinate transformation (a translation + rotation + optional reflection) that approximately aligns the maps. Such calculations can be used in an anytime, peer-to-peer process in which each node continually tries to align its own map with its neighbours'.

This report is organized as follows. First, the construction of a local map from distance information is discussed. Then the calculations required to align two overlapping maps are detailed. Then the peer-to-peer alignment process is discussed. Finally, two methods for trying to improve the speed of convergence of the peer-to-peer alignment process are considered.

2 Map Formation from Distance Information

This report assumes that nodes have some means to acquire appropriate inter-node distance information. The distance information may be subject to measurement errors. Such information may be thought of as a *distance graph*, i.e., an undirected graph whose (graph) nodes are labeled with node identifiers, and whose edges are labeled with distances (nonnegative real values), as in Figure 1. Nodes connected by a single edge are called *neighbours*.

A spatial map and a distance graph are (approximately) *consistent* when inter-node distances computed from the positions of the nodes recorded in the map are (approximately) the same as the inter-node distances recorded in the distance graph.

A distance graph is *rigid* when — up to isometry: translation, rotation and reflection — there is a unique spatial map that is optimally consistent with the distance graph.

Spatial maps should only be constructed for rigid distance graphs; in the case of a non-rigid graph, a rigid subgraph can be used.

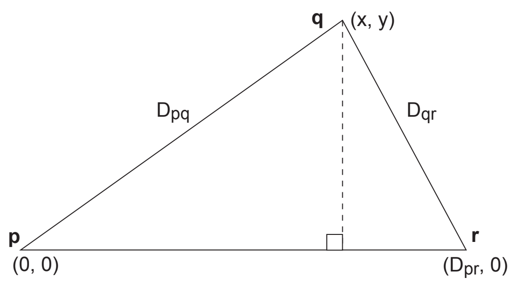
The distance information in a distance graph can be represented as a *distance matrix* mapping pairs of neighbouring nodes to distances, as in Figure 2. A distance matrix is thus a *partial* map from pairs of node identifiers to distances. The diagonal of a distance matrix cannot carry useful information — the distance of a node to itself is 0 — and is not used. It is further assumed that each distance matrix D is *symmetric* ($D_{pq} = D_{qp}$), which is enforced by defining the *domain* of a distance matrix D — denoted $\mathrm{dom}(D)$ — to be a set of *unordered* pairs $\{p, q\}$, $p{\neq}q$, of node identifiers. (Assuming some arbitrary but fixed linear ordering on the set of node identifiers, the elements of $\mathrm{dom}(D)$ can equivalently be viewed as ordered pairs of node identifiers (p, q) with $p < q$.) For example, taking D to be the distance matrix of Figure 2, $|\mathrm{dom}(D)| \;=\; 6$.

The objective is to compute a *Spatial Map* $M : Id \rightarrow \mathrm{Pos}$ which maps node identifiers to spatial positions. (Although it is assumed that space is two-dimensional, most of the calculations, algorithms and processes should readily extend to three dimensions.)

The notions of neighbour, consistency and rigidity are now reformulated in terms of distance matrices.

A node with identifier p is a neighbour of a node with identifier q in distance matrix D if and only if $\{p, q\} \in \mathrm{dom}(D)$.

A spatial map and a distance matrix are (approximately) *consistent* when the discrepancy between the inter-node distances computed from the positions of the nodes recorded in the



$$x^2 + y^2 = D_{pq}^2, \quad (D_{pr} - x)^2 + y^2 = D_{qr}^2$$

Figure 4: Initial map

map and the inter-node distances recorded in the distance matrix is low. The discrepancy can be quantified as, say, the root-mean-square of the differences between distances on the map and in the matrix:

$$\mathrm{error}(D, M) \hat{=} \sqrt{\frac{\sum_{\{p,q\} \in \mathrm{dom}(D)} (D_{pq} - |M_p - M_q|)^2}{|\mathrm{dom}(D)|}}$$

where subtraction of spatial positions is, of course, vector subtraction.

A distance matrix is *rigid* when — up to isometry: translation, rotation and reflection — there is a unique spatial map for which the discrepancy between the map and the distance matrix is minimal.

2.1 Map Formation Heuristic

Consider now the process of computing a spatial map from a distance matrix. The first step is to construct an initial map containing a triangle of three non-collinear pair-wise neighbouring nodes. Then more nodes are inserted into the map, one at a time, based on distances to nodes already in the map, in an iterative process. For a node to be inserted into a map, the node must have at least three non-collinear neighbour nodes. The process terminates when all nodes have been inserted into the map or when no uninserted node can be inserted.

The choice of the initial triangle may have a significant impact on the number of nodes that can be inserted. A heuristic is used that searches over triples of nodes that are pair-wise

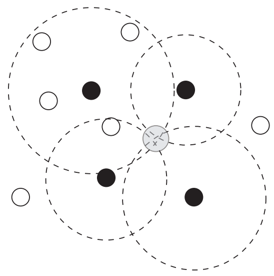


Figure 5: Inserting a node into a map: black nodes are reference points, dashed circles indicate distances from the reference points and the gray node indicates the position at which the new node will be inserted

neighbours, ruling out those that are nearly collinear[1] or too close together. Given a choice of three nodes, the initial map is computed as follows. (See Figure 4).

- Find the longest side (or one of them if there are several longest) and denote its end nodes as p and r. Align this side with the x-axis by setting p's position to the origin $(0, 0)$ and r's position to $(D_{pr}, 0)$.
- Let the third node be q. It is assigned the position (x, y) where, by simple geometry,

$$x = (D_{pq}^2 + D_{pr}^2 - D_{qr}^2)/2D_{pr}, \quad y = \sqrt{D_{pq}^2 - x^2} \quad .$$

Of course, there is an arbitrariness to these choices of positions — any translation, rotation or reflection of the whole map would do equally well.

Now consider the process by which additional nodes are inserted into an existing map. Nodes are inserted iteratively, one at a time. At each iteration, a node with the highest number of neighbours already in the map is chosen for insertion, with the process stopping when no remaining unmapped node can be found with at least three mapped neighbours that are non-collinear.

Figure 5 illustrates the multilateration computation used to determine the new node's position, based on its distance from nodes in the map that are neighbours. Let this set of neighbour nodes of the new node n be called the node's *reference set* R. Each node r in the

[1]Collinearity is, strictly speaking, a property of nodes in spatial maps, while at this stage only distance information is available. Instead, the geometrically equivalent requirement is used that the longest distance of the three is less then the sum of the other two. For robustness, the difference must be at least some number (e.g., 3.5) times the expected distance-measurement error.

Compute the following terms:

$$\begin{aligned} u &= \textstyle\sum_{pq} a_{pq}^2 \\ v &= \textstyle\sum_{pq} a_{pq} b_{pq} \\ w &= \textstyle\sum_{pq} b_{pq}^2 \\ g &= \textstyle\sum_{pq} a_{pq} c_{pq} \\ h &= \textstyle\sum_{pq} b_{pq} c_{pq} \end{aligned}$$

where each summation is over $p, q \in R \wedge p < q$ and

$$\begin{aligned} a_{pq} &= -2(x_p - x_q) \\ b_{pq} &= -2(y_p - y_q) \\ c_{pq} &= D_{pn}^2 - D_{qn}^2 - x_p^2 + x_q^2 - y_p^2 + y_q^2 \end{aligned}$$

Let $d = uw - v^2$. Then the new node's position is (x, y) where

$$x = (gw - hv)/d, \quad y = (hu - gv)/d \quad .$$

Figure 6: Computing the position of a node as a least-squares solution to a set of simultaneous equations arising from multilateration

reference set gives one quadratic equation in a set of simultaneous equations determining the new node's position (x, y):

$$(x - M_r.x)^2 + (y - M_r.y)^2 = D_{nr}^2$$

where $M_r.x$ indicates the x component of r's position in map M (and likewise for the y component).

Provided there are at least three reference nodes, the set of simultaneous equations can be solved for x and y. However, because there may be noise in the distance estimates, the equations may not have an exact solution. Instead, a least-squares solution may be computed. First, from each pair of quadratic equations for distinct reference nodes r and r' a linear equation in the unknowns x and y is obtained by subtraction[2]:

$$\{(x - M_r.x)^2 + (y - M_r.y)^2\} - \{(x - M_{r'}.x)^2 + (y - M_{r'}.y)^2\} = D_{nr}^2 - D_{nr'}^2 \quad .$$

[2]That this equation is linear may not be immediately apparent from the given form, but after expansion the squares of the unknown variables cancel.

In the non-degenerate case, the new set of linear equations is satisfied if and only if the original set of quadratic equations is. The least-squares solution for the new set is now computed as shown in Figure 6.

In the worst case, to place N nodes, this computation has time complexity $\Theta(N^3)$, but a more realistic measure — using the observation that the degree of the graph tends to be bounded — is $\Theta(N^2)$. In a practical implementation, the distance matrices and maps are confined to a small set of near nodes to preserve scarce memory on the actual computational nodes, and the time complexity is not an issue.

2.2 Comment on Completeness

The local map formation process may terminate with some unmapped nodes even when the distance matrix is rigid, for two reasons:

- Nodes are inserted into an existing map one at a time. Some distance matrices may require the insertion of multiple nodes simultaneously in order for all nodes to be inserted.
- At most, only one initial triangle is considered for map construction. The chosen triangle may fail to allow all nodes to be inserted in cases where a different initial triangle would succeed.

Given the real-time constraints under which the process would typically operate, it seems better to accept occasional failures and to compensate for them using maps received from neighbours.

3 Map Alignment

This section considers the problem of reconciling two maps that have at least some nodes in common but that differ on the positions of those common nodes. There are three main factors that can contribute to such discrepancies:

1. Alignment: Since maps computed solely from inter-node distances cannot have an absolute sense of position, direction or orientation, two maps may use different coordinate systems, and yet (restricted to their common nodes) be congruent in that they differ

only by a combination of a translation, rotation and optional reflection. A transformation from one map to the other can be computed provided that they have at least three non-collinear nodes in common.

2. Different Information Bases: The two maps may have been computed based on slightly different distance information. For example, the distance between any given pair of nodes may be estimated several times to reduce the effect of measurement error. Because communication may be unreliable, each estimate may be separately disseminated through a network, resulting in different sets of estimates arriving at different nodes, which then will subsequently compute different averages. Map discrepancies arising from such estimate differences should be small and can usually be handled by averaging the positions for a given node in the two maps. See however the next case.

3. Non-Rigidity: If a distance matrix is insufficiently rigid, several substantively different possible maps are consistent with the distance matrix. Therefore, map formation requires that the distance matrix is rigid. However, measurement error can make a non-rigid matrix appear rigid. When considering the distributed case, it is possible that different nodes acquire overlapping small fragments of the overall distance matrix, and that one or more of these fragments are not very rigid, but are made to appear rigid by measurement error. In combination with a different information base, these nodes may compute different, incompatible maps. Unfortunately, such errors are hard to correct without collecting a lot of information (to perform outlier rejection, for example). Repeated averaging may still remedy this, though.

Moreover, many physical processes that may be used for distance estimation are subject to *multipath* effects. For example, the distance between two nodes may be estimated from the time required for an acoustic signal to travel between them. In the simplest formulation, it is assumed that the estimate corresponds to the straight-line distance between the nodes, but if the signal is reflected then this assumption is no longer valid. Multipath effects are not considered in this report.

This section focuses on the map alignment problem. Specifically, given two spatial maps L and M, an isometric coordinate transformation is to be computed such that the discrepancy between positions in L and M is minimized, where the discrepancy is defined as the sum of the differences in positions of nodes that appear in both maps:

$$\text{discrepancy}(L, M) \hat{=} \sum_{p \in C} |L_p - M_p|^2$$

where $C \hat{=} \text{dom}(L) \cap \text{dom}(M)$ is the set of nodes that appear in both maps.

There are five steps in aligning M with L (Figures 7 to 10 illustrate each step except reflection):

1. Determine Common Nodes: The set of nodes that appear in both maps is determined. Let C denote this set and let L_C and M_C denote the *restriction* of maps L and M to C (i.e., any nodes not in C are removed).
2. Translation: The 'centre of gravity' (CoG) of the nodes in C is computed from their positions in L_C; denote this as $\overline{L_C}$. (See below for an equation defining CoG.) Likewise, the CoG of C in M_C is computed and denoted as $\overline{M_C}$. Then M is translated by the vector displacement $\overline{L_C} - \overline{M_C}$ so that the CoGs are aligned, at $\overline{L_C}$.
3. Rotation Without Reflection: Then a rotation about $\overline{L_C}$ is determined that minimizes the discrepancy with L. The residual discrepancy with L under this optimal rotation is also computed.
4. Rotation With Reflection: Similarly, an optimal rotation is computed after first reflecting the translated M in an axis through $\overline{L_C}$.
5. Choice of With or Without Reflection: A choice is then made for rotation with or without reflection, depending on which of steps 3 or 4 produced the lowest residual discrepancy.

Symbolically, the task is to find ϑ and φ that respectively minimize ϵ^ϑ and ϵ^φ where

$$\begin{array}{rcl}
\epsilon^\vartheta & = & \mathrm{discrepancy}(L, M^\vartheta) \\
\epsilon^\varphi & = & \mathrm{discrepancy}(L, M^\varphi) \\
M^\vartheta & = & \mathrm{rotate}(\overline{L_C}, \vartheta, M^T) \\
M^\varphi & = & \mathrm{rotate}(\overline{L_C}, \varphi, M^F) \\
M^F & = & \mathrm{reflect}_x(\overline{L_C}, M^T) \\
M^T & = & \mathrm{translate}(\overline{L_C} - \overline{M_C}, M) \\
\overline{M} & \hat{=} & \sum_{i \in \mathrm{dom}(M)} M_i / |\mathrm{dom}(M)|
\end{array}$$

where $\mathrm{rotate}(p, \alpha, M)$ rotates map M about point p through an angle α, $\mathrm{reflect}_x(p, M)$ reflects map M in an axis that passes through point p and is parallel to the y-axis, and $\mathrm{translate}(\delta, M)$ translates map M through the vector displacement δ.

According to which of ϵ^ϑ and ϵ^φ is the smaller, M^ϑ or M^φ is chosen as the optimal alignment of M to L.

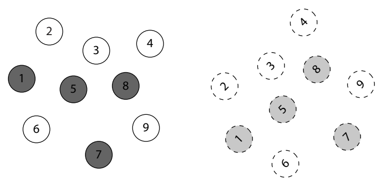


Figure 7: Unaligned maps

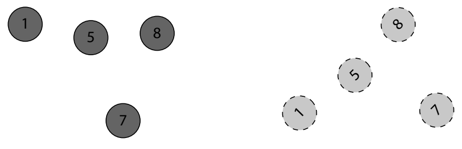


Figure 8: Common nodes identified

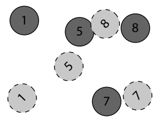
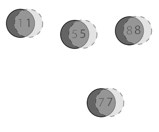


Figure 9: Centres of gravity aligned



N.B. Maps are shown slightly displaced for presentation

Figure 10: Optimal rotation

3.1 Computing an Optimal Rotation

Given two maps L and M with a non-empty set of common nodes C, a rotation of M about any point G can be computed that minimizes the discrepancy between L_C and the rotated M_C. (In degenerate cases, there may be more than one optimal rotation.) For the sake of simplicity, we write L for L_C and M for M_C in the rest of this subsection, so then $\mathrm{dom}(L) = \mathrm{dom}(M) = C$.

From the definition of discrepancy, it is easy to see that it is invariant under translation:

$$\mathrm{discrepancy}(\mathrm{translate}(L, \delta), \mathrm{translate}(M, \delta)) = \mathrm{discrepancy}(L, M) \quad .$$

A rotation about a point p can be realized as a sequence of three transformations: a translation by vector $-p$ so that p is the origin; then a rotation about the origin; then a translation by p:

$$\mathrm{rotate}(p, \alpha, M) = \mathrm{translate}(p, \mathrm{rotate}((0,0), \alpha, \mathrm{translate}(-p, M)) \quad .$$

So

$$\begin{aligned} & \mathrm{discrepancy}(\mathrm{rotate}(p, \alpha, M)), L) \\ = \; & \mathrm{discrepancy}(\mathrm{translate}(p, \mathrm{rotate}((0,0), \alpha, \mathrm{translate}(-p, M))), L) \\ = \; & \mathrm{discrepancy}(\mathrm{rotate}((0,0), \alpha, \mathrm{translate}(-p, M)), \mathrm{translate}(-p, L)) \quad . \end{aligned}$$

In other words, the discrepancy, and thus the optimal rotational angle, can be computed by first translating L and M so that the rotation point is the origin and then rotating about the origin. So in the following, only rotation about the origin is considered.

Let M^ϑ denote M rotated about the origin through an angle ϑ. The position $(x_p^\vartheta, y_p^\vartheta)$ of a node p in M^ϑ is related to the node's position (x_p, y_p) in M by

$$x_p^\vartheta = x_p \cos\vartheta - y_p \sin\vartheta, \quad y_p^\vartheta = x_p \sin\vartheta + y_p \cos\vartheta \quad .$$

Defining $x_p^L \mathrel{\hat{=}} L_p.x$ and $y_p^L \mathrel{\hat{=}} L_p.y$, the discrepancy with L for an angle ϑ is thus

$$\begin{aligned} & \mathrm{discrepancy}(M^\vartheta, L) \\ = \; & \textstyle\sum |M_p^\vartheta - L_p|^2 \\ = \; & \textstyle\sum (M_p^\vartheta.x - x_p^L)^2 + (M_p^\vartheta.y - y_p^L)^2 \\ = \; & \textstyle\sum (x_p \cos\vartheta - y_p \sin\vartheta - x_p^L)^2 + (x_p \sin\vartheta + y_p \cos\vartheta - y_p^L)^2 \end{aligned}$$

where all summations are over $p \in C$.

Differentiating with respect to ϑ, equating to zero, and setting $s \mathrel{\hat{=}} \sin\vartheta$ and $c \mathrel{\hat{=}} \cos\vartheta$ for brevity, gives:

$$\begin{aligned}
& 0 \\
= \; & \textstyle\sum \; 2(x_p c - y_p s - x_p^L)(-x_p s - y_p c) + 2(x_p s + y_p c - y_p^L)(x_p c - y_p s) \\
= \; & 2\textstyle\sum - csx_p^2 + s^2 x_p y_p + s x_p x_p^L - c^2 x_p y_p + csy_p^2 + c y_p x_p^L \\
& \quad + csx_p^2 + c^2 x_p y_p - c x_p y_p^L - s^2 x_p y_p - csy_p^2 + s y_p y_p^L \\
= \; & 2\textstyle\sum s x_p x_p^L + c y_p x_p^L - c x_p y_p^L + s y_p y_p^L
\end{aligned}$$

Hence,

$$\begin{aligned}
\sin\vartheta \textstyle\sum x_p x_p^L + y_p y_p^L &= \cos\vartheta \textstyle\sum x_p y_p^L - y_p x_p^L \\
\Rightarrow \qquad \sin\vartheta \; : \; \cos\vartheta &= (\textstyle\sum x_p y_p^L - y_p x_p^L) \; : \; (\textstyle\sum x_p x_p^L + y_p y_p^L)
\end{aligned}$$

This determines the optimal rotational angle, and from this the residual discrepancy can be calculated.

4 A Peer-to-Peer Alignment Process

The preceding sections showed how to compute a map from inter-node distance information and how to reconcile two maps that differ by a translation, rotation and optional reflection (using alignment) and by local noise (using averaging). This section discusses a peer-to-peer algorithm that uses these techniques to establish a common coordinate system throughout a network of communicating nodes.

The algorithm is staged into several phases, each of which executes more-or-less simultaneously throughout the network. However, in principle, the phases could be intermixed, repeated or continually executed as needed.

1. Collect and Diffuse Distance Information: Each node directly measures distances to some neighbours (nearby nodes) and locally broadcasts the distances it measures; this is repeated several times to reduce measurement error. Each node collects its own measurements and those communicated to it. It averages multiple measurements for the distance between a given pair of nodes and broadcasts such averages. To reduce storage and communication costs, each node continually prunes the information it stores and broadcasts by discarding information about more distant nodes.

   In this way, distance information is locally diffused through the network. It is expected that each node will eventually acquire estimates for the distance between itself and, say, 7 neighbours, and for the distances between those neighbours. However, in a practical application, this phase may be executed for only a limited duration and the distance information acquired by a node may be incomplete; i.e., it may receive some information regarding neighbours $p \in H$ but it may not have an estimate for every pair $p, q \in H$. Moreover, a given node may not have information about some nodes that are nearby even if it has information about nodes that are further away.

2. Form and Reconcile Local Maps: When a node has sufficient distance estimates, it computes and locally broadcasts a map of its neighbourhood. When a node receives a map from a neighbour, it reconciles its own map with its neighbour's and broadcasts its own map. When reconciling its map, a node may remove more distant nodes.

   In this way, each node should quickly acquire a map of its neighbourhood. Two nodes that are nearby should acquire maps that closely agree, both in terms of which nodes' positions are recorded in the maps and in terms of the closeness of the positions for a given node.

Eventually, this agreement should spread throughout the network so that a common coordinate system is formed. In particular, it should be possible, in principle, to gather all of the nodes' maps and construct a single composite map by averaging the positions for each node, and but for measurement error the composite should differ from a map of the nodes 'true' positions only by an isometric transformation. (The nodes' true positions may be determined using any suitable exogenous measurement process. For example, a pair of orthogonal axes may be arbitrarily fixed and the nodes' true positions determined by measuring shortest distances to those axes. This measurement process should, of course, be consistent with that used by the network to measure inter-node distances.)

Of course, some practical considerations need to be given to mechanisms to reduce communication contention and instability. For example, a node may limit the frequency with which it broadcasts distance estimates and maps.

4.1 Obtaining a Shared Coordinate System

The simple peer-to-peer algorithm described above should eventually result in a consistent 'world view' being established throughout the network. However, each node will still have its own coordinate system, and data involving locations exchanged between nodes must be subjected to a coordinate transformation from the sender's to the recipient's system.

For moderately large networks, a shared coordinate system can be obtained using a *wavefront* technique. On initialization, each node determines for itself a random priority with which it tags its own map when it is broadcast. Ideally, each node would receive a unique priority and priorities would be randomly distributed throughout the network.

When a node receives a map, it processes the map differently according to whether its own priority is higher or lower than the received map's:

- If the receiving node's priority is lower than the received map's, then the node aligns and averages its own map with the received map and increases its own priority to match that of the received map.
- If the receiving node's priority is higher than the received map's, then the node aligns the received map with its own map and averages its own map with the aligned map.

In this way, the coordinate system of the node with the highest priority acts as a fixed reference system to which the other nodes align themselves. (In effect, the nodes are executing an anytime leadership election algorithm combined with map reconciliation.)

However, for very large networks, the propagation of the common system throughout the full network can take a long time, because it is limited by the local nature of the information dissemination. Using only local broadcasts, the number of communication hops that information requires to cross the network grows like the spatial diameter of the network. (Assuming a bounded node density, for a roughly square or circular layout, the spatial dimension grows like the square root of the number of nodes; for a linear layout, the spatial dimension grows even linearly with the number of nodes.)

4.2 Convergence

Although the simple peer-to-peer algorithm should lead to a consistent world view, the rate of convergence can be rather slow, and for large networks it can take many iterations for long-distance discrepancies to ripple back-and-forth through the network and be smoothed out.

This is not so much due to any deficiency of the techniques described, but a generic problem in obtaining a global map by patching together local maps, in which many small measurement errors can add up to unbounded errors over large distances.

To fundamentally change the rate of convergence, a different communication paradigm is required, as discussed in the next section. It will also accelerate the establishment of a shared coordinate system.

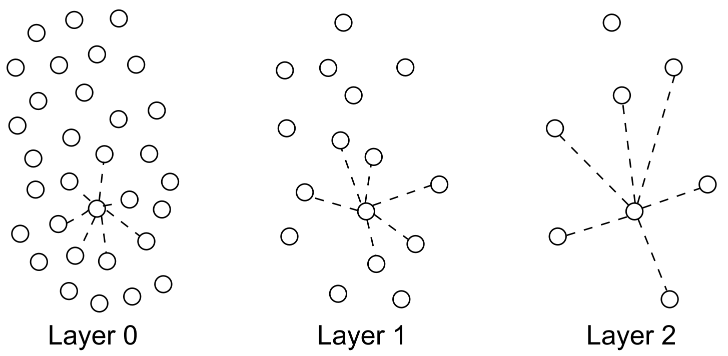


Figure 11: Nodes organized into layers

5 Hierarchical Localization

To overcome the limitations of strictly local broadcasts, a new organization is considered for the network:

- The nodes are organized into communication layers, designated by an index $k = 0, 1, \ldots, K$, where K is at most roughly equal to $\log_2(N/7)$, where N is the total number of nodes.
- All nodes belonging to layer k_1 also belong to each lower level $k_2 < k_1$.
- The number of nodes in each layer is (approximately) double the number in the next higher layer (if there is one).
- Layer 0 contains all of the nodes.
- Each layer is more-or-less evenly distributed throughout the network.
- A node in a given layer can communicate using local broadcast to other nodes in the same layer. Thus, each layer can form its own peer-to-peer network.
- Communication between two layers can occur through nodes that are in both layers.

Such an organization can be obtained in a distributed way if, on initialization, each node determines for itself a random bit string; it then belongs to layer k if its bit string starts with at least k digits `1`. So a node with random bit string `11010010`$\cdots$ belongs to layers 0, 1, and 2. The same bit string should be used for the random priority mentioned in Section 4.1 by taking it as a real number in the range from 0 to 1: $\texttt{0.11010010}\cdots_2 = 0.82\cdots$.

For localization, it is also assumed that a node in a given layer can estimate its distance to some other nodes in the same layer (see Figure 11), and that all nodes know the value of K. Hierarchical localization then proceeds layer by layer, starting with the highest layer.

- Each node in the highest layer estimates its distance to nearby nodes in the same layer and forms a map representing its own position and those of its neighbours within the same layer; these maps are reconciled as described in the preceding section so that overall the highest layer forms a large-scale global coordinate system.
- Once reconciliation has completed at the highest layer, the next highest layer begins. Each node in this layer executes a similar process of estimating distances, map formation and map reconciliation. However, first those nodes that also belong to the higher layer broadcast their existing maps within this layer. These maps are incorporated into the maps formed for this layer, and act as a large-scale skeleton onto which finer-scaled maps are attached.
- This process repeats for each successively lower layer.

Aside from the highest layer, nearby nodes within a given layer should start out with similar maps, arising from the next higher layer. Consequently, the number of iterations required to reach convergence is expected to be significantly reduced. Provided that the number of iterations is bounded as the size of the network grows, then the hierarchical organization should require time that increases only logarithmically with the network size. However, the underlying assumptions, that communication and distance estimation can be performed over arbitrarily large distances, presumably will fail at some point as physical limitations on communication or distance measurement become significant as the geometrical distance increases.