Abstract—This paper proposes a method of estimating a target-object shape, the location of which is unknown, through the use of location-unknown mobile distance sensors. The direction of the sensor is fixed from the moving direction. Typically, mobile sensors are mounted on vehicles. Each sensor continuously measures the distance from it to the target object. The estimation method does not require any positioning function, anchor-location information, or additional mechanisms to obtain side information such as angle of arrival of signal. Under the assumption of a polygon target object, each edge length and vertex angle and their combinations are estimated to completely estimate the shape of the target object.

I. INTRODUCTION

Cars are being implemented with various distance sensors such as mm-wave sensors to prevent traffic accidents and improve the comfort of driving. Because some of these sensors have ranges larger than 100 meters, they can gather environment information. This environment information is used by the car itself and can be useful even for other cars or people. If such information is used by other people for other applications, this is vehicular-based participatory sensing or crowd sensing.

Although such an estimation intuitively seems impossible due to too many unknown factors and some theoretical results shown in the next section suggest it is impossible, this paper proposes a theoretical method for successfully estimating the target-object shape by using mobile sensors that continuously measure the distance between individual sensors and the target object.

II. RELATED WORK

The fundamental questions related to the research topic of this paper is whether we can estimate the shape of a target object using many simple sensors without a positioning function or location information and how we estimate it if possible. Studies by Saito et al. suggested that we can estimate only a small number of parameters such as the size and perimeter length of a target object by using cameras that cannot cover the whole shape of the target object [21].

In addition, there has been research into capturing the shape of a target object by using cameras that cannot cover the whole shape of the target object [21].

III. MODEL

A target object $T$ is in a bounded convex set $\Omega \subset \mathbb{R}^2$. It is a polygon, and its boundary $\partial T$ is closed and simple (no holes or double points) and consists of directional edges $\{L_j\}_j$ where $j = 1, 2, \ldots, n_e$ (Fig. 1). Here, $n_e$ is the number of edges. Let $\lambda_j$ be the length of $L_j$, and let $\xi_j$ be the angle formed by $L_j$ and the reference direction where $0 \leq \xi_j < 2\pi$. Note that the inner angle formed by $L_j$ and $L_{j+1}$ is $\gamma_j = \pi - \xi_{j+1} - \xi_j$. Here, $\{L_j\}_j$ are counted counterclockwise along $\partial T$ and the head of $L_j$ is the tail of $L_{j+1}$. We do not know
any of \( \{\lambda_j, \xi_j, \gamma_j\}_j \). That is, we do not know the target-object shape, size, or location.

A vehicle is running at a speed \( v \) on a randomly placed straight line the direction of which is \( \phi \) from the reference direction and passes through \( \Omega \). (This can be extended to a time-variant speed, but, for simplicity, assume that \( v \) is time-invariant.) It is equipped with a directional distance sensor the direction of which is \( \theta \) from the moving direction. (In practice, the vehicle’s location (that is, the sensor’s location) \((x_s(t), y_s(t))\) may not be in \( T \), but the vehicle is assumed to run on a straight line passing through \( T \) for simplicity.) The sensor continuously measures the distance \( r(t) \leq r_{\text{max}} \) at \( t \) from the sensor to the target object and sends the sensing result. Here, \( r_{\text{max}} \) is the maximum range of the sensor. Because the direction of the sensing range is \( \phi + \theta \) from the reference direction, \( r(t) \) is given as follows:

\[
r(t) = \begin{cases} \hat{r}(t), & \text{if } r(t) \leq r_{\text{max}}, \\ 0, & \text{if } r(t) > r_{\text{max}}. \end{cases}
\]

For angles \( t_1, t_2, \langle t_1, t_2 \rangle \) is \( t_1 \) under mod \( 2\pi \) and \( \langle t_1, t_2 \rangle \) is an interval \([t_1, t_2]\) under mod \( 2\pi \). That is, \( \langle t_1, t_2 \rangle \) is an interval \([t_1, t_2]\) if \( t_1, t_2 < 2\pi \) and is intervals \([t_1, 2\pi]\) \( \cup [0, t_2 - 2\pi] \) if \( t_1 < 2\pi, 2\pi \leq t_2 < 4\pi \).

### IV. Basic properties

This section discusses basic properties of \( r(t) \). A simple example is illustrated in Fig. 2. An important observation of this figure is that there may be some jumps in \( r(t) \) from a certain value between 0 and \( r_{\text{max}} \) to another certain value. Only a single edge located nearest to a sensor is detected by the sensor, and its distance from the sensor is \( r(t) \). Even if other edges are within a sensing range, they are not detected or their distances to the sensor are not measured. That is, detection of an edge may be blocked by another edge. A jump down (up) of \( r(t) \) occurs when a block starts (finishes).

A sensor detects \( L_j \) at \( t \) if and only if the sensor is located in \( \omega_j(\theta + \phi) \) at time \( t \), where \( \omega_j(\theta + \phi) \) is a parallelogram attached to the right-hand side of \( L_j \) and one of edges is \( L_j \) and another has the length \( r_{\text{max}} \) and the direction \( \theta + \phi \) (Fig. 3). Note that the sensor detecting \( L_j \) needs to satisfy

\[
\phi + \theta \in [\xi_j, \xi_j + \pi].
\]

In the remainder of this section, we focus on the sensing results \( r(t) > 0 \). When a sensor keeps detecting an edge, \( r(t) \) becomes continuous and becomes a line segment while the sensor keeps detecting it (Fig. 4). When the period detecting the whole \( L_j \) with \( r(t) > 0 \) by a sensor the direction of which is \( \theta \) starts at \( t_e \) and ends at \( t_e \), an event corresponding to \( t_e \) is (i) a change of slope at \( r(t_e) > 0 \), (ii) a jump down of \( r(t) \) at \( t_e \), or (iii) \( r(t_e) < r_{\text{max}} \) and \( r(t_e - dt) = \emptyset \) and an event corresponding to \( t_e \) is (i) a change of slope at \( r(t_e) > 0 \), (ii) a jump up of \( r(t) \) at \( t_e \), or (iii) \( r(t_e - dt) < r_{\text{max}} \) and...
Thus, according to Fig. 5, results:

\[ \phi \]

\[ \lambda, \xi, \theta, v \]

\[ L \]

the whole \[ r \]

ending with the events mentioned above) of \[ L \] with \( r(t) > 0 \) for a given \( \theta \) unless we explicitly indicated otherwise. Let \( l_d(L|\theta) \) and \( s_d(L|\theta) \) be the length in time of \( p_d(L|\theta) \) and the slope of \( r(t) \) during \( p_d(L|\theta) \).

For edge \( L \) of length \( \lambda \) and direction \( \xi \) and the sensor the moving and sensing directions of which are \( \phi \) and \( \theta \), the following subsections provide (i) the relationships between the system parameters \( (\lambda, \xi, \phi, \theta, v) \) and the sensing results \( (l_d(L|\theta), s_d(L|\theta)) \), (ii) the probability that the sensor detects the whole \( L \) with \( r(t) > 0 \), and (iii) the probability that the sensor detects a vertex with \( r(t) > 0 \).

1) Relationships between system parameters and sensing results: According to Fig. 3

\[
vl_d(L|\theta) \sin \theta = \lambda \sin(\theta - \xi + \phi),
\]

\[
v l_d(L|\theta) s_d(L|\theta) \sin \theta = \lambda \sin(\xi - \phi).
\]

Thus,

\[
s_d(L|\theta) = -\sin(\xi - \phi) / \sin(\xi - \phi - \theta).
\]

Because of Eq. (4),

\[
\xi - \phi = \arctan \frac{s_d(L|\theta) \sin \theta}{s_d(L|\theta) \cos \theta + 1} + (\pi) s_d(L|\theta),
\]

where \((\pi) s_d(L|\theta)\) is 0 if \(\arctan \frac{s_d(L|\theta) \sin \theta}{s_d(L|\theta) \cos \theta + 1} - \theta \in [\pi, 2\pi]\) and is \(\pi\) if \(\arctan \frac{s_d(L|\theta) \sin \theta}{s_d(L|\theta) \cos \theta + 1} - \theta \in [0, \pi]\). Apply this to Eq. (3) and obtain

\[
\lambda = vl_d(L|\theta) \sqrt{s_d(L|\theta)^2 + 2s_d(L|\theta) \cos \theta + 1}.
\]

2) Probability that sensor detects whole \( L \) with \( r(t) > 0 \): According to Fig. 4, if line \( G \) on which the sensor moves is in the directional strip of width \( r_{\text{max}} | \sin \theta | - \lambda | \sin(\xi - \phi) | \) and if \( \phi \) satisfies Eq. (11) with \( \xi_2 = \xi \) (equivalently, \( \phi - \xi \in [-\eta, -\theta + \pi] \)), the sensor can detect the whole \( L \). Because the strip width must be non-negative, \( \phi - \xi \in [-\eta, \eta]\) for \( r_{\text{max}} | \sin \theta | < \lambda \), and if \( \phi - \xi \in [0, 2\pi] \) for \( r_{\text{max}} | \sin \theta | \geq \lambda \), where \( \eta(\lambda, \theta) \equiv \arcsin \frac{r_{\text{max}} | \sin \theta |}{\lambda} \in [0, \pi/2] \). (For simplicity, \( \eta(\lambda, \theta) \equiv \pi/2 \) for \( r_{\text{max}} | \sin \theta | < \lambda \) in the remainder of this paper.) Note that the measure of the set of \( G \) on
which sensors monitor $\Omega$ (Fig. 5) is given by Eq. (5.2) in [22] and is $[\Omega_1 + \pi \gamma_{r_{\max}} |\sin \theta|]$. Also note that the measure of the set of $G$ that is in this strip and has a direction satisfying Eq. (1) is $\int_{\Phi_1} r_{\max} |\sin \theta| - \lambda |\sin(\xi - \phi)|d\phi$ where

$$\Phi_1(\xi) \overset{\text{def}}{=} ([\xi - \eta, \xi + \eta] \cup [\xi - \pi + \eta, \xi + \pi + \eta]) \cap [\xi - \theta, \xi - \theta + \pi]$$

for $r_{\max} |\sin \theta| < \lambda$ and $\Phi_1(\xi) \overset{\text{def}}{=} [\xi - \theta, \xi - \theta + \pi]$ for $r_{\max} |\sin \theta| \geq \lambda$. Because the probability $q_d(\lambda)$ that the sensor detects the whole $L$ of length $\lambda$ is given by the ratio of these measures in accordance with the definition of geometric probability [22],

$$q_d(\lambda) = \frac{\int_{\Phi_1} r_{\max} |\sin \theta| - \lambda |\sin(\xi - \phi)|d\phi}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]} = \frac{2\lambda r_{\max} |\sin \theta| - 2\lambda (1 - \cos \theta)}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]}.$$  

(The denominator doubles because $G$ is directional.) Therefore, the expected number $E[n_d(\lambda)]$ of sensors detecting the whole $L$ of length $\lambda$ with $r(t) > 0$ is given by

$$E[n_d(\lambda)] = \sum_{i=1}^{n} \frac{2\gamma_i |\sin \theta_i| - 2\lambda (1 - \cos \gamma_i \theta_i)}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta_i|]}.$$  

3) Probability that sensor detects a vertex: Here, we pay attention to the number of sensors that have sensing results that cover a vertex of $T$. Such sensing results may not cover a whole edge.

Assume a vertex formed by $L_j, L_{j+1}$. As shown in Fig. 4 if line $G$ on which the sensor moves is in the directional strip of width $r_{\max} |\sin \theta|$ and if $\phi$ and $\xi_j (\xi_{j+1})$ satisfy Eq. (1), the sensor can detect the part around the vertex with $r(t) > 0$. Similar to in Subsection \[V-2\] the probability $q_d(\gamma_j)$ that the sensor can detect the vertex formed by $L_j, L_{j+1}$ with $r(t) > 0$ is given by the following.

$$q_d(\gamma_j) = \frac{\int_{\Phi_1} r_{\max} |\sin \theta| - \lambda |\sin(\xi - \phi)|d\phi}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]} = \begin{cases} \frac{\gamma_j r_{\max} |\sin \theta|}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]}, & \text{for } \gamma_j \in (0, \pi), \\ \frac{2\pi - \gamma_j r_{\max} |\sin \theta|}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]}, & \text{for } \gamma_j \in (\pi, 2\pi). \end{cases}$$

where $\gamma_j = \pi - \xi_{j+1} + \xi_j$ is the inner angle of the vertex. Therefore, the expected number $E[n_d(\gamma)]$ of sensors detecting a vertex of inner angle $\gamma$ with $r(t) > 0$ is given by

$$E[n_d(\gamma)] = \begin{cases} \sum_{i=1}^{n} \frac{\gamma_j r_{\max} |\sin \theta|}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]}, & \text{for } \gamma \in (0, \pi), \\ \sum_{i=1}^{n} \frac{2\pi - \gamma_j r_{\max} |\sin \theta|}{2[\Omega_1 + 2\pi r_{\max} |\sin \theta|]}, & \text{for } \gamma \in (\pi, 2\pi). \end{cases}$$

V. Estimation method

Now, we are in a position to discuss target-object shape estimation. The shape estimation method consists of four main parts and an additional part. The first main part, “edge length estimation part,” estimates the target object edge lengths $\{\lambda_j\}_j$. The second main part, “angle estimation part,” estimates the angle of vertexes $\{\gamma_j\}_j$. The third main part, “combining length and direction estimation part,” combines the results of the estimated edge lengths and angles and estimates a vertex formed by them. The fourth main part, “order estimation part,” estimates of the order of the edges. That is, it determines the consecutive edge of a certain edge. Because we have already obtained the lengths and directions of edges at the end of the second main part, the estimated shape of $T$ is expected to be obtained at the end of the four main parts. However, we need to compensate for estimation error when $T$ is not convex. The additional part makes up for errors of estimating edges forming concave parts of $\partial T$.

A. Edge length estimation part

This part estimates edge length and the number of edges of the estimated edge length. For a preliminary step, we need to obtain $(l_d, s_d)$ from the measured distance $r(t) > 0$. Assume that we obtain $(l_d(k, i), s_d(k, i))$ from the distance $r_t(t)$ measured by the $i$-th sensor where $l_d(k, i)$ $(s_d(k, i))$ is the $k$-th line segment derived from its sensing result observing a whole edge. That is, when the $i$-th sensor observes $j$ whole edges of $T$, $r_t(t)$ has $j$ line segments corresponding to individual whole edges of $T$ and $l_d(k, i)$ and $s_d(k, i)$ are the length and slope of the $k$-th line segment among them.

For a given $(l_d(k, i), s_d(k, i))$ and known $v_i$ and $\theta_i$, obtain the temporary estimate of the edge length $\lambda$ due to Eq. (6).

$$\hat{\lambda}(i, k) = v_i l_d(k, i) \sqrt{s_d(k, i)^2 + 2s_d(k, i) \cos \theta_i + 1}. \quad (11)$$

Intuitively, if the set $\{\hat{\lambda}(i, k)\}_{i,k}$ forms $n_e$ clusters, each cluster corresponds to an edge. To implement this intuition, classify the set of temporary edge length estimates. It is a good idea to apply a classification tool such as Mclus of R [23]. Let $\Lambda_m$ be the $m$-th classified subset of this set of temporary estimates (or the set of $(l_d, s_d)$ deriving $\hat{\lambda}(i, k) \in \Lambda_m$), and $n_\lambda$ be the number of the classified subsets (that is, $1 \leq m \leq n_\lambda$).

By using $\hat{\lambda}(i, k) \in \Lambda_m$, the mean of the temporary estimates in $\Lambda_m$ is adopted as the estimate of an edge length.

$$\hat{\lambda}(\Lambda_m) = \sum_{\hat{\lambda}(i, k) \in \Lambda_m} \hat{\lambda}(i, k)/n(\Lambda_m). \quad (12)$$

Here, note that $n_e = n_\lambda$ may not be valid. This is because several edges may have the same length or classification may
be incorrect. To overcome this point, use Eq. (5). Note that the observed \( n_d(\lambda) \) is \( \gamma(\Lambda_m) \) when \( \lambda = \lambda(i, k) \) for \( \forall (i, k) \in \Lambda_m \), and that, if the number of edges of length \( \lambda \) is \( m \), \( E[n_d(\lambda)] \) is given by \( m \) multiplied by the right-hand side of Eq. (5). Thus, the estimated number \( \tilde{n}_e(\Lambda_m) \) of edges of length \( \lambda(\Lambda_m) \) is shown below.

\[ \tilde{n}_e(\Lambda_m) \approx \gamma(\Lambda_m) / E[n_d(\lambda(\Lambda_m))] \]  

**B. Angle estimation part**

The angle estimation method proposed here applies Eq. (5) to two consecutive edges. For a preliminary step, we need to find the sensing results \{\( s_d(k, i, j) \)\} that cover a vertex. Note that \( r(t) \) is continuous when a sensor detects a vertex of which the direction on the edge is the origin. Therefore, \( r(t) \) becomes a line segment for each edge. \( r(t) \) becomes two consecutive line segments with different slopes for detected \( L_j, L_{j+1} \). Thus, we apply Eq. (5) to \( s_d(k, i, s_d(k+1, i)) \) corresponding to the slopes of \( r(t) \) detecting \( L_j, L_{j+1} \). Note that we can use \( s_d(k, i), s_d(k+1, i) \) that cover only parts of the two edges \( L_j, L_{j+1} \) (not the whole \( L_j, L_{j+1} \)).

By applying Eq. (5) to \( s_d(k, i), s_d(k+1, i) \), we obtain the temporary estimate of the inner angle \( \gamma_j \) formed by \( L_j, L_{j+1} \).

\[ \tilde{\gamma}_j(i, k) = \pi \pm (\arctan(s_d(k, i) \sin \theta_i) / s_d(k+1, i) \cos \theta_i + 1) \]  

where \( \pm \) becomes + if \( \sin \theta_i > 0 \) and becomes − otherwise. This is because \( s_d(k, i), s_d(k+1, i) \) corresponds to \( L_j, L_{j+1} \) if \( \sin \theta_i > 0 \) and corresponds to \( L_{j+1}, L_j \) otherwise.

Similar to the edge length estimation, classify the set of temporary inner angle estimates. Let \( \Gamma_m \) be the \( m \)-th classified subset of this set of temporary estimates (or the set of measured edge pairs \( s_d(k, i), s_d(k+1, i) \) deriving \( \tilde{\gamma}(i, k) \in \Gamma_m \), and let \( n_\gamma \) be the number of the classified subsets (that is, \( 1 \leq m \leq n_d(\lambda(\Lambda_m)) \)). By using \( \tilde{\gamma}(i, k) \in \Gamma_m \), the estimate of an edge length is derived.

\[ \tilde{\gamma}(\Gamma_m) = \sum_{\tilde{\gamma}(i, k) \in \Gamma_m} \tilde{\gamma}(i, k) / \tilde{n}(\Gamma_m) \]  

For each estimated \( \tilde{\gamma}(\Gamma_m) \), there may be multiple vertices. Use Eq. (10) to estimate the number of vertices corresponding to the estimated \( \gamma \). Because the observed \( n_d(\gamma) \) is \( \gamma(\Gamma_m) \) when \( \gamma = \tilde{\gamma}(i, k) \) for \( \forall (i, k) \in \Gamma_m \), the estimated number \( \tilde{n}_e(\Gamma_m) \) of vertices that have inner angle \( \tilde{\gamma}(\Gamma_m) \) is shown below.

\[ \tilde{n}_e(\Gamma_m) \approx \gamma(\Gamma_m) / E[n_d(\gamma(\Gamma_m))] \]  

**C. Combining length and direction estimation part**

A vertex is determined by its inner angle and the lengths of two edges forming the vertex. This part estimates the vertex by combining an estimated angle and estimated edge lengths obtained in the previous two parts.

If we can find such sensing results that \( (s_d(k, i), s_d(k+1, i)) \in \Gamma_m \), \( (s_d(k, i), s_d(k+1, i)) \in \Lambda_{m1} \), and \( (s_d(k, i), s_d(k+1, i)) \) are sensing results of consecutive edges, these are sensing results of a vertex angle \( \tilde{\gamma}(\Gamma_m) \) and of lengths \( \lambda(\Lambda_m) \) and \( \tilde{\lambda}(\Lambda_{m2}) \). We count the number of such sensing results and judge that such a vertex exists if the counted number of results is large enough.

Although the angle estimation requires a pair of slope \( s_d(k, i), s_d(k+1, i) \) that may not cover the whole edges, the edge length estimation requires \( l_d(k, i), l_d(k+1, i) \) that covers the whole edge. Therefore, it can happen that the angle of a vertex can be estimated but one of its edges (or any of its edges) cannot be estimated. For the estimated angle \( \tilde{\gamma}(\Gamma_m) \) and the edge length \( \tilde{\lambda}(\Lambda_{m'}) \), we count the number of sensing results satisfying \( (s_d(k, i), s_d(k+1, i)) \in \Gamma_m \), \( (l_d(k, i), s_d(k, i)) \in \Lambda_{m'} \), and judge the existence of the vertex angle \( \tilde{\gamma}(\Gamma_m) \) and of one of edge length \( \tilde{\lambda}(\Lambda_{m'}) \).

**D. Order estimation part**

To derive a method of identifying the order of edges, we use sensing results for consecutive edges.

Assume that \( (l_d(k, i), s_d(k, i)) \in \Lambda_m \) and \( (l_d(k+1, i), s_d(k+1, i)) \in \Lambda_{m'} \). If a sensor continuously detects multiple edges without jumps of \( r(t) \), they must be consecutive edges. Therefore, an edge of length which is estimated by \( (l_d(k, i), s_d(k, i)) \) likely connects to an edge the length of which is estimated by \( (l_d(k+1, i), s_d(k+1, i)) \). Let \( \Lambda(m, m') \) be the set of two sensing data pairs satisfying \( \{(l_d(k, i), s_d(k, i)) \in \Lambda_m, (l_d(k+1, i), s_d(k+1, i)) \in \Lambda_{m'} \}(i,k) \). We judge that an edge of length \( \tilde{\lambda}(\Lambda_{m'}) \) connects to an edge of length \( \tilde{\lambda}(\Lambda_{m}) \), if \( \gamma(\Lambda(m, m')) \) is large.

**E. Additional part**

This part may provide additional estimates of edges forming a concave vertex of \( T \). As described below, \( \tilde{n}_e(\Lambda_m) \) defined by Eq. (13) may underestimate the number of edges for a non-convex \( T \). This part compensates for this error.

When \( T \) has a concave part, \( E[n_d(\lambda(\Lambda_{m}))] \) given by Eq. (5) may not be correct. The reason is as follows: a sensor that should detect this edge may not detect another edge of \( \partial T \) is between this edge and this sensor. That is, an edge of \( \partial T \) blocks this sensor’s detection of this edge.

We take account of this blocking and modify Eq. (5) for a non-convex \( T \). We consider an event in which one consecutive edge \( L_{i-1}, L_i \) forming a concave vertex of \( T \) may block the detection of the other edge. We neglect other blocking events caused by other edges. As shown in the derivation of Eq. (8), \( \phi \in \Phi_1(\xi_i) \). This is because this sensor detects the whole \( L_i \) and \( t_{\text{max}}| \sin \theta - \lambda_i | |\sin(\phi - \xi)| > 0 \) (Fig. 10). In addition, the detection of \( L_i \) by the sensor the direction of which is \( \theta \) is not blocked only if \( \xi_{i-1} - \phi < 0 \) (Fig. 7). Hence, the probability that a sensor detects the whole edge with length...
\[ q_d(\lambda_i, \theta, \delta \xi_{i-1}) = \frac{f(\lambda_i, \theta, \delta \xi_{i-1})}{2|\Omega_1 + 2\pi r_{\text{max}}| \sin \theta} \]  

(17)

where \( \phi \in \Phi_2(\xi_i, \xi_{i-1}) \) and \( f(\lambda_i, \theta, \delta \xi_{i-1}) \) is the expected number of sensors detecting the whole \( \lambda_i \) of length \( \lambda_i \) with \( r(t) > 0 \) without blocking is given by

\[ E[n_d(\lambda_i, \xi_i, \xi_{i-1})] = \sum_{j} q_d(\lambda_i, \theta_j, \delta \xi_{i-1}). \]  

(18)

VI. NUMERICAL EXAMPLES

In the remainder of this section, the following conditions are used as the default conditions unless explicitly indicated otherwise. \( \Omega \) is a disk with a radius of 200 length units. \( r_{\text{max}} = 100 \), \( n_s = 2000 \). The sensing area direction is \( \theta = \pi/2 \) and the moving speed \( v = 1 \) is for all the vehicles.

In the simulation conducted, each sensor sends a sensing report every single time unit.

A. Basic properties

1) Impact of \( \theta \): The proposed method uses the sensing data observing a whole edge or a vertex. Therefore, the number of such data is very important for accurate estimation. When \( v \) and \( \theta \) are the same for all the vehicles, the expected number of such data for a single edge or vertex is \( E[n_d] = q_d n_s \) where \( q_d \) is given by Eq. 7 or 9. Because \( q_d \) is a function of \( |\sin \theta| \), it is plotted against \( \theta \) in Fig. 8.

As shown in this figure, \( q_d(\lambda) \) and \( q_d(\gamma) \) maximized at \( \theta = \pi/2 \). This seems to be because the part a sensor detects while it moves becomes smaller as \( |\sin \theta| \) becomes smaller. For example, for \( |\sin \theta| = 0 \), a sensor keeps detecting the same point of the target object and does not provide any information of other parts of the target object even though the vehicle moves. Thus, small \( |\sin \theta| \) results in a small amount of information that is useful in the proposed method. In particular, we should avoid \( |\sin \theta| \leq 0.5 \) if possible.

For a fixed \( \theta \), \( q_d \) increases as an edge length becomes shorter or a vertex angle becomes wider. This is because the shorter whole edge can be covered more easily than a longer one and because a wider angle can be detected more easily than a sharp one.

2) Impact of \( n_s \) and sensing error: In addition to \( \theta \), the number \( n_s \) of sensors is a key parameter to determine the number of sensing data useful for the proposed method. To evaluate the impact of \( n_s \) on the estimation results, a simulation was conducted where \( T \) is a right-triangle the edge lengths of which are 50, 50\( \sqrt{3} \), and 100. Ten simulation runs were used for each value of \( n_s \). Furthermore, sensing errors were intentionally added. Sensing errors for \( s_d \) are normally distributed random variable \( \epsilon_s \) with mean 0 and standard deviation (s.d.) 0.03. By adding an error, \( s_d \) becomes \( \tan(\arctan(s_d) + \epsilon_s) \). A sensing report was lost with probability \( \epsilon_l = 0.002 \). As a result, \( p_d(L|\theta) \) was divided at this epoch and \( l_d(L|\theta) \) became shorter.

Two types of estimation errors occurred. Type one is that the number of estimated edges and/or that of estimated angles became incorrect. That is, the edge lengths (angles) derived by the proposed method were not three in the simulation for the right-triangle. For example, the proposed method misjudged \( T \) to have four edges. Type two is that the estimated edge length or angle was inaccurate.

In the proposed method, \( n_s \) has a large impact on type-one errors. Figure 8 plots the ratio of the number of simulation runs the results of which show type-one errors to the total number of simulation runs. On the other hand, if there was no type-
one error, the estimated edge lengths and angles were fairly accurate and insensitive to $n_s$ with/without sensing errors. Figure 10 plots the standard deviation of $\lambda (\gamma)$ normalized by $\lambda (\gamma)$. Although it decreased as $n_s$ became larger, it was small even for small $n_s$. It was less than 1% for edge length estimates and several% or less for angle estimates. This also shows that a longer edge (wider angle) has better estimation accuracy than a shorter (sharper) one. (Although similar results were obtained with $\epsilon_s$ and $\epsilon_t$, they were omitted.) In addition, estimation bias was very small and fairly insensitive to $n_s$ with/without sensing errors.

As mentioned above, $n_s$ and $\theta$ have a large impact on the number of sensing data useful for the proposed method. Therefore, $n_s$ also has a large impact on estimating the consecutive edges and the combination of an angle and edges forming a vertex, although no figures are shown. As $n_s$ becomes smaller, the number of sensing results covering consecutive edges and those covering a vertex become smaller. Thus, they become more difficult to estimate appropriately.

As noise ($\epsilon_t, \epsilon_s$) became larger, the estimation became less accurate. Details are omitted due to the space limitations.

B. Shape estimation of buildings

The proposed method is applied to the buildings highlighted by thick blue lines in Fig. 11. One is convex, and the other is concave.

For building (a), the estimated edge lengths and angles are shown in Table III. The estimation errors were several% or less. More than ten samples observed for the vertex of an estimated angle and two estimated lengths were $(\hat{\gamma}(\Gamma_1), \hat{\lambda}(\Lambda_2), \hat{\lambda}(\Lambda_3)),$ $(\hat{\gamma}(\Gamma_1), \hat{\lambda}(\Lambda_3), \hat{\lambda}(\Lambda_2)),$ $(\hat{\gamma}(\Gamma_1), \hat{\lambda}(\Lambda_3), \hat{\lambda}(\Lambda_1)),$ $(\hat{\gamma}(\Gamma_1), \hat{\lambda}(\Lambda_1), \hat{\lambda}(\Lambda_3)),$ $(\hat{\gamma}(\Gamma_2), \hat{\lambda}(\Lambda_2), \hat{\lambda}(\Lambda_3)),$ $(\hat{\gamma}(\Gamma_2), \hat{\lambda}(\Lambda_3), \hat{\lambda}(\Lambda_2)),$ $(\hat{\gamma}(\Gamma_2), \hat{\lambda}(\Lambda_2), \hat{\lambda}(\Lambda_4)),$ and $(\hat{\gamma}(\Gamma_2), \hat{\lambda}(\Lambda_4), \hat{\lambda}(\Lambda_2)).$ Therefore, we can judge that a vertex of wide angle $(\hat{\gamma}(\Gamma_2))$ is formed by the longest edge $(\hat{\lambda}(\Lambda_1))$ and a short edge $(\hat{\lambda}(\Lambda_1)$ or $\hat{\lambda}(\Lambda_2))$. Because it is estimated that there are two vertexes of wide angle and a single longest edge, we can estimate that there are two of these vertexes and they are connected by the longest edge. Let $L_2$ be the edge of length $\hat{\lambda}(\Lambda_1)$. Then, what we have estimated so far is: the estimated $\lambda_3$ and $\lambda_4$ are $\hat{\lambda}(\Lambda_4)$ and $\hat{\lambda}(\Lambda_2)$, and both the estimated $\gamma_2$ and $\gamma_3$ are $\hat{\gamma}(\Gamma_2)$. In addition, we can judge that a vertex of approximately $\pi/2$ ($\hat{\gamma}(\Gamma_1)$) is formed by a short edge $(\hat{\lambda}(\Lambda_1))$ or $\hat{\lambda}(\Lambda_2)$ and a long edge $(\hat{\lambda}(\Lambda_3))$. Because there is a single edge of length $\hat{\lambda}(\Lambda_1)$ and there is a single edge of length $\hat{\lambda}(\Lambda_2)$, both the estimated $\lambda_1$ and $\lambda_5$ are $\hat{\lambda}(\Lambda_3)$ and both the estimated $\gamma_1$ and $\gamma_4$ are $\hat{\gamma}(\Gamma_1)$. Thus, we can estimate the shape of this building even though there were not enough observed samples of a vertex formed by two long edges $(\hat{\lambda}(\Lambda_3))$ or samples of consecutive long edges. The shape estimated is shown in Fig. 12(a).

For building (b), the estimated edge lengths and angles are shown in Table III. There are eight estimated edge lengths, although there are nine angles. Such inconsistency can occur...
because the former needs the sensing data containing the whole edge and the latter needs the sensing data containing a vertex. A wider angle generally has better estimation accuracy than a sharper one, but the concave angle had slightly poorer 

estimation accuracy than other convex angles in this example. A wider angle generally has better estimation accuracy than a sharper one, but the concave angle had slightly poorer estimation accuracy than other convex angles in this example.

More than ten samples observed for the vertex of a polygon target object and a straight line trajectory of each sensor, the generalization of these assumptions remains as a further study. An object and a straight line trajectory of each sensor, the generalization of these assumptions remains as a further study. A

TABLE II

| Estimated | Relative error | δ |
|-----------|---------------|---|
| λ(Λ₁)    | 23.17         | -0.073 | 1 |
| λ(Λ₂)    | 24.80         | -0.008 | 1 |
| λ(Λ₃)    | 99.77         | -0.002 | 2 |
| λ(Λ₄)    | 105.93        | -0.003 | 1 |
| γ(Γ₁)     | 1.610         | 0.025 | 3 |
| γ(Γ₂)     | 2.379         | 0.010 | 2 |

Fig. 12. Estimated shape of a building.

TABLE III

| Estimated | Relative error | δ |
|-----------|---------------|---|
| λ(Λ₁)    | 27.71         | -0.076 | 1 |
| λ(Λ₂)    | 29.71         | -0.010 | 2 |
| λ(Λ₃)    | 49.58         | -0.008 | 5 |
| γ(Γ₁)     | 1.615         | 0.028 | 6 |
| γ(Γ₂)     | 4.156         | -0.118 | 3 |

long edges (λ(Λ₃)) and that a vertex of approximately π/2 (γ(Γ₂)) is formed by the long edge (λ(Λ₃)) and a short edge (λ(Λ₁) or λ(Λ₂)). The shape of the target object is shown in Fig. 12 (b).

VII. CONCLUSION

By using location-unknown distance sensors, this paper proposed a method of estimating the shape of a target object that is at location is unknown. Each sensor moves on an unknown line at a known speed and continuously measures the distance between it and the target object. By collecting measured distances, the proposed method can estimate the shape of the target object. The estimation method does not require any positioning function, anchor-location information, or additional mechanisms to obtain side information such as angle of arrival of signal.

The successful development of this estimation method suggests that the possibility of software sensors implemented by participatory sensing under complete location privacy can be much wider. It also suggests that the secondary use of IoT (internet of things) information can be wider than expected.

Because the estimation method assumed a polygon target object and a straight line trajectory of each sensor, the generalization of these assumptions remains as a further study. An experiment using the proposed method is another future study.

REFERENCES

[1] J. Krumm, A survey of computational location privacy, Pers. Ubiquit. Comput., 13, pp. 391-399, 2009.
[2] Mauro Conti, et al., Providing Source Location Privacy in Wireless Sensor Networks: A Survey, IEEE Communications Surveys & Tutorials, 15, 3, pp.1238 - 1280, 2013.
[3] J. Gubbi, et al., Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems, 29, 7, pp. 1645-1660, 2013.
[4] L. D. Xu, et al., Internet of Things in Industries: A Survey, IEEE Trans. Industrial Informatics, 10, 4, pp. 2233-2243, 2014.
[5] H. Saito, et al., Wide Area Ubiquitous Network: The Network Operator’s View of a Sensor Network, IEEE Communications Magazine, 46, 12, pp. 112-120, 2008.
[6] Usman Raza, et al., Low Power Wide Area Networks: An Overview, IEEE Communications Surveys & Tutorials, 19, 2, pp. 855-873, 2017.
[7] H. Saito, et al., Shape Estimation Using Networked Binary Sensors, INFOCOM 2009.
[8] H. Saito, Local Information, Observable Parameters, and Global View, IEICE Trans. Communications, E96-B, 12, pp.3017-3027, 2013.
[9] H. Saito, et al., Geometric Analysis of Observability of Target Object Shape Using Location-Unknown Distance Sensors, arXiv.
[10] H. Saito, et al., Stochastic Geometric Filter and Its Application to Shape Estimation for Target Objects, IEEE Trans. Signal Processing, 59, 10, pp. 4971-4984, 2011.
When \(-\theta, \delta \xi - \theta \in Z_4, f = r_{max}|\sin \theta(2\eta - \delta \xi) - \lambda(2 - 2\cos \eta - \cos(\delta \xi - \theta) + \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_5, f = r_{max}|\sin \theta(\pi + \eta - \theta) - \lambda(2 - 2\cos \eta + \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_6, f = r_{max}|\sin \theta(\pi - \delta \xi) - \lambda(2 - 2\cos \delta \xi - \theta) + \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_1 \text{ or when } -\theta, \delta \xi - \theta \in Z_3, f = r_{max}|\sin \theta(2\eta - \delta \xi) + \lambda(2 - 2\cos \eta + \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_2, f = r_{max}|\sin \theta(\eta - \theta) - \lambda(2 - 2\cos \eta - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_3, f = r_{max}|\sin \theta(\pi - \delta \xi) - \lambda(2 + \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_4, f = r_{max}|\sin \theta(\pi + \eta - \delta \xi + \theta) - \lambda(2 + \cos(\delta \xi - \theta) - \cos \eta)\).

When \(-\theta, \delta \xi - \theta \in Z_5, f = r_{max}|\sin \theta(\pi + \eta - \delta \xi + \theta) - \lambda(2 - \cos(\delta \xi - \theta) - \cos \eta)\).

When \(-\theta, \delta \xi - \theta \in Z_6, f = r_{max}|\sin \theta(\eta - \theta) - \lambda(-2\cos \eta - \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_1, f = r_{max}|\sin \theta(2\eta - \delta \xi) - \lambda(-2\cos \eta - \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_2, f = r_{max}|\sin \theta(\eta - \theta - \pi) - \lambda(-2\cos \eta - \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_3, f = r_{max}|\sin \theta(\pi + \eta - \delta \xi + \theta) - \lambda(2 + \cos(\delta \xi - \theta) - \cos \eta)\).

When \(-\theta, \delta \xi - \theta \in Z_4, f = r_{max}|\sin \theta(\pi + \eta - \delta \xi + \theta) - \lambda(-2\cos \eta - \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_5, f = r_{max}|\sin \theta(\eta - \theta - \pi) - \lambda(-2\cos \eta - \cos(\delta \xi - \theta) - \cos \theta)\).

When \(-\theta, \delta \xi - \theta \in Z_6, f = r_{max}|\sin \theta(\eta - \theta) - \lambda(2 - 2\cos \eta - \cos(\delta \xi - \theta) + \cos \theta)\).