

Article

A Gravity Inspired Approach to Multiple Target Localization Through-the-Wall Using Non-Coherent Bi-Static Radar

Imran Mohammed , Iain B. Collings  and Stephen V. Hanly 

School of Engineering, Macquarie University, Sydney, NSW 2109, Australia; iain.collings@mq.edu.au (I.B.C.); stephen.hanly@mq.edu.au (S.V.H.)

* Correspondence: imran.mohammed@students.mq.edu.au

Abstract: This paper considers multiple target localization using a non-coherent bi-static radar with multiple receivers, where the targets are located behind a wall. This paper presents a new clustering algorithm inspired by Newtonian gravity that iteratively groups particles at target locations and eliminates particles at non-target locations. We first propose a histogram based pre-processing algorithm that imposes a grid over the region of interest and defines a particle with measurement-dependent mass for each grid square. We then calculate a Newtonian inspired force on each of the particles and move them in the direction of the force. We repeat the process until there is no further movement. The proposed algorithm works even when some of the measurements are unavailable or missing and when some of the measurements are false measurements. Location accuracy is shown to be in the order of 8 cm.

Keywords: non-coherent radar; through-the-wall radar; localization; multiple targets; newtonian gravity; gravity inspired algorithm



Citation: Mohammed, I.; Collings, I.B.; Hanly, S.V. A Gravity Inspired Approach to Multiple Target Localization Through-the-Wall Using Non-Coherent Bi-Static Radar. *Electronics* **2021**, *10*, 1524. <https://doi.org/10.3390/electronics10131524>

Academic Editors: Soo Young Shin and Giovanni Andrea Casula

Received: 31 May 2021
Accepted: 18 June 2021
Published: 23 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Through-the-wall radar (TWR) has attracted considerable interest in recent years because of its increasing applications in rescue and military operations [1–5]. There are two different processing techniques in TWR systems, coherent processing, and non-coherent processing. Most of the existing systems have employed coherent processing [6–11]; however, they require wide bandwidth and large aperture arrays [1,3]. They also need complex transmitter or receiver designs, which are expensive and typically cumbersome and not portable in emergencies. In contrast, non-coherent radar localization can be performed with simple hardware and allows flexible distributed deployment scenarios [12–15]. Non-coherent TWR uses the envelope of the received (reflected) signal, thus significantly relaxing the radar positioning and processing requirements and resulting in low-cost and low hardware complexity [3].

In this paper, we consider using a non-coherent bi-static radar to estimate the locations of multiple targets that are behind a wall. In the multiple target (N targets) scenario, the receiver lacks knowledge about which measurement is associated with which target. As such, there is a measurement target association problem. We will consider a deployment scenario with one transmitter and multiple receivers (R receivers), at geographically different locations outside the wall, and, therefore, there are up to $N^2 \binom{R}{2}$ possible target-measurement association combinations.

In the single target case there is no association ambiguity, and the localization is typically done using the time taken, t , by the transmitted signal to reach the receivers, after getting reflected from the target, and having travelled a distance (range) r . For a given transmitter and receiver pair, there is a set of potential target locations that satisfy a given range measurement. These locations form an ellipse. The task is to find where the actual target lies on the ellipse. When there are two different transmitter-receiver pairs, the two ellipses intersect at the target location. Estimating the target location using ellipses is called

ellipse cross localization (ECL). When transmitting through a wall, the transmitted signal takes a longer time to travel (through the wall) than in free space and the path is different, due to the dielectric constant of the wall. For a given transmitter-receiver pair, the target no longer lies on the same (free space) ellipse, however without knowing the exact dielectric constant of the wall, the wall thickness, the distance to the wall, and the angle of incidence, it is impossible to know where the actual ellipse is. Therefore, it is necessary to use the free space ellipse and accept that the intersection points will not locate the target exactly.

In [4], an algorithm was proposed to estimate a single target location through a wall using ECL with a particular characterization of the influence of the wall thickness and its dielectric constant. In [5], an algorithm was proposed based on a hyperbola cross localization (HCL) technique. However, both of these techniques required knowledge about the distance of the wall from the transmitter and receiver pair and the wall parameters, such as the wall thickness, material of the wall, etc., which are generally not available in real-time applications. In [2], an Ellipse-Hyperbola-Localization approach was proposed based on ECL and HCL. The approach composed of two steps, rotation cross localization approach and determination of optimal range based on the statistical properties of the ECL and HCL in different target scene regions. However, as with [4,5], it is also restricted to localizing only a single target and cannot be directly extended to multiple target localization.

For multiple (N) targets it is necessary to solve both the noise displacement and the measurement association problem together. At each receiver, there will be N range measurements (assuming there are no lost measurements). As mentioned above, the receivers have a target-measurement association ambiguity, and, therefore, there is an associated target-ellipse association problem. When there are R receivers, there will be up to $N^2 \binom{R}{2}$ intersection points. Only $N \binom{R}{2}$ of these correspond to the actual targets. The first approach to solve the multi-target problem was proposed in [16] in the free space scenario. However, it is a greedy algorithm and does not perform well in the through-the-wall scenario. The algorithm often fails to select N target estimates in a through-the-wall scenario. The complexity of the algorithm increases exponentially with the number of receivers (number of measurements). In [17], an algorithm was proposed to find multiple target locations by iteratively enforcing a set of constraints. It is limited by the fact that the complexity of the algorithm increases exponentially with an increase in measurements, and it also requires all the measurements to be available, i.e., no missing measurements.

An important practical limitation of all the ECL based approaches is that they require the receivers to be co-linear. When the receivers are not co-linear, the resulting ellipses will be rotated, and calculating the intersection points of two rotated ellipses is known to be a hard problem due to the cosine terms in the second order polynomial. As the number of receivers increases, the number of intersection points increases, giving the problem even greater computational complexity. This co-linear limitation is problematic for practical applications requiring accuracy in two dimensions, since the ellipse crossings are oblique for targets at broadside, as well as those in the axis of the array, and are hence sensitive to measurement errors. For practical applications with targets in two dimensions, a new non-ECL based approach is needed.

Adding to the challenge is that, even if it were possible to calculate all the crossing points for an ECL-based algorithm, it is then necessary to decide which points correspond to real targets (i.e., where the crossing ellipses are both from the same target), and which points have arisen from ellipses from two different targets. One approach would be to apply a clustering algorithm, however, since there is no known model for the statistical distribution of the locations of the ellipse intersection points in the multiple target scenario, there is no existing clustering algorithm that is suitable. Existing clustering algorithms, such as k-means, are based on the assumption that the data comes from a source with additive noise. As most of the intersection points in our case do not belong to any source (such as the intersection of two ellipses belonging to two different targets), existing clustering algorithms fail to estimate the target location.

In this paper, we propose a new multiple target location estimation algorithm consisting of a new (non-ECL) pre-processing step which we call Ellipse-Histogram (EH) pre-processing, coupled with a new clustering algorithm motivated by Newtonian gravity.

Our proposed EH pre-processing algorithm imposes a grid over the region of interest, and places a virtual *particle* of zero mass at the centre of each grid square. We then map the transmitter–receiver-pair ellipses onto the grid, by considering each ellipse, and increasing (by one unit) the mass of the particles in each grid-square that the ellipse passes through. Note that this operation is computationally much simpler than calculating the intersection of two ellipses. At the end of this process, each grid-square will, therefore, contain a particle with a mass equal to the number of ellipses that pass through the square. Any particles that remain with zero mass are removed.

Note that the particles in the grid-squares around the target locations will have high mass compared to other particles in non-target areas. The task then is to identify clusters of high-mass particles, corresponding to target locations. Our EH pre-processing algorithm is presented in detail in Section 3.1.

At this point it would be possible to consider applying traditional clustering algorithms such as k-means clustering [18], or fuzzy c-means clustering [19]. However both of these algorithms operate under the assumption that each data point (particle in our case) belongs to one of the clusters, and is randomly perturbed from the cluster centroid according to a particular distribution. As mentioned previously, this is not the case for our particles.

An alternate clustering algorithm based on a universal gravity rule was proposed in [20]. In that algorithm, initially the cluster-centroids were randomly chosen, and the data points were associated with the clusters based on Euclidean distance to the centroids. After association, a ‘force’ was calculated on the centroid, due to the particles that were associated with it. The centroid was then moved in the direction of the force. The distance each centroid moved was proportional to the magnitude of the force. After moving the centroids to the new location, the data points were reassociated with the updated centroids (based on the new locations). This process was repeated until there was no further movement of centroids. As with the other algorithms, a characteristic of this approach was that there was a significant influence on the centroids due to particles at non-target locations.

Our proposed approach to clustering (which forms the second part of our new proposed algorithm) is also inspired by gravitational ideas. We call it the gravity inspired clustering algorithm (GICA). A key difference is that our approach is not based on calculating cluster centroids. Instead, we calculate a Newtonian inspired force on each of the particles individually, and move each particle in the direction of the corresponding force. When particles collide, they combine into a single particle with the mass of the original two particles. This is analogous to the formation of stars in the early phase of the universe. Our proposed algorithm iteratively eliminates any particles that have low mass. We repeat this process iteratively until the particles have coalesced into clusters and there is no further movement in the particles. Our new algorithm is presented in detail in Section 3.2.

We note that the overall complexity of our proposed approach is limited by the number of grid squares, and unlike other approaches, the complexity of ours does not increase exponentially with the number of measurements. Additionally, our proposed algorithm can be applied even when some of the measurements are missing, and works even when the receivers are not collinear. The proposed algorithm can also be applied when some of the measurements are false. False measurements arise when there is a strong reflection from objects that are not targets (i.e., not of interest, e.g., reflection from walls.).

2. System Model

The objective is to estimate the target locations that are behind a wall. In this paper, we assume the region of interest is a region of size 20 m by 15 m. We consider the bi-static radar with multiple receivers (R). For example, when trying to map an office that is behind a wall, with a corridor in between, as shown in Figure 1. Unlike many ray-tracing studies,

we do not assume a straight line reflector for internal walls; instead, we model the practical situation where these walls are made of plasterboard, and the RF reflectors come from the vertical metal studs that support the walls and doors. Here, the targets are the metal studs, and the aim is to estimate the target location of the metal studs.

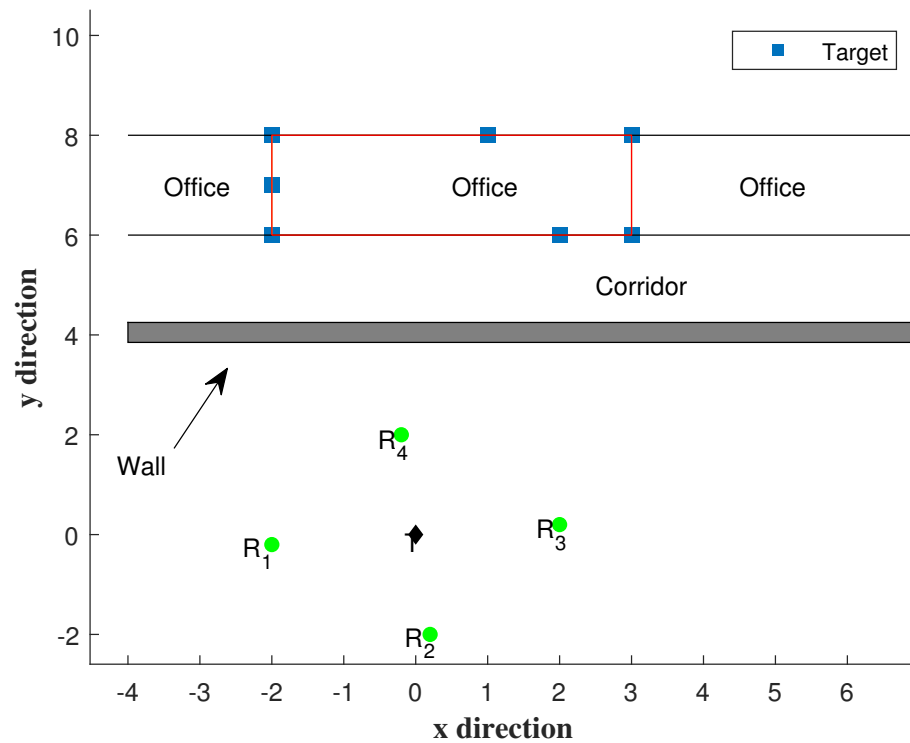


Figure 1. Location map showing target location and transmitter and receiver locations, for a scenario with a 10 m² office and a corridor in between.

As shown in Figure 1, the targets of interest are on one side of the wall, and the transmitter-receivers are on the other side. The receivers (R) are placed as shown in Figure 1. Receivers in this configuration produce better results compared to when they are collinear.

As mentioned previously, for a given transmitter and receiver pair, there is a set of potential target locations that satisfy a given range measurement. These locations form an ellipse. Let Tx and Rx be the transmitter and receiver locations for a specific pair, and let r be the range measurement from that pair. Then the ellipse is defined by the set of points, e , such that $\|Tx - e\|_2 + \|Rx - e\|_2 = r$.

When the receivers are in x and y directions as shown in Figure 1, the ellipses that belong to the same target, and come from different receivers, intersect at greater angles resulting in better target estimation. When the receivers are collinear, the ellipses intersect at lower angles. Even with small measurement noise, the ellipses intersect at a location far from the true target location.

The measurements are taken as shown in Figure 2. The transmitter and receivers move together, from one place to another. At intermediate points ($\phi'_k, k = 1, 2, \dots, M$) the receivers collect range measurements. In practice, an example is when the transmitter and receivers are mounted on a vehicle moving from one point to another collecting measurements at intermediate locations. At each intermediate point ϕ_k , $N \times R$ range measurements are collected (N range measurements corresponding to N targets at each receiver). As the range measurements are collected at M intermediate points, there will be a total of $M \times N \times R$ range measurements.

The range measurements in the presence of a wall is modeled as $\tilde{r}_{i,j,\phi_k} = r_{i,j,\phi_k} + \Delta r_{i,j,\phi_k}$ [2]. Where \tilde{r}_{i,j,ϕ_k} represents the range measurement corresponding to the to the

i th target at j th receiver when the transmitter and receivers are at intermediate location ϕ_k in the presence of a wall. Where $\Delta r_{i,j,\phi_k}$'s are the i.i.d Gaussian random variables with mean μ and variance σ^2 ($\Delta r \sim \mathcal{N}(\mu, \sigma^2)$), and r_{i,j,ϕ_k} represents the distance from the transmitter to the i th target and from i th target to the j th receiver at the ϕ_k^{th} intermediate location. In this paper, we do not use any specific knowledge about the wall parameters, and we use average values for the mean and standard deviation ($\mu = 0.48$ m and $\sigma = 0.1$ m) [2]. These are typical values, and details about the calculation of μ and σ can be found in [2] as a function of frequency and bandwidth. Without loss of generality, mean μ , is removed from all range measurements. Indeed, when the knowledge about the wall parameters, such as wall thickness, dielectric constant, etc., are known, the proposed algorithm works better. For a given transmitter and receiver pair, the set of all points that satisfy a range measurement is an ellipse. There are a total of $M \times N \times R$ ellipses, as this is the number of range measurements.

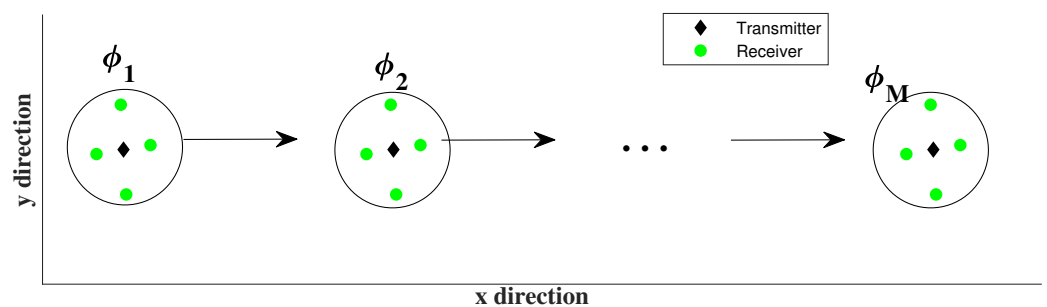


Figure 2. Scenario showing where both the transmitter and receivers are moving together.

Figure 3 shows the ellipses for the scenario in Figure 1. The number of intermediate points are taken to be 5, i.e., $M = 5$ (number of ellipses equals to $(M =) 5 \times (N =) 7 \times (R =) 4$). From the figure, it can be seen that each ellipse intersects with other ellipses at various locations. Some of the intersection points represent target locations, i.e., intersection points corresponding to the ellipses from the same target and different receivers. Some of the intersection points are from the ellipses belonging to different targets. As noted, the number of intersection points from the ellipses arising from different targets is very large compared to the number of intersection points of ellipses belonging to the same target. From Figure 3 we can see that it is hard to estimate the target locations. The aim is to find the target locations from the available range measurements information.

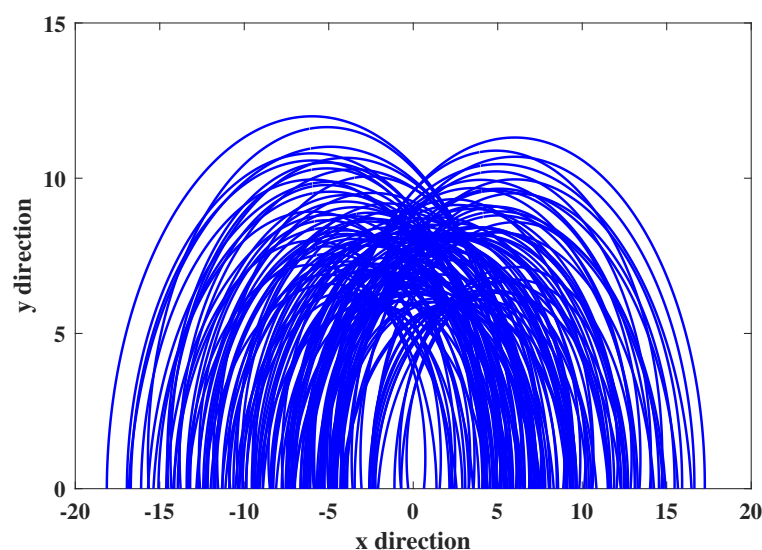


Figure 3. Ellipses plotted for the scenario shown in Figure 1, with measurements taken at $M(= 5)$ different locations.

3. Proposed Algorithm

Our proposed algorithm consists of two elements. We first propose a pre-processing algorithm called Ellipse-Histogram (EH) pre-processing, which generates a set of particles. We then propose a new clustering algorithm inspired by Newtonian gravity to cluster the particles and hence identify the target locations. In the following subsections, we present our algorithm and describe it using pseudocode.

3.1. Ellipse-Histogram (EH) Pre-Processing

We propose a pre-processing approach to generate particles based on ellipses and histograms. The first pre-processing step is to impose a grid over the region of interest, (as shown in Figure 4), and place a particle of zero mass at the centre of each grid square. In this paper, we use a grid square that is 7 cm by 7 cm (the choice of 7 cm is for the office-type scenario we are considering in this paper, but these values could be adjusted for other building scenarios of different scale). We then map the transmitter-receiver-pair ellipses onto the grid, by considering each ellipse, and increasing (by one unit) the mass of the particles in each grid-square that the ellipse passes through. At the end of this process, each grid-square will therefore contain a particle with a mass equal to the number of ellipses that pass through the square. The higher the particle's mass, the higher the probability that particle location corresponds to the target location. In the algorithm, we represent the entire region with matrix $Mass$ with the mass at grid location $[i j]$ stored in the element $Mass[i j]$.

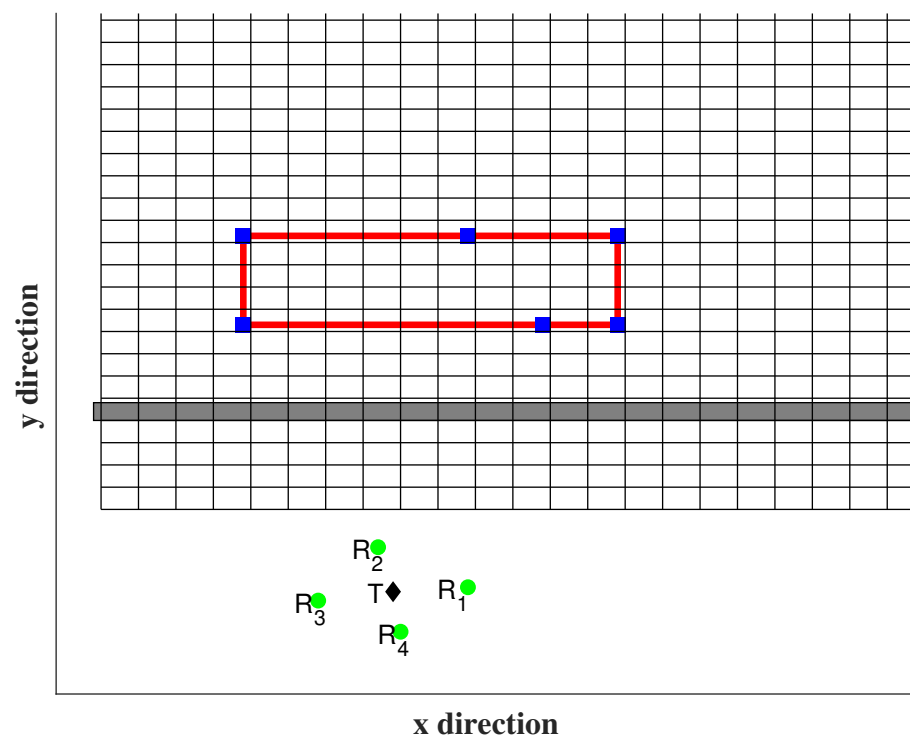


Figure 4. Figure showing imposing the grid over the area of interest.

Figure 5 is plotted for the scenario in Figure 3 with height as the mass of the particle. From the figure, it can be observed that the particles around the target location have higher mass compared to non-target locations. However, there are many particles at non-target locations whose mass is relatively high. This makes the problem challenging, i.e., identifying the particles corresponding to the target locations among various local maximas.

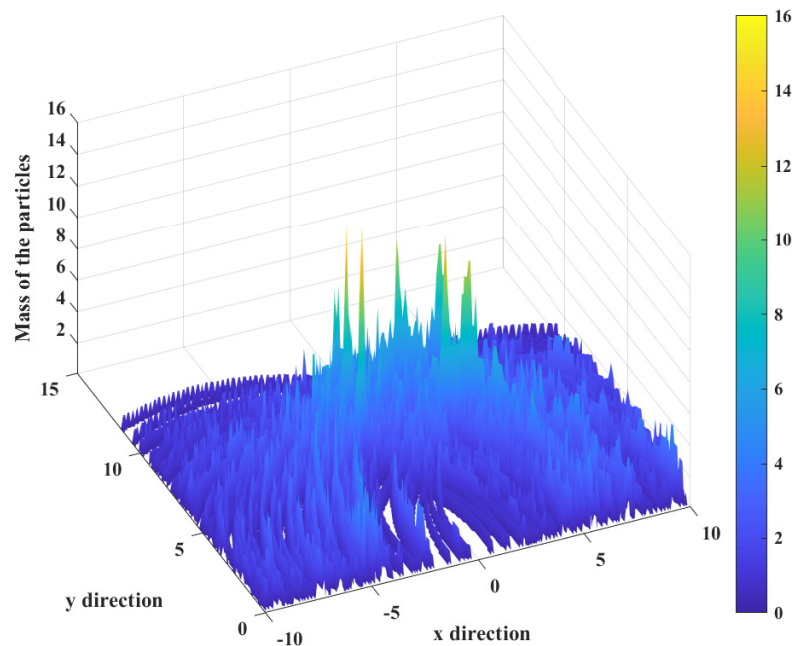


Figure 5. Plot of particle weightings for the 2-D grid, derived from the ellipses in Figure 3.

The second pre-processing step is to remove particles that have less than the half the mass of the largest mass particle, thus restricting attention to the particles corresponding to the highest concentration on ellipses.

After this EH pre-processing, we then performed clustering via our novel gravity inspired algorithm, which we now present.

3.2. Gravity Inspired Clustering Algorithm (GICA)

We propose a new algorithm motivated by Newtonian gravity. We calculate a Newtonian inspired force on each of the particles from the Ellipse-Histogram pre-processing and move them in the direction of the force. We repeat this iteratively until the particles have coalesced into clusters and there is no further movement in the particles. This is in contrast to the other approaches that are based on iteratively estimating a known number of centroids.

First we select a non-zero mass particle randomly from the entire region. An important element of our approach is the definition of a box around the chosen particle. We define the box to be square, where the box-size is defined to be the distance from the particle to the edge of the box, as shown in Figure 6. Within the box we will be calculating a Newtonian inspired force on the chosen particle from all the other particles within the box.

Our Newtonian Force is calculated in the procedure DISPLACEMENT in Algorithm 1. The position of the randomly selected particle is located at the grid coordinates $[i j]$ and the masses of all the particles in the whole region is stored in the matrix *Mass*. The size of the box is chosen randomly between two values r_{low} and r_{high} . The box size should not be too large or otherwise particles belonging to two different targets will merge into one. The box size should not be too small or else there will no other particle and no gravitational force will be felt. The minimum box size depends on the density of ellipse intersection points, which is heterogeneous across the entire region. For this reason, we use a random box size chosen between r_{low} and r_{high} . We choose $r_{low} = 2$ and $r_{high} = \lceil \frac{q}{2l} \rceil$, where q is the minimum distance between any two targets, and l is the length of a grid square.

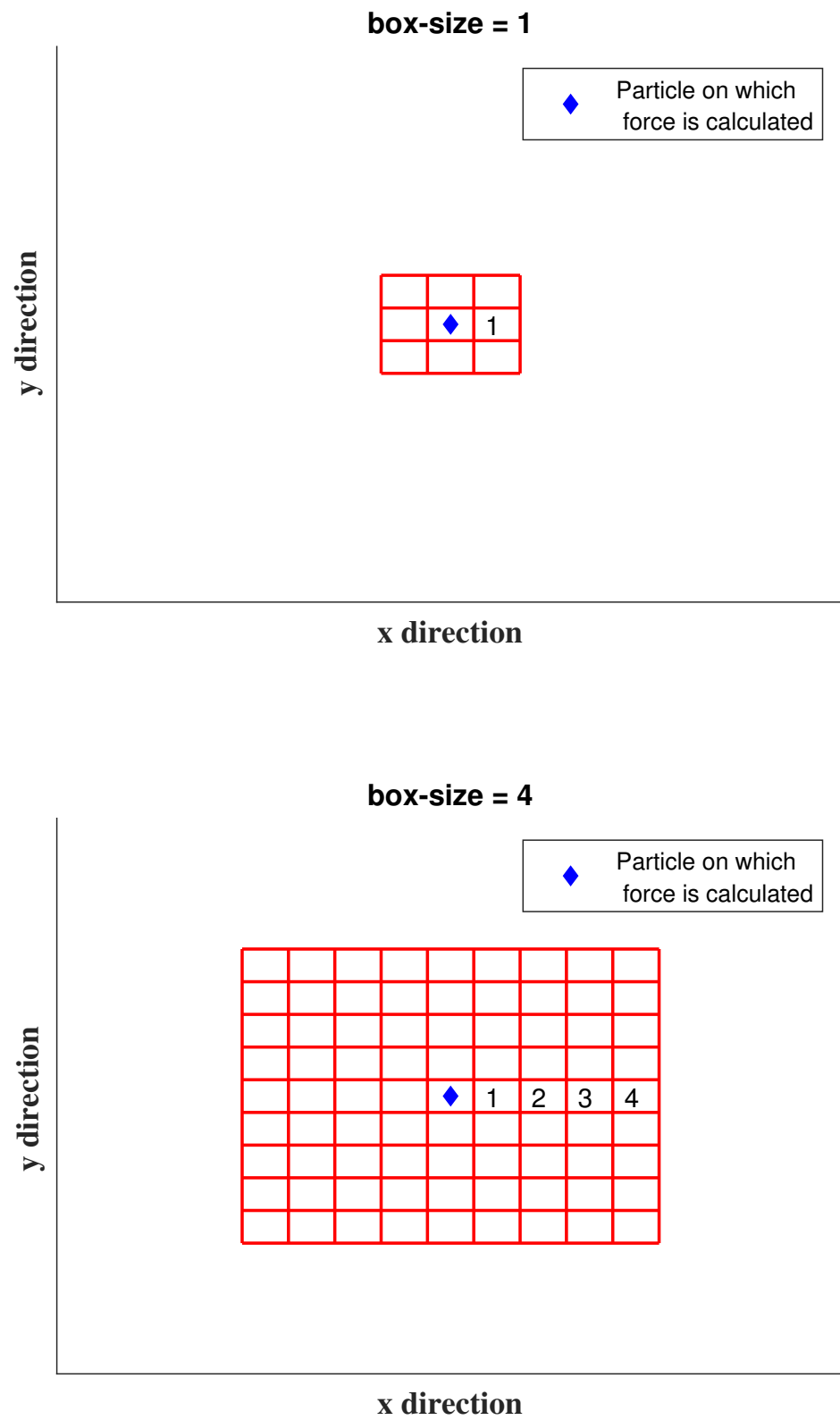


Figure 6. Figure showing different box-sizes.

To calculate the force, we use the traditional Newtonian force formula, and propose to set the gravitational constant equal to one. In particular, the force on particle p_1 because of the particle p_2 is defined as $F_{p_1,p_2} = \frac{m_1 \times m_2}{\|l_2 - l_1\|_2^3} \times (l_2 - l_1)$. Where l_1 , and l_2 are the physical locations (measured in meters), and m_1 , and m_2 are the masses of the particles p_1 and p_2 , respectively. Similarly, we calculate the force from all the particles within the box. The resultant force on the selected particle is the vector addition of all the forces from individual particles.

In the procedure DISPLACEMENT, the double loop looks at all the grid squares (each which may contain a particle) inside the given box. The box is cropped so that we only look at the part of the box that lies in the rectangular region of interest where targets can occur. On the edge of the region, the box can go outside, hence the need for cropping.

Once the force is calculated, the acceleration of the selected particle is calculated using the formula $\frac{F_{p_1}}{m_1}$, where m_1 is the mass of the selected particle p_1 . Upon calculation of acceleration, displacement of the selected particle in time $\sqrt{2}$ seconds is calculated (assuming initial velocity is zero).

The procedure MOVE actually moves the particle, and adjusts the mass matrix *Mass*. If the particle moves to a grid square where there is another particle, the two particles merge into a single particle with the combined mass of the two original particles. Movement only occurs if the magnitude of the displacement is more than half of the length of the grid square (which is 7 cm) containing the particle. If so, the selected particle moves one step in the direction of the force (i.e., we move the particle to one of the eight adjacent grid squares). After moving to a new location, if there already exists a particle in that location, we merge the two particles into one. In other words, the mass of the particle in the new location increases to be the sum of the two masses. After merging the two particles, we remove any particles that are likely to be related to non-targets. To achieve this, we remove particles whose mass is less than one-seventh of the largest mass particle. We have performed extensive simulations across a wide range of system configurations, and show in Section 4 that the choice of one-seventh produces better results compared to other thresholds such as $\frac{1}{5}$, $\frac{1}{6}$, $\frac{1}{8}$, $\frac{1}{9}$, and $\frac{1}{10}$.

The main Algorithm 2 consists of two loops. The inner loop repeatedly selects a particle at random, chooses a random-sized box, computes the gravitational force, and associated movement and mass change (if any). All of this is done in Algorithm 1 as described above, which is called from Algorithm 2. If the particle moves, the mass matrix changes, because the mass at the original location goes to zero, and the mass at the new location increases by the mass of the moving particle. The inner loop of Algorithm 2 repeats until it is deemed that there is no further change. Even if one particle does not move, another randomly selected particle may yet move. An equilibrium is deemed to have been reached if there is no change after $N \times 50$ randomly selected particles are tested. When the number of targets is unknown, we output the location of the non-zero mass particles after the equilibrium reached. When we have additional information regarding how many targets are there, we output the locations of non-zero mass particles if the number of non-zero mass particles equals to the number of source targets. Otherwise, when they are not equal, we repeat the process again. This is done using the outer loop in Algorithm 2.

The reason for the outer loop is because the problem we are solving is hard. It is a non-convex problem, and the equilibrium reached can be a locally optimal solution, but not globally optimal. We find that when we run the algorithm for a second time, from the same starting conditions, a different equilibrium can be reached. The randomness of particle selection and box size selection can lead to different equilibria being reached. The number of targets found can be different in different iterations. The algorithm stops when the number of targets found equals the true number of targets, in which case it outputs the target locations found in the final equilibrium. The algorithm terminates after a maximum of $10 \times N$ trials. If it runs for $10 \times N$ trials then it has failed by definition.

We find that Algorithm 2 almost always (98% of the time over 5000 realizations) terminates before reaching $10 \times N$ iterations, and the results presented in Section 4 show how accurate the output target locations are. The fact that the number of trials is upper bounded by $N \times 10$ tells us that the number of local optima is limited, and grows no more than linearly with the number of targets.

Ideally, we would run the gravity algorithm in continuous time, and all forces would be calculated simultaneously. Since we are running a discrete-time algorithm, and considering each particle one at a time, it is important that we do select each particle in a systematic way (e.g., from right to left) as this would impose a bias on the process of movement. For this reason, each particle is selected randomly each time. The maximum movement step of one grid square is imposed as a way to approximate a continuous movement.

Algorithm 1 Procedures

```

1: procedure DISPLACEMENT([i j], M, X)
2:    $F_{total} \leftarrow [0, 0]$ 
3:    $m_1 \leftarrow \mathbf{M}(i, j)$ 
4:    $(x_i, y_j) \leftarrow$  corresponding physical location of the element at  $[i, j]$  from X.
5:    $l_1 = (x_i, y_j)$ 
6:    $q \leftarrow$  minimum distance between any two targets.
7:    $r_{low} \leftarrow 2$  (minimum box-size value)
8:    $r_{high} \leftarrow \lceil \frac{q}{2l} \rceil$  (maximum box-size value)
9:    $r \leftarrow$  a random number between  $r_{low}$  and  $r_{high}$ 
10:   $c \leftarrow \text{maximum}(1, i - r)$ 
11:  for  $c \leq \text{minimum}(\text{size}(\mathbf{M}, 1), i + r)$  do
12:     $d \leftarrow \text{maximum}(1, j - r)$ 
13:    for  $d \leq \text{minimum}(\text{size}(\mathbf{M}, 2), j + r)$  do
14:      if  $\{(c = i) \text{ and } (d = j)\}$  then
15:         $m_2 \leftarrow \mathbf{M}(c, d)$ 
16:         $(x_c, y_d) \leftarrow$  corresponding physical location of the element at  $[c, d]$  from X.
17:         $l_2 = (x_c, y_d)$ 
18:         $F_{p_1, p_2} = \frac{m_1 \times m_2}{\|l_2 - l_1\|_2^3} \times (l_2 - l_1)$ 
19:         $F_{total} \leftarrow F_{total} + F_{p_1, p_2}$ 
20:         $d \leftarrow d + 1$ 
21:       $c \leftarrow c + 1$ 
22:     $\mathbf{a} \leftarrow \frac{F_{total}}{m_1}$ 
23:     $\mathbf{s} \leftarrow \frac{1}{2} \mathbf{a} (\sqrt{2})^2$ 
24:    return s
25: procedure MOVE(d, [i, j], M,  $m_1, l$ )
26:   if  $\|\mathbf{d}\|_2 \geq \frac{l}{2}$  then
27:     Move the particle to one of 8 adjacent square in the direction of force.
28:   return M
29: procedure UPDATE(M)
30:    $m_{max} \leftarrow \text{max}(\mathbf{M})$ 
31:    $[i, j] \leftarrow \text{size}(\mathbf{M})$ 
32:    $a, b \leftarrow 0$ 
33:   for  $a \leq i$  do
34:      $b = 0$ 
35:     for  $b \leq j$  do
36:       if  $\mathbf{M}(a, b) \leq \frac{m_{max}}{7}$  then
37:          $\mathbf{M}(a, b) = 0$ 
38:        $b \leftarrow b + 1$ 
39:      $a \leftarrow a + 1$ 
40:   return M

```

Algorithm 2 Gravity Inspired Algorithm

```

1: Create a matrix Mass such that each element in matrix represents mass of the particle.
2: X vector consists of the corresponding physical locations of the elements in the matrix Mass.
3:  $\text{nnz}(\mathbf{Y})$  - represents the number of non zero entries in matrix Y.
4:  $l \leftarrow$  length of the square's side
5: iteration  $\leftarrow$  0
6: for iteration  $\leq N \times 10$  do
7:   no-change = 0
8:   Mass-Previous = Mass
9:   while no-change  $\leq N \times 50$  do
10:     $z \leftarrow$  randomly pick a non-zero element from matrix Mass
11:     $[i \ j] \leftarrow$  indices of  $z$  w.r.t to matrix Mass
12:    s  $\leftarrow$  DISPLACEMENT( $[i \ j]$ , Mass, X)
13:    Mass  $\leftarrow$  MOVE(s,  $[i \ j]$ , Mass,  $z$ ,  $l$ )
14:    if Mass == Mass-previous then
15:      no-change = no-change + 1
16:    else
17:      Mass = UPDATE(Mass)
18:      no-change = 0
19:      Mass-previous = Mass
20:    if  $\text{nnz}(\mathbf{Mass}) = N$  then
21:      break
22:    else
23:      iteration = iteration + 1
24: Output the location of the non-zero elements of the Matrix Mass from the Matrix X.

```

3.3. General Application of GICA

We note that our proposed gravity inspired clustering algorithm can be applied much more generally to any data clustering task (i.e., not limited to TWR radar). For example, any system where measurements are being made from a finite number of sources or classes and where those measurements are inaccurate or noisy, and where there can be additional spurious measurements. This includes many more general classification problems.

4. Results

The range measurements in the presence of a wall is modeled as $\tilde{r}_{i,j,\phi_k} = r_{i,j,\phi_k} + \Delta r_{i,j,\phi_k}$ [2]. Where \tilde{r}_{i,j,ϕ_k} represents the range measurement corresponding to the i th target at j th receiver when the transmitter and receivers are at intermediate location ϕ_k in the presence of a wall. $\Delta r_{i,j,\phi_k}$'s are the i.i.d Gaussian random variables with mean μ and variance σ^2 ($\Delta r \sim \mathcal{N}(\mu, \sigma^2)$). r_{i,j,ϕ_k} represents the distance from the transmitter to the i th target and from i th target to the j th receiver at the ϕ_k^{th} intermediate location. Figure 7 is plotted for the scenario in Figure 1, i.e., 7-target scenario.

It is difficult to compare our proposed algorithm with existing algorithms because none have been proposed for the novel problem formulated in this paper. Given just the range measurements, a model would typically be needed for the locations of ellipse intersection points before a suitable algorithm could be applied, and we do not currently have such a model. As noted earlier, calculating intersection points of two rotated ellipses is a hard problem. The number of intersection points increases as the number of targets increases, which further increases the complexity. For the same seven target scenario, using "solve" function in MatLab to solve for all the intersection points took one day for just one realization. This shows that it is not practical. The simulation is performed on the desktop computer with 32 gigabytes of RAM and Intel Xeon @3.50 GHz processor.

Instead, we will first utilize our EH pre-processing step to obtain data consisting of the histogram of ellipse crossings over grid squares, and then consider alternatives to our

proposed gravity clustering algorithm. We consider two standard clustering algorithms: k-means and fuzzy c-means, and we also consider the alternative clustering algorithm using gravity proposed in [20]. In all cases, the measurements are taken at $M = 9$ equally spaced intermediate points between -5 and 5 m, as the system moves from left to right across 10 m.

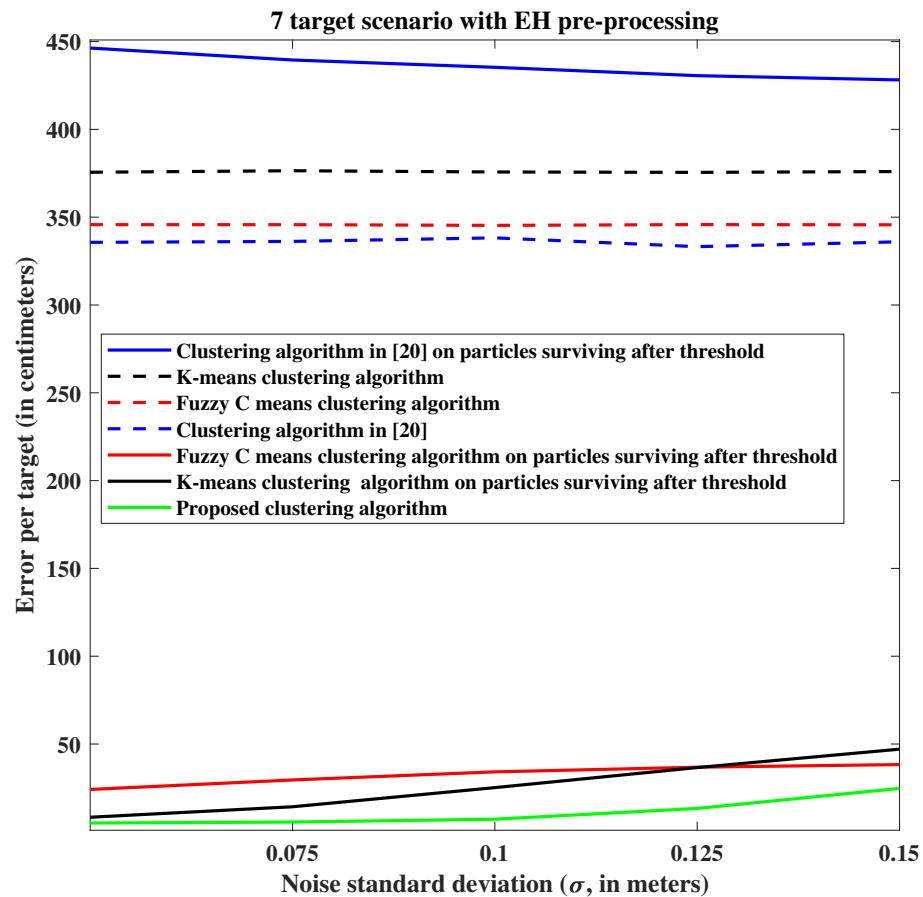


Figure 7. Performance comparison with proposed ellipse-histogram (EH) pre-processing.

Our proposed algorithm uses a thresholding operation (in EH pre-processing step) to remove those particles with mass less than half that of the largest mass particle. For the existing algorithms, we depict in Figure 7 the performance both with and without this initial thresholding operation. The dashed red curve represents the performance of the fuzzy c-means algorithm without thresholding, and the solid red curve represents the performance with thresholding. It can be seen from the results that thresholding improves the performance of the two standard clustering algorithms. Thresholding does not benefit the gravity-based clustering algorithm [20], and it appears that this algorithm does not work for our problem.

From Figure 7, we can see that proposed algorithm performs better than the other algorithms. For $\sigma = 0.1$, the proposed algorithm has 70% less error per target when compared to the best of the other three algorithms, which is either k-means or fuzzy c-means, depending on the level of noise. The proposed algorithm has an accuracy of 8 cm.

In simulations, the maximum number of iterations for convergence in the k-means and fuzzy c-means algorithm is set to 100. In [20], they used the following values for the parameters in their algorithm, $G_0 = 0.02$ (the initial gravity constant) and $T = 200$ (the number of iterations), which resulted in good performance for the datasets considered in that paper. However, we have observed that for the datasets we are considering in this paper, the gravity constant needs to be increased to $G_0 = 0.5$ in order for the centroids in

their algorithm to move, although, as remarked above, this is still not effective for finding the target locations.

For the seven target scenario in Figure 1, we compare the error performance of our algorithm for different threshold levels. Table 1 tabulates the error per target for different threshold levels. From the table, we can see that one-seventh has slightly better performance; however, we can see that performance is robust to the change in threshold levels. The results are for the noise level $\sigma = 0.1$.

Table 1. Impact of different threshold level on performance of the proposed algorithm.

Threshold	Error per Target (in cm)
1/5	8.72
1/6	7.75
1/7	7.69
1/8	7.76
1/9	8.47
1/10	8.07
1/11	8.22

Figure 8 shows that the proposed algorithm performs better than the other two, even when some of the measurements are missing. From the figure, we can see that as the percentage of missing measurement increases, the error per target also increases. Results show that the proposed algorithm outperforms the existing algorithms and produce better results. The results are for the noise level $\sigma = 0.1$.

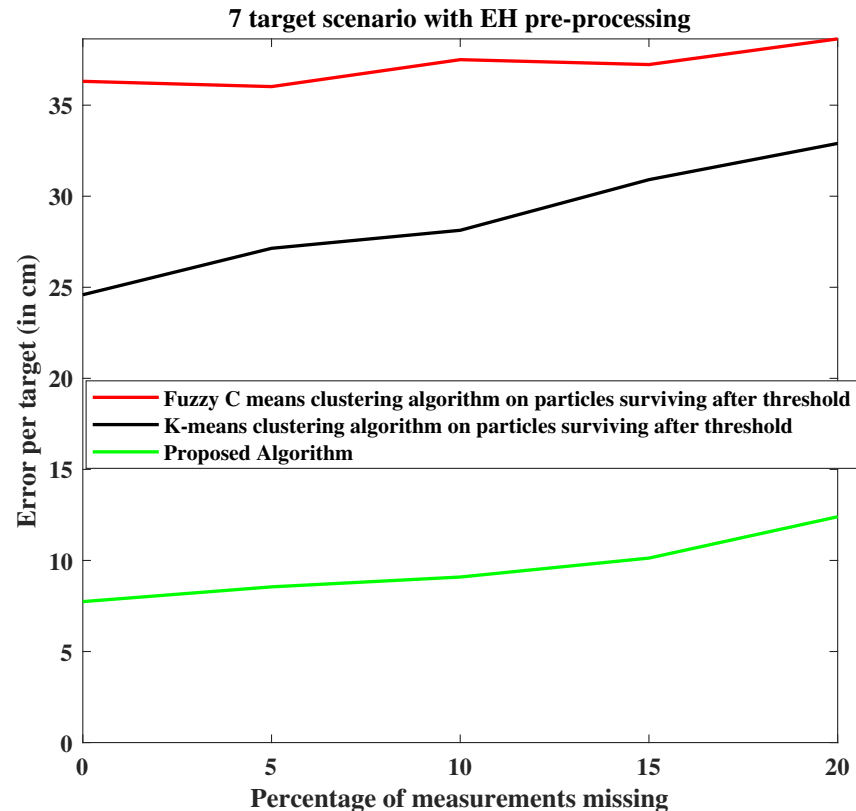


Figure 8. Performance comparison with missing measurements and for $\sigma = 0.1$.

To show the effect of the choice of square size's length on the proposed algorithm, we have tabulated the error performance for different grid square side's length. Table 2 shows

that the performance is robust to the changes but performs slightly better when the square side's length is 7 cm.

Table 2. Impact of the grid square side's length on the proposed algorithm.

Length of Grid Square (in cm)	Error per Target (in cm)
6	8.4
7	7.69
8	7.91
9	9.28
10	9
11	10

Figure 9 shows that the proposed algorithm performs better than the existing algorithm when some of the measurements are false measurements for the scenario in Figure 1 (i.e., 7-target scenario). False measurements arise when there is a strong reflection from the objects that are not targets (i.e., not of interest, e.g., reflection from the walls.) Figure 9 is plotted when there are no missing measurements. Figure 10 shows the proposed algorithms perform better than the existing algorithms when 25% of measurements are missing and contains false measurements. From the figures, we can see that as the percentage of false measurement decreases, error per target also decreases. We also see that in both the scenarios (with and without missing measurements), the proposed algorithm outperforms the existing algorithms. The simulations are performed for the noise level $\sigma = 0.1$.

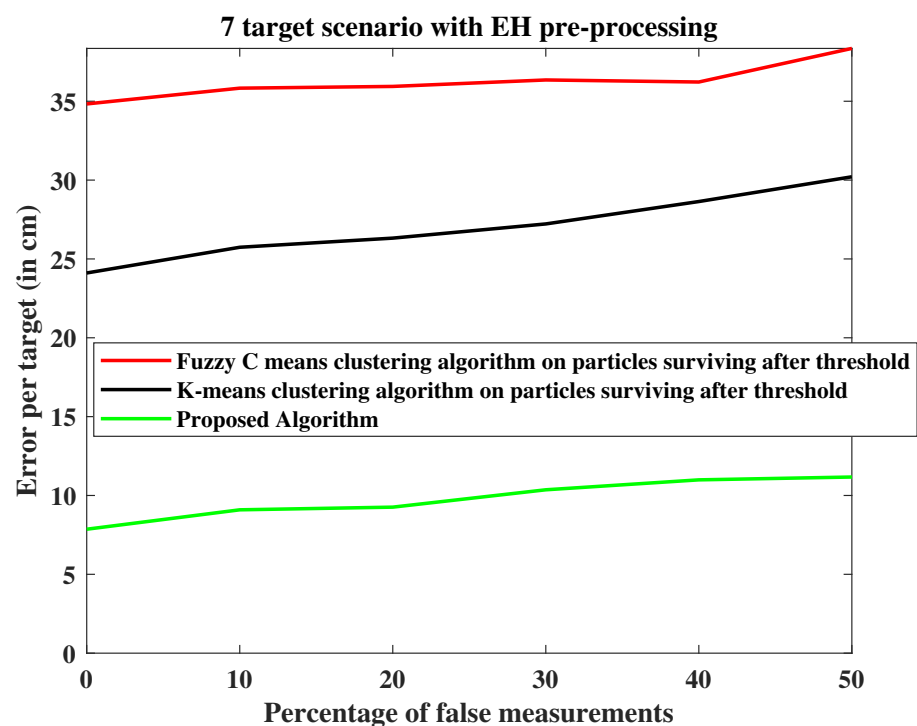


Figure 9. Performance comparison with false measurements and for $\sigma = 0.1$ and no missing measurements.

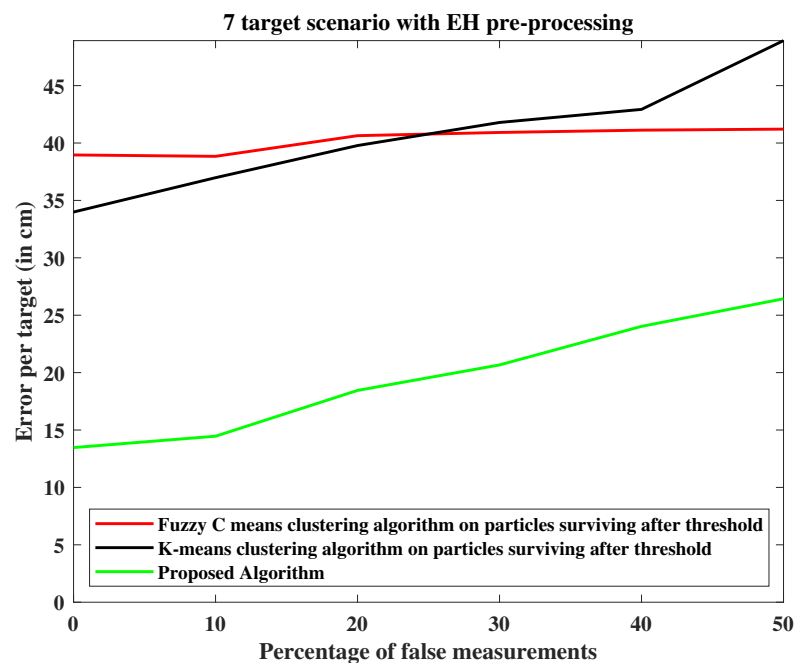


Figure 10. Performance comparison with false measurements and for $\sigma = 0.1$ and 25% of missing measurements.

Figure 11 shows the performance of the proposed algorithm for a different number of intermediate locations. From the figure, we can see that the proposed algorithm performs better than existing algorithms. The figure shows that as the number of intermediate locations increases, the error per target decreases.

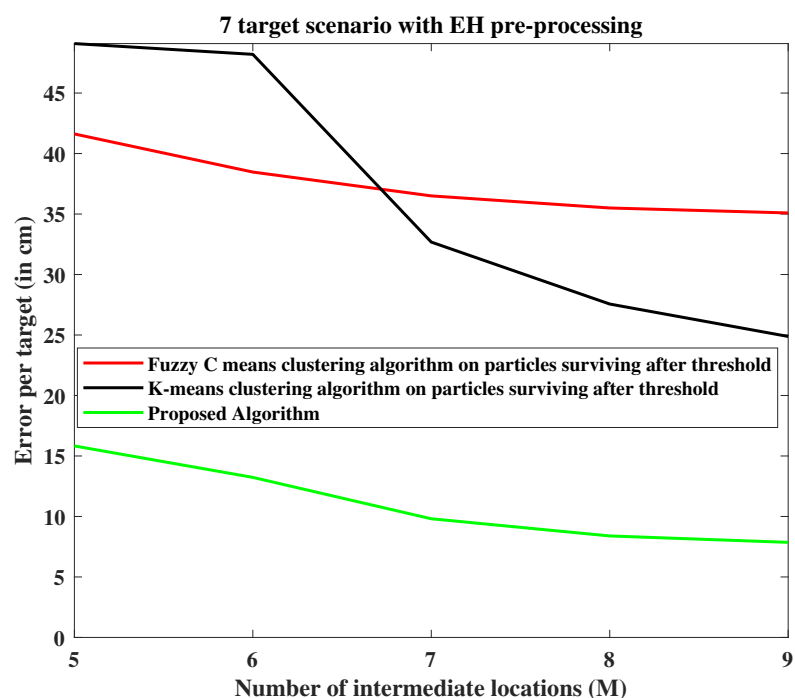


Figure 11. Performance comparison with different number of intermediate locations for $\sigma = 0.1$.

Figure 12 shows that the proposed algorithm performs better than existing for the different number of receivers for the scenario in Figure 1 (i.e., 7-target scenario). From the figure, we can see that error per target decreases as the number of receivers increases. The

simulations are done for the noise level, $\sigma = 0.1$ and number of intermediate locations, $M = 7$.

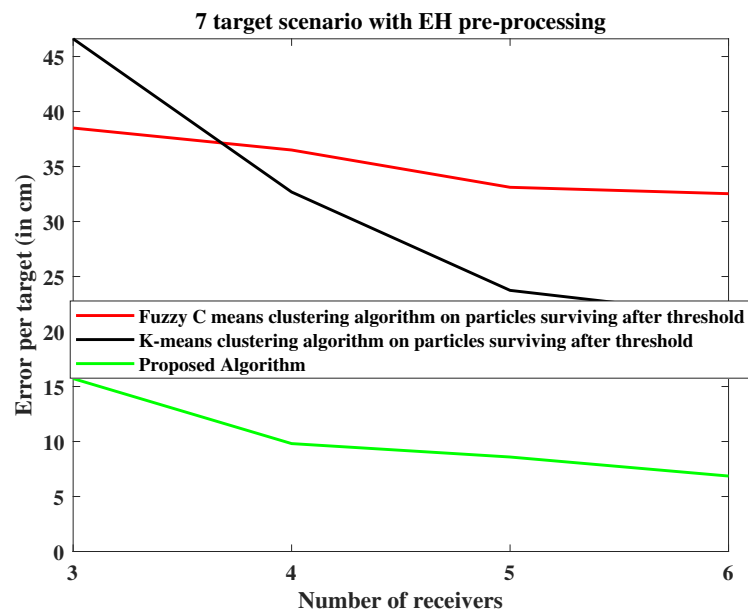


Figure 12. Performance comparison with different number of receivers for $\sigma = 0.1$ and number of intermediate locations, $M = 7$.

To illustrate how the proposed algorithm progresses, we have shown different stages of the algorithm in Figures 13–16, for a 9 target scenario with noise level $\sigma = 0.1$ and 10% of measurements are missing. From the figures, it can be seen that as the proposed algorithm progress, the particles at non-target locations are removed iteratively. It can also be seen that as the algorithm progresses, particles around the target locations merge together (mass of the particles increasing from Figures 13–16).

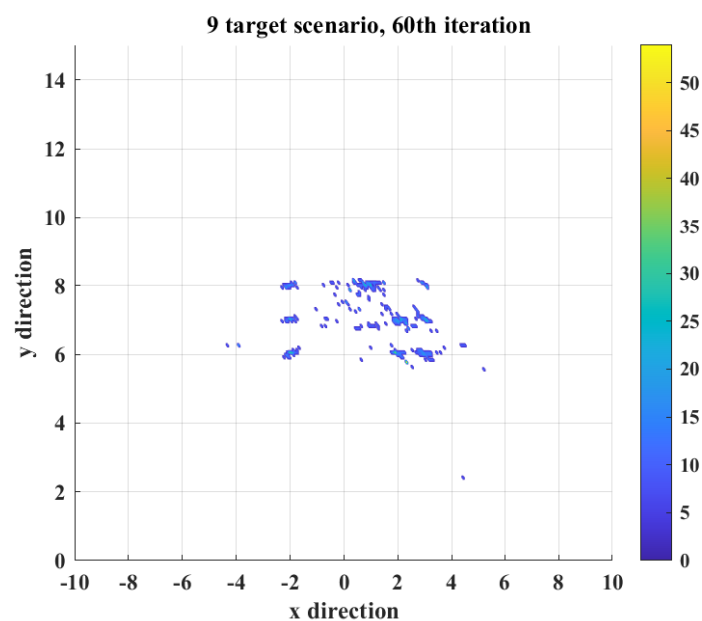


Figure 13. Surviving particles of the proposed algorithm at the end of 60th iteration. Mass of the particles is represented using the color. The simulation is performed for a 9 target scenario with 10% of missing measurements and with noise level $\sigma = 0.1$. The color scale indicates the mass of the particles.

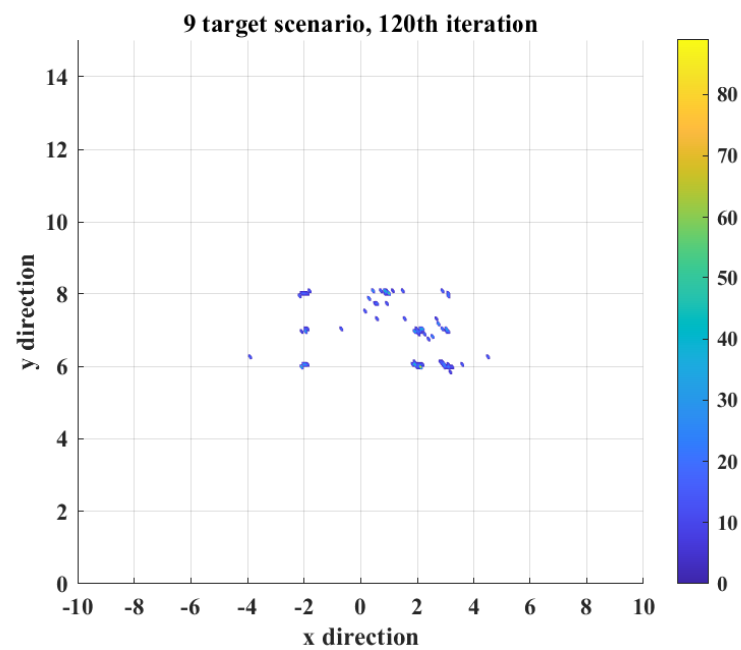


Figure 14. Surviving particles of the proposed algorithm at the end of 120th iteration for the same scenario as in Figure 13. The color scale indicates the mass of the particles.

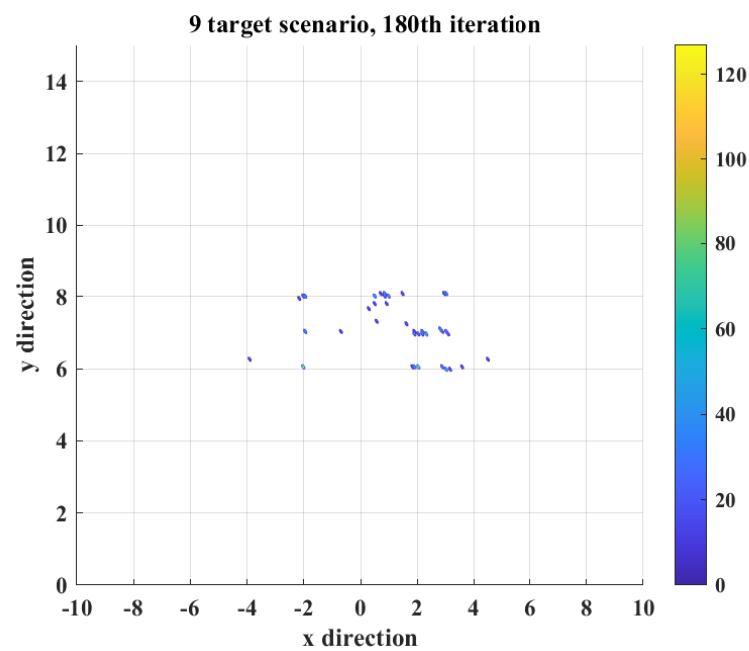


Figure 15. Surviving particles of the proposed algorithm at the end of 180th iteration for the same scenario as in Figure 13. The color scale indicates the mass of the particles.

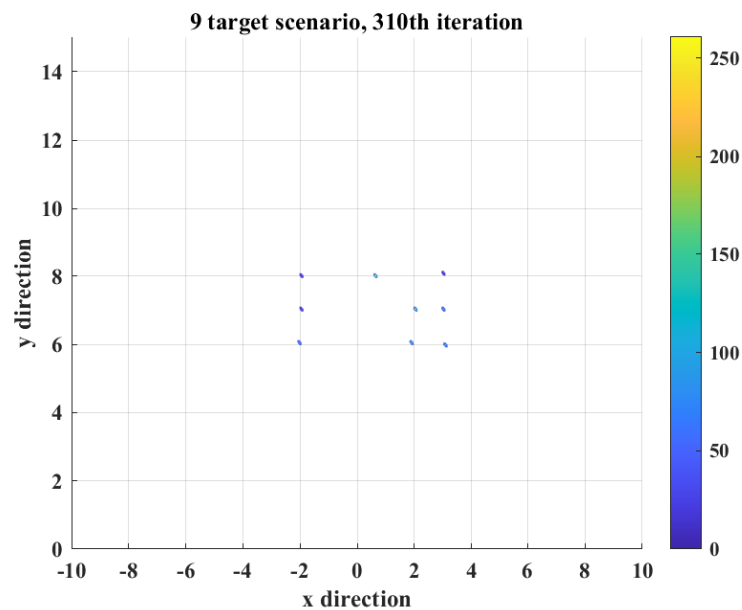


Figure 16. Surviving particles of the proposed algorithm at the end of 310th iteration for the same scenario as in Figure 13. The color scale indicates the mass of the particles.

For a range of different 7-target, 8-target, and 9-target scenarios simulated, on average, error per target is 7.5 cm, 8 cm, and 11 cm.

5. Conclusions

This paper presented a gravity inspired algorithm for non-coherent localization of multiple targets. The proposed algorithm iteratively groups particles at target locations and eliminates particles at non-target locations. The overall complexity of our proposed approach is limited by the number of grid squares and does not increase exponentially with the number of measurements. We showed that the proposed algorithm can be applied even when some of the measurements are unavailable or missing, and when some of the measurements are false measurements. Location accuracy of the proposed algorithm is shown to be in the order of 8 cm.

Author Contributions: I.M., I.B.C., S.V.H. contributed to the problem formulation and algorithm development; I.M. wrote the simulator code; I.M., I.B.C., S.V.H. authors contributed to results analysis and findings; I.M., I.B.C., S.V.H. authors contributed to writing and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Narayanan, R.; Gebhardt, E.; Broderick, S. Through-Wall Single and Multiple Target Imaging Using MIMO Radar. *Electronics* **2017**, *6*, 70. [[CrossRef](#)]
2. Huang, X.; Cui, G.; Guo, S.; Kong, L.; Yang, X. A novel noncoherent localization approach for through-the-wall radar. In Proceedings of the 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, 8–12 May 2017; pp. 1245–1249.
3. Ahmad, F.; Amin, M.G. Noncoherent approach to through-the-wall radar localization. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 1405–1419. [[CrossRef](#)]
4. Jia, S.; Kong, L.; Liu, B. Ellipse-cross-localization accuracy analysis of through-the-wall radar. In Proceedings of the 2009 IEEE Radar Conference, Pasadena, CA, USA, 4–8 May 2009; pp. 1–4.
5. Yuan, X.; Kong, L.; Jia, Y. Through-wall-radar target localization based on random sparse array. In Proceedings of the 2011 IEEE CIE International Conference on Radar, Chengdu, China, 24–27 October 2011; Volume 2, pp. 1331–1334.
6. Sostanovsky, D.; Boryssenko, A. Experimental UWB sensing & communication system. *IEEE Aerosp. Electron. Syst. Mag.* **2006**, *21*, 27–29.

7. Dilsavor, R.; Ailes, W.; Rush, P.; Ahmad, F.; Keichel, W.; Titi, G.; Amin, M. Experiments on wideband through-the-wall radar imaging. In *Algorithms for Synthetic Aperture Radar Imagery XII*; International Society for Optics and Photonics: Bellingham, WA, USA, 2005; Volume 5808, pp. 196–209.
8. Benjamin, R.; Craddock, I.; McCutcheon, E.; Nilavalan, R. Through-wall imaging using real-aperture radar. In Proceedings of the URSI General Assembly, Maastricht, The Netherlands, 17–24 August 2002.
9. Hunt, A.R. Image formation through walls using a distributed radar sensor network. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IV*; International Society for Optics and Photonics: Bellingham, WA, USA, 2005; Volume 5778, pp. 169–174.
10. Lai, C.P.; Narayanan, R.M. Through-wall imaging and characterization of human activity using ultrawideband (UWB) random noise radar. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IV*; International Society for Optics and Photonics: Bellingham, WA, USA, 2005; Volume 5778, pp. 186–195.
11. Immoreev, I.Y.; Samkov, S.; Tao, T.H. Short-distance ultra wideband radars. *IEEE Aerosp. Electron. Syst. Mag.* **2005**, *20*, 9–14. [[CrossRef](#)]
12. Falconer, D.G.; Steadman, K.N.; Watters, D.G. Through-the-wall differential radar. In *Command, Control, Communications, and Intelligence Systems for Law Enforcement*; International Society for Optics and Photonics: Bellingham, WA, USA, 1997; Volume 2938, pp. 147–151.
13. Chen, J.C.; Yao, K.; Hudson, R.E. Source localization and beamforming. *IEEE Signal Process. Mag.* **2002**, *19*, 30–39. [[CrossRef](#)]
14. Hayes, M.P.; Gough, P.T. Broad-band synthetic aperture sonar. *IEEE J. Ocean. Eng.* **1992**, *17*, 80–94. [[CrossRef](#)]
15. Murino, V.; Regazzoni, C.S.; Trucco, A.; Vernazza, G. A noncoherent correlation technique and focused beamforming for ultrasonic underwater imaging: A comparative analysis. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **1994**, *41*, 621–630. [[CrossRef](#)]
16. Paradie, M.J.; Labitt, B.D. Method and Apparatus for Identifying Complex Objects Based on Range Readings from Multiple Sensors. US Patent 6,664,918, 16 December 2003.
17. Mohammed, I.; Collings, I.B.; Hanly, S.V. Multiple Target Localization Through-the-Wall using Non-Coherent Bi-static Radar. In Proceedings of the 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, Australia, 16–18 December 2019; pp. 1–8.
18. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [[CrossRef](#)]
19. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [[CrossRef](#)]
20. Bahrololoum, A.; Nezamabadi-pour, H.; Saryazdi, S. A data clustering approach based on universal gravity rule. *Eng. Appl. Artif. Intell.* **2015**, *45*, 415–428. [[CrossRef](#)]

Short Biography of Authors



Imran Mohammed received the B.Tech degree in electrical and electronics engineering from the Rajiv Gandhi University of Knowledge Technologies (RGUKT) Nuzvid in 2015, the M.Tech degree in electrical engineering from the Indian Institute of Technology (IIT) Kanpur in 2017, and the MRes degree from Macquarie University, Sydney in 2018. He is currently working towards the Ph.D. degree in the school of engineering with Macquarie University, Sydney, Australia. His research interests include optimization, localization, and UAV deployment analysis.



Iain B. Collings (Fellow, IEEE) received the B.E. degree in Electrical and Electronic Engineering from the University of Melbourne in 1992, and the Ph.D. degree in Systems Engineering from the Australian National University in 1995. Currently he is Deputy Dean Research, School of Engineering, Macquarie University, Sydney Australia. Prior roles include Deputy Chief (Research), Computational Informatics Division, Australian Commonwealth Scientific and Industrial Research Organization; and academic positions at the University of Sydney and the University of Melbourne. He has published over 340 research papers on wireless digital communications. In 2009, he was awarded the Engineers Australia IREE Neville Thiele Award for outstanding achievements in engineering, and in 2011 he was a recipient of the IEEE CommSoc Stephen O. Rice Best Paper Award for IEEE Transactions on Communications. Prof. Collings served as an Editor for the IEEE Transactions on Wireless Communications (2002–2009), the Elsevier Physical Communication Journal PHYCOM (2008–2012), and Sensors (2020–).



Stephen V. Hanly (Fellow, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees from the University of Western Australia and the Ph.D. degree from Cambridge University, U.K. He was a Post-Doctoral Researcher and member of Technical Staff at AT&T Bell Laboratories, Murray Hill, USA, then on the Research and Teaching Staff at the University of Melbourne, and, later, the National University of Singapore. Since 2012, he has been a Professor at Macquarie University. He is a recipient of the INFOCOM Best Paper Award, the IEEE Information Theory Society and the IEEE Communication Society Joint Paper Award, and the IEEE Communications Society Tutorial Paper Award. He has been an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and a Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS twice. He has taken major roles at several IEEE conferences and workshops, including IEEE ISIT, and IEEE CTW.