Scalar Waves: Designing Perception Frameworks and Path Planning for Swarm Robot Systems Operating in a Scalar Field

Poondru Prithvinath Reddy

October 18, 2024

Scalar Waves: Designing perception frameworks and path planning for swarm robot systems operating in a scalar field

Poondru Prithvinath Reddy

ABSTRACT

Scalar waves hold the potential to revolutionize computing and communications by offering advantages like instantaneous data transmission and immunity to electromagnetic interference. In this paper, we will explore the world of scalar waves and also their potential to reshape the future of computing, and robotics. In our work, we used scalar fields as a global level of guidance for the robots in operation. A scalar field associates a value with every point in the working region. Here we show how scalar fields can be used to guide swarm of robots to execute a specific task. We present an example of tasks using the scalar field, such as constructing shapes from ambient objects, which involves finding and collecting ambient objects to the collection area and aggregating to desired shapes. First, we show the scalar field mapping with scalar function using SLAM algorithm to guide the robots in their movements. Second, the development of AR model of a sensor data by integration to achieve detailed object sensing of robots. Here, each robot to find an estimate of material location and an estimate of the value of the scaler field is iteratively updated during the movement of the robot and a central control command is also updated regarding the mobile robots to form a network and cover the field. Also, entire scalar field is used it as a pathway network among the swarm of robots to perform the aggregation task. Experiments have been conducted to demonstrate the workability of the proposed algorithms.

INTRODUCTION

Scalar waves are often described as standing waves, meaning they do not move through space but exist as stationary patterns of energy. Unlike conventional EM waves, scalar waves are believed to be non-

Hertzian, meaning they do not travel through space in the same way as traditional electromagnetic waves. Scalar waves are thought to exist beyond the limitations of Hertzian waves, which have a specific frequency and wavelength. Scalar waves are believed to have zero frequency, meaning they do not oscillate in the traditional sense. This property allows them to transcend the constraints of space and time.

Scalar waves are hypothetical waves, which differ from the conventional electromagnetic transverse waves by one oscillation level parallel to the direction of propagation, they thus have characteristics of longitudinal waves. Scalar waves are superluminal, which means they move faster than the speed of light, because they are unbounded by the limitations of 3D space. Also, since they don't exist in the third dimension in the same way that matter does, they move through the empty space between all matter.

A scalar quantity is defined as the physical quantity that has only magnitude. On the other hand, a vector quantity is defined as the physical quantity that has both magnitude as well as direction. Scalar, a physical quantity that is completely described by its magnitude. Temperature is a scalar quantity as it is independent of direction at a point. Wavelength is a scalar quantity. It has magnitude but not direction. Examples of scalars are volume, density, speed, energy, mass, and time. Other quantities, such as force and velocity, have both magnitude and direction and are called vectors.

Mechanical are those which propagate through a physical matter such as sound wave, ocean wave or earthquake wave while electromagnetic wave do not require a medium and they can even travel through vacuum such as light wave. As electromagnetic wave does not need a medium, it travels faster than mechanical wave.

METHODOLOGY

Scalar Field

A scalar field is a name we give to a function defined in some sort of space. Thus, in ordinary three dimensional space the following are examples of scalar fields: sin xyz, cos z, $x^2 + y^2 + z^2$. A linear field is one of the form ax + by + cz + d for some constants a, b, c and d.
The difference between a scalar field and a scalar function:
For example, given a fixed point A, f(P)= Distance of P from A, is an example of a scalar field, because distances between points do not

change under rotation of axes or shifting of the origin. On the other hand, f(x,y,z)=(x+y) is a scalar function, which is not a scalar field. Therefore, the unit of scalar field: Under this convention the units of the scalar field physical quantities are [ϕ] = $(\text{energy/length})^{1/2}$ and [˜ma]=length−1 [m ~ a] = length - 1 .

Scalar Field Function

In our work, we used scalar fields as a global level of guidance for the robots. A scalar field associates a value with every point in the working region. We show how scalar fields can be used to guide a low-cost and limited-capability swarm of robots to execute a specific task. We present examples of tasks using the scalar field, such as constructing shapes from ambient objects, finding the connected network among the robots, aggregating to a predefined area and foraging by finding and collecting ambient objects to the collection area and finally, aggregating to a predefined work area. This work is divided into two parts; first, we show how the scalar field can help divide labour among the robots and guide them in their movements. Second, we investigate combining a scalar field with augmented reality to overlay real-life objects that enhances the real world with computer-generated perceptual information. Finally, we design scalar field mapping and use it as a road network for the swarm to reduce spatial interference among the robots. We practically build our robots based on the practical examples to perform the aggregation task.

Robotics

Scalar Field and their gradients, which are vector fields, can be used in robotics for motion planning. Consider a robot which needs to move in an area to a desired point avoiding some obstacles. The so-called navigation function is constructed for this purpose which is a continuously differentiable scalar field defined on the obstacle free inside of the area, has a unique minimum at the goal point and attain its maximum value at the boundary of the operational area and the obstacles. A robot moving in the direction of the gradient of the navigation function can avoid obstacles and reach to the goal without hitting.

Navigation function usually refers to a function of position, velocity, acceleration and time which is used to plan robot trajectories through the environment. Generally, the goal of a navigation function is to create feasible, safe paths that avoid obstacles while allowing a robot to move from its starting configuration to its goal configuration.

For robots, like drones or self-driving cars, Euclidean distance helps calculate the simplest route from one point to another. This helps robots and other automated systems move efficiently and safely, avoiding obstacles and calculating the easiest paths to their destinations.

Let us consider an operational area of spherical shape centered at the target point $q_0 = (x_0 , y_0)^r$ with radius r_0, with three obstacles located at the spherical level with radii r_1, r_2 and r_3 and centres $q_1 = (x_1 , y_1)^r$, $q_2 = (x_2 , y_2)^r$, $q_3 = (x_3, y_3)^r$. In mathematics, the Euclidean distance is defined as **the distance between two points**. In other words, the Euclidean distance between two points in the Euclidean space is defined as the length of the line segment between two points. Let $d(q_a, \ q_b)$ denote the Euclidean distance between $q_a = (x_a, y_a)^r$ and $q_b = (x_b, y_b)^r$, namely,

$$d\left(q_a , \ q_b\right) = \sqrt{\ (x_a - \ x_b)^2 + (\ y_a - y_b)^2}$$

The navigation function can then be constructed as

$$\varphi(q) = \frac{d(q, \ q_0)^2}{[\ d\left(\ q, q_0\right)^{2k} + \beta(q)]^{1/k}}$$ where k is the large enough positive number

And $\beta_i = d(q, q_i)^2 - r_i^2$ for i = 1,2,3. The number β_i should be the same as number of objects.

SLAM algorithm with Augmented Reality

SLAM is a technology used in vision technologies to get the visual data from the physical environment in the form of points and dots to feed the data into machines. **SLAM** provides an optical input for devices and computers, making them understand what is going on in the physical world. This data also helps **AR (Augmented Reality)** developers to create interactive and realistic experiences for the audience or the central command. The technology can be used in different scenarios like self-driving cars, games, robotics, artificial intelligence, and virtual reality.

The simplest form of SLAM technology is understanding the floor, barriers, and walls. Currently, most AR SLAM technologies use floor recognition and position tracking to place AR friendly objects around the working area. Advanced SLAM technologies like Google Tango create a web of the real-time environment and notify us about the floor, walls, and objects in the environment allowing everything around us to act as an intractable element.

Visual simultaneous localization and mapping (vSLAM) algorithms use device camera to estimate agent's position and reconstruct structures in an unknown environment. As an essential part of augmented reality (AR) experience, vSLAM enhances the real-world environment through the addition of virtual objects, based on localization (location) and environment structure (mapping). From technical perspectives, visual SLAM algorithm proposed in this paper cater to its applications in augmented reality, mapping, navigation, and localization.

SLAM algorithms to track the robot’s position and overlay ambient objects onto the real world. Augmented reality is an interactive experience that enhances the real world with computer-generated perceptual information. Augmented reality works by overlaying digital objects, information, or other sensory elements on top of the physical world to provide users with a beneficial, or informative experience and using software, augmented reality overlays digital content onto real-life environments and objects. SLAM is the foundation of augmented reality (AR.) It allows AR devices to perceive the world in three dimensions. AR software can then identify objects or images in the real-world environment and project virtual content on the AR displays so it appears in the real world.

Simultaneous localization and mapping (SLAM) algorithms come from robotics research and provide a geometric position for the AR system. SLAM algorithms can build 3D maps of an environment while tracking the location and position of the robot camera in that environment.

Most modern AR devices, come with built-in cameras that can be used for AR applications. Some AR devices may have specialized cameras that provide more advanced tracking and sensing capabilities, such as depth sensing or infrared sensors. AR devices are less restrictive and typically include devices like **phones, glasses, projections and HUDs**.

SLAM algorithm and its application for AR, mapping, localization and wayfinding

Simultaneous localization and mapping (SLAM) algorithms come from robotics research and provide a geometric position for the AR system. SLAM algorithms can build 3D maps of an environment while tracking the location and position of the camera in that environment

SLAM is the estimation of the pose of a robot and the map of the environment simultaneously. SLAM is hard because a map is needed for localization and a good pose estimate is needed for mapping

- **Localization**: inferring location given a map.
- **Mapping**: inferring a map given locations.
- **SLAM**: learning a map and locating the robot simultaneously.

SLAM has multiple parts and each part can be executed in many different ways:

- Landmark detection
- Data association
- State Estimation
- State Update
- Landmark Update

SLAM process consists of the following steps:

- In the first step, it uses the environment to update the position of the robot. We can use Odometry or laser scans of the environment to correct the position of the robot.
- Thus, the position of the robot can be better identified by extracting features from the environment.

Extended Kalman Filter

The Extended Kalman Filter (EKF) is the core of the SLAM process. It is an estimation of non-linear processes or measurement relationships. It is responsible for updating where the robot thinks it is based on the Landmarks.

Laser and Odometry data

Laser data is the reading obtained from the scan whereas, the goal of the odometry data is to provide an approximate position of the robot.

Landmarks

Landmarks are the features that can easily be re-observed and distinguished from the environment. These are used to localize the robot. Landmark should be easily available, distinguishable from each other, should be abundant in the environment and stationary

Landmark Extraction

After selecting and deciding on the landmarks, we need to extract landmarks from inputs of robot sensors. The basic landmark extraction used by randomly sampling the readings and then using the using a least-squares approximation to find the best fit that runs through these readings.

Data Association

Data association or data matching is that of matching observed landmarks from different (laser) scans with each other.

There are few approaches to perform data association, we will be using the nearest neighbour algorithm:

- First, when we get the data from the laser scan use landmark extraction to extract all visible landmarks.
- After that, we associate all the extracted landmarks to the closest landmark that can be observed >N times.
- Now, we input the list of extracted landmarks and list of previously detected landmarks that are in the database, if the landmark is already in the database then, we increase their count by N, and if they are not present then set their count to 1.

After the above step, we need to perform the following update steps:

- **State Estimation:** In this step, we use the odometer data to get the current state estimate.
- **State update:** In this stage, we update our new estimated state by re-observing landmarks.
- **Landmarks update:** In this step, we add new landmarks that are detected in current stage.
- **Scalar Field Function Update:** In this step, we update scalar field value
- **Landmarks augmentation:** In this step, we add augmented reality visualizations to landmarks that are detected

ARCHITECTURE

IMPLEMENTATION OF SLAM

There are many ways to implement a solution for **SLAM** (Simultaneous Localization and Mapping), but the algorithm choosen to implement SLAM is based on Graph theory..

SLAM is a technique used in robotics to simultaneously estimate a robot's trajectory over time and estimate the positions of landmarks or objects in the environment by representing them as nodes and edges as a graph network. The graph consists of nodes representing robot poses at different time steps and landmark positions, and edges representing the constraints between them, such as initial location, relative motion, and relative measurement constraints. By optimizing the network, SLAM aims to find the most likely trajectory and object(s) positions that best explain the sensor measurements.

- **Graph Model Representation**

In SLAM, the environment and the robot's trajectory are depicted using a graph structure. The graph consists of nodes and edges, where nodes represent the robot's poses (positions and orientations) at different points in time, and edges represent the constraints or measurements between these poses.

In addition to representing the robot's poses, the network also includes nodes that represent the landmarks in the environment. These landmarks can be objects, points of interest, or any other distinctive features that the robot can perceive and use for mapping.

The graph representation connects the robot's poses and landmarks through edges, which represent the measurements obtained from sensors. These measurements can include range measurements, or any other type of sensor data that provides information about the relative positions and orientations of the robot and the landmarks.

By incorporating these measurements into the graph network, the SLAM algorithm can estimate the most likely trajectory of the robot and the map of the environment that best satisfies the constraints imposed by the measurements. The graph optimization process involves adjusting the poses and landmark positions in the graph network to minimize the errors between the predicted measurements and the actual measurements obtained from the sensors.

- **Matrix and Vector Depiction**

In SLAM, we use matrix and vector representations to model the relationships between robot poses and landmarks in a map. These representations help us solve the SLAM problem.

In Graph SLAM, we create a matrix called the information matrix. This matrix represents the relationships between different variables in the SLAM problem. Each variable corresponds to a robot pose or a landmark in the map. The information matrix is a square matrix, and its size depends on the number of variables in the SLAM problem. The elements of this matrix encode the information about the relationships between variables.

Now, let's move on to the vector representation. In Graph SLAM, we also create a vector called the information vector. This vector represents the measurements or observations we have made in the SLAM problem. Each element of the vector corresponds to a specific measurement or observation. The information vector contains the information about the measurements and their relationships with the variables in the SLAM problem. It helps us incorporate the measurements into the SLAM problem and update our estimates of the robot poses and landmarks.

- **SLAM System in Graph Mode**

Once we declared our information matrix and vector, then we need to apply the initial location constraint to the matrix and vector. For example, to update the information matrix with an initial location, we would create a simple linear equation: Then we take simple linear equation and its coefficients and add it the row corresponding to location constraint.

- **Graph Optimization**

Once we have have the initial matrix and vector representations of the graph, we need to perform graph optimization. Graph optimization in Graph SLAM is the process of refining the estimates of robot poses and landmark positions by iteratively updating the graph based on sensor measurements. It involves two main steps: the measurement update and the state update. In the measurement update step, we iterate over the edges of the graph and add constraints into the information matrix. These constraints represent the measurements obtained from sensors, such as range measurements or bearing measurements. In the state update step, we solve a system to estimate the optimal robot poses and landmark positions that minimize the error in the graph. This is done by taking the inverse of the information matrix and multiplying it by the information vector.

Measurement Update: In the Measurement Update, it uses the Graph data to define the information matrix and vector data and the information matrix that represents the coefficients of the linear equations, whereas the information vector that represents the constant terms in those equations, and the graph that contains the edge weights representing the measurements (e.g., distance).

State Update: In the state update, we need to performs the state update by multiplying the inverse of the covariance matrix with the measurement vector. The result, therefore, represents the state estimate of the robot's pose and landmark positions. It is a vector that contains the values of all the variables that define the state of the system.

- **Scalar Field Update**

In the scalar field update step, we solve a system to estimate the operational field based on the landmarks that minimize the movement and interference in the connected network.

- **Landmarks Visualizations**

In this step, we use visualization tools to augment the landmarks and to create porotypes for easily rearranging them into desired shapes.

RESULTS

The Graph SLAM algorithm has not given exactly the correct answer for the simulated experiments, but it was within range to predicted outcomes. Therefore, the result of the SLAM algorithm depends on the quality of measurements that we feed the algorithm for achieving right fit.

Also the results were based on just a simple one-dimensional reproduction system however, this can be easily extended into 3-dimensional space to explain exactly which directional path the robot is moving in.

CONCLUSION

Scalar waves hold the potential to revolutionize computing and robotics by offering numerous advantages besides immunity from electromagnetic interference. In this paper, we used scalar fields as a global level of guidance for the robots in operation as scalar field associates a value with every point in the working region and we have shown how scalar fields can be used to guide swarm of robots to execute a specific task. This paper proposed a scalar field with scalar function using SLAM algorithm that drives a team of robots to explore ambient objects and learn to move in scalar field to guide them in their movements and the development of AR model of a sensor data by integration to achieve detailed object sensing of robots. The algorithm is based on a graph-inspired SLAM approach for distributed ambient objects seeking problems. And Our algorithm leverages a Central Command Process model to predict scalar field values as robots explore. By measuring the field values, agents move along the gradient of the field model while simultaneously improving the SLAM process model. The experimental results in simulation using vector robots movement and updating of scalar field values demonstrate our approach.

REFERENCE

1. **Hugh Durrant-Whyte, Fellow, IEEE, and Tim Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. https://people.eecs.berkeley.edu › readings › Durr...**