

Scalable underwater assembly with reconfigurable visual fiducials

Samuel Lensgraf, Ankita Sarkar, Adithya Pediredla, Devin Balkcom, Alberto Quattrini Li
Dartmouth College

Abstract—We present a scalable combined localization infrastructure deployment and task planning algorithm for underwater assembly. Infrastructure is autonomously modified to suit the needs of manipulation tasks based on an uncertainty model on the infrastructure’s positional accuracy. Our uncertainty model can be combined with the noise characteristics from multiple sensors. For the task planning problem, we propose a layer-based clustering approach that completes the manipulation tasks one cluster at a time. We employ movable visual fiducial markers as infrastructure and an autonomous underwater vehicle (AUV) for manipulation tasks. The proposed task planning algorithm is computationally simple, and we implement it on AUV without any offline computation requirements. Combined hardware experiments and simulations over large datasets show that the proposed technique is scalable to large areas.

I. INTRODUCTION

Autonomous assembly of structures using drones or free-floating robots is a promising direction for creating rapidly deployable, flexibly designed structures [1]. In most real-world systems, localization relative to a reference is achieved using calibrated and fixed positional infrastructure such as motion capture systems or visual fiducials [2]–[5]. Unfortunately, these systems are not scalable as the coverage area is fixed and scaling beyond the coverage area requires redesigning the positioning technology.

To overcome the limited coverage area of the positioning technologies, we propose to design the positioning infrastructure as a dynamic component of the construction plan. Our method allows localizing against large structures with minimal modification to the area around them and can also be integrated with existing underwater construction structures to make them scalable – see Fig. 1 that shows our robot in action while moving a fiducial marker.

Localization infrastructure is often considered to have a constant noise distribution, allowing coverage algorithms to plan based on the noise properties of the fixed infrastructure. However, the properties of localization infrastructure often depend on environmental factors: distance from the infrastructure, reflections, or water temperature gradients influence the accuracy of the positioning systems [6]. As the accuracy depends on relative positioning between the infrastructure and the robot, high-accuracy positioning can only be provided in a small fixed area resulting in either large infrastructure requirements, or small structures. Instead,

This project was partially supported by the NSF GRFP, CNS-1919647, 2024541, 2144624.

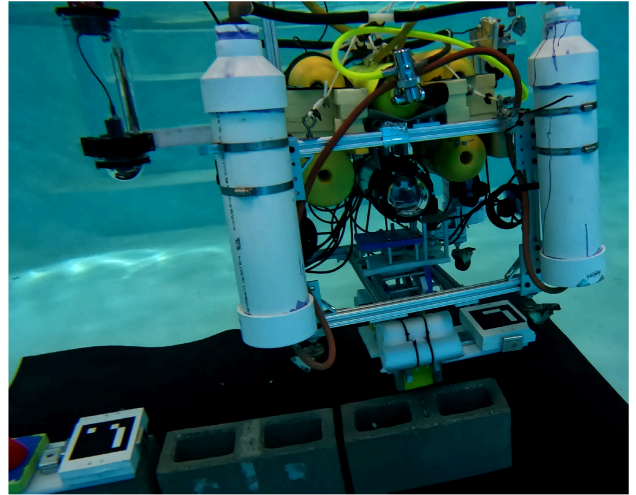


Fig. 1: AUV placing a reconfigurable fiducial marker on a foundation while localizing using another marker following our proposed method.

by understanding the noise properties of the infrastructure components and modeling them accurately as a function of environmental factors, we show that it is possible to dynamically reconfigure the infrastructure to maximize the positioning accuracy of any region. Our noise formulation model is also conducive to sensor fusion techniques and, hence, can be extended for the case of multiple sensors.

For a given construction structure manipulation task, we have to plan the movement of infrastructure such that the repositioned infrastructure guarantees the accuracy of the manipulation task. This planning is challenging and results in the combined deployment and sequencing problem.

To solve this problem, we group the manipulation tasks using a clustering algorithm that guarantees that the radius of any cluster is within the high localization accuracy achievable with the dynamic markers. We reposition the markers to have high accuracy for the cluster and execute the manipulation tasks, one cluster at a time.

For implementation, we use visual fiducial markers mounted on plates that are movable on our previously developed error-correcting construction foundations [7]. The AUV views these markers with a downward-facing wide-angle camera and can compute the position accuracy using our noise model. The implementation of our clustering-based planning algorithm is computationally simple, and we implemented it on the AUV itself without any external cloud computing requirements. We validated the proposed technique both experimentally and on large simulation datasets.

Our technique is particularly interesting for the case where

deploying large amounts of infrastructure is impractical or expensive. In real-world scenarios, it is often the case as the sparse area of interest is often near a protected region that we cannot permanently mar. For example, in the construction of artificial reefs, the surrounding areas are critically important to protect. In hard to reach places such as caves, or very deep waters, transporting large amounts of infrastructure can be impossible. Our technique is a foundational component towards a scalable solution for these cases. A number of practical challenges, such as flexibility in terms of material of connected components per layer, will be subject of future studies as discussed at the end of the paper.

II. RELATED WORK

Uncertainty-based planning. Infrastructure placement for small scale scenes is a well-studied problem. In particular, Magnago *et al.* [8] define a landmark placement algorithm that computes a static landmark placement based on certainty requirements. We consider the problem of dynamically altering landmark placements. We also develop a direct model of positional certainty based on visual fiducial measurements.

Russell *et al.* [9] develop a swarm robot foraging algorithm that dynamically deploys and re-deploys sensor motes. While the sensors are moved to provide certainty implicitly, they do not model localization quality during their deployment.

Sensor coverage problems also directly relate to our work. In particular, dynamic sensor coverage problems consider moving sensors. Liu *et al.* [10] consider randomly moving mobile sensors and analyze coverage properties. Clustering has been applied to dynamic coverage problems [11], but the sensors themselves were considered as mobile units rather than beacons, which are deployed and re-deployed by a moving robot. Sensor deployments via actuators are also similar to our problem [12]. However, the problems of deployments of actuators often focus on coverage and do not discuss the coupling of localization and deployment.

In sensor coverage problems, some works incorporate a continuous sensor field intensity [13], [14]. In the attenuated disk model, sensing quality decays with the distance to the sensor. This modeling is similar in spirit to our modeling of visual fiducials, but our model focuses as much on directionality as on distance in determining the noise model.

Deployment planning algorithms for heterogeneous robot teams also relate to our work. Such algorithms consider both cases where rewards are known apriori and those in which rewards are randomly distributed [15]–[17]. Our problem statement is similar in that there is competition for resources and dependencies imposed by deployment decisions, but we judge the quality of the assignment by the time required to execute the deployment plan rather than as rewards accumulated at each assignment.

Localization. Mainstream underwater localization relies on acoustic sensors, such as Doppler Velocity Log (DVL), long/short/ultrashort baseline acoustic positioning systems, and multibeam or sidescan sonars [6], [18], [19]. While such sensors allow the robot to navigate in large areas, their accuracy depends on a number of external factors, including

multipath effect and their overall resolution, making acoustic sensors not best suited to support manipulation tasks. Vision-based perception is ubiquitously adopted for many robotics tasks [20], including underwater [21], given camera’s low cost and ability to capture rich information of the surrounding. The literature classified state estimation methods according to different axes, with one being on whether they minimize reprojection errors of tracked features – indirect methods, such as ORB-SLAM [22] – or the alignment error considering image intensity values – direct methods, e.g., DSO [23]. Adding IMUs [24] can improve state estimation and including loop closure will allow the odometry estimate to be corrected. Underwater, however, vision-based perception still remains a challenge mainly due to the haze, color loss, and featureless environments [25], [26]. Given the precision required by the underwater construction task, we rely on visual fiducial markers and extend the operation area of the robot by allowing the robot to move them.

Fisheye cameras are often used to localize mobile robots [27], [28], but there are no techniques for modeling the quality of features detected using a visual fiducial marker.

Visual fiducial markers have been developed specifically for fisheye cameras [29] with the purpose of providing better position information. Other visual fiducial markers have been developed to reduce positioning noise [30]. To our knowledge, no attempt has been made to directly model the uncertainty of detecting visual fiducials. An exploration of the noise properties of visual fiducials is presented by Kalaitzakis *et al.* [31], but the geometry of the noise distribution is unexplored.

Free-floating construction systems. We are inspired by the limitations of our autonomous underwater construction system [7]. This work builds on and extends our autonomous underwater construction robot. Previously, it localized using a single visual fiducial that provided a limited coverage area.

Existing aerial free-floating construction systems commonly make use of fixed-place motion capture systems [1], [2], [32]. These motion capture systems provide precise, low latency position information but require numerous precisely calibrated cameras with limited coverage area. We want to provide coverage to large areas with limited need for complex fixturing.

III. PROBLEM MODEL

We consider the problem of deploying and moving infrastructure dynamically to provide high quality localization information for a set of tasks at known positions in global coordinates. The robot localizes using m beacons which can be placed and moved throughout the mission. The quality of information coming from the beacons depends on how the robot is positioned relative to the beacons. Information from each of the beacons can be combined to increase the localization accuracy. Our goal is to find a mission plan, $\mathcal{A}$, which consists of an ordered set of actions a_i . Each a_i can correspond to picking up a beacon, placing a beacon, or completing a task.

Each of the n tasks, located at $t_i \in \mathbb{R}^3$, requires a high enough precision of localization information. We model the quality of information coming from a beacon b_i using a function $\Sigma(r_i) \mapsto \Sigma_{r_i} \in \mathbb{R}^{3 \times 3}$ which maps relative positions (r_i) into covariance matrices that describe the noise distribution of the information coming from the beacon. We assume zero mean error. Information from multiple sensors can be combined by using sensor fusion equations. We use the equations described by Smith and Cheeseman [33]. Algorithm 1 shows how we combine noise distributions from multiple sensors.

Algorithm 1 Procedure to fuse covariance matrices of uncertain positions [33].

Require: Covariance matrices $\Sigma_1, \dots, \Sigma_k$

Ensure: Fused covariance matrix Σ

```

1:  $\Sigma \leftarrow \Sigma_1$ 
2: for  $i \in 1 \dots n$  do
3:    $K \leftarrow \Sigma(\Sigma + \Sigma_i)^{-1}$ 
4:    $\Sigma \leftarrow \Sigma - K\Sigma$ 
5: return  $\Sigma$ 
```

Each task t_i requires a high enough precision to be completed. We model the precision requirements using a scalar C_i which is obtained by applying a certainty approximation function $C(\Sigma) \mapsto C_i \in \mathbb{R}$ to the fused covariance matrix Σ . We define $C(\Sigma)$ as a function which approximates the probability that location readings are inside of a given error range. Receiving a location reading outside of the error range could cause the robot to fail at its task.

To move a beacon b_i , the robot must have precise enough location information for both pickup and placement. This means that beacons must be clustered to provide coverage of one another. For simplicity in our initial exploration, we assume perfect placement of the beacons. This assumption is reasonable for moving beacons on error-correcting foundations. In future work, we plan to extend our method to model a decay in the quality of information of moved beacons because of small placement errors.

For simplicity, we also assume a known relative orientation R of the AUV. In practice, the orientation of a free-floating robot can be sensed with a high accuracy using out-of-the-box AHRS boards. An initial calibration step can be used to measure the relative rotation for the set of beacons.

1D example. Consider a robot operating in a 1D world with no collisions. Two beacons, red and blue in Figure 2, provide coverage with precision $\Sigma_r(r) = r^2$, that is the quality of information decays quadratically with the distance to the beacon. We set $C(\Sigma) = 1 - \Sigma$ because, in 1D, Σ is a scalar and can be used directly. The beacons start at positions $b_1 = -0.1$ and $b_2 = 0.1$. The robot is given one task to complete at position $t_1 = .7$. Our fusion function in Algorithm 1 becomes $\Sigma = r_1^2 - \frac{r_1^2}{r_1^2 + r_2^2}$. The task has a requirement $1 - \Sigma \geq 0.95$. Moving a beacon requires the same certainty. Figure 2 shows an example of the problem.

We can compute a coverage area for a single beacon and a pair of beacons to guide our creation of a simple mission plan: $[b_i - 0.224, b_i + 0.224]$. To reach and cover

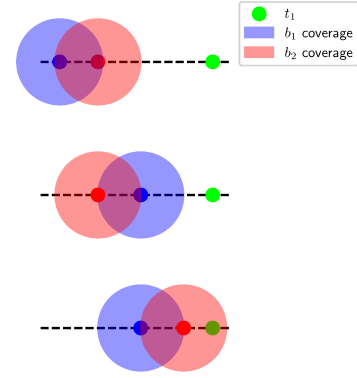


Fig. 2: Series of steps to cover a task at t_1 with the red and blue beacons.

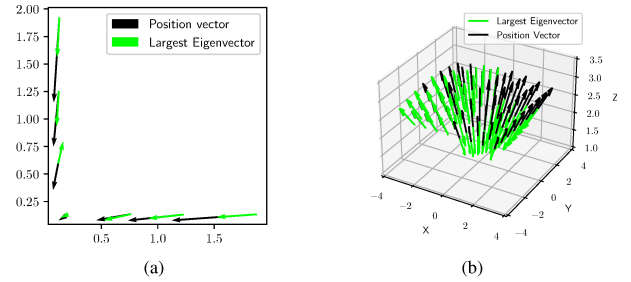


Fig. 3: (a) Predicted and measured largest eigenvector directions in real world experiment. The arrows extend from the marker's position. (b) Results from simulated corner noise. In both cases the predicted and measured directions closely match.

our task at position 0.7, we need to move one beacon to position $.7 - 0.224 = 0.476$. Moving the beacons will require multiple hops due to their limited coverage area. Our final plan $\mathcal{A}$ is then $\mathcal{A} = \text{MOVEBEACON}(b_1, 0.324), \text{MOVEBEACON}(b_2, 0.548), \text{COMPLETE}(t_1)$.

In the specific case of assembly a task t_i will represent placing a block at t_i 's location. For this application, we write $\text{PLACEBLOCK}(t_i)$ to mean placing a block at location t_i . We also replace MOVEBEACON with MOVEMARKER when we are dealing with a reconfigurable visual fiducial marker.

IV. NOISE CHARACTERIZATION OF VISUAL FIDUCIALS

The first step to implementing our assembly planning and localization method is to accurately model the noise distribution of visual fiducial markers. To understand how the noise distribution varies based on the relative position between the fisheye camera and a marker, we built a simulator. The simulator applies Gaussian distributed noise to the distorted corner positions of the visual fiducial, then undistorts them using the Kannala-Brandt Camera model [34] – typically used for fisheye lenses – and solves the Perspective-n-point problem. This simulation captures the important sources of noise: sensor noise and barrel distortion. We experimentally validate our simulator and noise model in Section VI-C.

Figure 3 shows outputs from a real world experiment (a) and from our simulation (b). We found that the noise distribution was highly structured and could be predicted using only two values: the largest eigenvector and eigenvalue.

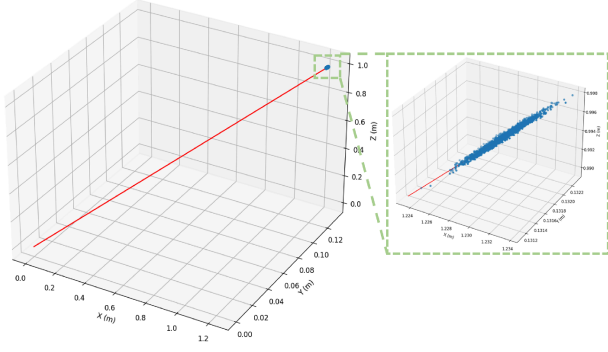


Fig. 4: Position readings from a static fisheye lens camera for a static visual fiducial marker. Blue points are detected positions and the red line represents the ground truth value for r .

Further, the eigenvector is parallel to the position vector. In the remainder of this section, we discuss how we predict the two components.

A. Scale noise

In our experiments on fiducial marker noise when viewed through a fisheye lens, we found that the axis of largest noise very consistently pointed towards the camera. Figure 4 shows an example of a set of relative position measurements for a single visual fiducial measured by a static camera. The position vector is marked in red and aligns closely with the largest eigenvector of the noise distribution (blue). We noticed this phenomenon to be consistent across various positions. The largest eigenvector dominates the noise distribution, and the other two are an order of magnitude smaller. We provide empirical evidence for this observation in Section VI-C.

B. A definition of $\Sigma(r)$

Algorithm 2 shows our covariance matrix prediction procedure. It accepts as arguments r , the relative position, β , a predictor of the largest eigenvalue and λ_i , an upper bound on the two smaller eigenvalues.

The largest eigenvalue, λ_* , is predicted using a spline which is calibrated on experimental data. v_1 , v_2 , and v_3 are our predictions of the eigenvectors. The largest eigenvector, v_1 , is predicted to be the normalized position vector. The other two eigenvectors, v_2 , and v_3 are predicted to be orthogonal to r and one another. Line 6 accumulates the eigenvalues into a 3×3 diagonal matrix. The matrices Λ and S are multiplied together to produce a predicted covariance matrix which has the predicted eigenvalues and eigenvectors.

Algorithm 2 Find covariance matrix Σ_r for a relative position r .

Require: r, β, λ_i

Ensure: Σ_p

- 1: $\lambda_* \leftarrow \beta(r)$
 - 2: $v_1 \leftarrow \frac{r}{\|r\|}$
 - 3: $v_2 \leftarrow \text{ORTHOgonAL}(r)$
 - 4: $v_3 \leftarrow \text{ORTHOgonAL}(r, v_2)$
 - 5: $\Lambda \leftarrow \text{HSTACK}(v_1, v_2, v_3)$
 - 6: $S \leftarrow \text{DIAG}(\lambda_*, \lambda_i, \lambda_i)$
 - 7: **return** $\Lambda S \Lambda^{-1}$
-

V. FINDING FEASIBLE ASSEMBLY PLANS IN PRACTICE

We model the assembly process as a set of manipulation tasks located at points in 3D space. The probability of completing a task $C(\Sigma_r)$ is the probability that a block dropped will be inside of the acceptance area of the slot it is aimed at. As a structure is built, it can occlude the view of markers, therefore, the markers must be continually moved as the structure is erected to increase $C(\Sigma_r)$.

A. Computing $C(\Sigma_r)$

To enable planning during the construction process, we define $C : \Sigma_r \mapsto \mathcal{R}$, which takes as input a covariance matrix and outputs the probability of successfully dropping a block. A block or marker is successfully placed if the robot decides to initiate the placement action within an acceptable range of the ideal position. This acceptable range (α) is dictated by the design of the error-correcting construction foundation. Assuming that the uncertainty in position is well approximated with a multivariate Gaussian, $C(\Sigma_r)$ is the probability that a random sample position drawn from $\mathcal{N}(0, \Sigma_r)$ is within the sphere of radius α .

Analytically computing this probability is challenging. Instead, we use a conservative approximation. The largest eigenvalue (λ_*) is a conservative estimate of the standard deviation of noise in any direction. So, we assume that the noise distribution along all three coordinate axes is independent and is equal to λ_* . This approximation results in the closed form estimate $C^*(\Sigma_r)$:

$$C(\Sigma_r) \geq \text{erf} \left(\frac{\alpha}{\sqrt{\lambda_*} \sqrt{2}} \right)^3 = C^*(\Sigma_r). \quad (1)$$

B. A layer-based approach for assembly

To assemble a structure, the robot must manipulate markers which can rest on top of blocks in order to localize while blocks are placed. When placed, the blocks can obscure sight of the markers. Planning around obscured markers introduces difficult nonlinearities into the constraints for any solver hoping to find feasible solutions.

To construct feasible plans efficiently without needing to model occlusions between the structure as it is erected and the markers, we propose a layer-by-layer algorithm. Algorithm 3 shows our strategy for generating feasible plans for a structure with n slots, using m markers.

Our algorithm works by dividing the blocks into layers $l_1, \dots, l_h$, where l_1 is the bottom layer of blocks, l_2 is the layer to be placed above l_1 , and so on with l_h being the topmost layer of blocks. For each $i \in \{1, 2, \dots, h\}$, we first cluster l_i to obtain clusters of width at most r , where r is an empirical determination of a marker's coverage radius based on the bound in Equation 1. In our implementation, this is achieved using a subroutine `CLUSTERUNTILRADIUS`(l_i, r). This subroutine performs k -means clustering on l_i using a value of k that is tuned, via binary search, to be the minimum possible such that all cluster widths are at most r .

The sub-procedure `EXTRACTCENTERS` finds the center point of each cluster. The cluster centers are then passed into

Algorithm 3 Layer-by-layer traversal.

Require: Structure with slots $S = \{s_1, \dots, s_n\}$, marker positions $M = \{b_1, b_2, \dots, b_m\}$

Ensure: Assembly plan $\mathcal{A}$

```

1: Divide  $S$  into layers  $l_1, \dots, l_h$ 
2:  $\mathcal{A} \leftarrow \emptyset$ 
3: for  $i \in \{1, \dots, h\}$  do
4:    $\mathcal{C} \leftarrow \text{CLUSTERUNTILRADIUS}(l_i, r)$ 
5:    $C' \leftarrow \text{EXTRACTCENTERS}(\mathcal{C})$ 
6:    $c_1, c_2, \dots, c_k \leftarrow \text{FINDTOUR}(C') \triangleright \text{efficient tour on cluster centers}$ 
7:   for  $j = 1$  to  $k$  do  $\triangleright$  process clusters in tour order
8:      $C_j \leftarrow \text{CLUSTEROF}(c_j)$ 
9:      $\mathcal{A} \leftarrow \mathcal{A}.\text{extend}(\text{WALKTOCOVERAGE}(M, c_j)) \triangleright \text{move markers into cluster}$ 
10:    for  $s \in C_j \setminus M$  do  $\triangleright$  place blocks in slots unoccupied by markers
11:       $\mathcal{A} \leftarrow \mathcal{A}.\text{append}(\text{PLACEBLOCK}(s))$ 
12:    for  $s \in M$  do  $\triangleright$  move markers to place remaining blocks
13:       $p \leftarrow \text{NEARESTNEIGHBOR}(s, C_j)$ 
14:       $p \leftarrow p + (0, 0, 1)$ 
15:       $\mathcal{A} \leftarrow \mathcal{A}.\text{append}(\text{MOVEMARKER}(s, p))$ 
16:       $\mathcal{A} \leftarrow \mathcal{A}.\text{append}(\text{PLACEBLOCK}(s))$ 
17: return  $\mathcal{A}$ 

```

FINDTOUR which computes a tour of the cluster centers O which has elements that index the clusters.

In each cluster produced by CLUSTERUNTILRADIUS(l_i, r), we select m points to serve as marker destinations. We choose the marker destinations to be the m farthest points from each other in the cluster. We then use the sub-procedure WALKTOCOVERAGE to transition the markers between destinations. We achieve this via a simple hopping strategy, like the one discussed in Section III. This strategy repeatedly hops one marker to the outside of the other marker’s coverage area, resembling a “gait” if one imagines the markers to be a robot’s feet. In this way, we position the markers at the m destinations within a cluster, place the blocks within that cluster, and repeat for successive clusters in that layer. Since the maximum cluster width is the coverage radius of a marker, we can ensure that each marker is always covered by another marker, enabling us to continue the gait after the blocks in a cluster have been placed.

After blocks have been placed around each of the markers in the cluster, the markers must be moved to make room for the remaining blocks. To do this, the loop in lines 12 to 16 iterates over each marker, moves it on top of the nearest block to it, and then places a block where the marker used to be. If the cluster has fewer than m points, two markers might share the same nearest neighbor and cause the markers to be placed in the same location. We avoid this edge case by requiring a minimum cluster size of m in CLUSTERUNTILRADIUS(l_i, r); this is feasible via a reasonable assumption that m markers can be placed within radius r .

VI. EXPERIMENTS

We implement our reconfigurable visual fiducial localization system in both hardware and simulation, and the results are described in the following.

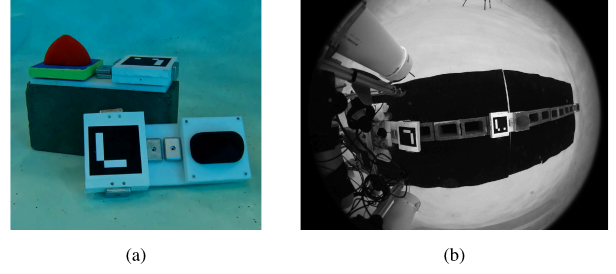


Fig. 5: (a) Reconfigurable visual fiducial design. (b) Robot’s eye view during assembly with the reconfigurable fiducials.

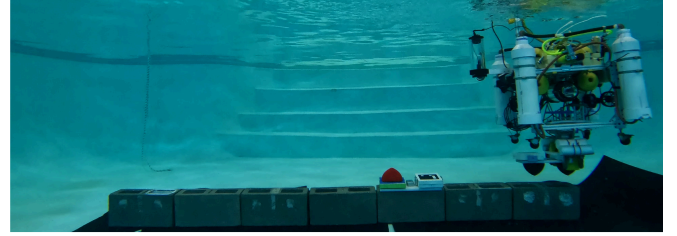


Fig. 6: The AUV performing the last placement of the two hop maneuver.

A. Experimental setup

Our hardware implementation is deployed on the Droplet AUV system [5], [7]. A ROS node holds the known global position of the visual fiducial markers. As the markers are moved, the node is notified and then marker readings are offset accordingly before performing the sensor fusion steps described in Algorithms 1 and 2.

For viewing visual fiducial markers, the robot is equipped with FLIR Blackfly camera with a Senko fisheye lens mounted facing downwards. The camera is mounted in a 3 inch Blue Robotics acrylic enclosure with a dome. Figures 6, 1 show the camera mounted on the robot and Figure 5 (b) shows how the scene looks through the camera.

B. Hardware testing

To validate the concept of reconfigurable visual fiducials in practice, we mounted visual fiducial markers on our error correcting connector geometry [7]. We show the design in Figure 5. The fiducials provide error correction during both pickup via the top handle (red in Figure 5) and placement.

We tested the concept of walking behaviors by implementing a 1D hopping strategy. The robot was able to successfully perform manipulation tasks in a large area with a side of 2.8 meters using cement blocks placed by hand. This side is 57% longer than the widest side usable by our previous implementation. Note that the side of the line covered was bounded by the physical width of the pool and not the robot’s ability to stack blocks successfully. Figure 6 shows the AUV completing the maneuver.

To verify that the visual fiducial marker fusion procedure reduces the localization noise, we recorded the measured positions and the number of valid marker readings when the robot was still. When only one marker was determined as valid, the standard deviation of position measurements on the X axis was 4.3 cm, but when two markers were valid, the standard deviation was reduced to 0.69 cm; around $6 \times$

improvement. On the Y axis, we observe similar improvements: from 6.0 cm to 1.5 cm; around $4\times$ improvement.

C. Validation of noise model

To validate our prediction model of covariance matrices given in Algorithm 2, we conducted both real world and simulation experiments. Our simulator projects a marker at a given relative position into the camera and offsets the corners according to a Gaussian distribution. This noise addition simulates the effects of pixel flicker on corner detection.

Figure 3 (a) shows the results of our physical testing. We arrayed visual fiducials in an area spanning about 1.4 meters on the positive X and Y axes. This setup mimics the distances between markers used in practice. We predict the direction of the largest eigenvector as the median of the measured relative positions. We found that the predicted direction is accurate to within 4 degrees. This result shows that computing the direction of the largest eigenvector as the position vector is effective in practice.

To more extensively test our prediction algorithm, we conducted a test using a spline predictor of the largest eigenvalue trained on 6250 noise distributions generated at known relative positions in our simulator. We set the upper bound for the smallest two eigenvalues to 10^{-4} . We set α to 2 cm for Equation 1. We tested the trained predictions from Algorithm 2 using 5184 test relative positions not present in the training data.

To evaluate the quality of Algorithm 2, we checked whether its predictions were more or less conservative than measured covariance matrices in the test dataset. We evaluated whether Algorithm 2 produced more or less conservative results by comparing the bound in Equation 1 for the predicted covariance matrix Σ_r^* against the measured covariance matrix Σ_r . We found that in 98.3% of cases $C^*(\Sigma_r^*) \leq C^*(\Sigma_r)$. In the other 1.7% of cases, $C^*(\Sigma_r)$ was only smaller than $C^*(\Sigma_r^*)$ by at most 3.4%. To avoid this case, we configure our planning algorithm so two markers are always visible.

When combining the two worst predictions using Algorithm 1, we find that the fused covariance matrix is a conservative estimate. We combined the covariance matrices of the two worst over-predictions of $C^*(\Sigma_r)$ for both measured and predicted covariances. We found that the predicted $C^*(\Sigma_r)$ is 40% lower (59%) than the measured bound (99%).

Our estimate of the noise is often very conservative, but we will show in the following section that large structures still have a high predicted success probability when planned using Algorithm 3.

D. Assembly algorithm

To test Algorithm 3 we use both a hand crafted plan checker and a full underwater robot 3D simulator called DAVE [35]. Our plan checker checks lines of sight to visual fiducials and computes the predicted certainty of every step. We use the underwater robot simulator to determine whether a plan which is feasible according to the plan check is also feasible under realistic control noise. We ran a construction

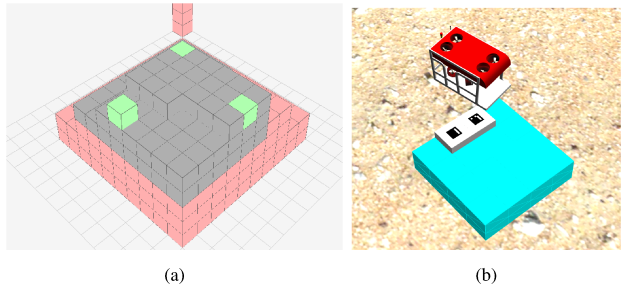


Fig. 7: (a) Visualization of plan checker. Green blocks represent markers. (b) Robot moving markers in the DAVE simulator.

TABLE I: Effect of increasing the cluster radius on plan efficiency and probability of success. After a certain cluster size, occlusions make construction impossible.

r	Predicted $P(\text{success})$	# steps
2.5	0	226
2.0	0.82	240
1.5	0.96	259
1.0	0.99	345

process with 74 build steps. Figure 7 (b) shows the partially completed structure in the DAVE simulator. With realistic control noise, the robot kept at least two markers in view 100% of the time.

As the number of manipulation steps increases, the time to build a structure increases. Our layer-based construction (Algorithm 3) takes an input a parameter r which describes the radius of clusters used for marker placement. Between each cluster, the markers are moved in an expensive walking procedure, so increasing r could improve the efficiency of the construction process but at the cost of reducing the reliability. Table I shows how changing r affects the number of steps required to build a 200 block structure and the certainty afforded during construction with three markers. We measure the predicted $P(\text{success})$ as the product of the probability of success of every state in the construction process. We also tested a pyramidal structure containing 1800 blocks. The assembly planning algorithm took about 5 minutes to plan the structure and the predicted probability of success for $r = 1.5$ is 91%.

VII. CONCLUSIONS & FUTURE WORK

This paper proposes a novel strategy for localizing relative to error correcting structures while planning the construction process. Our method is shown to work in practice at small scale and at large scale in simulation. We also show the robot being able to reliably complete the “hopping” strategy for moving the markers, extending the area for assembly.

We plan to improve the scale and quality of our hardware implementation. Planning for large scale construction with heterogeneous materials will require adaptive and flexible clustering strategies. Materials which are not well described by a bounding box may require more sophisticated strategies for avoiding occlusions.

Our assembly process is currently limited to structures which have only a single connected component per layer. If there is more than one connected component in a layer, the markers can become stranded as they are lifted up the

structure. In the future, we plan to explore ways to increase the flexibility of our method.

REFERENCES

- [1] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems*, vol. 34, no. 4, pp. 46–64, 2014.
- [2] G. Hunt, F. Mitzalis, T. Alhinai, P. A. Hooper, and M. Kovač, "3D printing with flying robots," in *Icra*, IEEE, 2014, pp. 4493–4499.
- [3] B. Dams, S. Sareh, K. Zhang, P. Shepherd, M. Kovac, and R. J. Ball, "Aerial additive building manufacturing: Three-dimensional printing of polymer structures using drones," *Proceedings of the Institution of Civil Engineers - Construction Materials*, vol. 173, no. 1, pp. 3–14, 2020.
- [4] S. Goessens, C. Mueller, and P. Latteur, "Feasibility study for drone-based masonry construction of real-scale structures," *AUTOMAT CONSTR*, vol. 94, pp. 458–480, 2018.
- [5] S. Lensgraf, A. Sniffen, E. Honnold, J. Jain, Z. Zitzewitz, W. Wang, A. Q. Li, and D. Balkcom, "Droplet: Towards Autonomous Underwater Assembly of Modular Structures," in *RSS*, 2021.
- [6] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of oceanic engineering*, vol. 39, no. 1, pp. 131–149, 2013.
- [7] S. Lensgraf, D. Balkcom, and A. Q. Li, "Buoyancy enabled autonomous underwater construction with cement blocks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 5207–5213.
- [8] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii, "Effective Landmark Placement for Robot Indoor Localization With Position Uncertainty Constraints," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4443–4455, Nov. 2019.
- [9] K. Russell, M. Schader, K. Andrea, and S. Luke, "Swarm Robot Foraging with Wireless Sensor Motes,"
- [10] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic Coverage of Mobile Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 301–311, Feb. 2013.
- [11] D. Yu, H. Xu, C. L. P. Chen, W. Bai, and Z. Wang, "Dynamic Coverage Control Based on K-Means," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 5333–5341, May 2022.
- [12] X. Li, A. Nayak, D. Simplot-Ryl, and I. Stojmenovic, "Sensor Placement in Sensor and Actuator Networks," in *Wireless Sensor and Actuator Networks*, A. Nayak and I. Stojmenovic, Eds., 1st ed., Wiley, Jan. 2010, pp. 263–294, ISBN: 978-0-470-17082-3 978-0-470-57051-7.
- [13] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, "Exposure in Wireless Sensor Networks: Theory and Practical Solutions," *Wireless Networks*, vol. 8, no. 5, pp. 443–454, Sep. 2002.
- [14] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–53, Oct. 2011.
- [15] C. Y. H. Lee, G. Best, and G. A. Hollinger, "Stochastic Assignment for Deploying Multiple Marsupial Robots," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, Cambridge, United Kingdom: IEEE, Nov. 2021, pp. 75–82, ISBN: 978-1-66542-926-9.
- [16] C. Y. H. Lee, G. Best, and G. A. Hollinger, "Optimal Sequential Stochastic Deployment of Multiple Passenger Robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China: IEEE, May 2021, pp. 8934–8940, ISBN: 978-1-72819-077-8.
- [17] C. Mitchell, G. Best, and G. Hollinger, "Sequential Stochastic Multi-Task Assignment for Multi-Robot Deployment Planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom: IEEE, May 2023, pp. 3454–3460, ISBN: 9798350323658.
- [18] Y. R. Petillot, G. Antonelli, G. Casalino, and F. Ferreira, "Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles," *IEEE Robotics & Automation Magazine*, vol. 26, no. 2, pp. 94–101, 2019.
- [19] F. Maurelli, S. Krupiński, X. Xiang, and Y. Petillot, "Auv localisation: A review of passive and active techniques," *International Journal of Intelligent Robotics and Applications*, pp. 1–24, 2021.
- [20] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [21] J. McConnell, I. Collado-Gonzalez, and B. Englot, "Perception for Underwater Robots," *Current Robotics Reports*, vol. 3, no. 4, pp. 177–186, Dec. 2022.
- [22] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [23] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
- [24] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
- [25] A. Quattrini Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane, and I. Rekleitis, "Experimental Comparison of Open Source Vision-Based State Estimation Algorithms," in *2016 International Symposium on Experimental Robotics*, D. Kulić, Y. Nakamura, O. Khatib, and G. Venture, Eds., ser. Springer Proceedings in Advanced Robotics, Cham: Springer International Publishing, 2017, pp. 775–786, ISBN: 978-3-319-50115-4.
- [26] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaos, and I. Rekleitis, "Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 7227–7233.
- [27] S. Houben, M. Neuhausen, M. Michael, R. Kesten, F. Mickler, and F. Schuller, "Park marking-based vehicle self-localization with a fisheye topview system," *Journal of Real-Time Image Processing*, vol. 16, no. 2, pp. 289–304, Apr. 2019.
- [28] Y. Huang, J. Zhao, X. He, S. Zhang, and T. Feng, "Vision-based Semantic Mapping and Localization for Autonomous Indoor Parking," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 636–641.
- [29] J. Hajjiami, J. Caracotte, G. Caron, and T. Napoleon, "ArUcOmni: Detection of highly reliable fiducial markers in panoramic images," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 2693–2699, ISBN: 978-1-72819-360-1.
- [30] B. Benligiray, C. Topal, and C. Akinlar, "STag: A Stable Fiducial Marker System," *CoRR*, vol. abs/1707.06292, 2017.
- [31] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaos, "Fiducial Markers for Pose Estimation," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, p. 71, Mar. 2021.
- [32] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D'Andrea, "Building tensile structures with flying machines," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 3487–3492.
- [33] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
- [34] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.
- [35] M. M. Zhang, W.-S. Choi, J. Herman, D. Davis, C. Vogt, M. McCarrin, Y. Vijay, D. Dutia, W. Lew, S. Peters, and B. Bingham, "Dave aquatic virtual environment: Toward a general underwater robotics simulator," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, 2022.