

On Experimental Emulation of Printability and Fleet Aware Generic Mesh Decomposition for Enabling Aerial 3D Printing

Marios-Nektarios Stamatopoulos, Avijit Banerjee, and George Nikolakopoulos

Abstract—This article introduces an experimental emulation of a novel chunk-based flexible multi-DoF aerial 3D printing framework. The experimental demonstration of the overall autonomy focuses on precise motion planning and task allocation for a UAV, traversing through a series of planned space-filling paths involved in the aerial 3D printing process without physically depositing the overlaying material. The flexible multi-DoF aerial 3D printing is a newly developed framework and has the potential to strategically distribute the envisioned 3D model to be printed into small, manageable chunks suitable for distributed 3D printing. Moreover, by harnessing the dexterous flexibility due to the 6 DoF motion of UAV, the framework enables the provision of integrating the overall autonomy stack, potentially opening up an entirely new frontier in additive manufacturing. However, it's essential to note that the feasibility of this pioneering concept is still in its very early stage of development, which yet needs to be experimentally verified. Towards this direction, experimental emulation serves as the crucial stepping stone, providing a pseudo mockup scenario by virtual material deposition, helping to identify technological gaps from simulation to reality. Experimental emulation results, supported by critical analysis and discussion, lay the foundation for addressing the technological and research challenges to significantly push the boundaries of the state-of-the-art 3D printing mechanism. - Full mission video available at <https://youtu.be/gfZuYCA8jAw>

I. INTRODUCTION

In recent years, the field of 3D printing has witnessed significant advancements at an unprecedented pace, transforming it into a powerful tool for rapid prototyping. The application of the 3D printing technology has transformed what was once relegated to the realm of science fiction into a tangible reality [1], [2]. Its current impact spans across diverse industries, including construction, agriculture, healthcare, automotive, and aerospace, unveiling exciting prospects for the future. Notably, it holds potential for constructing full-scale infrastructures, such as emergency shelters and post-disaster relief accommodations in remote, harsh, and hard-to-reach environments exposed to extreme climates, unstructured terrain, and distant military locations [3]. Conventional manufacturing and 3D printing demand substantial machinery and infrastructure, limiting their utility in far-flung and inaccessible regions [4], [5], while also constraining the size of built objects based on machinery dimensions, which hinders scalable construction, particularly in areas lacking the necessary infrastructure [6].

The Authors are with the Robotics and Artificial Intelligence Group, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden

Coorsrponing author's e-mail: marsta@ltu.se

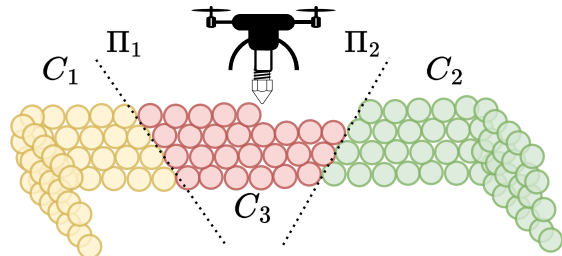


Fig. 1: Conceptual representation of aerial 3D printing UAV during the manufacturing process of chunk C_3 . The whole construction is decomposed through planar cuts Π_1 and Π_2 .

By leveraging significant advancements in autonomous Unmanned Aerial Vehicle (UAV) technology, this visionary framework is emerging as a groundbreaking concept offering unparalleled flexibility, accessibility, and efficiency in autonomous and scalable manufacturing constructions. Nevertheless, aerial 3D printing is a pioneering concept still at its very early stage of development, with limited existing research, mostly in its conceptual phase [7], [8]. This article introduces an experimental emulation of a versatile and adaptable generic aerial 3D printing framework designed to decompose construction into individual manufacturing sub-tasks (as shown in Fig. 1), laying the foundation for more efficient and scalable construction operations, while it should be highlighted that both aspects in the field of material science and extruder technology are left out of the scope of this article as sequential future works.

II. RELATED WORK

The context of multi-robot automated construction has been investigated mostly through ground robots equipped with manipulators [9]–[11], but are limited to movements in a 2D plane and the reachability of the arm. Aerial additive manufacturing is posed as an alternative to it, although it is still in an early stage. Specifically, the approach in [8] examined the feasibility and proof of concept of the hybrid of aerial robotics and additive layer manufacturing which primarily involves a single UAV in operation. In the context of constructing full large-scale constructions in hard-to-access areas, in [12] the use of aerial platforms is examined where both scenarios with single and multiple UAVs are evaluated. The process of multi-UAV construction is executed through a conventional 2.5D tactic, which entails the segmentation of the given mesh into discrete layers, subsequently dividing each layer into an equitably distributed set of curves, referred to as print jobs that are assigned to each UAV. For the multi-UAV scenario, a virtual printing environment was

employed. Luminescent light markers were mounted on the UAVs, wherein alternations in their color would indicate the extrusion of the material. Towards the deployment of multiple UAVs for the realization of any given large-scale construction, an early-stage conceptual framework is presented in [13] aiming to decompose any given mesh that is intended to serve as input for the subsequent manufacturing process. The decomposition is carried out in the form of an iterative search, while upon the generation of the sub-parts, they are sequentially assigned to each UAV. The framework is evaluated in Gazebo software by tracking the extruder tip and visualizing the material extrusion whenever the printing action is commanded from the UAV.

III. CONTRIBUTIONS

In this article, a novel, generic multi-axis chunk-based aerial 3D printing framework supported by experimental emulation is presented. In this context, a recently introduced chunk-based methodology [13] is extended to incorporate an additional heuristic function in the iterative beam search process. This extension is strategically designed to increase the number of "seed" chunks, which hold significant importance due to their direct influence on the parallelizability of the manufacturing process. These seed chunks exhibit minimal manufacturing interdependencies, thereby facilitating the parallelization of printing tasks and bringing the framework closer to achieving the milestone of collaborative printing. In contrast with the previous state-of-the-art, the proposed framework enables seamless integration of distributed aerial 3D printing with multiple UAVs in a multi-axis layered manufacturing fashion. Moreover, the experimental emulation essentially considers the overall autonomy, incorporating precise coordinated motion planning of a UAV, while traversing a series of planned space-filling paths. It effectively utilizes a Model Predictive Control (MPC) strategy, with careful consideration of the unique attributes of the UAV model, ensuring necessary movements for successful execution. Experimental emulation results, focusing on the framework evaluation, motion planning excluding the material science and extruder technology are critically analyzed and discussed. Thus, paving the way and providing a foundational basis as a bridge between sim to real, identifying the technological challenges necessary to realize this innovative approach.

IV. METHODOLOGY

The initial step in realizing aerial 3D printing involves an innovative optimization-based mesh decomposition approach. This process dissects the user-provided mesh into smaller segments called "chunks" by strategically selecting cutting planes. This selection employs a search-based methodology [13], wherein potential cutting planes are sampled and assessed. The most favorable combinations of planes advance to subsequent iterations, incorporating additional cuts. To facilitate evaluation, a heuristic function is introduced maximizing volume dispersion among the chunks and enhancing the parallelizability of chunk printing tasks. A

Binary Space Partitioning Tree (BSP) [14] is established to track the planar cuts and resulting chunk structure. Post-search phase, the BSP tree acts as a task priority sequencer due to its inherent relationships within its structure, which dictate dependencies between the chunks. Subsequently, each chunk is allocated to a UAV in a responsive manner. Every chunk undergoes slicing through a commercially available slicer and is transformed into a path, which the assigned UAV executes. An overview of this approach is depicted in Fig. 2.

V. CHUNKS GENERATION

The chunk generation process is carried out by intersecting the mesh with a set of planes, fragmenting it into smaller printable sub-parts. The cutting planes are denoted by $\Pi^{i,j}$, defined with the surface-normal vector $\vec{n}_i \in R^3$, perpendicular to the plane, originated at $\vec{p}_j \in R^3$ and laying on respective cutting surfaces. A BSP Tree \mathcal{T} is utilized to keep track of the cuts and the resulting chunks. The original mesh is located at the root node, all the generated chunks C_i are located at the leaves of \mathcal{T} , while all the cutting planes are in the rest of the nodes. When a new planar cut $\Pi_i \in \mathcal{P}$, where \mathcal{P} denotes the set of all planar cuts, is executed, a recursive process identifies all affected chunks C_j and are extended into child nodes. Specifically, intermediate children nodes are labeled as $C_{j,i}^-$ and $C_{j,i}^+$, representing the outcome of the cut between Π_i and C_j . Among these, $C_{j,i}^+$ resides in the half-space aligned with the normal vector \vec{n}_i and is referred to as a positive chunk. Conversely, $C_{j,i}^-$ is termed a negative chunk. This article follows the convention of positioning the negative chunk, $C_{j,i}^-$, as the left child of the parent node.

Towards finding the optimal combination of planar cuts $\Pi^{j,i}$ that will result into generated chunks with specified characteristics to facilitate aerial collaborative printing, an iterative beam-search approach [15] is employed.

Initially, normal vectors \vec{n}_i are uniformly sampled within a spherical space subject to constraints in the polar angle $\phi_s \in [-\phi_s^{max}, \phi_s^{max}]$, while $r = 1$, and $\theta_s \in [0, 2\pi)$. A family of planes is calculated for each sampled normal vector along its axis with a predefined distance offset $\delta \in \mathbb{R}$ between them. Each planar cut $\Pi^{i,j}$ of the aforementioned sampling is applied to the corresponding tree \mathcal{T}_i that requires expansion. Every newly generated tree $\mathcal{T}_{i,j}$, resulting from this extension process, is collected and in the sequel evaluated through a heuristic function. The top W_{inner} ones are selected to be added to the set containing all the new trees generated in the current expansion iteration. After the evaluation of the new cuts is executed for all \mathcal{T}_i trees, the top W_{outer} ones are selected to proceed to the next iteration.

A. Heuristics

Each BSP tree containing both the optimal cuts and the chunks, needs to be evaluated and be given a cost value in order to compare it among the rest and keep the best ones for the next round of expansions. For this purpose, a heuristic function is assigned incorporating two different sub-heuristics. Uniform volume distribution of the generated

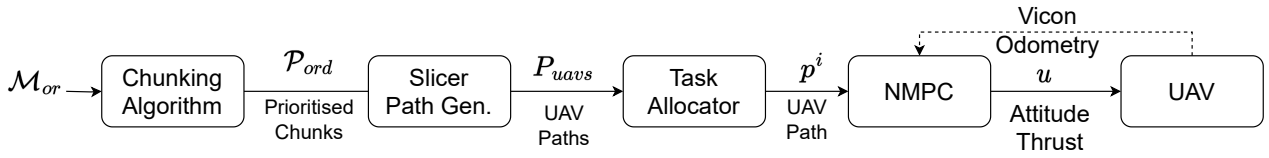


Fig. 2: Framework Block Diagram

chunks along with increase in the manufacturing parallelability of the structure are targeted.

1) *Volume Dispersion*: Uniformity in volume yields an evenly distributed arrangement of the constituent parts, with their geometric centers adequately spaced apart. This uniformity is translated into reduced waiting times between printing operations, as each constituent element's printing duration closely aligns with that of its counterparts. Thus, the volume dispersion $c_v = \sigma/\mu$ is used as heuristic, where $\mu = \sum_{i=1}^N V_i/N$, $\sigma = \sum_{i=1}^N (V_i - \mu)^2/N$ where V_i corresponds to the volume of the i -th chunk.

2) *Seed Chunks*: In accordance with the notation established in [16], a "seed chunk" is precisely delineated as a chunk for which all resultant faces from the planar cuts exhibit positivity. The maximization of the presence of such chunk types within the final hierarchical structure is of great importance, given that their sole prerequisite is to be fabricated either at ground level or atop a flat chunk. In this configuration, all of their faces act as supports for subsequent chunks within the sequence. This particular attribute enhances the potential for parallel execution of printing tasks, as it eliminates any additional inter-dependencies. To determine the seed status of a given chunk, denoted as C_k , a set $\mathbf{C} = \{\Pi_i, i = 1, 2, \dots, M\}$ is assembled, where each Π_i is co-planar with it. Via set \mathbf{C} it is verified whether the chunk aligns positively or negatively with the corresponding plane Π_i . This determination is achieved by the projection of an arbitrary vertex $v_j^k \in \mathbb{R}^3$, from the C_k mesh, onto the normal vector \vec{n}_i of Π_i . It is calculated according to $proj_{\vec{n}_i}(v_j^k) = (v_j^k - \vec{p}_i) \cdot \vec{n}_i$ where, \vec{p}_i represents the origin of plane Π_i . The final classification is determined by the value $s_k \in \mathbb{R}$, defined as follows:

$$s_k = \begin{cases} 1 & \text{if } proj_{\vec{n}_i}(v_j^k) < 0, \forall \Pi_i \in \mathbf{C} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Hence, a chunk C_k qualifies as a seed whenever $s_k = 1$. This attribute is utilized by including the term s_k in the heuristic function. Furthermore, an extension is introduced where the count of positive faces of a chunk is rewarded, provided that it is classified as a seed. A positive face of a seed chunk C_k is defined as a face that shares a co-planar alignment with a plane Π_i and is represented by the term $f_{i,j}^{k+}$ being equal to 1, having $proj_{\vec{n}_i}(v_j^k) < 0$ as a positivity condition. Consequently, the term $p_{faces}^k = \sum_{i=1}^M f_{i,j}^{k+} s_k$ denoting their number, is introduced as a reward in the heuristic. Both of the previously mentioned heuristics are integrated into the unified heuristic function:

$$g(C_k) = G_{part} s_k + G_{faces} p_{faces}^k, C_k \in \mathcal{T} \quad (2)$$

where $G_{part}, G_{faces} \in \mathbb{R}$, gains adjusted to align with the goals of the particular application.

B. Combined Heuristic

Fusing the aforementioned heuristics in a single one, the total heuristic function $h(\mathcal{T})$, for the evaluation of any tree \mathcal{T} , is given by:

$$h(\mathcal{T}) = G_{disp} c_v - \sum_{k=1}^M g(C_k) \quad (3)$$

where M represents the number of chunks within the specific BSP tree denoted as \mathcal{T} . The summation of functions $g(C_k)$ for each chunk is negated to stress the necessity of rewarding these values rather than imposing penalties.

C. Printability Constraints

During the phase of normal vector sampling, the polar angle ϕ of the sampling surface \mathbf{S} , and consequently, the inclination of each planar cut, is constrained by considerations related to both chunk inter-connectivity and potential extruder collisions [13]. Specifically, an upper angular limit ϕ_{max}^{conn} is imposed to ensure the creation of adequate printable overlap regions between adjacent surfaces. This overlap is critical for providing the requisite support and eventual adhesion of the printed parts. Additionally, the extruder's geometric characteristics, approximated as a bounding rectangle of length l and height h , are considered to prevent any collisions with previously printed components. This consideration results in the determination of a maximum allowable cutting slope, denoted as $|\phi_{max}^{extr}| \leq \tan^{-1}(\frac{h}{l})$. Finally, a unified upper bound, expressed as $|\phi_{max}| = \max(\phi_{max}^{conn}, \phi_{max}^{extr})$ is used.

D. Tree Terminal Condition

The initial UAVs' fleet setup is considered known before the chunking phase. Specifically, the volume of the carried extrusion material from each UAV is given as input to the algorithm. Both UAVs' carrying material d_m and chunks volumes c_k are sorted in descending order and placed in the sets $\mathbf{D}_{mat} = [d_1, d_2, \dots, d_n]$ and $\mathbf{C} = [c_1, c_2, \dots, c_k]$ for the UAVs and chunks correspondingly. A tree is considered to be terminated when all the chunks generated at its leaf nodes can be printed by at least one of the available UAVs. Hence, the algorithm stops searching for further cuts to extend it when a non-empty set exists and is defined as:

$$S = \{p_l = (d_j, c_i) : d_j > c_i\} \neq \emptyset, \forall l = [1, \dots, k] \quad (4)$$

The sets C, D and S are updated recursively after the construction of each chunk.

VI. UAV EXECUTION

Once a chunk is designated for printing, a prescribed sequence of actions must be executed to initiate the printing process. Each chunk mesh is given to the slicer [17] that is configured with precise parameters including UAV and extruder dimensions and material deposition variables, adjusted to suit the project's specific requirements. It is transformed into the adequate G-code commands for effectively manufacturing the chunk. A block diagram depicting the sequential workflow associated with this process is illustrated in Fig. 2.

A. G-Code to UAV Path

A post-processing step is necessary to eliminate extraneous data from the G-code and to convert the path into a simplified format compatible with the UAV. All G-code segments are transformed into equivalent waypoints that the UAV must traverse, resulting in a path P_{ex}^i for the i -th chunk. Since this path corresponds to the extruder's path and not the UAV's, a transformation converting the extruder tip's frame to the UAV's body frame is applied. To facilitate this transformation, an extruder of a length l_{ex} is employed for the printing operation, with the extruder tip vertically oriented toward the ground. Finally, the path P_{uav}^i referring to the i -th chunk is obtained.

B. Task Allocation

The BSP tree organizes planar cuts during the search phase, with chunks allocated to its leaves for orderly mesh creation. Constraining the plane's slope with ϕ_{max} ensures a positive vertical component (along z axis) in normal vectors \vec{n}_i . This results in a dependency between the two resulting chunks after a cut: the positive chunk $C_{j,i}^+$ lies above the plane Π_i , while the negative chunk $C_{j,i}^-$ lies below. Strategically selecting the preceding printed chunks as supports for the following ones, the left child nodes and their subordinates in the BSP tree \mathcal{T} take precedence over the right ones. To eliminate dependency on plane selection order, planar cuts in the BSP tree \mathcal{T} are reordered based on origin point \vec{p} z -coordinates, from low to high. After executing the cuts with the aforementioned order, the same priority rule is applied and the priority order set \mathcal{P}_{ord} is generated by traversing the tree in-order. \mathcal{P}_{ord} comprises the requisite sequence in which the individual chunks must be printed to facilitate the construction in a feasible manner.

Throughout the execution of the printing process, the allocation of each chunk to the UAVs is conducted in a reactive fashion. The chunks are printed sequentially following the rule that each chunk must be printed by only one UAV. Whenever a UAV is in an idle state, no other UAV is printing at the time and there are still chunks pending to be printed, it is assigned the next chunk in order and the corresponding chunk is excluded from the \mathcal{P}_{ord} set becoming no longer available. In the case that a UAV is running low on battery, it sets itself to an inactive state and is no longer able to get assigned chunks to print. The aforementioned process is executed iteratively until there are no more chunks to be manufactured.

C. UAV Motion Planning Framework

For precise execution of aerial 3D printing, each UAV must follow the designated path P_{uav}^i , while ensuring uniform material deposition. To achieve this, a Nonlinear Model Predictive Control (NMPC)-based optimal path following controller [18] is employed and designed to accurately track the path generated by the G-code. The UAV's dynamics are described using two coordinate frames: the world frame $W \in \mathbb{R}^3$ and the body-fixed frame $B \in \mathbb{R}^3$. The body-fixed frame is centered on the UAV's mass center, while the inertial frame is assumed to be fixed in the origin. The UAV's motion is governed by a set of non-linear dynamic equations presented as:

$$\begin{aligned} \dot{p}(t) &= v(t) \\ \dot{v}(t) &= R_{x,y}(\theta, \phi)\mathbf{T} + \mathbf{G} - \mathbf{A}v(t) \\ \dot{\phi}(t) &= \frac{1}{\tau_\phi}(K_\phi\phi_{ref}(t) - \phi(t)) \\ \dot{\theta}(t) &= \frac{1}{\tau_\theta}(K_\theta\theta_{ref}(t) - \theta(t)) \end{aligned} \quad (5)$$

where $p = [p_x, p_y, p_z]^T \in \mathbb{R}^3$ denotes the position, $v = [v_x, v_y, v_z]^T \in \mathbb{R}^3$ represents the velocity of UAV in world frame. The UAV roll and pitch angles are defined by $\phi, \theta \in [-\pi, \pi]$ respectively. $R(\phi, \theta) \in SO(3)$ denotes the rotation matrix that transforms the acceleration due to thrust acting on the UAV body frame to its equivalent components into the world frame, $\mathbf{T} = [0, 0, T]^T$, $\mathbf{A} = \text{diag}(A_x, A_y, A_z)$, where $A_x, A_y, A_z \in \mathbb{R}$ represent linear damping terms, and $\mathbf{G} = [0, 0, g]^T$, with $g \in \mathbb{R}$ the Earth gravitational acceleration. The attitude dynamics is approximated by a first-order system with time constants τ_ϕ and τ_θ . The overall motion control system includes a low-level attitude controller with constant gains K_ϕ and K_θ to track the reference orientation commands ϕ_{ref} and θ_{ref} in \mathbb{R} . These orientation references, along with the thrust magnitude $T > 0 \in \mathbb{R}$, form the control input. This input is designed using an NMPC-based high-level position controller to achieve accurate reactive path following. The UAV dynamics described in Equation 5 undergo discretization via the forward Euler method, utilizing a sampling time $\delta_t \in \mathbb{R}^+$. The discretized model serves as the prediction model for the NMPC framework for each time instant $(k + j|k)$, which denotes the prediction of the time step $k + j$ produced at the time step k , with a prediction horizon denoted as N . The control design's objective is to determine an optimal sequence of $\mathbf{u}_k = [T_{in}^k, \phi_{in}^k, \theta_{in}^k]$ at each time step to minimize the cost function given by:

$$\begin{aligned} J &= \sum_{j=1}^N \underbrace{(x_{ref} - x_{k+j|k})^T Q_x (x_{ref} - x_{k+j|k})}_{\text{state cost}} \\ &\quad + \underbrace{(u_{ref} - u_{k+j|k})^T Q_u (u_{ref} - u_{k+j|k})}_{\text{input cost}} \\ &\quad + \underbrace{(u_{k+j|k} - u_{k+j-1|k})^T Q_{\Delta u} (u_{k+j|k} - u_{k+j-1|k})}_{\text{input smoothness cost}} \end{aligned} \quad (6)$$

where $Q_x \in \mathbb{R}^{8 \times 8}$, $Q_u, Q_{\Delta u} \in \mathbb{R}^{3 \times 3}$. The *state cost* penalizes the deviation of the predicted future positions with the reference, the *input cost* is to minimize deviations from the reference control input $u_{ref} = [g, 0, 0]$, aiming to minimize the energy of the control actuation. The *input smoothness cost* penalizes deviations between successive inputs, promoting smooth control action by minimizing the control rate. This optimization process aims to minimize the cost function while adhering to the system dynamics and input constraints. The input constraints are defined as $u_{min} \leq u \leq u_{max}$, where $u_{min}, u_{max} \in \mathbb{R}^3$. Additionally, constraints are imposed on the maximum rate of change of ϕ_{ref} and θ_{ref} , bounded as $\Delta\phi_{max}$ and $\Delta\theta_{max}$. The optimization problem is numerically solved using PANOC with a single-shooting approach [19] through the Optimization Engine (OpEn) platform [20].

VII. RESULTS AND DISCUSSIONS

The efficacy of the proposed framework is investigated by executing real-life emulation experiments in a lab environment using an in-house designed and built quad-copter. The Vicon motion-capturing system is responsible for tracking the UAV and providing odometry feedback. Neither the extruder arm nor the extruder tip are addressed within the scope of this article since material science and extruder technology are deliberately excluded from consideration for this specific application. However, a suitable candidate for the specific task could be considered a foam spraying mechanism similar to [8], [12] Assuming the extruder's length $l_{ex} = 0.5m$ and considering it rigidly mounted on the bottom of the UAV, its position is calculated using the real-time data received from Vicon. Upon the issuance of a command instructing the initiation of material extrusion from the extruder, the deposition of the material is emulated by placing small spherical markers of radius $r_m = 0.3cm$ in a vertical offset of $0.1m$ to account for the inherent spray effect generated by the extruder.

The mesh used for the experiment is a $2m$ wide, $2m$ long and $0.5m$ height hollow rectangle having walls with 0.1 m width as depicted in Fig. 3. The bounding angles of the search for the planar cuts sampling space \mathcal{S} are set to $\phi_{sample}^{max} = 45^\circ$ and $\theta \in [0, 2\pi)$ while the weights of the heuristic function are designed as follows: $G_{disp} = 200$, $G_{part} = 10$ and $G_{faces} = 20$. The planar cuts search algorithm resulted in 5 planes $[\Pi_1, \dots, \Pi_5]$ as the optimal set that generated 18 chunks. The generated chunks are shown in Fig. 3a composing the whole given mesh, in a color-coded format. The search result is illustrated in the BSP tree in Fig. 3b having all the cuts as intermediate nodes and all the generated color-codes chunks as leaf nodes. The final priority order calculated by the in-order traversal of the tree is shown at the bottom of Fig.3, starting from chunk C_0 to C_{17} .

The model parameters as mentioned in the model 5 are selected as following: $A_x = 0.1$, $A_y = 0.1$, $A_z = 0.2$ and $\tau_\phi = 0.4$, $\tau_\theta = 0.47$ and $K_\phi = K_\theta = 1$. The sampling time used is 50 msec while the prediction horizon time-steps number is set to $N = 40$ corresponding to a 2 sec.

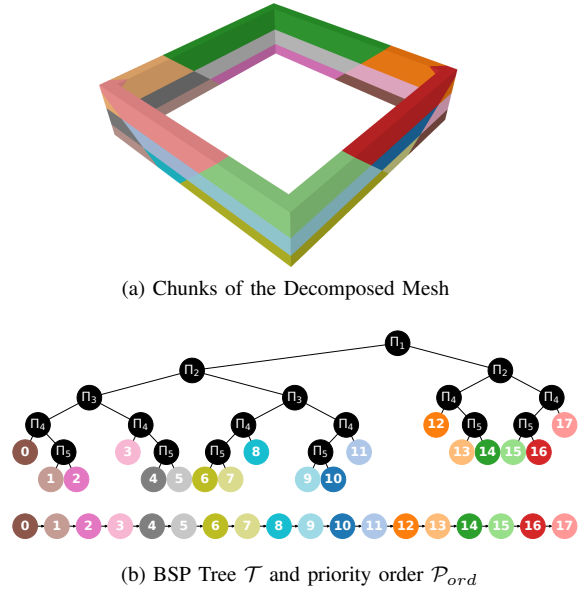


Fig. 3: Color-coded chunking Results for the rectangle, stacked together forming the decomposed mesh(a). BSP tree \mathcal{T} visualizing the cutting planes Π_1, \dots, Π_5 in the intermediate nodes and the chunks C_1, \dots, C_{17} on the leaf nodes. The printing sequence graph \mathcal{P}_{ord} is extracted at the bottom (b).

prediction. The cost function weights are set as follows $Q_x = \text{diag}(15, 15, 25, 4, 4, 4, 8, 8)$, $Q_u = \text{diag}(3, 15, 15)$ and $Q_{\Delta u} = \text{diag}(3, 15, 15)$. Additionally, the control input bounds are set to $u_{min} = [3, -0.2, -0.2]$ and $u_{max} = [15.5, 0.2, 0.2]$ (SI). Finally, the input rate thresholds were set to $\Delta\phi_{max} = \Delta\theta_{max} = 0.04$ rad. The reference and measured position of the UAV through Vicon along with the reference position of the extruder (l_{ex} m below the UAV) and its projected position as calculated from the pose of the UAV, are depicted in Fig. 4. Observations from this plot reveal that the UAV adheres to the prescribed flight paths, executing linear movements in accordance with the slicing process. Notably, the extruder's positional data exhibits occasional oscillations and overshoots, particularly noticeable during the initiation and conclusion of movement sequences. This phenomenon can be attributed to the extruder's length, as slightly small changes to the UAV's roll and pitch angles result in big errors proportional to the length. Despite the aforementioned overshoots, overall the measured positions are bounded within an offset of 5 to 6 cm per axis. The data outlined are also graphically presented per axis in Figure 6. The tracking of reference points within each axis exhibits a satisfactory level of performance, notwithstanding occasional extruder oscillations during movement and hovering. It is imperative to acknowledge that such oscillations are primarily stemming from model inaccuracies or potential manufacturing imbalances within the aerial platform. Additionally, it can be observed that the position error exhibits an increased magnitude during the start and the end of each linear segment's movement. This occurrence is a predictable

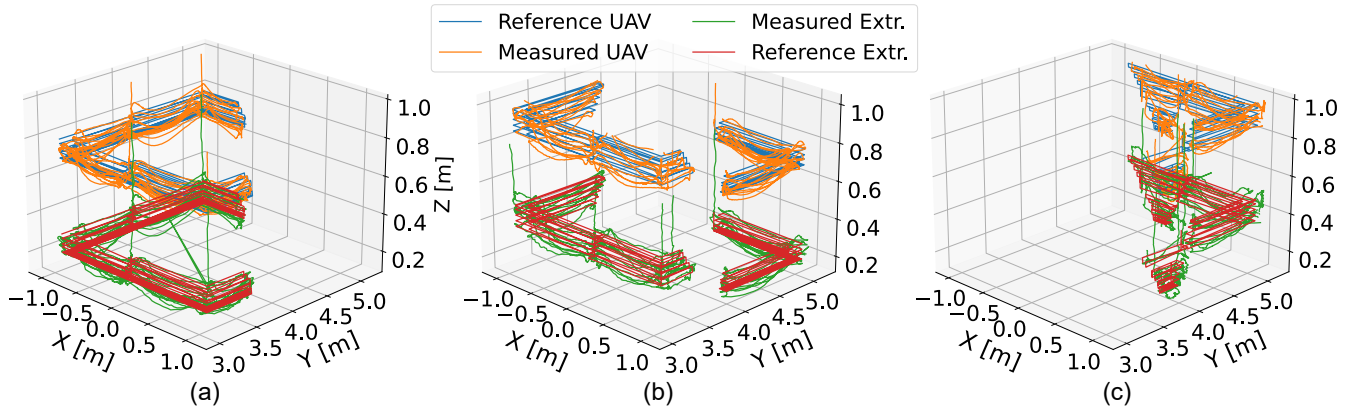


Fig. 4: Sequential 3D Plots for chunks C_0-C_2 (a), C_3-C_4 (b), C_7-C_9 (c) of Reference UAV position (blue), Measured UAV position (green), Extruder reference position (red) and calculated Extruder position (orange) during the printing procedure.

outcome attributable to the momentary rolling and pitching angle of the UAV during the movement initiation aiming to accurately track the reference waypoint. Whenever this occurs, a fast increase in the velocity is noticed. Not maintaining a constant speed during the execution of the printing, may lead to discontinuities and uneven material deposition throughout the traversal of the manufacturing path. Despite these variations and the presence of a deviation of the order of 6 cm per axis from the designated waypoints, the diameter of the deposited material is sufficiently big to compensate for it and not compromise the integrity of the whole structure as shown in Fig. 5. A video with the full mission can be found in <https://youtu.be/gfZuYCA8jAw>

VIII. CONCLUSIONS

This article introduces a novel aerial 3D printing framework and demonstrates it through experimental emulation. In the selected evaluation scenario a rectangular wall has been emulated for printing carried out by deploying real-life tracked UAVs executing the planned print task in the adequate sequential order imposed by the decomposition process. This framework has the potential to revolutionize additive manufacturing by enabling large-scale structures through its capability to divide complex 3D models into manageable components, enabling distributed 3D printing. However, it's in the early stages, and further validation is needed. The use of experimental emulation helps identify gaps between simulation and reality, providing a foundation for addressing challenges in 3D printing.

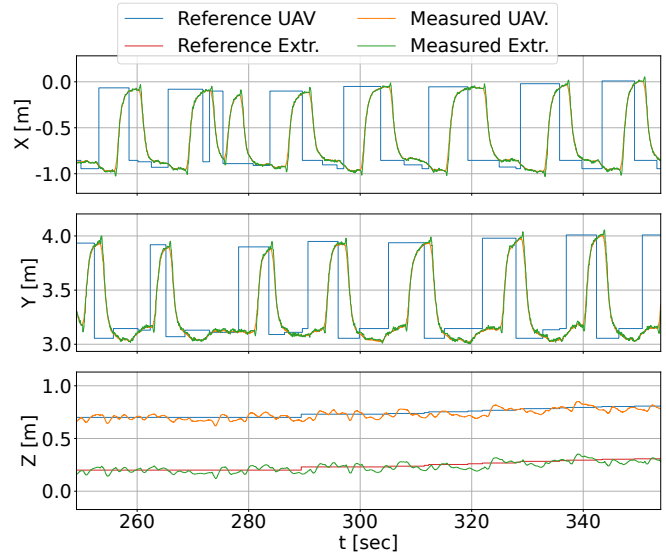


Fig. 6: Reference, Measured UAV and Measured Extruder Coordinates per Axis X and Y from $t = 271$ sec until $t = 360$ sec during the emulation of the printing in the real-life UAV.

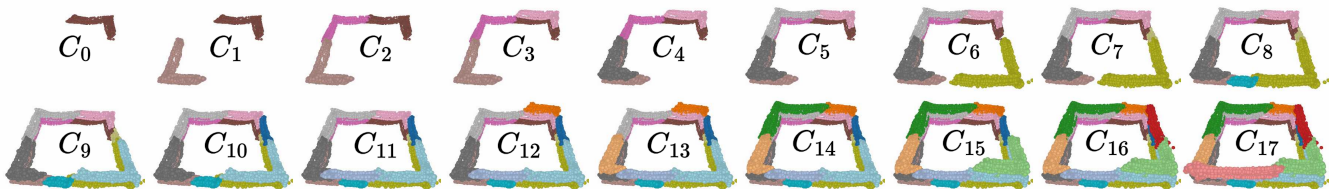


Fig. 5: Sequential visualizations of the chunks C_0 to C_{17} printing process via real-time data gathered from the motion capture system.

REFERENCES

- [1] E. Karasik, R. Fattal, and M. Werman, "Object partitioning for support-free 3d-printing," in *Computer Graphics Forum*, vol. 38, no. 2. Wiley Online Library, 2019, pp. 305–316.
- [2] C. Balletti, M. Ballarin, and F. Guerra, "3d printing: State of the art and future perspectives," *Journal of Cultural Heritage*, vol. 26, pp. 172–182, 2017.
- [3] Z. Xu, T. Song, S. Guo, J. Peng, L. Zeng, and M. Zhu, "Robotics technologies aided for 3d printing in construction: A review," *The International Journal of Advanced Manufacturing Technology*, vol. 118, no. 11-12, pp. 3559–3574, 2022.
- [4] H. Al Jassmi, F. Al Najjar, and A.-H. I. Mourad, "Large-scale 3d printing: the way forward," in *IOP Conference Series: Materials Science and Engineering*, vol. 324, no. 1. IOP Publishing, 2018, p. 012088.
- [5] M. Bazli, H. Ashrafi, A. Rajabipour, and C. Kutay, "3d printing for remote housing: Benefits and challenges," *Automation in Construction*, vol. 148, p. 104772, 2023.
- [6] D. Zhao, W. Guo, and Y. Han, "General function sets theory-based type synthesis of the collaborative 3d printer for planar and non-planar printing," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 236, no. 16, pp. 8965–8979, 2022.
- [7] S. Goessens, C. Mueller, and P. Latteur, "Feasibility study for drone-based masonry construction of real-scale structures," *Automation in Construction*, vol. 94, pp. 458–480, 2018.
- [8] G. Hunt, F. Mitzalis, T. Alhinai, P. A. Hooper, and M. Kovac, "3d printing with flying robots," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 4493–4499.
- [9] L. Poudel, L. G. Marques, R. A. Williams, Z. Hyden, P. Guerra, O. L. Fowler, Z. Sha, and W. Zhou, "Toward Swarm Manufacturing: Architecting a Cooperative 3D Printing System," *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 144, no. 8, pp. 1–15, 2022.
- [10] "Large-scale 3D printing by a team of mobile robots," *Automation in Construction*, vol. 95, no. April, pp. 98–106, 2018.
- [11] J. Sustarevas, D. Kanoulas, and S. Julier, "Autonomous Mobile 3D Printing of Large-Scale Trajectories," no. July, 2022.
- [12] K. Zhang, P. Chermprayong, F. Xiao, D. Tzoumanikas, B. Dams, S. Kay, B. B. Kocer, A. Burns, L. Orr, C. Choi *et al.*, "Aerial additive manufacturing with multiple autonomous robots," *Nature*, vol. 609, no. 7928, pp. 709–717, 2022.
- [13] M.-N. Stamatopoulos, A. Banerjee, and G. Nikolakopoulos, "Flexible multi-dof aerial 3d printing supported with automated optimal chunking," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3033–3039.
- [14] H. Fuchs, Z. M. Kedem, and B. F. Naylor, "On visible surface generation by a priori tree structures," *SIGGRAPH Comput. Graph.*, vol. 14, no. 3, p. 124–133, jul 1980.
- [15] I. Baran and S. Rusinkiewicz, "MIT Open Access Articles Chopper : Partitioning models into 3D-printable parts," 2012.
- [16] J. McPherson and W. Zhou, "A chunk-based slicer for cooperative 3D printing," *Rapid Prototyping Journal*, vol. 24, no. 9, pp. 1436–1446, nov 2018.
- [17] Ultimaker B.V., "Ultimaker cura." [Online]. Available: <https://ultimaker.com/software/ultimaker-cura>
- [18] B. Lindqvist, S. S. Mansouri, P. Sotasakis, and G. Nikolakopoulos, "Collision avoidance for multiple micro aerial vehicles using fast centralized nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9303–9309, 2020, 21st IFAC World Congress.
- [19] A. Sathya, P. Sotasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using panoc," in *2018 European control conference (ECC)*. IEEE, 2018, pp. 1523–1528.
- [20] "Open: Code generation for embedded nonconvex optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6548–6554, 2020, 21st IFAC World Congress.