ISSN 1828-5961

Lorenzo Ceccon
PhD Candidate in the field of ICAR/17 Drawing at the Polytechnic University of Milan.
The research activity focuses on parametric modeling and simulation, the application of artificial intelligence systems, photogrammetry, and virtual/mixed reality for the built environment.

Matteo Cavaglià
PhD Candidate in the field of ICAR/10 Architectural Engineering at the Polytechnic University of Milan. The research activity focuses on the development of integrated visual analysis methods for designing the built environment, particularly within the context of parametric design.

An iterative algorithmic UAV path optimization process for Structure-for-Motion modelling

The use of unmanned aerial vehicles (UAVs) for 3D reconstruction through photogrammetry has gained significant attention in recent years. With the advancement of technology and the availability of affordable drones with high-resolution cameras, capturing aerial images for creating detailed 3D models has become more accessible, however, UAV survey flight planning still presents challenges. The planning stage is essential in aerial photogrammetry as it sets the foundation for efficient and accurate surveying. Proper predictive planning ensures a smooth workflow on site, generating high-quality datasets for reconstruction while minimizing the need for repeat surveys. This approach not only reduces costs but also mitigates potential errors and delays during the survey process.
Within the presented frame of reference, the present study explores the use of UAVs for 3D reconstruction through photogrammetry, focusing on optimizing flight paths and view planning. It addresses challenges such as safety, navigation, and image dataset optimization. The study presents the current advancement of custom parametric workflow developed in Rhino/Grasshopper. The workflow is targeted for average users, aiming to simplify the process and integrate it with architectural and planning workflows. The approach involves four algorithms, including proxy model generation, visibility analysis, path generation, and camera position estimation. The iterative process enhances precision through progressive refinement of the proxy model, offering potential for predictive modelling and effective photogrammetry utilization in UAV planning. Further research and testing are needed to validate real-world performance.

Keywords:
Structure from Motion; Unmanned Aerial Vehicle; Path Planning; GIS; Visual analysis

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

INTRODUCTION

Over the course of the last two decades significant development of "unmanned aerial vehicles" (UAV) reached a sufficient degree of maturation to effectively spawn a wide range of novel drone applications. Regarding civil applications, the relative affordability of high-end camera drones allowed for a more accessible implementation of aerial photography and aerial survey across multiple industries. While many uses may be cited as an example of this revolution, it is possible to group most of them in one of the following three major categories: 3D reconstruction, environment exploration, and aerial imaging (Zhou et al., 2020). In this paper we will focus on the domain of 3D reconstruction via drone-photogrammetry, as a tool to realize a reliable 3D model of an existing environment or building. However, although very different among one another, each of the main applications mentioned tends to face common challenges because of the common role of UAV technology as the cornerstone for the many implemented workflows and processes.
While the frame of reference may slightly change in accordance with the regulations applied in each country, camera drones generally pose different issues that often require careful planning ahead of the actual flight. Regarding UAV operations in urban environments, Radišić (Radišić et al., 2018) details the main issues to address when piloting a drone: awareness of safety issues, optimal navigation and avoidance of dangerous collisions are among the many points discussed. These aspects, while intuitively simple, may pose quite a challenge when planning a flight. In addition, the specific field of 3D reconstruction via drone photogrammetry adds to the equation an additional set of specific issues to consider (Dilbaryan, 2017; Toffanin, 2019).
The basis of any 3D photogrammetry survey is the picture dataset, and the quality of the final output of the 3D reconstruction is largely linked to how much the dataset exhaustively captured the entirety of the surveyed item. Any parts that won't be accurately captured within the dataset, will likely fail to be accounted for in the final model, resulting in a hole in the reconstruction. On the other hand, any data redundancy may cause unnecessary waste of computational power (Factors Affecting Accuracy in Photogrammetry, n.d.). Almost all such problems can be effectively translated into visual relationships between the subjects to digitize and the points of view used to capture it into the base pictures of the referenced dataset.
Hence, the concepts of "path planning" and "view planning", generally associated with the seminal work of Scott (Scott et al., 2003), as the study of evaluation systems to support decision-making strategies linked to visual data. While the topic has been largely discussed, it is still today a very interesting aspect of UAV practices. For instance, Zhou recently regarded path and view planning as a key issue in UAV research (Zhou et al., 2020).

AIM OF THE STUDY

This study aims at inspecting different optimization strategies regarding drone photogrammetry flight planning, focusing on applying view planning concepts within the widespread architectural modelling environment Rhinoceros, and its algorithmic plugin Grasshopper. While general research on view planning itself is well documented and certainly robust, practical applications are still hard to apply by the average user. Because of this, by performing said flight optimization, within such an accessible software package, we aim to achieve a twofold result. On the one hand, the process is manageable without extra skills and extra software packages. On the other hand, an entire software ecosystem, aimed at architectural and planning workflows, can be seamlessly integrated allowing to perform more layered and complex analyses. The main underlying issue is finding a sweet spot within the trade-off between minimizing the number of pictures to be used in the photogrammetric process, while achieving the desired prospected output resolution/error.

MAIN APPROACHES IN PATH/VIEW PLANNING

As Scott shortly summarized, the informally shared underlying challenge of view planning is regarded as follows: "the view planning problem (VPP) is [...] for a given image environment and target object, find a
suitably short view plan N, where $N \subset V$, satisfying the specified reconstruction/inspection goals and achieve this within an acceptable amount of time. The VPP involves reasoning about object surface space S, viewpoint space V and imaging workspace I" (Scott, 2009). In essence, view planning's main objective is to simulate visual conditions of a certain object to develop the optimal survey strategy to generate, among other outputs, also proper picture datasets suitable for 3D reconstructions.
We here note that path planning – devising a specific path the UAV should follow for optimal coverage/accuracy and least redundancy – stems from a conceptually prior assessment of the view optimization of the UAV cameras vis-à-vis the surveyed site, so to ideally place the exact minimum views needed to cover the site effectively.
As any typical evaluation procedure, view planning requires to set both specific evaluation metrics and goals for filtering the optimal solutions to complete the task at hand. Concerning these aspects, the most used evaluation parameter is the object coverage, which was described by Cabreira (Cabreira et al., 2019) as the propriety of a flight path to cover/see every point of the selected area of interest. This is a relevant aspect within the field of drone photogrammetry, as the base input of any 3D reconstruction is the picture dataset of the object to reconstruct, any part that won't be accurately captured within the dataset, will likely fail to be accounted for in the final model, resulting in a hole in the reconstruction. An optimal survey, therefore, needs to cover as many parts of the desired object as possible.
In addition, implementation of assessment processes based on coverage may ultimately vary depending on the simulation setup used to acquire the related data to test. A simulation setup usually falls into one of two main categories: model-based

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

ISSN 1828-5961

simulations, and non-model-based simulations (Munkelt et al., 2010). A model-based simulation, in opposition to non-model-based simulation, develops a test environment where the object to reconstruct via drone photogrammetry is the main data source to develop view planning strategies. This approach could be criticized as having a "chicken and egg" problem, since a base model of the object to reconstruct via photogrammetry must be already available before the actual reconstruction process. However, as already anticipated, it is possible to solve the issue by working with a "proxy model" (Zhou et al., 2020), i.e., a rough model derived from a previous survey, or a simplified geometry extracted from a public database or from a course preliminary survey. Moreover, an iterative approach – whereby the proxy-model is recursively completed, adjusted, and perfected – seems able to guarantee a result which converges to an increasingly accurate model, even though such convergence can be assessed only relative to some chosen parameters, as we will explain further on.
Finally, the form of the final output can also differ greatly based on the designed process. A fundamental reference on the topic can be found within Tarbox' and Gottshlich's (Tarbox et al., 1995) research, where the testing viewpoints to be used for generating optimal view groups were solely picked up on the surface of a specific "view sphere" surrounding the target object. However, many other forms of arbitrary constraints can be linked together, thus varying the analysis results which may range from single optimal points of view (LEE, 1991), to volumetric point clouds (Lartigue et al., 2014), to UAV optimal flight path.

METHODOLOGY

As mentioned earlier, this study aims at devising an integrated workflow within the widespread software package Rhino/Grasshopper for UAV flight optimization in view of a 3D reconstruction process based on the Structure from Motion paradigm.

In view of such an overall goal, we have divided the workflow in a series of subsequent steps or "processes" tackling key aspects of the proposed overall drone photogrammetry workflow, so that the result of each process can be used as a basis for the subsequent ones.
Algorithm A. As a very first step, a preliminary virtual assessment of the spatial conditions based on GIS and other available spatial datasets is performed in Rhino/Grasshopper, dealing with the mere geometrical feasibility of the UAV flight in the surveyed areas. The aim of this step is to offer the tools to better anticipate potential flight issues – including no-fly-zones – and improve the ability of the drone's user to plan more viable flights once on the field.
Algorithm B. A second process aims at analyzing the overall combined model of the survey site – also called "proxy-model" to determine optimal flight volume for covering the surveyed site in its currently expected shape. A gradient-map is hence created in Rhino/Grasshopper as a basis for the view/path planning.
Algorithm C. Based on the analysis stemming from algorithm B, as well as on the proxy model - as amended in the iterative process described here below - a more specific algorithm deals with the automatic creation of a flight path as a starting point for a subsequent phase where it can be checked and fine-tuned.
Algorithm D. This path is then evaluated in its potential coverage capabilities, expected resolution and overall effectiveness by another process, whereby a heuristic optimization approach tries to fine-tune the flying path.
Drone Flight Phase. The next step within the workflow is the real-world testing of the UAV path on the survey site. This step is crucial, as it sets new foundations to the subsequent iterative workflow, by providing pictures and a new "proxy" model based on the photogrammetric reconstruction. Subsequently, in an iterative workflow, the process is repeated based on the reconstructed model resulting from each subsequent dataset as the new starting "proxy" model, which shall then gradually become more refined and complete as the iterations run, filling reconstructions gaps and improving the overall quality (Fig.1). The idea is that every additional flight would help refine the model along a convergent cobweb model.

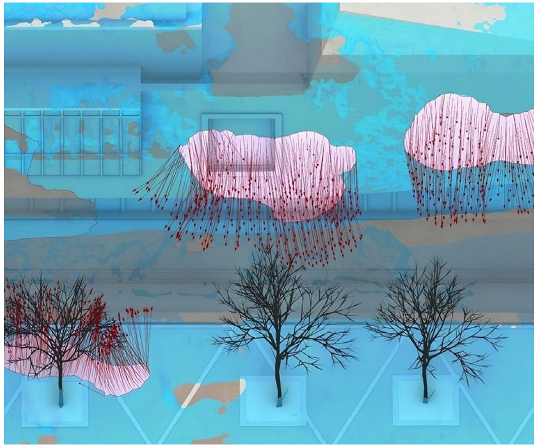

Fig. 1 - SfM reconstruction missing areas to be further investigated through UAV's flights.

ALGORITHM A - PROXY MODEL GENERATION IN GRASSHOPPER FOR RHINO (@IT PLUGIN)

As seen, the chosen starting point for the workflow is the creation of a proxy model of the surveyed area/object. Among the many possible sources of said model - which may also include preliminary, randomly performed UAV flights and the coarse photogrammetric reconstruction stemming therefrom - we chose to adopt a GIS-based solution inside of Grasshopper for Rhino, hereinafter "GH".
More specifically, among the many GIS plugins for GH, we adopted @IT, as it is a free, lightweight and easy to use solution to import GIS data within the GH/Rhinoceros environment.

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

As a precondition to the use of a GIS database, we need to have saved the relevant data about the surveyed area in a series of files along the Shapefile standard, including *.shp, *.dbf, *.shx and other file formats containing the geometrical as well as non-geometrical data related to the area at stake.
For the sake of providing an example, we have chosen a portion of the urban area around the Polytechnic of Milan Leonardo campus in Milan, and downloaded the dataset from the Lombardy Region Geoportal (Geoportale della Lombardia, n.d.). We then imported the dataset in QGIS, where we selected a portion thereof around the campus, and exported it as Shapefile(s).
In GH, we then proceeded to import separately each *.shp geometric file by its database categories. For instance - according to the taxonomy adopted by the Geoportal - we extracted different sets of 2D geometries referring to, among others, the main building ground extension, and the various paths and roads for pedestrians, bicycles and all other transportation means.
The advantage of such an approach in GIS data acquisition in a NURBS CAD modelling environment, as Rhinoceros, is that we can now - at least indirectly - exploit data multidimensionality as stemming from the classifications extracted from the database: we can in fact create and treat geometries according to their classification beyond their pure geometrical shape. Moreover, such classification also allows for a proper matching of many other characteristics, ranging from the materials to be attributed to each category (e.g. asphalt, grass, stone, ...), but potentially extending to other "dimensions", such as the legal classification of space: no-fly-areas, urban planning zoning, etc. Thus, Grasshopper becomes a means to add further informational dimensions to the "pure" CAD-NURBS model, overcoming the mostly geometrical nature of such modelling environments. The extra layer of information can be best exploited in Rhinoceros by grouping geometries in layers and sub-layers based on said extra, non-geometrical, characteristics.
We would here like to stress the great potential ushered by such automatic/parametric process, which does not require a direct data gathering and classification by the user, who can hence directly focus on the task at hand, i.e. the photogrammetric UAV survey planning.
As a last remark, and as already mentioned, such an approach as an automatized way to provide a starting proxy model for the surveyed area is not the only possibility: in case the user does not have access to reliable GIS data of the area, or they are incomplete or not trustworthy, or in case the user already has a rough, proxy model of the area at stake, it is possible to start with any such geometrical proxy model. However, the added value of the GIS/CAD approach lies in the extra data richness which could be extracted from the GIS database, upon which more advanced and grounded algorithmic choices may be built, especially as to the spatial legal features.
An even greater degree of flexibility is the opportunity of merging GIS data with specific 3D models the user may partially own about the surveyed items: such combination makes the proxy model open to be subsequently amended and enriched and can be exploited in iterative workflows aimed at increased precision and in-depth survey. A simple outcome of such a mixed-source model can be seen in Fig.2.

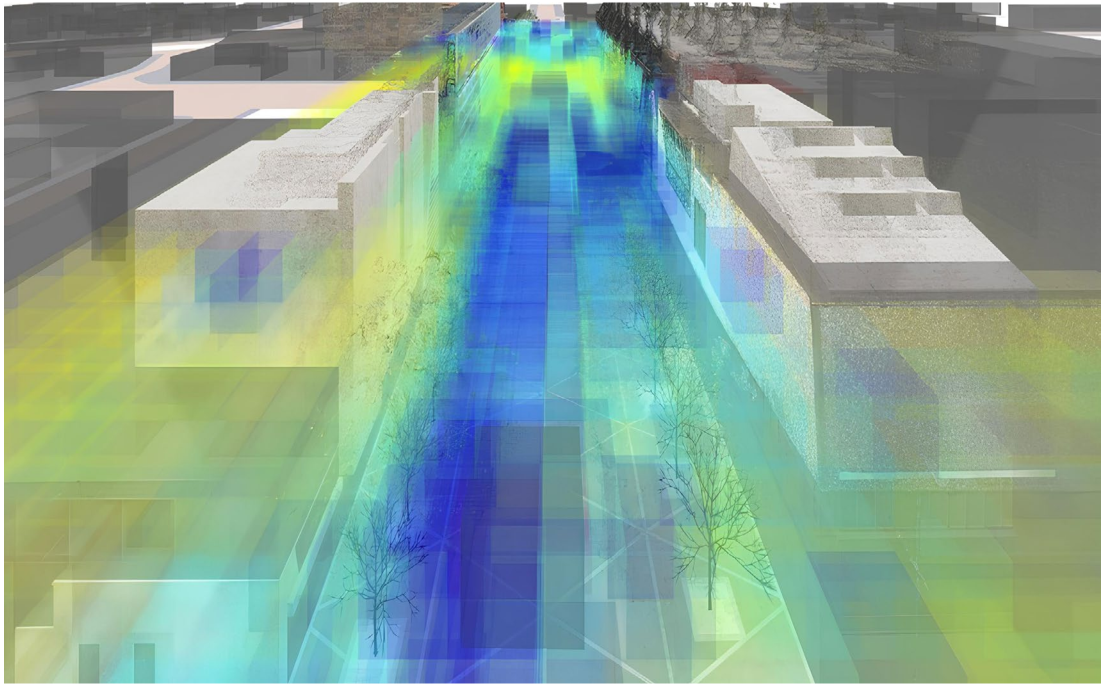

Fig. 2 - A preliminary view-planning evaluation of the surveyed site coverage is conducted. The urban proxy model is generated from a GIS dataset implemented in the Rhinoceros environment to create a mixed database. The color scheme is here aplied to a voxel-based subdivision of space, to create a spatial heatmap about potential coverage. The voxel-based subdivision may also map more complex data based on GIS features.

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

ALGORITHM B - FLIGHT VOLUME EVALUATION VIA A VIEW-BASED PROCESS

The subsequent algorithm detailed, here defined as algorithm B, is the first process to directly engage UAV operations. Algorithm B aims to allow the user to access a rough estimate of the flight visual conditions inside the required environment, as such, the following process can output a preliminary evaluation of the effectiveness of specific "flight volume" in performing the survey. A "flight volume" represents a spatial volume where the drone will operate once in the field, and depending on its extension, the user may be required to request additional authorizations for flying over the necessary areas. A preliminary evaluation on the matter is therefore useful to accurately shape a viable and conscious flight plan.
As the previously cited studies highlighted, a fundamental interaction within view planning is the subject view coverage, which is essentially a collection of visual tests with a binary output in the form of a "visible" or "not-visible" type of feature. The Grasshopper VPL via its strongly geometry-oriented syntax allows generating basic, yet efficient, visual tests which can deliver the desired outputs (Cavaglià, 2023). Custom scripts are ultimately coupled with the built-in view analysis functions of the Ladybug (Ladybug Tools | home Page, n.d.)to achieve a more accurate visual analysis. Ladybug is a renowned solution to perform environmental analysis inside the Rhinoceros environment via Grasshopper definitions (SADEGHIPOUR ROUDSARI et al., 2013). View-oriented analysis is also part of Ladybug capabilities, and although it constitutes a less researched implementation of the tool, it exhibits promising performance in dealing with visual-related issues.
The scripts here detailed mainly exploit the "view percent" analysis component (Visibility Percent, n.d.) which allows determining the visibility of a selected geometry from a specific set of viewpoints. The analysis procedure can be briefly summarized like this: a selected geometry is converted to a mesh (LB Generate Point Grid.py, n.d.; LB Visibility Percent.py, n.d.) and its vertices are then parsed one by one examining if each of them is visible from a list of selected viewpoints. The analysis may account for visual obstructions, in the form of solid geometry located in the virtual space, therefore the visual relationship may result in a "not-visible" output when such obstructions, effectively intersect the line of sight of a specific viewpoint. The results are ultimately mapped on the test geometry via the assignment of a percentage value to each vertex equal to the percentage of viewpoints that could output a "visible" outcome over all accounted viewpoints. Finally, when just a singular viewpoint is selected, the process essentially provides a clear report of the visible parts of the selected test geometry from that singular viewpoint.
Ladybug visual analyses essentially treat any provided viewpoint as a 360° camera; therefore, the process will test the virtual environment across any visual direction around the viewpoint. While this basic setup possesses different merits when developing generalized preliminary analysis, it lacks the capabilities to simulate specific camera conditions. To cope with this limitation, it is used a specific visual obstruction that we define as the "occlusion sphere" (Fig.7). The occlusion sphere is a type of obstructive geometry that is used to mask the virtual space that cannot be viewed via the visual constraints of a real camera, such as a UAV's digital camera. The occlusion sphere is generated from a basic sphere object with its center overlapped with the viewpoint to test, and it presents a hole along the line of sight of the simulated camera. This hole is generated from the specific field of view (FoV) (Christenson, 2011) of the simulated camera, and it is calculated from the parameters of sensor width, sensor height, and focal length (Abbas et al., 2019; Al-Zuky et al., 2015). Although essential, this setup effectively creates a visual constraint that allows to simulate basic camera visual conditions in the working environment of Grasshopper.

The first step in Algorithm B is the generation of the tridimensional flight volume to test the view efficiency of the subject. The flight volume can be defined by the user, and subsequently, it is automatically filled with test viewpoints. Viewpoints are positioned via a cube-shaped unit cell system with edge length given by an interactive parameter "distance". Modifying the "distance" parameters allows controlling the viewpoints density and consequently managing the whole algorithm output quality.
The algorithm subsequently tests the proxy model visibility via one viewpoint at a time. The visual analysis is performed via nested recursion of Ladybug percentage analysis implemented via the Anemone plugin[I], without the use of an occlusion sphere. Therefore, the FoV of each viewpoint is treated as an ideal "omnidirectional camera" (Scaramuzza, 2014) able to cover the full visual sphere. This allows performing a visual analysis completely free of physical constraints aside from positioning.
The algorithm outputs two types of data: the object coverage, along with a second data here defined as "surface inclination". Surface inclination measures how much a visible portion of the proxy model is seen frontally from an associated viewpoint. The value is calculated as the angle between the visual ray linking the viewpoint with a visible face of the proxy model mesh, and the plane's normal of the proxy model's surface in the contact point. It is here speculated that surface inclination may be a valuable indicator to identify areas where a highly steep angle of view may generate distorted pictures. The resulting graphical information may prove insufficient for a correct reconstruction, therefore locally mapping such a feature, may help to predict critical locations when performing the actual survey.
The obtained datasets can be ultimately used to generate heatmaps using as a medium both the proxy model and the viewpoint cluster around it. Data mapping over the viewpoints can also account for the application of certain weights, to generate a hierarchical ranking system based on user-defined criteria, for example, the distance of said viewpoint from the proxy model.
The heatmaps generated from this output can be used as a reference to filter the most effective vol-

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

ISSN 1828-5961

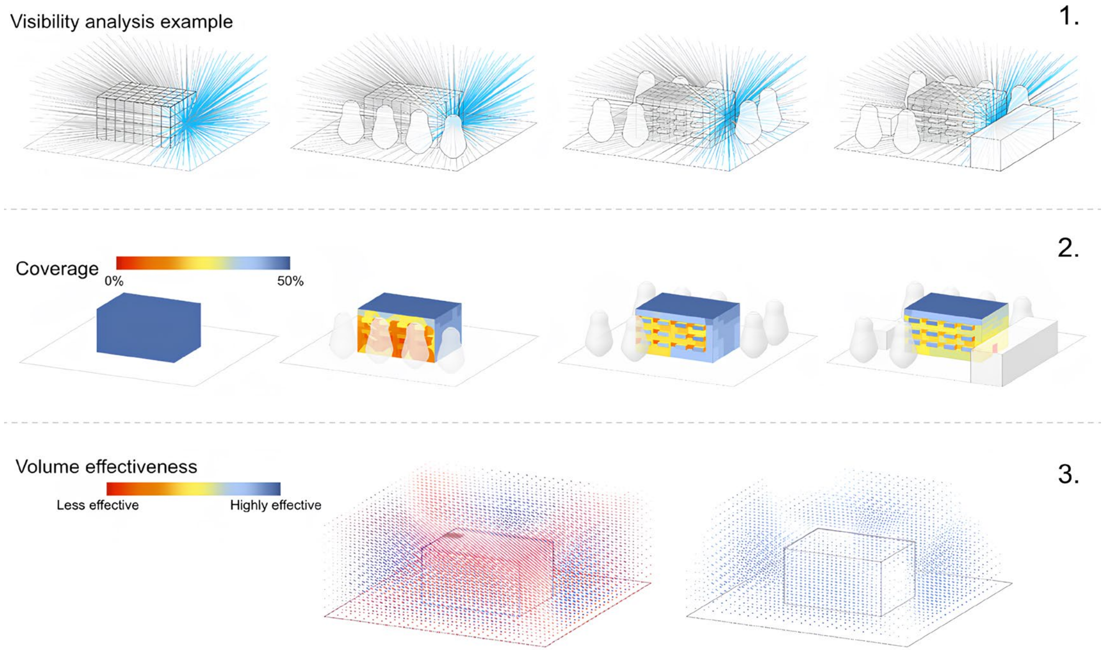

Fig. 3 - Preliminary flight planning assessments rely on visibility analysis within selected flight volumes. They utilize a proxy model representing the target structure to be reconstructed and simplified volumes of the primary obstructions to generate data outputs highlighting the parts of the target structure with reduced visibility accessibility. The picture summarizes the various operational outputs of Algorithm B: [1] Definition of the spatial grid for visual testing of the proxy model. [2] Estimation of potential coverage of the proxy model within the spatial volume previously defined as accessible for navigation. In this image, different proxy models are listed to emphasize the impact of environmental obstructions on predicted coverage. [3] Filtering of potential viewpoints within the spatial volume based on the potential visual coverage offered by each location based on percentage covera, distance, and viewing angle.

umes to use for planning a flight path, and foresee possible problems related to the reconstructions of hard-to-see areas (Fig.3).

ALGORITHM C - PATH GENERATION ALGORITHM BASED ON PRELIMINARY PROMISING VIEWPOINT GENERATION

We now can start building a UAV path which - based and referring to the outputs from the previous algorithms - would assure a reasonable starting point to further deepen the geometrical knowledge of the surveyed site through drone photogrammetry.
In order to do so, all the spatial features, including the possible physical and legal obstacles, as well as all the geometrical analysis about what the most critical survey areas are shall be taken into account. Hence, the new algorithm starts from the proxy model, as generated by Algorithm A - and by any subsequent refinement during the iterative process illustrated hereinafter - as well as the 3D heatmap ranking the relative importance of prospected UAV photographic viewpoints in view of the photogrammetric reconstruction stemming from Algorithm B.
To start with, a mesh encompassing the whole proxy model is created, as both a volume outside of which the UAV shall maintain its flight, and to calculate the intricacy of the geometries to be surveyed. As to this latter aspect, in fact, key parameters of the proxy model can be analyzed, such as the curvature at the specific mesh vertex (we used the Gaussian curvature, through the GH component "Mesh Curvature".).
From this first analysis, the mesh is offset outwards, to simulate a flying "plane" whereon the actual flying path and UAV photo vantage points shall be positioned. Such a process is a sort of preliminary, rule-of-thumb, shortcut to start building a flying path which can ensure some degree of uniformity among pictures, e.g., as to the distance from the surveyed items.
Once the new offset mesh is obtained, a series of geometrical operations are performed to obtain a series of promising photogrammetric vantage points to be included within the UAV path. Such points are derived from the mesh vertices: not all of them, though, as we must take into consideration at least two main aspects. The first aspect, which can be extracted from the pure geometrical features of the mesh at hand, stems from the geometrical complexity of such mesh: the lower the complexity (e.g., a flat area with no reliefs), the less vantage points we need for the reconstruction, up to the threshold depending on the chosen UAV camera focal length and distance, and from the final resolution of the photogrammetric process. A reasonable proxy for such complexity can be extracted from the mesh curvature at the vertices, along with from the presence of discontinuity within the mesh grid. An easy way to account for such aspects is combining a curvature analysis with a mesh simplification component.
To achieve those aims, we tried a series of different components, corresponding to different approaches and related parameters. A first series of similar components we tested are "Mesh Reduce", "Reduce Mesh", "Mesh Reducer", and the new and powerful remeshing component "QuadRemesh". The result is a much-reduced mesh, according to the target face count, and the overall shape depends on some constraints, for instance, using the existing edges as hard edges, or fixed boundaries. The QuadRemesh component even allows for a flexible face count based on the mesh curvature,

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

which seemed a good match to our aims. However, the results were somewhat disappointing, as only when fixing the edges as constraint the result would maintain the main geometric features, but it would also produce a too dense mesh. Same can be said as to random mesh reduction, which does not even guarantee that the overall shape is retained. The component "Decimate Mesh", still acting on the simple mesh simplification, manages to both achieve a much more relevant reduction, and to maintain the overall shape, as described by the mesh edges.

However, while such approaches may yield some interesting results, we realized that a more advanced, physics-based approach would be needed, as eliminating mesh vertices would require a recursive readjustment of all the others. Hence, we adopted such an approach through a specific add-on called "MeshMachine" by Daniel Piker(Piker, 2014; Plankton, n.d.) in turn running in connection with the Kangaroo(Kangaroo 2, n.d.) physics engine. More specifically, we run the optimization algorithm so that only the vertices where the mesh curvature substantially changes, i.e., is high, are preserved, apart from some vertices which - despite having low curvature - have a meaningful and necessary topological position for approximating the overall shape. A further Catmull-Clark subdivision was added to that aim.

The yielded results were satisfying, as the output mesh would, at the same time, approximate the overall shape well, maintaining its edges, as well as having a considerably reduced number of vertices, especially in areas where they could be considered redundant, as explained (Fig. 4).

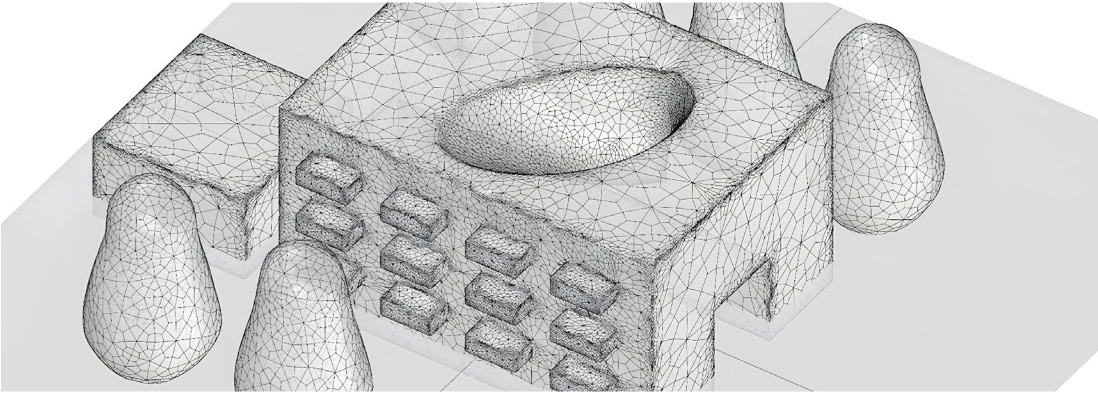

Fig. 4 - The path generating mesh, optimised based on geometric complexity.

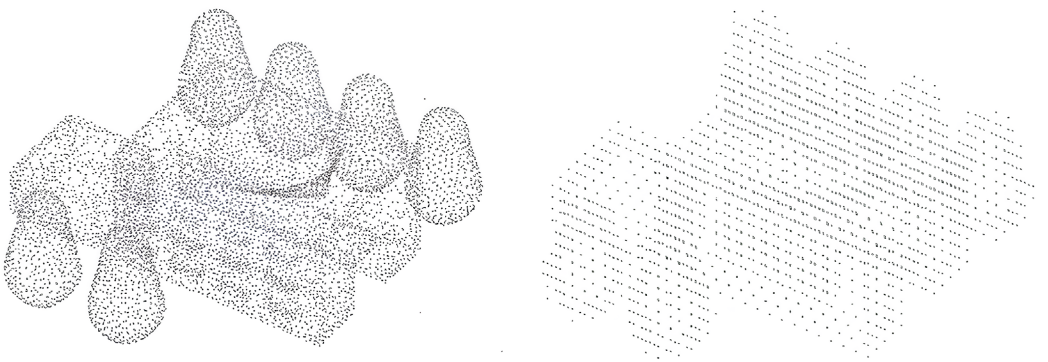

Fig. 5 - The optimized mesh vertices, and a regularized version thereof, adjusted on a regular 3D grid.

The outcome is not yet the whole picture, though. In fact, it must be compared to that of Algorithm B, i.e., the prospected degree of needed coverage of the proxy model by a sufficient number of vantage points, in order to avoid that the mesh is oversimplified, e.g., where the relative lack of detail and changes in the mesh structure would risk implying an insufficient number of points from a photogrammetric point of view. Further research on such integration between Algorithm B and C shall be carried forward in future work. Moreover, the list of the mesh vertices was refined, cleaning out those points which might be interfering with specific extra limitations the user might want to introduce, such as extra no-fly areas and security margins. Lastly, as a final "touch", we performed a regularization of the point positions by approximating the relevant vantage points using a regular isotropic 3D grid, so that the subsequent outputs could be easier to check at a glance, given increased regularity of the geometry, but also avoiding excessive point density beyond a threshold based on the chosen UAV camera characteristics (Fig. 5).

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

ISSN 1828-5961

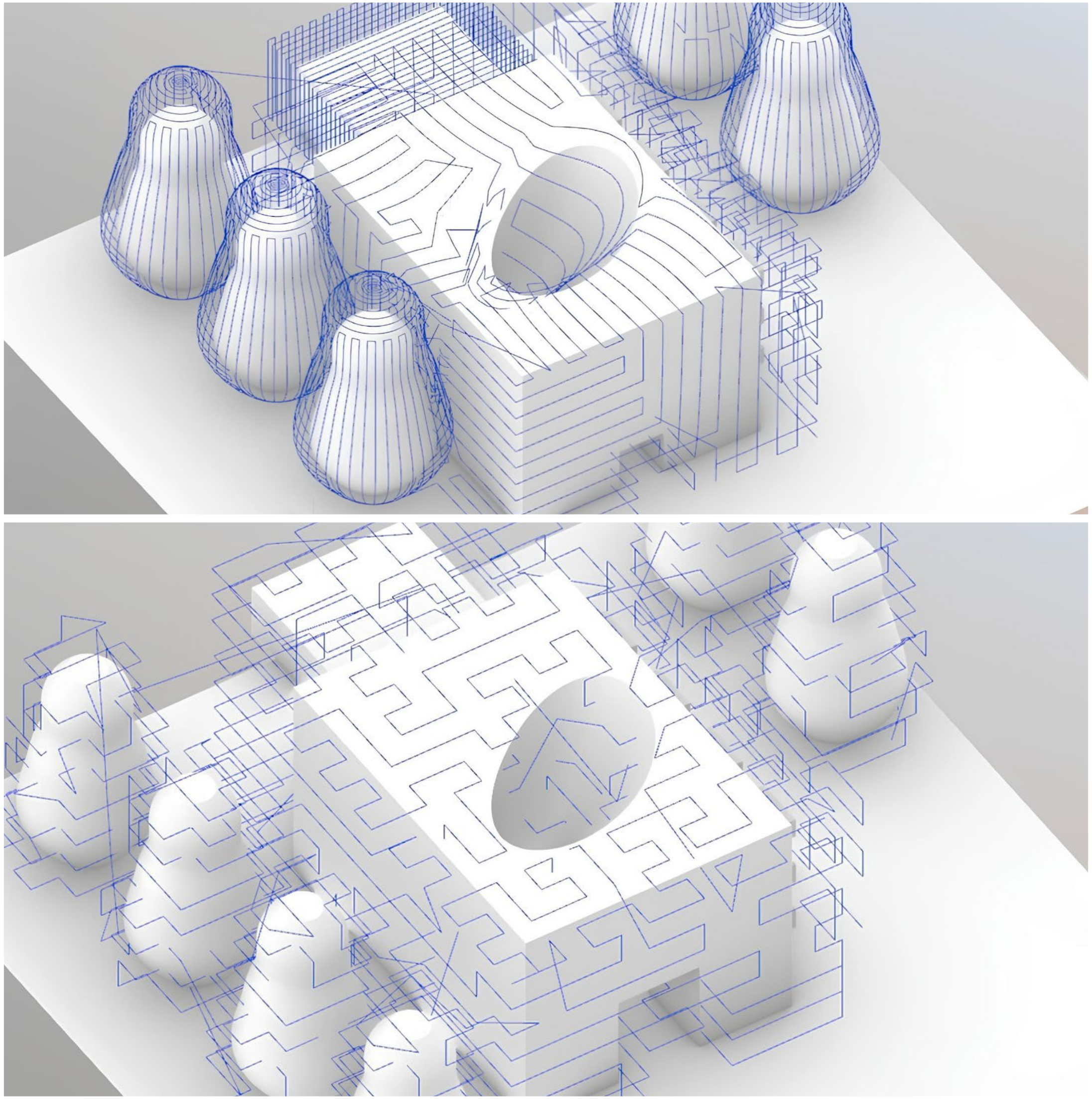

Fig. 6 - [Above] The flight path generated based on the mesh topology. [Below] The optimized flight path developed based on the consideration of geometrical complexity and vantage point adequacy.

To sum up, at the end of the foregoing process, we obtained a simplified offset mesh whose vertices could be used as a good proxy for initial vantage points for the UAV path. Hence, a path connecting those points could now be traced, also considering that those points - since they stem from a mesh - are intrinsically ordered along a specific topology, hence already ordered. In such topology it is in fact always possible to perform analyses about the direct connections between and among connected vertices. The whole topology may also be seen as a graph and be even converted to graph within GH (using and testing the components "Graph from Mesh", "Mesh to Graph" and "Mesh Graph" (Nejur, n.d.)), unleashing all the potential of graph theory (Trudeau, 1993) when it comes to path finding.
However, this final step within Algorithm C is a problematic one, since there is no easy solution to the issue of finding a unique path connecting all the points, and some constraints may hugely impact the final length and complexity of the outcome, and even on its feasibility. For instance, if we set the constraint of not ever passing from the same point(s) more than once, we are dealing with the classical Euler's Travelling Salesman Problem: this may imply that not always is there a solution to the issue, as demonstrated in the literature. However, relaxing that constraint might cause the path to grow indefinitely, and may also imply that the final path might not look like a single, continuous path, but rather as a "leaf" of main paths and contingent "detours".
Graph theory, and a handful of GH plugins - especially "Dijkstra Search" (Dijkstra Shortest Path algorithm for Grasshopper and Rhinoceros, n.d.), "Hamiltonian Cycle" (Hamiltonian Path, n.d.; LEAFVEIN, n.d.) and most importantly "Travelling Salesman Problem (Traveling Salesman Problem, n.d.) Solver" - were precious allies to try analyzing and solve such issues.
As a first attempt to flight path finding, we used Galapagos (Rutten, n.d.) - a heuristic solver for GH (Rutten, 2013) - to minimize the path length, while maintaining all the points as part of the flight. The topological aspects of spatial nearness among each vantage point would ensure an optimized

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

(even though not necessarily the optimal) solution. However, after trying the approach, it seems that the sheer number of points, along with the implicit constraints which would have to be coded within the GH definition to avoid the path crisscrossing the mesh volume result in an inefficient solution to the issue, even though further research would be needed to assess if and how these issues could be addressed.

We then tried another, more straightforward approach, which could deliver a path just based on the point nearness and their topology, ignoring the typical constraints of the Travelling Salesman Problem. Such an approach is clearly not ensuring that the path would be the optimal one, i.e., the shortest passing just once by all points. In fact, there is not even any check, not even a heuristic one, about the existence of other, better solutions. However, this solution has an advantage: it is fast, and it somewhat just depends on the points we provide it with as an input. This means that the process of path optimization may be linked to just one variable - the number and position (and topology) of the vantage points - while the path-creation could be left on the background, at least for the moment being. (See Fig.6).

Despite all the foregoing limitations, we decided this latter solution would better help focusing on the main topic of optimizing the planning of the appropriate vantage points, while the proper path optimization would be researched more in depth in a later phase.

The chosen approach thus constitutes a sound starting point for the next Algorithm D, where a specific a priori assessment and simulation of the prospected photogrammetric results stemming from those vantage points is performed in view of optimizing their number and relative position, in an iterative process which would produce increasingly optimized paths through Algorithm C.

ALGORITHM D - ESTIMATION OF PHOTO LOCATION ALONG A FLIGHT PATH VIA CONSECUTIVE VISUAL OVERLAPPING EVALUATION

Algorithm D aims to ideally "close the loop" by taking on one of the last levels of detail in flight planning methodology, which is the issue of camera positioning.

Photogrammetry involves a few basic guidelines for selecting pictures suitable to be processed within 3D reconstruction operations. Among them, an important feature to consider is image overlap. Image overlap identifies the area of overlapping visibility between different pictures in the dataset and represents a key feature within 3D reconstructions workflows (Sadeq, 2019). Various studies in the field typically recommend an overlap of at least 60% or higher (Róg et al., 2021; Wang et al., 2022) between adjacent images in aerial strips, although the optimal overlap may vary based on environmental conditions (Wolf et al., 2014).

Ladybug view percent analysis component can be used to perform predictive simulation of image overlapping. To simulate this feature, it is possible to perform a view percent analysis for each camera to examine. At this stage of the work, it is also more effective to add occlusion spheres (Fig.7) along the analysis inputs to account for the physical constraints of the camera FoV. Each viewpoint outputs an individual visibility list, where the proxy model vertices are examined and the corresponding statuses as "visible" or "not visible" are reported according to the specific visibility outcome. Changing the camera viewpoints changes the resulting outcomes for each vertex visibility, but the order used to compile the visibility list never changes as long as the proxy model is the same.

This allows to compare visibility lists extracted from different viewpoints to search for matching visibility between elements in the same list position. Any element visible from multiple viewpoints will appear at the same position of all the corresponding visibility lists with the same visible status. Matching visible statuses can be determined via a single AND Boolean operator where all the visibility lists to test are used as inputs. The output of this operation collects a new visibility list, where only the proxy model vertices whose visibility is shared along all the lists used as inputs are recorded with a "True" Boolean value. The list obtained can effectively describe the portion of visibility overlap between the different pictures virtually simulated.

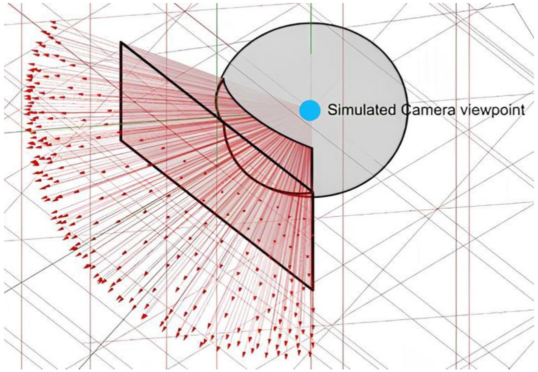

Fig.7. Occlusion sphere setup. The viewpoint visibility is limited only to the visible area of the studied camera.

Algorithm D exploits this and simulates viewpoints overlap in two distinctive stages. Firstly, overlap is simulated over the vertical axis and via a recursive process, the flight path obtained from algorithm C is optimized to account for the vertical overlap of pictures shot along its track. While this step is still unrefined, this process will ultimately boost the effectiveness of the previous results.

As the final step, the flight path is elaborated in a recursive process, where the overlap of consecutive pictures simulated on the path itself are evaluated in pairs. At any iteration, one viewpoint is statically fixed on the track, while a second view-

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

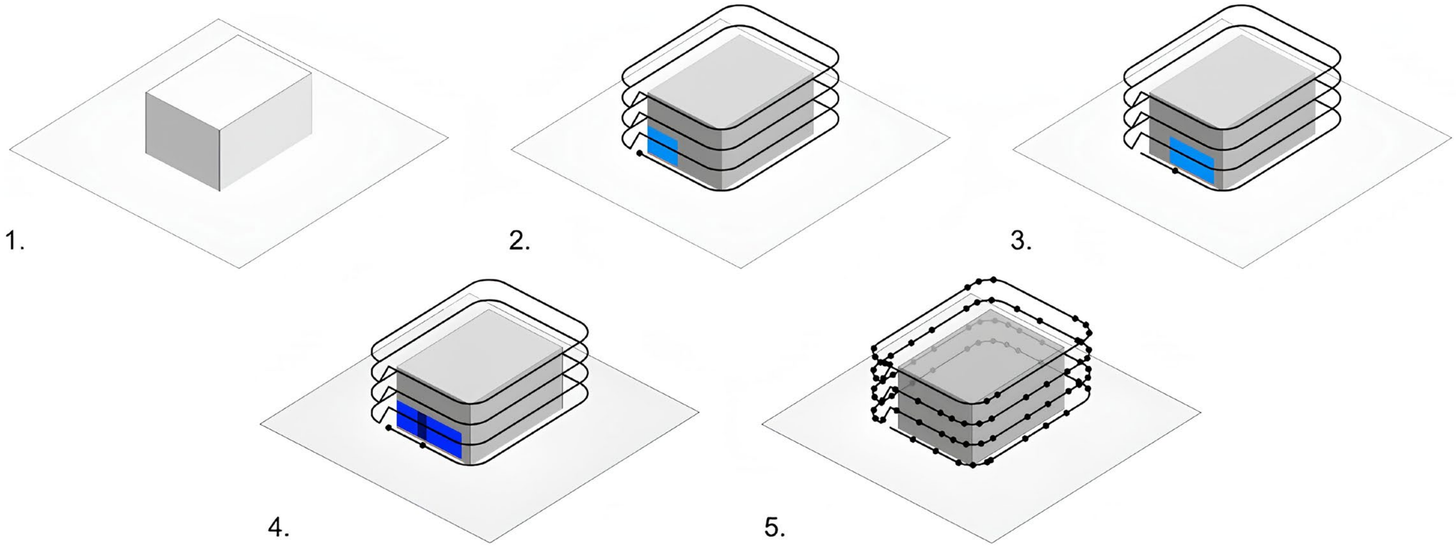

Fig. 8 - Algorithm D workflow: [1] proxy model configuration, [2] fixed viewpoint insertion, [2] mobile viewpoint translation, [4] overlap goal validation, [5] repeat until path saturation. The base flight path exploited here is a simplified test-proxy used for better communication. The overlapping percentage of pictures is defined as a variable parameter, which may be changed by the user; therefore, the analyses can be implemented in different operative scenarios.

point is slowly moved along the path. When specific fitness goals related to overlap are satisfied, the second viewpoint position is saved to be later used as a static point, while another movable point is added next to it. The comparative process repeats until the endpoint of the flight path.
What is obtained from algorithm D, is a collection of points placed along the previously studied flight path, with a proper positioning able to keep a constant visual overlap value. Said value can be parametrically adjusted by the user to increase, or decrease pictures overlap. The viewpoint positioning is linked to the input data used, among others the geometrical fidelity of the proxy model to the real object. The camera positioning, therefore, gets more precise as the input data becomes more polished. Nevertheless, this process, even at the most preliminary stages, can offer a rough prediction of the number of photos needed to properly describe the survey environment via the chosen equipment (Fig.8).

CONCLUSION AND FUTURE RESEARCH

Via the presented study it has been compiled a complete workflow to elaborate multiple relevant stages of UAV's flight planning within the software environment of Grasshopper. The preliminary results show that Grasshopper VPL, along with the robust array of available plugins, can be exploited in detailing versatile predictive workflow regarding visual planning and path planning on UAVs' operations. While the methodology is still far from being accessible or fully tested, further research on the matter can surely improve the current procedures. Considering the many advantages of seamlessly performing complex preliminary planning operations in the same working environment where the subsequent geometrical modelling may take place, it is here regarded that pursuing a deeper understanding on the matter could be a meaningful research topic.
Future elaborations will test the algorithm's performances in real UAV's photogrammetry applications.

DOI: https://doi.org/10.20365/disegnarecon.30.2023.22

REFERENCES

Book with one author: Author, A. (2016). *Title*. City: Publisher.

Abbas, H. Kh., Mohamad, H. J., Al-Saleh, A. H., & Al-Zuky, A. A. (2019). Modelling vision angles of optical camera zoom using image processing algorithm. IOP Conference Series: Materials Science and Engineering, 571(1), 012117. doi: 10.1088/1757-899X/571/1/012117

Al-Zuky, A., & K.Abbas, H. (2015). Estimate Mathematical Model to Calculate the View Angle depending on the Camera Zoom. International Journal of Scientific & Engineering Research.

Cabreira, T., Brisolara, L., & Ferreira Jr., P. R. (2019). Survey on Coverage Path Planning with Unmanned Aerial Vehicles. Drones, 3(1), 4. doi: 10.3390/drones3010004

Cavaglià, M. (2023). A Multi-scale Investigation of Visual Interactions in the Built Environment via the Generation of Parametric Procedures (pp. 509–518). doi: 10.1007/978-3-031-13588-0_44

Christenson, M. (2011). On the architectural structure of photographic space. Architectural Science Review, 54(2), 93–100. doi: 10.1080/00038628.2011.582365

Dijkstra Shortest Path algorithm for Grasshopper and Rhinoceros. (n.d.). Retrieved from https://github.com/gyetis/dijkstra-shortest-path-gh

Dilbaryan, V. (2017). Investigations about the use of UAV photogrammetry and Laser Scanning: Investigation about UAV Photogrammetry and Laser Scan for control of construction works by comparison with CAD model. AV Akademikerverlag.

Factors Affecting Accuracy in Photogrammetry. (n.d.). Retrieved from https://www.photomodeler.com/kb/factors_affecting_accuracy_in_photogramm/

Geoportale della Lombardia. (n.d.). Retrieved from https://www.geoportale.regione.lombardia.it/

Hamiltonian Path. (n.d.). Retrieved from https://www.grasshopper3d.com/video/hamiltonian-path

Kangaroo 2. (n.d.). Retrieved from https://grasshopperdocs.com/addons/kangaroo-2.html

Ladybug Tools | home Page. (n.d.). Retrieved from https://www.ladybug.tools/

Lartigue, C., Quinsat, Y., Mehdi-Souzani, C., Zuquete-Guarato, A., & Tabibian, S. (2014). Voxel-based Path Planning for 3D Scanning of Mechanical Parts. Computer-Aided Design and Applications, 11(2), 220–227. doi: 10.1080/16864360.2014.846096

LB Generate Point Grid.py. (n.d.). Retrieved from https://github.com/ladybug-tools/ladybug-grasshopper/blob/master/ladybug_grasshopper/src/LB%20Generate%20Point%20Grid.py

LB Visibility Percent.py. (n.d.). Retrieved from https://github.com/ladybug-tools/ladybug-grasshopper/blob/master/ladybug_grasshopper/src/LB%20Visibility%20Percent.py

LEAFVEIN. (n.d.). Retrieved from https://www.food4rhino.com/en/app/leafvein

LEE, J. (1991). Analyses of visibility sites on topographic surfaces. International Journal of Geographical Information Systems, 5(4), 413–429. doi: 10.1080/02693799108927866

Munkelt, C., Breitbarth, A., Notni, G., & Denzler, J. (2010). Multi-View Planning for Simultaneous Coverage and Accuracy Optimisation. Procedings of the British Machine Vision Conference 2010, 118.1-118.11. doi: 10.5244/C.24.118

Nejur, A. (n.d.). Ivy. Retrieved from https://www.grasshopper3d.com/group/ivy

Piker, D. (2014, June 9). MeshMachine update. Retrieved from https://www.grasshopper3d.com/m/blogpost?id=2985220%3ABlogPost%3A1085830&maxDate=2018-11-01T08%3A28%3A26.338Z

Plankton. (n.d.). Retrieved from https://www.grasshopper3d.com/group/plankton

Radišić, T., Vidović, A., Ivošević, J., & Wang, B. (2018). Challenges and Solutions for Urban UAV Operations. International Scientific Conference “Science and Traffic Development” (ZIRP 2018).

Róg, M., & Rzonca, A. (2021). The Impact of Photo Overlap, the Number of Control Points and the Method of Camera Calibration on the Accuracy of 3D Model Reconstruction. Geomatics and Environmental Engineering, 15(2), 67–87. doi: 10.7494/geom.2021.15.2.67

Rutten, D. (n.d.). Galapagos. Retrieved from https://www.grasshopper3d.com/group/galapagos

Rutten, D. (2013). Galapagos: On the Logic and Limitations of Generic Solvers. Architectural Design, 83(2), 132–135. doi: 10.1002/ad.1568

SADEGHIPOUR ROUDSARI, M., PAK, M., & VIOLA, A. (2013, August 28). Ladybug: A Parametric Environmental Plugin For Grasshopper To Help Designers Create An Environmentally-conscious Design. doi: 10.26868/25222708.2013.2499

Sadeq, H. A. (2019). Accuracy assessment using different UAV image overlaps. Journal of Unmanned Vehicle Systems, 7(3), 175–193. doi: 10.1139/juvs-2018-0014

Scaramuzza, D. (2014). Omnidirectional Camera. In Computer Vision (pp. 552–560). Boston, MA: Springer US. doi: 10.1007/978-0-387-31439-6_488

Scott, W. R. (2009). Model-based view planning. Machine Vision and Applications, 20(1), 47–69. doi: 10.1007/s00138-007-0110-2

Scott, W. R., Roth, G., & Rivest, J.-F. (2003). View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys, 35(1), 64–96. doi: 10.1145/641865.641868

Tarbox, G. H., & Gottschlich, S. N. (1995). Planning for Complete Sensor Coverage in Inspection. Computer Vision and Image Understanding, 61(1), 84–111. doi: 10.1006/cviu.1995.1007

Toffanin, P. (2019). OpenDroneMap: The Missing Guide (1st Edition). Retrieved from https://odmbook.com/1/

Traveling Salesman Problem. (n.d.). Retrieved from https://www.grasshopper3d.com/video/traveling-salesman-problem

Trudeau, R. J. (1993). Introduction to Graph Theory (Second edition). Dover Pubns.

Visibility Percent. (n.d.). Retrieved from https://docs.ladybug.tools/ladybug-primer/components/3_analyzegeometry/visibility_percent

Wang, F., Zou, Y., Del Rey Castillo, E., & Lim, J. B. P. (2022). Optimal UAV Image Overlap for Photogrammetric 3D Reconstruction of Bridges. IOP Conference Series: Earth and Environmental Science, 1101(2), 022052. doi: 10.1088/1755-1315/1101/2/022052

Wolf, P., DeWitt, B., & Wilkinson, B. (2014). Elements of Photogrammetry: With applications in GIS (4th Edition). McGrawHill Education.

Zhou, X., Yi, Z., Liu, Y., Huang, K., & Huang, H. (2020). Survey on path and view planning for UAVs. Virtual Reality & Intelligent Hardware, 2(1), 56–69. doi: 10.1016/j.vrih.2019.12.004