A Directional Visual Descriptor for Large-Scale Coverage Problems

M. Tamassia, A. Farinelli, V. Murino and A. Del Bue

***Abstract*— Visual coverage of large scale environments is a challenging problem that has many practical applications such as large scale 3D reconstruction, search and rescue and active video surveillance. In this paper, we consider a setting where mobile robots must acquire visual information using standard cameras, while minimizing associated movement costs. The main source of complexity for such scenario is the lack of a priori knowledge of 3D structures for the surrounding environment. To address this problem, we propose a novel descriptor for visual coverage that aims at measuring the orientation dependent visual information of an area, based on a regular discretization of the 3D environment in voxels. Next, we use the proposed visual descriptor to define an autonomous cooperative exploration approach, which controls the robot movements so to maximize information accuracy and minimizing movement costs. We empirically evaluate our approach in a simulation scenario based on real data for large scale 3D environments, and on widely used robotic tools (such as ROS and Stage). Experimental results show that the proposed method significantly outperforms a baseline random approach and an uncoordinated one, thus being a valid proposal for visual coverage in large scale outdoor scenarios.**

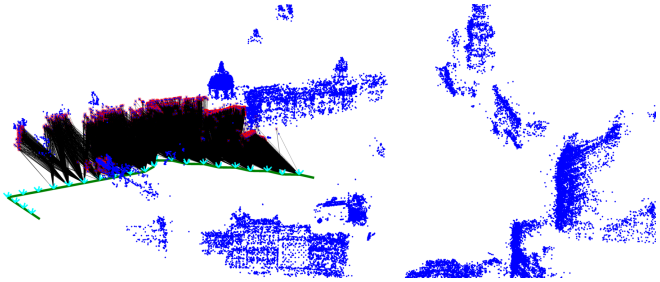


Fig. 1. The image shows a view of Trafalgare Square 3D reconstruction with a robot moving using our strategy that maximise the coverage. The blue dots represent the 3D input data, dark green line shows few moves of the robot path while the black lines represent the projecting ray form the camera to the observed 3D points (shown with red squares). This information is used to define our visual coverage descriptor proposed in this work and the strategies for maximizing the coverage. Notice that in this example the optimization of the proposed utility function makes the robot following the walls of the National Gallery Museum building thus providing a reasonable coverage of the area.

I. Introduction

Visual sensing of large-scale environments has recently attracted increasing research and industrial interests. In particular, image based large scale 3D reconstruction systems have demonstrated strong potentials in obtaining accurate maps for cultural heritage and entertainment purposes [1], [2], [3]. However these systems are normally not parsimonious in the sense that they require thousands/millions/billions of images in order to obtain a satisfactory reconstruction. Moreover, in classical applications, these images are collected from image-based social networks thus accounting only for the most popular tourist attractions in the world.

Now, mobile robotic platforms constitute a promising technology for large scale visual sensing. In fact, mobile robots have been often engaged in applications that involve sensing operations for large scale, dangerous or hostile environments (e.g., search and rescue, surveillance, etc.). However, to date, much of this work focuses either on building an accurate map of the environment by using sensors that provide dense measurements (such as laser range finder) [4] or on searching for important elements in partially unknown or unstructured environments.

In this paper, we take a different perspective and focus explicitly on the visual coverage problem of outdoor environments. Our aim is to provide accurate 3D coverage by using a team of mobile robots equipped only with cameras. We focus on cameras because they are very well suited for large scale environments, as their field of view typically span several meters, moreover, cameras are relatively cheap and by far the most ubiquitous sensors nowadays. Specifically, our mobile platforms build a local 3D map of the environment at each time instance by running a structure from motion [5], [6] procedure on the acquired images. Our goal is then to choose target points for the robots so to maximize the amount of information acquired about the 3D structure of the scene. Figure 1 provides an overview of our application scenario, where a single robot navigates in a large 3D environment and acquire visual observations to maximise the overall coverage[1].

In more detail, this paper makes the following contributions to the state of the art: i) we propose a novel coverage descriptor for 3D environments that encodes crucial statistics such as the number of 3D points observed at a given orientation, the transparency of the voxel and the number of times the voxel has been observed from a given direction. These three key elements convey rich information to decide what would be the best position to observe the voxel in the next move; ii) we propose a cooperative strategy to drive a team of robots in the environments so to maximize visual information encoded with the descriptor presented above; iii) we empirically evaluate our approach by building a

M. Tamassia, A. Del Bue and V. Murino are with PAVIS, Istituto Italiano di Tecnologia (IIT), Genova, Italy. A. Farinelli is with the Department of Computer Science, University of Verona, Italy.

[1]The 3D data used in this paper are available online at `http://grail.cs.washington.edu/projects/bal/` and `http://www.diegm.uniud.it/fsusiello/demo/samantha/`

custom simulation environment that provides to the robot the observed features in a 3D reconstruction of large scale environment, our results prove the effectiveness and applicability of our method.

II. Related work

Visual coverage based on cameras has been addressed from several different perspectives in various fields such as sensors network, computer vision and robotic itself. The recent comprehensive review of Mavrinac and Chen [7] reveals that, even if the camera coverage problem is an active field, many problems are still unsolved. This is because the coverage problem in 3D is ill-posed. In the most general operational scenarios there is no a priori information about the 3D structure of the scene and possible occluders.

Now, the coverage problem has strong relations with the classical *Art Gallery* problem (AGP) [8] – especially for the case of multiple agents/guards and with known scene geometry. In such scenario, a set of agents/guards has to be placed at the vertexes of a known map in order to maximise the coverage of the area. This creates a combinatorial problems for which exact solutions have been actively investigated for some specific [9] and more general configurations [10].

In a more practical scenario, in recent years, there has been a growing interest towards autonomous robotic systems that can explore their surrounding environments to perform various sensing and surveying tasks, with applications ranging from surveillance and security, to environmental surveying. In particular, a large body of such work focuses on exploration strategies for robot and multi-robot systems, using dense sensors that can provide accurate information on the environments, such as 2D or 3D laser range finders [11], or more recently the Kinect system [12]. The idea of frontier based exploration, originally proposed by Yamauchi [13], is a widely used approach to address autonomous exploration and information gathering problems. For example, Burgard and colleagues [4] propose a multi-robot exploration approach where robots cooperatively choose next sensing positions by considering both the utility (in terms of information) of frontier points as well as the cost that robot would incur to reach such position. Our approach is similar to this work because we also define a utility function to drive the robots, however, we do not use a frontier-based method. This is because, in our setting, directional information are crucial to assess the level of coverage of a given voxel. Hence, in our case, frontier voxels can not be easily extracted from the map (e.g., we can not directly consider a voxel as observed when it falls inside the range of the sensor).

In this perspective, the work by Stachniss and Burgard [14] proposes an autonomous approach for exploration that considers coverage maps: an extension of occupancy maps that maintain occupancy probability for each map cell. Based on such representation they proposed a decision-theoretic method for autonomous exploration based on the concept of information gain. Instead, we consider a different concept of coverage, as we are interested in *visual* coverage which measure the number of 3D visual features observed by the robots rather than a probabilistic measure of occupancy.

Surmann and colleagues in [11] propose an approach to determine the next best view of a mobile platform for digitalization of 3D indoor environments using a 3D laser scanner. Finally, Dornhege and Kleiner in [12] propose a frontier-like exploration strategy for a 3D environment based on the Kinect system, focusing on unstructured scenarios (typical of rescue applications). With respect to such previous approach, here we focus on visual coverage, hence explicitly restricting our attention to cameras. In such regard this work provide a similar application as in [15], however the descriptor and simulation scenario proposed here is fundamentally different since it encodes a strong directional information as a spherical histogram of the viewing directions. This also gives a new strategy for coverage since the planned moves take into account the orientation for which the 3D structure is visible.

III. A Novel Visual Coverage Descriptor

We start by introducing our main contribution with a visual coverage descriptor that can encode a directional measure of coverage for each voxel of the 3D map. The basic idea for the descriptor is that a certain 3D volume is covered if it is possible to *view through it*, hence the measure of coverage is related to how much of the voxel volume is "penetrated" by the bundle of rays projected from the camera center. In particular, the descriptor should encode both a classical information of occupancy as in laser range systems and also orientations from which the camera observes a transparent voxel. A voxel volume of a given area might have several 3D reconstructed points only from specific orientations since the number of 2D features extracted at such orientations is higher. On the other hand, voxels without points might be considered empty only after checking all the viewing directions or if there is a large amount of penetrating rays at every viewing direction (i.e. a voxel is transparent because the camera can see through the voxel from every direction).

Given this, our aim is to propose a descriptor which encodes explicitly the viewing direction. To do so we first define a visibility model given a generic 3D point, the camera position and orientation. In general, a 3D point is considered visible if the point is subject to specific constraints that simulates the imaging conditions of a real system. If any 3D point in the map satisfies these constraints, it is considered as observed in our model.

A. The camera visibility model

To formally describe the concept of coverage with visual sensors we adopt the *General Camera Model* [16], [17] which defines the imaging model as a set of rays travelling in a straight line. This is a convenient formalisation[2] for modelling coverage using ray bundles. We also define a robot position that coincides with the camera optical center $\mathbf{t}$. Given all the possible rays departing from the camera

[2]For full details about the camera model check [17], [15] and the graphical description in Figure 2

center, we need now to define a set of criteria to compute the visibility of a 3D point $\mathbf{x} \in \mathcal{P}$ given the camera position $\mathbf{t}$ and camera orientation defined as the pair of angle ϑ, φ. For visual sensors, there are three predominant criteria to impose: field of view, resolution and focus [7].

Field of View [7]. We model the field of view constraint by considering visible only those 3D points which lie within a pyramid having the bottom base corresponding to the image frame and being oriented accordingly to the camera. In order to do so, we first define the angles given the relative position $\mathbf{r} = \mathbf{x} - \mathbf{t}$ of the camera center $\mathbf{t}$ and the 3D point position $\mathbf{x}$ as:

$$\text{pitch}(\mathbf{r}) = \text{atan2}(r_y, r_x), \;\; \text{yaw}(\mathbf{r}) = \text{atan2}\left(r_z, \sqrt{r_x^2 + r_y^2}\right)$$

where $\mathbf{r} = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^\top$.

Now given a camera tilt ϑ and orientation φ, we define the set of the 3D points that are visible such that:

$$\mathcal{P}^{fov}_{\mathbf{t},\vartheta,\varphi} = \Bigg\{ \mathbf{x} \in \mathcal{P} : 0 \leq \text{ang}\Delta(\vartheta, \text{pitch}(\mathbf{x} - \mathbf{t})) \leq \frac{fov}{2} \wedge 0 \leq \text{ang}\Delta(\varphi, \text{yaw}(\mathbf{x} - \mathbf{t})) \leq \frac{fov}{2} \Bigg\}, \quad (1)$$

where *fov* is the *field of view* of the camera and $\text{ang}\Delta(\omega_1, \omega_2)$ is the angular distance between angles ω_1 and ω_2.

Resolution [7]. In most systems, resolution is modelled as a distance constraint that limits the visibility of faraway points [18], [19]. The points $\mathcal{P}^{res}_{\varphi} \subset \mathcal{P}$ that have enough resolution to be detected can be defined as:

$$\mathcal{P}^{res}_{\mathbf{t}} = \{\mathbf{x} \in \mathcal{P} : \|\mathbf{x} - \mathbf{t}\| < \delta_{max}\}, \quad (2)$$

where the maximum range δ_{max} can be fixed for a specific camera model and $\|\cdot\|$ is the euclidean norm. An analysis of image feature detectors recall with respect to resolution can be found in [20] and it can be used as a guideline for setting the parameter δ_{max}.

Focus [7]. Likewise, the scene has to be imaged at the proper focus in order to avoid misdetection of the 2D image features. In practice this constraint mostly holds for elements in the scene that are too close to the camera. This can be implemented as a minimum range constraint [21] such that:

$$\mathcal{P}^{foc}_{\mathbf{t}} = \{\mathbf{x} \in \mathcal{P} : \|\mathbf{x} - \mathbf{t}\| > \delta_{min}\}. \quad (3)$$

Angle of incidence. Here we extend the previous model by considering a further aspect, which we call *angle of incidence*: a 3D point can be reconstructed only if the associated 2D image feature is observable. This further constraint does not only accounts for obvious effects, e.g. an image feature cannot be seen from behind, but also for more subtle 2D image matching reasons. In particular, as empirically observed in [22], 2D image features can be detected and matched only if observed under a limited range of orientations.

For this reason, we first define a normal $\mathbf{n}_p$ associated to each 3D point, and then compute the angle of incidence between the point viewing direction and the normal, to obtain a measure of the camera orientation with respect to the point. If this angle is higher than ϵ the point is not visible because at such camera orientation the image patch support is too warped to be detected. Notice that, with respect to [7] and [15], this is a novel constraint in modelling coverage of points. The formal definition of the constraint is as follows:

$$\mathcal{P}^{aoi}_{\mathbf{t}} = \left\{ \mathbf{x} \in \mathcal{P} : \text{acos}\left(\frac{\mathbf{n}_p \cdot (\mathbf{x} - \mathbf{t})}{|\mathbf{n}_p| \cdot |\mathbf{x} - \mathbf{t}|} \right) < \epsilon \right\}. \quad (4)$$

These four constraints as defined in Eq. (1), (2), (3), (4) give the visibility of a 3D point given a certain camera position and orientation such that:

$$\mathcal{P}^{vis}_{\mathbf{t},\vartheta,\varphi} = \mathcal{P}^{fov}_{\mathbf{t},\vartheta,\varphi} \bigcap \mathcal{P}^{res}_{\mathbf{t}} \bigcap \mathcal{P}^{foc}_{\mathbf{t}} \bigcap \mathcal{P}^{aoi}_{\mathbf{t}}. \quad (5)$$

Given all the presented criteria, if $\mathbf{x} \in \mathcal{P}^{vis}_{\mathbf{t},\vartheta,\varphi}$, we say that point $\mathbf{x}$ is *observed*.

It is possible to apply the same concept of observability with the center of a voxel $\mathbf{v}_k$ to determine whether voxel k is visible (observable) or not. Thus, we determine, at each camera position and orientation, the set $\mathcal{V}^{vis} \subseteq \mathcal{V}$ of visible voxels, where $\mathcal{V}$ is the set of all voxels center:

$$\mathcal{V}^{vis}_{\mathbf{t},\vartheta,\varphi} = \mathcal{V}^{fov}_{\mathbf{t},\vartheta,\varphi} \bigcap \mathcal{V}^{res}_{\mathbf{t}} \bigcap \mathcal{V}^{foc}_{\mathbf{t}}. \quad (6)$$

Similarly, if $\mathbf{v}_k \in \mathcal{V}^{vis}_{\mathbf{t},\vartheta,\varphi}$ we say that voxel k is *observed*. Notice that while the robot is moving, a voxel can be observed more than once and this effect will be modelled explicitly in the visual coverage descriptor.

For brevity, from now on, we will denote with $\mathbf{h}$ the combination of robots coordinates, orientation and tilt:

$$\mathbf{h} = (\mathbf{t}, \vartheta, \varphi). \quad (7)$$

B. Defining the directional coverage descriptor

Our approach considers the information related to the direction from which a voxel is observed. Since we want to store a finite amount of data, we discretize the space of possible 3D directions by dividing the ranges of both vertical component (pitch) and horizontal component (yaw) in a number of intervals[3]. These intervals are indexed in two dimensions, one for the vertical angle (pitch) and one for the horizontal angle (yaw) of the ray coming through the voxel.

To decide which interval contains the coverage information of voxel k about a particular observation direction, the only information needed is the relative position $\mathbf{r}_k = \mathbf{v}_k - \mathbf{t}$ of the voxel center $\mathbf{v}_k$ with respect to the camera center $\mathbf{t}$. Given $\mathbf{r}_k$ we can define two helper functions to compute the indices of the correct interval:

$$\text{bvi}(\mathbf{r}_k) = \left\lfloor \frac{\text{pitch}(\mathbf{r}_k)}{\sigma} \right\rfloor \qquad \text{bhi}(\mathbf{r}_k) = \left\lfloor \frac{\text{yaw}(\mathbf{r}_k)}{\sigma} \right\rfloor, \quad (8)$$

where σ is the angular sampling rate. A graphical representation of the division in intervals and the selection of an interval is shown in Figure 2.

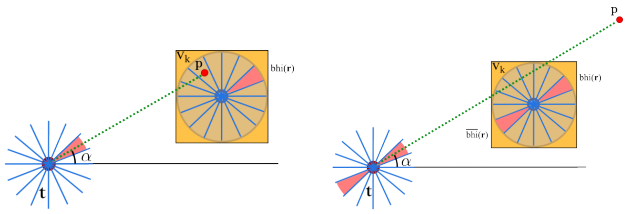


Fig. 2. The image on the left shows a voxel $\mathbf{v}_k$ viewed from the top with a ray departing from the camera position $\mathbf{t}$ and connecting a point $\mathbf{p}$. The discrete angle intervals are given by a sampling step of $\sigma = 22.5° = \frac{\pi}{8}$. This plot also shows how the $\text{bhi}(\mathbf{r})$ index is determined from the viewing direction represented by $\mathbf{r}$ giving an angle $\alpha = \text{yaw}(\mathbf{r})$. The image on the right shows a line that is fully penetrating the voxel and for this reason we both select the interval from "the front" $\text{bhi}(\mathbf{r})$ and the respective interval from "the back" $\overline{\text{bhi}}(\mathbf{r})$ of the voxel.

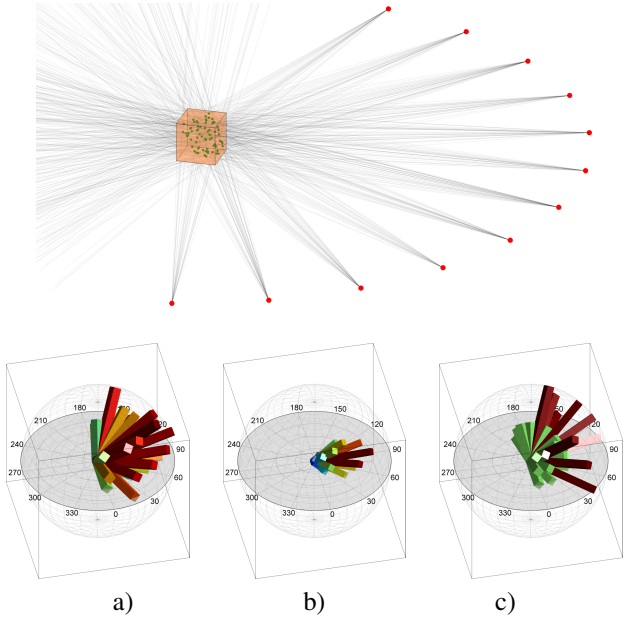


Fig. 3. The figure on top shows a simulated scenario with a voxel (orange cube) and a set of green points contained inside the voxel. The camera centers are represented by red points and black lines departs from such centers and connects the visible 3D points given our visibility model. The three bottom figures show a visualisation of our visual coverage descriptor, for such voxel, as three spherical histograms. The bars of the histogram refer to each viewing angle covered given this specific configurations of voxel position, 3D points location and camera displacements. Histogram a) presents the maximum number of points observed at each angular interval ($\mathcal{F}_{k,vi,hi}$). Histogram b) shows the number of penetrating rays at each angle interval ($\mathcal{T}_{k,vi,hi}$). Finally histogram c) gives the information of the number of times the voxel has been seen from a particular angular interval ($\mathcal{O}_{k,vi,hi}$). Notice that the we have less values in $\mathcal{T}_{k,vi,hi}$ because it refers only to complete penetrations of the voxel.

C. The coverage descriptor features

Our descriptor is associated at each voxel k and it contains different information about the coverage of the voxel for each angular interval (vi, hi) (see Figure 3 for a graphical representation of the descriptor):

- the maximum *number of 3D points* $\mathcal{F}_{k,vi,hi}$ that have been observed inside voxel k from the set of directions included in the angular interval (vi, hi).
- the *number of times* $\mathcal{T}_{k,vi,hi}$ voxel k has been completely *penetrated* by lines with directions included in the angular interval (vi, hi) where a penetration is a line connecting the camera center with a 3D point outside the voxel (we will address this concept in Section IV).
- the *number of times* $\mathcal{O}_{k,vi,hi}$ voxel k has been *observed* from any direction in the angular interval (vi, hi).

In particular here we stress the directionality information stored in the coverage descriptor. This information is crucial to motion strategies that account for the direction where the voxel has not been observed. This is particularly important when we use cameras as the main sensor modality, because the 3D reconstruction of points inside a voxel is highly dependent on 2D image features and such features might be observable only if the camera is oriented with certain angles (as discussed in section III-A). Moreover, since a camera has a longer range than a standard laser sensor, the bundle of rays intersecting voxels are a strong cue to understand if the intersected voxels are free of obstacles. This is a problem for vision sensors because the information extracted after a 3D reconstruction module is far sparser then a standard laser system. In particular this is true with environments having homogeneous textures since they might not be reconstructed because it is not possible to extract a reasonable number of 2D image descriptions (e.g. a wall with homogeneous texture). In contrast, if we know that a high number of rays are passing trough a set of voxels, it is probable that we can see through and no obstacles are present.

Regarding the computational costs, computing $\mathcal{F}$ consists in counting the points for each visible voxel which requires $O(max(|\mathcal{P}^{vis}_{\mathbf{t},\vartheta,\varphi}|, |\mathcal{V}^{vis}_{\mathbf{t},\vartheta,\varphi}|))$. The computation of $\mathcal{T}_{k,vi,hi}$ requires for each visible point $\mathbf{p} \in \mathcal{P}^{vis}_{\mathbf{t},\vartheta,\varphi}$, to perform ray-tracing by computing the line from camera center to 3D point and then to compute voxel penetration. The overall update requires $O(|\mathcal{P}^{vis}_{\mathbf{t},\vartheta,\varphi}|)$ steps. The cost of performing ray tracing can be highly optimized using GPU implementations of such operation. Finally, the computation of $\mathcal{O}_{k,vi,hi}$ to increase the observation counter for each visible voxel can be done in $O(|\mathcal{V}^{vis}_{\mathbf{t},\vartheta,\varphi}|)$.

IV. Coverage Approach

We now use the features of the visual coverage descriptor to define our approach for driving the robots. Similar to previous work in exploration [4] our approach encodes the value of future moves of the robot by defining a utility function. We then perform a greedy maximization of such utility function to choose the next move for the robot. The worth of a move in terms of coverage depends on the specific applications, however, here we assume that it is useless to observe a 3D point from the same position (including orientation) more than once. This is a reasonable assumption if the environmental conditions do not significantly change when two different observations are made (e.g. light condition) and

[3]For ease of notation, we assume that both divisions consist in a number of intervals which is divisible by 2

allows us to decouple the coverage process from the 3D point extraction method used to generate the input data.

A. Single robot mechanism

Our utility function is based on the current state of the descriptors and the current location $\mathbf{t}$ of the robots. Such utility function, is designed to have higher values for parts of the map for which we have less 3D information. Since the location and orientation varies in a continuum range, in order to maximize the utility function we should have a closed form solution for the utility. However, given the complexity of the 3D structure for our reference applications, we discretize the possible locations that robot can take and compute an estimation of the utility function only for such feasible locations.

The utility function is then formed by three elements: a gain component, representing an estimation of the coverage gain given a possible next location, and two cost components, representing the cost that the robot incurs when translating and rotating. In more detail, calling $\mathbf{h} = (\mathbf{t}, \vartheta, \varphi)$ the current position and $\mathbf{h}' = (\mathbf{t}', \vartheta', \varphi')$ the position for which the utility has to be computed, we have that the utility function is defined as follows:

$$\mathrm{u}(\mathbf{h}, \mathbf{h}') = A \cdot \mathrm{g}(\mathbf{t}', \vartheta', \varphi') - B \cdot |\mathbf{t} - \mathbf{t}'| - C \cdot (\mathrm{ang}\Delta(\vartheta, \vartheta') + \mathrm{ang}\Delta(\varphi, \varphi')), \quad (9)$$

where A, B and C are weighting parameters to be tuned depending on the platform, and $\mathrm{g}(\cdot)$ is the gain function. Such gain function considers two main elements: the number of points observed in each voxel (that we want to maximize) and a measure of coverage for a voxel which encodes how many observations we obtained for a voxel (regardless of the number of points that we observed inside such voxel).

Since our main goal is to cover voxels with a presence of 3D points, it is important to focus on portions of the map that contain structured 3D information (e.g., as given by walls, building facades etc.). In more detail, we want to capture the concept of **spatial locality** of points in neighbouring voxels. Since 3D points are typically located on structures that occupy several voxels, it is reasonable to assume that the number of 3D points that can be observed from a given angular direction for neighbouring voxels is correlated. To model this, we use a Gaussian function N based on the distance $\mathbf{d} = \begin{bmatrix} d_x & d_y & d_z \end{bmatrix}^\top$ between voxel centers to define an **expectation of the number of points** $E_{\mathrm{b}(k,\mathbf{t})}$ observable in a voxel k from a direction in (vi, hi) where $(k, vi, hi) = \mathrm{b}(k, \mathbf{t})$ such that:

$$\mathrm{b}(k, \mathbf{t}) = \Big(k, \mathrm{bvi}(\mathbf{t} - \mathbf{v}_k), \mathrm{bhi}(\mathbf{t} - \mathbf{v}_k)\Big), \quad (10)$$

where bvi and bhi have been defined in equation (8).

Then for a voxel k we compute $E_{\mathrm{b}(k,\mathbf{t})}$ as follows:

$$E_{\mathrm{b}(k,\mathbf{t})} = \frac{\sum_{\bar{k}} \left(N(\overbrace{\mathbf{v}_k - \mathbf{v}_{\bar{k}}}^{\mathbf{d}}) \cdot \mathcal{F}_{\bar{k},vi,hi} \right)}{K}, \quad (11)$$

where $\bar{k} \neq k$ is the index of a neighbouring voxel, K is the number of neighbouring voxels, $\mathcal{F}_{\bar{k},vi,hi}$ is the number of points observed in voxel $\bar{k}$ from a direction in (vi, hi), and $\mathbf{d} = \mathbf{v}_k - \mathbf{v}_{\bar{k}}$ is the distance between the voxel centers.

Next, we define a measure of **coverage**, that accounts for the number of observations $\mathcal{O}$. In general, we wish to move to a location if such location offers useful information on 3D points contained in a voxel, hence, the more observations a voxel may contain, the better. Now, to minimize robot movements, it is crucial to model the concept of *transparent* voxels, i.e. voxels that do not contain objects and hence will not provide interesting information. In particular, we want to model the fact that if a voxel is transparent, observing it from a specific viewing direction makes it unnecessary to observe it again from the specular direction.

However, in general the robot is not able to precisely detect whether a voxel is transparent (i.e. empty) or whether it contains an object that has no visual features (e.g. a building with a monochromatic facade). Furthermore, in presence of occlusions, no information about the voxel can be obtained. To address these issues we consider penetrations measured as the number of points behind the voxel that have been seen through it, and we consider a high number of penetrations to be a hint of transparency. Notice that, when deciding whether a voxel is transparent we must consider directional information as a voxel can be transparent from some viewing directions and not transparent from others.

Considering all this, to model our concept of directional coverage we use both the number of observation $\mathcal{O}$ and penetrations $\mathcal{T}$ stored in the coverage descriptor. Specifically, when a voxel k is observed from location $\mathbf{t}$, our method takes into account the number of penetrations $\mathcal{T}_{\mathrm{b}(k,\mathbf{t})}$ from the observation direction and also from the specular direction $\mathcal{T}_{\overline{\mathrm{b}}(k,\mathbf{t})}$ where $\overline{\mathrm{b}}(k, \mathbf{t})$ is given by

$$\overline{\mathrm{b}}(k, \mathbf{t}) = \Big(k, \overline{\mathrm{bvi}}(\mathbf{t} - \mathbf{v}_k), \overline{\mathrm{bhi}}(\mathbf{t} - \mathbf{v}_k)\Big), \quad (12)$$

where $\overline{\mathrm{bvi}}$ and $\overline{\mathrm{bhi}}$ select the indices for the opposite intervals as graphically shown in Figure 2.

Then, if the sum of these two terms is higher than a given threshold τ (which is necessary to filter out noise), the voxel is considered transparent and the coverage $\mathrm{cov}(k, \mathbf{t})$ from the observation direction is defined as the sum of the number of observations from "the front" and from "the back". In the opposite case, the coverage is defined as the number of observations from "the front" only. In such way, $\mathrm{cov}(k, \mathbf{t})$ can be formalised as:

$$\mathrm{cov}(k, \mathbf{t}) = \begin{cases} \mathcal{O}_{\mathrm{b}(k,\mathbf{t})} + \mathcal{O}_{\mathrm{ob}(k,\mathbf{t})} & \text{if } \mathcal{T}_{\mathrm{b}(k,\mathbf{t})} + \mathcal{T}_{\mathrm{ob}(k,\mathbf{t})} > \tau \\ \mathcal{O}_{\mathrm{b}(k,\mathbf{t})} & \text{otherwise} \end{cases}. \quad (13)$$

Now, recall that in our framework once a 3D point was observed a further observation will not provide useful information. Hence, we need to model the degree of knowledge that the robot acquired about a voxel. In particular, we

measure the lack of observations for a voxel as follows:

$$\mathrm{U}(k,\mathbf{t}) = \begin{cases} 1 & \text{if } \mathrm{cov}(k,\mathbf{t}) = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

We can now define the **gain function** by combining the estimation on the number of 3D points for a voxel k (i.e. $E_{\mathrm{b}(k,\mathbf{t})}$) with the information on whether such voxel was ever observed (i.e. $\mathrm{U}(k,\mathbf{t})$). Specifically, the gain function is formulated as follows:

$$\mathrm{g}(\mathbf{t},\vartheta,\varphi) = \sum_{k \in \mathcal{V}^{vis}_{\mathbf{t},\vartheta,\varphi}} \left(\left(c + \beta \cdot \rho_{\mu_{occ},p}(E_{\mathrm{b}(k,\mathbf{t})}) \right) \cdot \mathrm{U}(k,\mathbf{t}) \right), \tag{15}$$

where c is a constant (we set this to 1 in the experiments), $\rho_{\mu_{occ},p}(\cdot)$ is a ramp function, μ_{occ} is the average number of points observed for each voxel and direction from the beginning of the entire process (computed selecting only strictly positive values), p is a parameter to tune according to the scenario (we set this to 0.1 in our experiments). The ramp function performs a soft thresholding to consider equally interesting voxels with very high number of points. In fact, for high values of observations, the number of points is only a consequence of the type of surface and not of the presence or absence of objects inside a voxel. Similar considerations hold for low observations that essentially indicate noise.

Notice that, the gain function is designed to have a null value if the system has already collected information regarding voxel k from an orientation in interval (ϑ,φ). Otherwise, it assumes a value in $[c, \beta + c]$, the value being higher if the voxel is considered interesting. Finally, the constant c in the utility function provides a positive value for voxels which are not observed and surrounded by empty or unobserved voxels. This gives the robot an incentive to move also in the initial steps of the coverage process.

B. Multi-Robot Visual Coverage

We extend our approach to multi-robot coverage by using a centralised greedy method to distribute a set of robots R.

In particular, a central controller stores a visual coverage descriptor for each voxel in the map, and updates such structures with the information communicated by the robot. Specifically, each robot communicates its position and the observed 3D points to the central controller and since robots have homogeneous sensors the controller can directly update the descriptors for the visible voxels. Moreover, whenever a robot reaches a target point it sends the acquired visual information and queries for the next point to reach.

Now, to maximise the efficiency of the coverage process we must spread the robots across the map so to avoid visiting positions which have redundant information (i.e., observing the same 3D points more than once). To do so we discount the amount of redundant information from the utility of a future move of a robot i given the next positions of all other robots, by not considering the utility yielded by voxels that will be observed by the other robots.

As our experimental Section will show, this coordination approach significantly improves the performance of our system with respect to not coordinated robots. This is more evident in the first initial moves of the robots where the increase of the coverage is steepest.

V. Experiments

Each robot is equipped with a camera and navigates with the associated battery costs for rotation and translation. In such case, the robots are completely autonomous and work in both coordinated and uncoordinated modalities.

A. Simulation Setup details

We test our system using the 3D reconstruction of three different large scale environments obtained from real world images: Piazza Bra (Verona, Italy), Trafalgar Square (London, United Kingdom) and Piazza San Marco (Venice, Italy). Since such 3D reconstructions are not aligned to a particular reference system, we fix a floor plane by detecting the first two dominant eigenvectors obtained from a Principal Component Analysis (PCA) applied on the 3D point clouds. This procedure works particularly well for man-made structures where most of the reconstructed 3D points have clearly a relevant set of points at the basement. The third PCA axis represents the elevation of the 3D reconstruction. This registration is then followed by a filtering of sparse (outlying) 3D points that is tuned to remove isolated points. This stage is performed by creating a 3D grid by sampling uniformly the three axes. Next, we consider the number of 3D points inside each voxel and filter out all the points that do not reach a particular quantity in a grid voxel. Notice that this stage also creates the voxel grid used for computing and updating the visibility coverage descriptors. After this removal stage we re-align the reference system to eliminate the influence of gross outliers in the first PCA computation.

Moreover, as required by our visibility model, the 3D points are augmented with a visibility normal to simulate the orientation from which the image feature point is observable. This is done interactively with a semi-automatic labelling processes using human operators. First, the operator selects, a group of 3D points on the map by preferring points belonging to a consistent architectural element (e.g. a bulding facade, a frontal arch). In order to define the normal direction we then perform a local PCA on the set of points an we select the third most dominant eigenvector assuming that the locally selected points are mostly planar. The process continues until most of the points have been selected. The few unselected points are then considered as fully visible from every direction. An example of the viewing normals for the Piazza Bra 3D reconstruction scenario are shown in Figure 4. Finally, we compute the 2D map for the navigation module by projecting the 3D points onto the (x, y) plane. We also project the previous voxel grid and check if a 2D map square is occupied by counting the number of points inside. Regarding the navigation setup, we simulate a mobile robotic platform that is able to localize itself and navigate autonomously in a 2D map. To this end, we use *ROS (Robot*

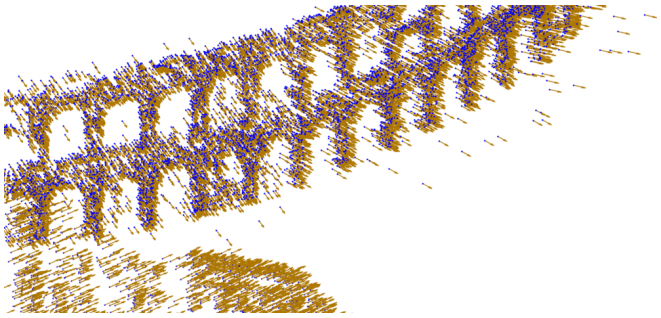


Fig. 4. The figure shows a detail of the 3D reconstruction of Piazza Bra and the associated normals at each 3D point.

Operating System) to control our simulated platforms, and we simulate the 2D environment by using *Stage 2D*, a ROS module that simulates virtual 2D worlds. We use Videre erratic platforms as models for our robots.

B. Empirical methodology

In what follows, we provide details about the parameters used in the experiments for the three scenarios. Notice that all these 3D reconstructed maps vary in size and complexity thus creating a favourable test bed for evaluating the algorithms.

In more detail, we used a uniform length of 4 meters per voxel side and a sampling rate of of $45°$ for orientations. For the visibility model (see Section III-A), we set the camera field of view to $90°$, the focus constraint to a minimum distance of 1 meter and a maximum distance of 30 meters. As for the angle of incidence constraint (Eq. (4)), we set $\epsilon = 70°$. Regarding the descriptor, we set the minimum number of penetrations to consider a voxel transparent (Eq. (13)) to 5. As for the coefficient of the utility function (Eq. 9) we tuned the values for A, B and C for our Videre model through a tuning phase performed with a single robot only on the Piazza Bra scenario, and used such value for all the experiments.

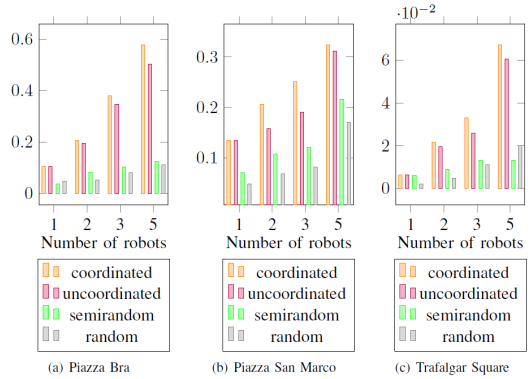
We then run experiments with 1, 2, 3 and 5 robots, spawning them close to each other. Finally, the value for the parameter β in the gain function (Eq. (15)) is fixed to 2000.

We evaluate our approach by fixing a maximum travel distance (i.e., how many meters the robot/camera can move in total) which accounts for battery limitations. We fix different maximum travel distances (measured in meters) to account for the different sizes of the three maps: Piazza Bra is 28773 m^2, Piazza San Marco is 31515 m^2 and Trafalgar Square is 66708 m^2. Specifically, we use a maximum travel cost of 400 meters for Piazza Bra, 600 meters for Piazza San Marco and 800 meters for Trafalgar Square. We then compute a metric which measures the ratio of 3D points that have been observed over the total number. Formally, if we call H the set of all the positions used by all the robots, we consider a point $\mathbf{p}$ visible if $\mathbf{p} \in \bigcup_{(\mathbf{t},\vartheta,\varphi)\in H} \mathcal{P}^{vis}_{\mathbf{t},\vartheta,\varphi}$. Hence, the more 3D points were observed the higher the performance of the coverage approach.

We then benchmark our approach against a baseline random method that performs a walk in the environment by randomly choosing the next target at a fixed distance. We also propose a comparison with a semi-random approach which again performs a random walk but always performs all the possible rotations for each location. Moreover, for multi-robot scenarios, we provide results for both the coordinated coverage strategy and an uncoordinated approach where robots do not share any information.

C. Experimental Results

Figure 5 shows a comparison for the coordinated, uncoordinated, semi-random and random approaches for increasing number of robots. These results show that the coordinated approach has the best performance with respect to uncoordinated and random strategies with a significance gap in performance.



(a) Piazza Bra (b) Piazza San Marco (c) Trafalgar Square

Fig. 5. Results for the coordinated, uncoordinated semi-random and random strategies.

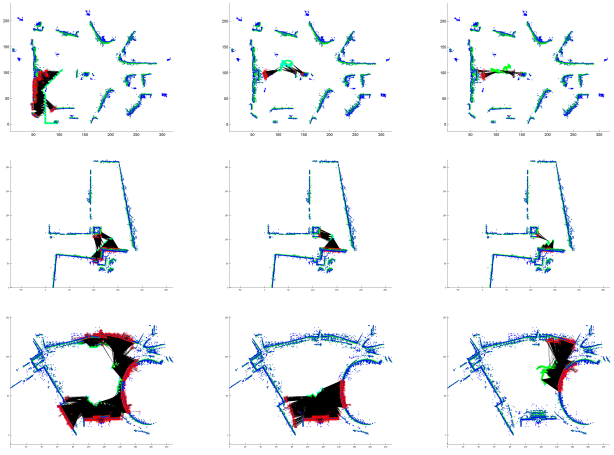
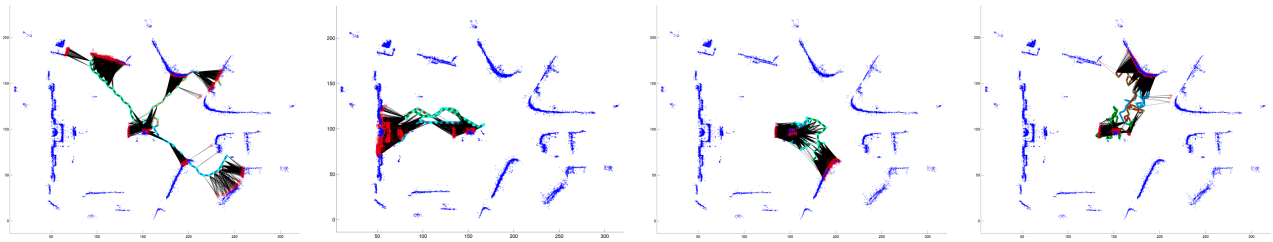


Fig. 6. The figure shows examples of runs of our approach against the baselines in the three different environments from top to down: Trafalgar Square, Piazza San Marco and Piazza Bra (best viewed in color).

In more detail, Figure 6 shows some examples of the final coverage results on the three scenarios. Notice that for these examples, the optimized path clearly follow the predominant



(a) Cooperative (b) Uncooperative (c) Semi-random (d) Random

Fig. 7. The figure shows the behaviour of the cooperative, uncooperative, random and semi-random strategies in an experiment with three robots moving in Trafalgar Square. Robot paths are encoded with colours (best viewed in color).

3D structure close to the respective initial points where the robots start their navigation. The first row of Figure 6 shows results for Trafalgare Square, the largest map in our dataset. Given that the map has relevant empty spaces, random and semi-random approaches struggle to provide even a minimal coverage of the area. Differently, our proposed strategy locks into the main 3D structure and it goes towards observing voxels with a continuous structure and orientations such as the one given by the National Gallery Museum walls and facade. The second row shows the results for the Piazza San Marco experiment (we only display few moves for clarity). Notice that, our approach is the only on to observe both sides of the tower, thus achieving a better coverage in the area. Finally, in the third row we present results for the smallest scenario, Piazza Bra. Since this is a more compact area both random and semi-random approaches increase their performance. However, our approach drives the robot along the profile of the buildings while the random approaches are moving in a limited, local area.

Finally, we also present a qualitative example for a coordinated and uncoordinated approach trial using a fleet of three robots for the Trafalgar Square scenario. Figure 7 shows that our coordinated approach spreads the robots towards relevant elements of the 3D structure and it avoids overlapping between field of views of the robots.

VI. Conclusions

In this paper we propose a novel descriptor for visual coverage in large scale outdoor environments. Based on such descriptor we propose a cooperative strategy to drive a team of robots in the environments so to maximize visual information. We empirically evaluate our approach in a simulation environment that uses real data from large-scale outdoor scenarios (i.e., Piazza Bra, Trafalgar Square and Piazza San Marco) and widely used robotic tools (such as ROS and Stage) for 2D navigation. The empirical results show that our approach is indeed able to provide effective strategies for visual coverage in outdoor environments.

References

[1] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys, "Building rome on a cloudless day," in *Computer Vision ECCV 2010*, 2010, pp. 368–381.

[2] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski, "Bundle adjustment in the large," *Computer Vision–ECCV 2010*, pp. 29–42, 2010.

[3] R. Gherardi, M. Farenzena, and A. Fusiello, "Improving the efficiency of hierarchical structure-and-motion," in *CVPR*, 2010.

[4] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.

[5] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[6] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *Vision algorithms: theory and practice*. Springer, 2000, pp. 298–372.

[7] A. Mavrinac and X. Chen, "Modeling coverage in camera networks: A survey," *IJCV*, 2013.

[8] J. O'Rourke, *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987, vol. 57.

[9] M. Couto, P. de Rezende, and C. de Souza, "An exact algorithm for minimizing vertex guards on art galleries," *Int. Transactions in Operational Research*, 2011.

[10] A. Kröller, T. Baumgartner, S. P. Fekete, and C. Schmidt, "Exact solutions and bounds for general art gallery problems," *J. Exp. Algorithmics*, vol. 17, no. 1, pp. 2.3:2.1–2.3:2.23, May 2012.

[11] H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 34, pp. 181 – 198, 2003.

[12] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," in *SSRR*, 2011.

[13] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97*, 1997.

[14] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Proc. of the Int. Conference on Artificial Intelligence (IJCAI)*, 2003.

[15] A. Del Bue, M. Tamassia, F. Signorini, V.Murino, and A. Farinelli, "Visual coverage using autonomous mobile robots for search and rescue applications," in *SSRR*, Linkoping, Sweden, 2013.

[16] P. Sturm, "Multi-view geometry for general camera models," in *Computer Vision and Pattern Recognition, 2005, (CVPR 2005).*, vol. 1. IEEE, 2005, pp. 206–212.

[17] G. Schweighofer and A. Pinz, "Fast and globally convergent structure and motion estimation for general camera models," in *BMVC*, 2006.

[18] A. Mittal and L. S. Davis, "A general method for sensor planning in multi-sensor systems: Extension to random occlusion," *Int. Journal of Computer Vision*, vol. 76, no. 1, pp. 31–52, 2008.

[19] Y. Yao, C.-H. Chen, B. Abidi, D. Page, A. Koschan, and M. Abidi, "Sensor planning for automated and persistent object tracking with multiple cameras," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[20] D. Q. Huynh, A. Saini, and W. Liu, "Evaluation of three local descriptors on low resolution images for robot navigation," in *Image and Vision Computing New Zealand, 2009. IVCNZ'09. 24th Int. Conference*. IEEE, 2009, pp. 113–118.

[21] J. Park, P. C. Bhat, and A. C. Kak, "A look-up table based approach for solving the camera selection problem in large camera networks," in *Proc. of the Int. Workshop on Distributed Smart Cameras (DCS06)*, 2006.

[22] H. Aanæs, A. L. Dahl, and K. S. Pedersen, "Interesting interest points: A comparative study of interest point performance on a unique data set," *IJCV*, 2012.