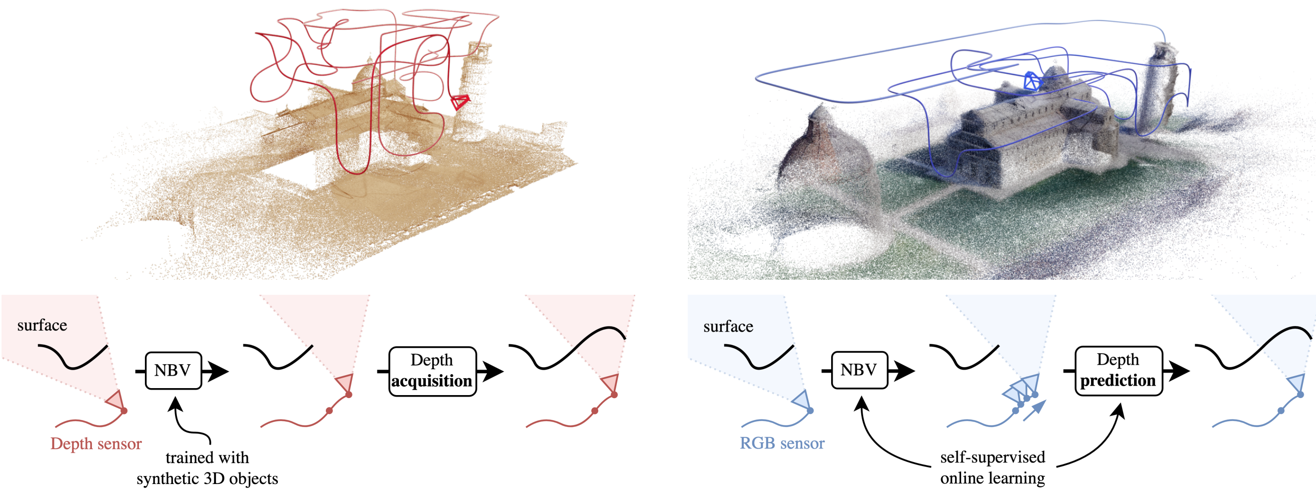
MACARONS: Mapping And Coverage Anticipation with RGB Online Self-Supervision

Antoine Guédon Tom Monnier Pascal Monasse Vincent Lepetit
LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France
https://imagine.enpc.fr/~guedona/MACARONS/



(a) NBV methods with a depth sensor (*e.g.*, [27]) (b) Our approach MACARONS with an RGB sensor

Figure 1. **Mapping and Coverage Anticipation with RGB Online Self-Supervision. (a)** NBV methods such as [27] rely on a depth sensor to perform path planning (bottom) and scan the environment (top). They need to be trained with explicit 3D supervision, generally on small-scale meshes. **(b)** Our approach MACARONS instead simultaneously learns to efficiently explore the scene and to reconstruct it (top) using an RGB sensor only. Its self-supervised, online learning process scales to large-scale and complex scenes.

Abstract

We introduce a method that simultaneously learns to explore new large environments and to reconstruct them in 3D from color images only. This is closely related to the Next Best View problem (NBV), where one has to identify where to move the camera next to improve the coverage of an unknown scene. However, most of the current NBV methods rely on depth sensors, need 3D supervision and/or do not scale to large scenes. Our method requires only a color camera and no 3D supervision. It simultaneously learns in a self-supervised fashion to predict a "volume occupancy field" from color images and, from this field, to predict the NBV. Thanks to this approach, our method performs well on new scenes as it is not biased towards any training 3D data. We demonstrate this on a recent dataset made of various 3D scenes and show it performs even better than recent methods requiring a depth sensor, which is not a realistic assumption for outdoor scenes captured with a flying drone.

1. Introduction

By bringing together Unmanned Aerial Vehicles (UAVs) and Structure-from-Motion algorithms, it is now possible to reconstruct 3D models of large outdoor scenes, for example for creating a Digital Twin of the scene. However, flying a UAV requires expertise, especially when capturing images with the goal of running a 3D reconstruction algorithm, as the UAV needs to capture images that together cover the entire scene from multiple points of view. Our goal with this paper is to make this capture automatic by developing a method that controls a UAV and ensures a coverage suitable to 3D reconstruction.

This is often referenced in the literature as the "Next Best View" problem (NBV) [14]: Given a set of already-captured images of a scene or an object, how should we move the camera to improve our coverage of the scene or object? Unfortunately, current NBV algorithms are still not suitable for three main reasons. First, most of them rely on a voxel-based representation and do not scale well with the size of the scene. Second, they also rely on a depth sensor, which is in practice not possible to use on a small UAV in outdoor conditions as it is too heavy and requires too much power. Simply replacing the depth sensor by a monocular depth prediction method [49, 56, 77] would not work as such methods can predict depth only up to a scale factor. The third limitation is that they require 3D models for learning to predict how much a pose will increase the scene coverage.

In this paper, we show that it is possible to simultaneously learn in a self-supervised fashion to efficiently explore a 3D scene and to reconstruct it using an RGB sensor only, without any 3D supervision. This makes it convenient for applications in real scenarios with large outdoor scenes. We only assume the camera poses to be known, as done in past works on NBV [31,48,80]. This is reasonable as NBV methods control the camera.

The closest work to ours is probably the recent [27]. [27] proposed an approach that can scale to large scenes thanks to a Transformer-based architecture that predicts the visibility of 3D points from any viewpoint, rather than relying on an explicit representation of the scene such as voxels. However, this method still uses a depth sensor. It also uses 3D meshes for training the prediction of scene coverage. To solve this, [27] relies on meshes from ShapeNet [6], which is suboptimal when exploring large outdoor scenes, as our experiments show. This limitation can actually be seen in Figure 1: The trajectory recovered by [27] mostly focuses on the main building and does not explore the rest of the scene. By contrast, we use a simple color sensor and do not need any 3D supervision.

As our experiments show, we nonetheless significantly outperform this method thanks to our architecture and joint learning strategy. As shown in Figure 2, our architecture is made of three neural modules that communicate together:

1. Our first module learns to predict depth maps from a sequence of images in a self-supervised fashion.
2. Our second module predicts a "volume occupancy field" from a partial surface point cloud. This field is made of the probability for any input 3D point to be occupied or empty, given the previous observed images of the scene. We train this module from past experience, with partial surface point cloud as input and aiming to predict the occupancy field computed from the final point cloud.
3. Our third module predicts for an input camera pose the "surface coverage gain", *i.e.*, how much new surface will be visible from this pose. We improve the coverage estimation model introduced by [27] and propose a novel, much simpler loss that yields better performance. We rely on this module to identify the NBV.

While exploring a new scene and training our architecture, we repeat the three following steps: (1) We identify the Next Best View where to move the camera; (2) We move the camera to this Next Best View, collect images along the way, and build a self-supervision signal from the collected images, which we store in the "Memory"; (3) We update the weights of all 3 modules using Memory Replay [50]. This avoids catastrophic forgetting and significantly speeds up training compared to a training procedure that uses only recent data, as such data is highly correlated. This last step establishes a synergy between the different parts of the model, each one providing inputs to the other parts.

We compare to recent work [27] on their dataset made of large scale 3D scenes under the CC license. We evaluate the evolution of total surface coverage by a sensor exploring several 3D scenes. Our online, self-supervised approach that learns from RGB images is able to have better results than state-of-the-art methods with a perfect depth sensor.

To summarize, we propose the first deep-learning-based NBV approach for dense reconstruction of large 3D scenes from RGB images. We call this approach MACARONS, for Mapping And Coverage Anticipation with RGB Online Self-Supervision. Moreover, we provide a dedicated training procedure for online learning for scene mapping and automated exploration based on coverage optimization in any kind of environment, with no explicit 3D supervision. Consequently, our approach is also the first NBV method to learn in real-time to reconstruct and explore arbitrarily large scenes in a self-supervised fashion. We experimentally show that this greatly improves results for NBV exploration of 3D scenes. It makes our approach suitable for real-life applications on small drones with a simple color camera. More fundamentally, it shows that an autonomous system can learn to explore and reconstruct environments without any 3D information *a priori*. We will make our code available on a dedicated webpage for allowing comparison with future methods.

2. Related Work

We first review prior works for next best view computation. We then discuss depth estimation literature, from which we borrow techniques to avoid the need for depth acquisition.

2.1. Next Best View (NBV)

Approaches to NBV can be broadly split into two groups based on the scene representation. On the one hand, volumetric methods represent the scene as voxels used as inputs of traditional optimization schemes [2, 8, 13, 15, 55, 64, 65, 68] or more recently, neural networks [31, 48, 66]. On the other hand, surface-based approaches [9, 35, 36, 41, 80] operate on dense point clouds representing the surface as computed by the depth sensor. Although modeling surfaces allows to preserve highly-detailed geometries, it does not scale well to complex scenes involving large point clouds, thus limiting their applicability to synthetic settings of isolated centered objects with cameras lying on a sphere. The closest work to ours is Guédon *et al*. [27] which proposes an hybrid approach called SCONE that maximizes the surface coverage gain using a volumetric representation. Our proposed approach yet differs in two ways. First, although SCONE processes real complex scenes with free camera motions at inference, it can only be trained on synthetic datasets [5]. Our approach instead benefits from a new online self-supervised learning strategy, which is the source of our better performances. Second, like most of NBV methods, SCONE assumes to have access to a depth sensor whereas our framework relies on RGB images only.

To relax the need for depth acquisitions, we propose a self-supervised method that learns to predict a depth map from color images captured by an arbitrary RGB sensor such as a flying drone while exploring a new environment.

2.2. Depth estimation

Monocular. Classical monocular deep estimation methods are learned with explicit supervision, using either dense annotations acquired from depth sensors [17, 18, 20] or sparse ones from human labeling [10]. Recently, other works used self-supervision to train their system in the form of reprojection errors computed using image pairs [21, 22, 74] or frames from videos [26, 84, 85]. Advanced methods even incorporate a model for moving objects [1, 11, 24–26, 34, 42, 43, 57, 63, 78]. However, all these approaches are typically self-trained and evaluated on images from a specific domain, whereas our goal is to obtain robust performances for any environment and any RGB sensor.

Sequential monocular. A way to obtain better depth predictions during inference is to assume the additional access to a sequence of neighboring images, which is the case in our problem setup. Traditional non-deep approaches are efficient methods developed for SLAM [19, 51, 52, 76], which can further be augmented with neural networks [3, 40, 62]. Deep approaches typically refine at test time monocular depth estimation networks to account for the image sequence [4, 11, 39, 46, 47, 61]. Other methods instead modify the architecture of monocular networks with recurrent layers to train directly with sequences of images [37, 54, 69, 70, 75, 83]. Inspired by deep stereo matching approaches [7, 12, 33, 45, 60, 79, 81, 82], another line of works utilizes 3D cost volumes to reason about the underlying geometry at inference [28, 32, 44, 71–73]. In particular, the work of Watson *et al*. [71] introduces an efficient cost volume based depth estimator that is self-supervised from raw monocular videos and that provides state-of-the-art results. In this work, we adapt the self-supervised learning framework from [71] to jointly learn our NBV and depth estimation modules.

3. Problem setup

The general aim of Next Best View is to identify the next most informative sensor position(s) for reconstructing a 3D object or scene efficiently and accurately. Like previous works [27], we look for the view that increases the most the total coverage of the scene surface. Optimizing such criterion makes sure we do not miss parts of the target scene.

We denote the set of occupied points in the scene by $\chi \subset \mathbb{R}^3$; its boundary $\partial\chi$ is made of the surface points of the scene. During the exploration, at any time step $t \geq 0$, our method has built a partial knowledge of the scene: It has captured *observations* $(I_0, ..., I_t)$ from the *poses* $(c_0, ..., c_t)$ it visited. The 6D poses $c_i = (c_i^{\text{pos}}, c_i^{\text{rot}}) \in \mathcal{C} := \mathbb{R}^3 \times SO(3)$ encode both the position and the orientation of the sensor. In our case, observations I_i are RGB images.

To solve the NBV problem, we want to build a model that takes as inputs $(c_0, ..., c_t)$ and $(I_0, ..., I_t)$ and predicts the next sensor pose c_{t+1} that will maximize the number of new visible surface points, *i.e*., points in $\partial\chi$ that will be visible in the observation I_{t+1} but not in the previous observations $I_0, ..., I_t$. We call the number of new visible surface points the *surface coverage gain*. We assume the method is provided a 3D bounding box to delimit the part of the scene it should reconstruct.

4. Method

Figure 2 gives an overview of our pipeline and our self-supervised online learning procedure. During online exploration, we perform a training iteration at each time step t which consists in three steps.

First, during the *Decision Making step*, we select the next best camera pose to explore the scene by running our three modules: the *depth module* predicts the depth map for the current frame from the last capture frames, which is added to a point cloud that represents the scene. This point cloud is used by the *volume occupancy module* to predict a volume occupancy field, which is in turn used by the *surface coverage gain module* to compute the surface coverage gain of a given camera pose. We use this last module to find a camera pose around the current pose that optimizes this surface coverage gain.

Second, the *Data Collection & Memory Building* step, during which the camera moves toward the camera pose previously predicted, creates a self-supervision signal for all three modules and stores these signals into the Memory.

Third and last, the *Memory Replay* step selects randomly supervision data stored into the Memory and updates the

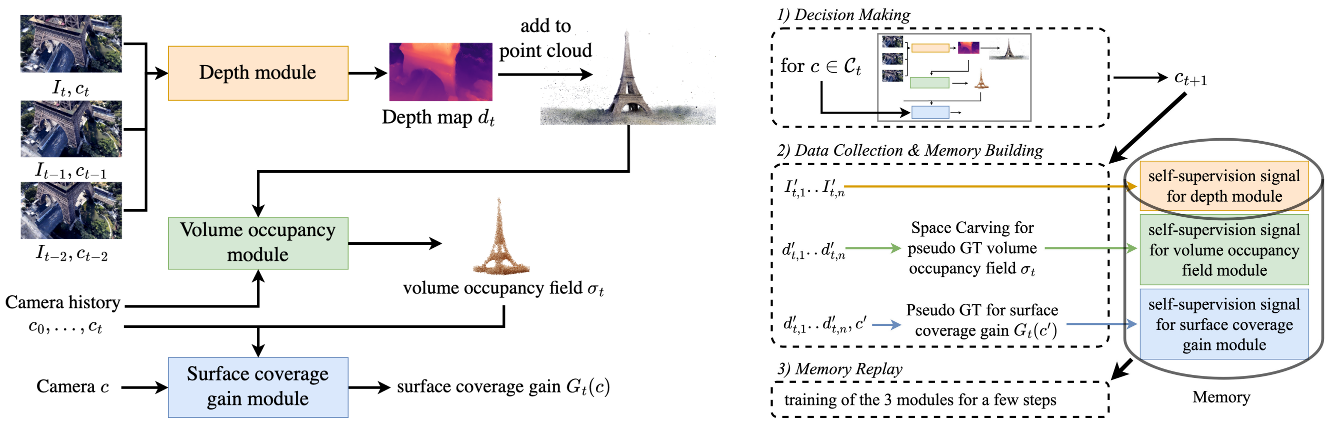


Figure 2. **Our architecture and the three steps of our self-supervised procedure.**

weights of each of the three modules.

We detail below our architecture and the three steps of our training procedure.

4.1. Architecture

Depth module. The goal of this module is to reconstruct the surface points observed by the RGB camera in real time during the exploration. To this end, it takes as input a sequence of recently captured images $I_t, I_{t-1}, .., I_{t-m}$ as well as the corresponding camera poses $c_t, c_{t-1}, .., c_{t-m}$ with $0 \leq m \leq t$ and predicts the depth map d_t corresponding to the latest observation I_t.

We follow Watson *et al.* [71] and build this module around a cost-volume feature. We first use pretrained ResNet18 [29] layers to extract features f_i from images I_i. We define a set of ordered planes perpendicular to the optical axis at I_t with depths linearly spaced between extremal values. Then, for each depth plane, we use the camera poses to warp the features $f_{t-j}, 0 < j \leq m$ to the image coordinate system of the camera c_t, and compute the pixelwise L1-norm between the warped features and f_t. This results in a cost volume that encodes for every pixel the likelihood of each depth plane to be the correct depth. We implement this depth prediction with a U-Net architecture [59] similar to [71] that takes as inputs f_t and the cost volume to recover d_t. More details can be found in [71]. Contrary to [71], we suppose the camera poses to be known for faster convergence. We use $m = 2$ in our experiments. In practice, the most recent images of our online learning correspond to images captured along the way between two predicted poses c_t and c_{t+1} so we use them instead of $I_0, .., I_t$.

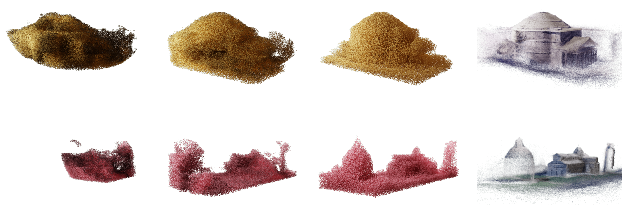
We then backproject the depth map d_t in 3D, filter the point cloud and concatenate it to the previous points obtained from $d_0, .., d_{t-1}$. We filter points associated to strong gradients in the depth map, which we observed are likely to yield wrong 3D points: We remove points based on their value for the edge-aware smoothness loss appearing in [23, 30, 71] that we also use for training. We hypothesize such outliers are linked to the module incapacity to output sudden changes in depth, thus resulting in over-smooth depth maps. We denote by S_t the reconstructed surface point cloud resulting from all previous projections.

Volume occupancy module. This module computes a "volume occupancy field" σ_t from the predicted depth maps. Given a 3D point p, $\sigma_t(p) = 0$ indicates that the module is confident the point is empty; $\sigma_t(p) = 1$ indicates that the module is confident the point is occupied. As shown in Figure 3, during exploration, $\sigma_t(p)$ evolves as the module becomes more and more confident that p is empty or occupied.

We implement this module using a Transformer [67] taking as input the point p, the surface point cloud S_t and previous poses c_i, and outputting a scalar value in $[0, 1]$. The exact architecture is provided in the supplementary material. This volumetric representation is convenient to build a NBV prediction model that scales to large environments. Indeed, it has a virtually infinite resolution and can handle arbitrarily large point clouds without failing to encode fine details since it uses mostly local features at different scales to compute the probability of a 3D point to be occupied.

Surface coverage gain module. The final module computes the surface coverage gain of a given camera pose c based on the predicted occupancy field, as proposed by [27] but with key modifications.

Similar to [27], given a time step t, a camera pose c and a 3D point p, we define the visibility gain $g_t(c; p)$ as a scalar function in $[0, 1]$ such that values close to 1 correspond to occupied points that will become visible through c and values close to 0 correspond to points not newly visible through c. In particular, the latter includes points with low occupancy, points not visible from c or points already



(a) Volume occupancy at t_0 (b) Volume occupancy at $t_1 > t_0$ (c) Volume occupancy at $t_2 > t_1$ (d) Reconstructed surface

Figure 3. **Evolution of the volume occupancy field and final surface** estimated by MACARONS on two examples.

visible from prior poses. We model this function using a Transformer-based architecture accounting for both the predicted occupancy and the camera history.

Specifically, we first sample N random points $(p_j)_{1\leq j\leq N}$ in the field of view F_c of camera c with probabilities proportional to $\sigma_t(p_j)$ using inverse transform sampling. Second, we represent the camera history of previous camera poses $c_0, ..., c_t$ for each point p_j by projecting them on the sphere centered on p_j and encoding the result into a spherical harmonic feature we denote by $h_t(p_j)$. Finally, we feed the camera pose c, a 3D point p_j, its occupancy $\sigma_t(p_j)$ as well as the camera history feature $h_t(p_j)$ to the Transformer predicting the corresponding visibility gain $g_t(c; p_j)$. Note that the self-attention mechanism is useful to deal with potential occlusions between points.

The visibility gains of all points are aggregated using a Monte Carlo integration to estimate the surface coverage gain $G_t(c)$ of any camera pose c:

$$G_t(c) = V_c \cdot \frac{1}{N}\sum_{j=1}^{N} g_t(c; p_j) \cdot l(c; p_j) \;, \tag{1}$$

where V_c and $l(c; p_j)$ are two key quantities we introduce compared to the original formula of [27]. First, we multiply the sum by $V_c = \int_{F_c} \sigma_t(p)dp$ (*i.e.*, the volume of occupied points seen from c) to account for its variability between different camera poses, which is typically strong for complex 3D scenes. Second, since the density of surface points visible in images decrease with the distance between the surface and the camera, we also weight the visibility gains with factor $l(c; p_j) = \min(1/\|c^{\text{pos}} - p_j\|^2, \tau)$ inversely proportional to the distance between the camera center and the point, to give less importance to points far away from the camera. We also made several minor improvements to the computation of the surface coverage gain, which we detail in the supplementary material.

4.2. Decision Making: Predicting the NBV

At any time t, the Decision Making step simply consists in applying successively the three modules of the model, as described in Section 4.1. Consequently, we first apply the depth prediction module on the current frame I_t and use the resulting depth map d_t to update the surface point cloud S_t. Then, for a set of candidate camera poses $\mathcal{C}_t \subset \mathcal{C}$, we apply the other modules to compute in real time the volume occupancy field and estimate the surface coverage gain $G_t(c)$ of all camera poses $c \in \mathcal{C}_t$. In practice, we build $\mathcal{C}_t$ by simply sampling around the current camera pose but more complex strategies could be developed. We select the NBV as the camera pose with the highest coverage gain:

$$c_{t+1} = \operatorname*{arg\,max}_{c\in\mathcal{C}_t} G_t(c) \;. \tag{2}$$

We do not compute gradients nor update the weights of the model during the Decision Making step. Indeed, since the camera visits only one of the candidate camera poses at next time step $t + 1$, we do not gather data about all neighbors. Consequently, we are not able to build a self-supervision signal involving every neighbor. As we explain in the next subsection, we build a self-supervision signal to learn coverage gain from RGB images only by exploiting the camera trajectory between poses c_t and c_{t+1}.

4.3. Data Collection & Memory Building

During Data Collection & Memory Building, we move the camera to the next pose c_{t+1}. This is done by simple linear interpolation between c_t and c_{t+1} and capture n images along the way, including the image I_{t+1} captured from the camera pose c_{t+1}. We denote these images by $I'_{t,1}, ..., I'_{t,n}$ so $I'_{t,n} = I_{t+1}$, and write $I'_{t,0} := I_t$.

Then, we collect a self-supervision signal for each of the three modules, which we store in the Memory. Some of the previous signals can be discarded at the same time, depending on the size of the Memory.

Depth module. We simply store the consecutive frames $I'_{t,1}, ..., I'_{t,n}$, which we will use to train the module in a standard self-supervised fashion.

Volume occupancy module. We rely on Space Carving [38] using the predicted depth maps to create a supervision signal to train the prediction of the volume occupancy field. Our key idea is as follows: When the whole surface of the scene is covered with depth maps, a 3D point $p \in \mathbb{R}^3$ is occupied iff for any depth map d containing p in its field of view, p is located behind the surface visible in d. Consequently, if we had images covering the entire scene and their corresponding depth maps, we could compute the complete occupancy field of the scene by removing all points that are not located behind depth maps.

In practice, we only have access to the depth maps $d'_{t,1}, ..., d'_{t,n}$ predicted for the images captured so far. We can still compute an intermediate occupancy field, which is an approximation but can be used as supervision signal. Since it is not reliable far away from the depth maps when the whole surface has not been covered, we only sample points around the newly reconstructed surface within a margin that increases with the total number of depth maps.

Finally, we store in the Memory some of the sampled points with their occupancy value for future supervision, and update the value of points already stored in the Memory.

Surface coverage gain module. The process to build supervision values for training the surface coverage gain prediction is as follows. Using the data collected at time steps $i \leq t$, we apply the surface coverage gain module to predict the surface coverage gain of camera poses $c'_{t,1}, ..., c'_{t,n-1}$. Then, for each $c'_{t,i}$, we compute a supervision value for the coverage gain by counting the number of new visible surface points appearing in the depth map $d'_{t,i}$. We consider a surface point to be new if its distance to the previous reconstructed point cloud S_t is at least ϵ, where ϵ is a hyperparameter used for coverage evaluation.

We finally update the surface point cloud S_t stored in the Memory. We also store the poses $c'_{t,i}$ and the depth maps $d'_{t,i}$, in order to recompute supervision values for surface coverage gain when sampling from the Memory.

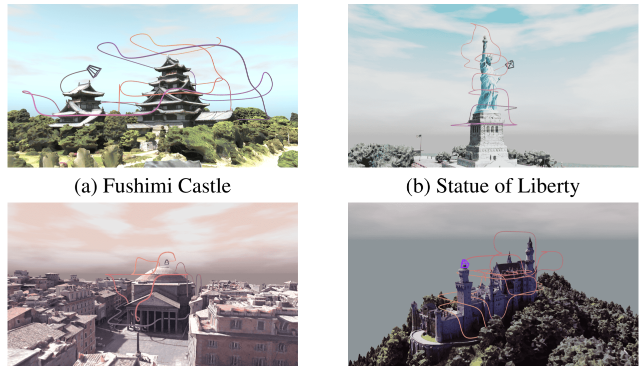
4.4. Memory Replay

During this step, we randomly sample the data stored in the Memory to train each of the modules as follows. We add to these samples the newly acquired data, to make sure the model learns from the current state of the scene. The more memory replay iterations, the faster the model learns and converges but the slower it explores.

Depth module. We follow a standard loss from self-supervised monocular depth prediction literature [23,25,71] to train the depth prediction module from RGB images. The only difference is that in our case, we use multiple input images. We thus predict the depth map for the current image from the m previously captured images. The loss compares these images after warping using the predicted depth map with the same reconstruction loss as [23, 25, 71], which is a combination of SSIM, L1 and an edge-aware smoothness loss. We also include in this loss the next image as suggested in [71], since it greatly improves performance.

Volume occupancy module. We train this module by comparing its predictions, computed from S_t and the previous camera poses, to the updated carved occupancy field computed from the newly acquired data with the MSE loss.

Surface coverage gain module. We rely on a loss different from [27] to train the surface coverage gain, which improves performance and interpretability. [27] showed that its formalism can estimate the surface coverage gain by integrating over the volume occupancy, but only to a scale factor that cannot be computed in closed form. Moreover, their training approach requires to have a dense set of cameras for each forward pass, since they compute the surface coverage gain as a distribution over the whole set of camera poses to compute their loss. They solve this requirement using many renderings of ShapeNet objects, but such a dense set is not available in our online self-supervised setting. Also, their normalization using softmax does not enforce the lowest visibility gain values to be close to zero.



(a) Fushimi Castle (b) Statue of Liberty (c) Pantheon (d) Neuschwanstein Castle

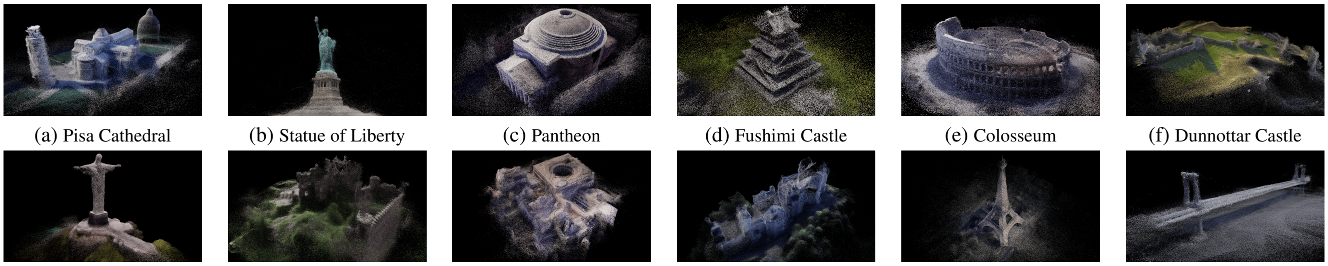
Figure 4. **Trajectories computed in real-time by MACARONS in large 3D structures.** MACARONS predicts a NBV at each time step t to build trajectories that consistently cover most of the surface of the scene. MACARONS performed 100 NBV iterations in these images.

Since the predicted coverage gains are supposed to be proportional to the real values, we propose a much simpler approach that consists in dividing both predicted and supervision coverage gains by their respective means on a potentially small set of cameras. We then compare these normalized coverage gains directly with a L1-norm. This simpler loss also enforces the lowest visibility gain values to be equal to zero. Overall, this loss function applies better constraints on the model to target meaningful visibility gains, and allows for training with less camera poses, which is essential to let our model learn in an online self-supervised fashion where only a few coverage gain values are available at real-time. Additional figures showing the proportionality of predicted and true coverage gains are available in the supplementary material.

5. Experiments

5.1. Implementation

We implemented our model with PyTorch [53] and use 3D data processing tools from PyTorch3D [58], such as ray-casting renderers to generate RGB images as inputs to our model. MACARONS learns online to explore large, un-



(a) Pisa Cathedral (b) Statue of Liberty (c) Pantheon (d) Fushimi Castle (e) Colosseum (f) Dunnottar Castle

(g) Christ the Redeemer (h) Bannerman Castle (i) Alhambra Palace (j) Neuschwanstein Castle (k) Eiffel Tower (l) Manhattan Bridge

Figure 5. **Automated reconstruction of large 3D structures from RGB images with our approach MACARONS.** Our model reconstructs the surface in real time during exploration: We show here the reconstruction after 100 NBV iterations. The model has been trained on a set of previous scenes; all weights are frozen and we only perform inference computation. The first 8 images depict scenes that were already seen by the model during its online, self-supervised training. The last 4 images depict scenes the model has never seen before. MACARONS is not only able to reconstruct surfaces thanks to its depth prediction module (even for scenes it has never seen before), but is also able to optimize its path around the structure and consistently cover most of the surface of the scene thanks to its NBV prediction.

known environments thanks to its self-supervised pipeline that does not need any 3D input data; After being trained long enough, we can either freeze the weights and deactivate online learning to save computation time for future exploration, or let the model continue its training to further finetune it to novel scenes. We perform online training with up to 4 GPUs Nvidia Tesla V100 SXM2 32 Go to let the model explore 4 different scenes in parallel and speed up the convergence, but we used a single GPU Nvidia GeForce GTX 1080 Ti for the inference experiments presented below. In our experimental setup, after each NBV selection step, we perform 5 memory replay iterations for the depth module and up to 3 for the other modules. We provide extensive details in the supplementary material.

5.2. Exploration of large 3D scenes

We compare our method to the state of the art for learning-based NBV computation for dense reconstruction in large environments. All methods use perfect depth maps as input except for MACARONS, which takes RGB images as input. We generate input data from 3D meshes of large scenes (courtesy of Brian Trepanier and Andrea Spognetta, under CC License; all models were downloaded from the website Sketchfab). This dataset was introduced in [27]. To compare the different methods, we follow [27] and compute the area under the curve of the evolution of the total surface coverage during exploration, after 100 NBV iterations, as presented in Table 1. The surface coverage is computed using the ground truth meshes. AUCs are averaged on multiple trajectories in each scene: We use the same starting camera poses and same sets of candidate camera poses C_t for each method for fair comparison. For this experiment, MACARONS was trained on a set of several scenes: The 8 scenes above the bar in Table 1 were seen during online training (but with different, random starting camera poses and trajectories), and the 4 scenes below the bar were not. The other methods were trained on ShapeNet [6] with 3D supervision since their learning process cannot scale to unknown, large environments.

During this experiment, we freeze all weights of MACARONS and only perform inference computation to better demonstrate the ability of our model to generalize to novel scenes, even when online learning is deactivated. Even if it only uses RGB images, our model is able to outperform the baselines in large environments since, contrary to other methods, its self-supervised online training strategy allows it to scale its learning process to any kind of unknown environment, where no ground truth is available and data has to be acquired with a camera. Figures 4 and 5 show examples of trajectories as well as of surface reconstructions computed with MACARONS.

5.3. Ablation study

Apart from adapting learning-based NBV prediction to RGB inputs, we proposed both a novel loss to learn the surface coverage gain compared to [27], and a new online training strategy to let the model learn from any kind of environment in a self-supervised fashion. To quantify the benefits of each of these two improvements, we perform two additional experiments.

First, we use our new loss function to train our volume occupancy module and surface coverage gain module for isolated, single objects with explicit 3D supervision on the ShapeNet dataset [6], similar to [27]. We call the resulting model MACARONS-NBV. We compare with other methods in Table 2 and verify that our new loss does not lower but slightly increases performance compared to the state of the art for the specific case of single object reconstruction.

Then, we reconstruct 3D scenes with both our full model and MACARONS-NBV. For the latter, we use perfect depth maps rather than the depth prediction module. We compare two versions of MACARONS-NBV: One is trained on

3D scene	Random Walk	SCONE-Entropy [27]	SCONE [27]	MACARONS (Ours)
Dunnottar Castle	0.330 ± 0.106	0.381 ± 0.041	0.650 ± 0.093	$\mathbf{0.784} \pm 0.025$
Colosseum	0.312 ± 0.098	0.477 ± 0.023	0.532 ± 0.032	$\mathbf{0.629} \pm 0.011$
Bannerman Castle	0.316 ± 0.106	0.534 ± 0.012	0.552 ± 0.020	$\mathbf{0.716} \pm 0.029$
Pantheon	0.206 ± 0.064	0.338 ± 0.030	0.401 ± 0.030	$\mathbf{0.499} \pm 0.019$
Christ the Redeemer	0.518 ± 0.058	0.836 ± 0.028	0.833 ± 0.037	$\mathbf{0.865} \pm 0.037$
Statue of Liberty	0.296 ± 0.188	0.673 ± 0.025	0.695 ± 0.020	$\mathbf{0.708} \pm 0.014$
Pisa Cathedral	0.327 ± 0.121	0.431 ± 0.024	0.555 ± 0.033	$\mathbf{0.661} \pm 0.008$
Fushimi Castle	0.513 ± 0.091	0.791 ± 0.020	0.806 ± 0.029	$\mathbf{0.842} \pm 0.019$
Alhambra Palace	0.361 ± 0.045	0.425 ± 0.046	0.528 ± 0.030	$\mathbf{0.634} \pm 0.019$
Neuschwanstein Castle	0.396 ± 0.090	0.491 ± 0.039	0.662 ± 0.041	$\mathbf{0.760} \pm 0.020$
Eiffel Tower	0.417 ± 0.074	0.702 ± 0.016	0.753 ± 0.013	$\mathbf{0.789} \pm 0.010$
Manhattan Bridge	0.382 ± 0.092	0.422 ± 0.055	0.745 ± 0.069	$\mathbf{0.825} \pm 0.034$
Average on all scenes	0.364	0.542	0.643	**0.726**

Table 1. **AUCs of surface coverage on large 3D scenes.** All methods use perfect depth maps as input except for MACARONS, which takes RGB images as input. We follow [27] and compute the area under the curve representing the evolution of the total surface coverage during exploration. The 8 scenes above the bar were seen by MACARONS during self-supervised training (but with different, random starting camera poses and trajectories), and the 4 scenes below the bar were not. Other methods are trained on ShapeNet [6] with 3D supervision. Even if it only uses RGB images, our model MACARONS is able to outperform the baselines in large environments since, contrary to other methods, its self-supervised online training strategy allows it to scale its learning process to any kind of environment.

	Categories seen during training								Categories not seen during training								
Method	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel	Bus	Bed	Bookshelf	Bench	Guitar	Motorbike	Skateboard	Pistol	Mean
Random	0.745	0.545	0.542	0.724	0.770	0.589	0.710	0.674	0.609	0.619	0.695	0.795	0.795	0.672	0.768	0.614	0.679
Proximity Count [16]	0.800	0.596	0.591	0.772	0.803	0.629	0.753	0.706	0.646	0.645	0.749	0.829	0.854	0.705	0.828	0.660	0.723
Area Factor [65]	0.797	0.585	0.587	0.751	0.801	0.627	0.725	0.714	0.629	0.631	0.742	0.827	0.852	0.718	0.799	0.660	0.715
NBV-Net [48]	0.778	0.576	0.596	0.743	0.791	0.599	0.693	0.667	0.654	0.628	0.729	0.824	0.834	0.710	0.825	0.645	0.706
PC-NBV [80]	0.799	0.612	**0.612**	**0.782**	0.800	0.640	0.760	0.719	0.667	0.662	0.740	**0.845**	0.849	0.728	0.840	0.672	0.733
SCONE [27]	0.827	0.625	0.591	**0.782**	0.819	**0.662**	0.792	0.734	0.694	0.689	0.746	0.832	**0.860**	0.728	0.845	0.717	0.746
MACARONS-NBV	**0.830**	**0.639**	0.595	0.771	**0.826**	**0.662**	**0.810**	**0.741**	**0.702**	**0.690**	**0.758**	0.829	0.852	**0.734**	**0.848**	**0.732**	**0.751**

Table 2. **AUCs of surface coverage for several NBV selection methods for single object reconstruction, as computed on the ShapeNet test dataset following the protocol of [27, 80].** MACARONS-NBV is trained with 3D supervision on the ShapeNet dataset using the new loss we introduced. Even if our loss is designed for large environments, it still maintains state of the art performance for the specific case of isolated, single object reconstruction.

	MACARONS-NBV		MACARONS
	Loss from [27]	Our loss	
AUC of surface coverage	0.561 ± 0.162	0.635 ± 0.148	$\mathbf{0.719} \pm 0.109$

Table 3. **Contribution of our new loss and self-supervised learning process.** AUCs of surface coverage in large 3D scenes, averaged over multiple trajectories in all 12 scenes of the dataset. Both our new loss and the self-supervised learning process of the full model lead to dramatic increase in performance.

ShapeNet using the loss from [27], the other is trained using our new loss. Table 3 shows how our new loss is crucial to increase performance in large scenes, and how self-supervised learning increases performance even further.

6. Conclusion

Our method can explore large scenes to efficiently reconstruct them using only a color camera. Beyond the potential applications, it shows that it is possible to jointly learn to explore and reconstruct a scene without any 3D input.

We assume the scene to be static, which can be a limitation, however several self-supervised depth prediction models already showed how to be robust to moving objects [71]. Another limitation is that we assume the pose to be known as in previous works on NBV prediction. This is reasonable as the method controls the camera but such control is never perfect. It would be interesting to estimate the camera pose as well. Since we control the camera, we already have a very good initialization, which should considerably help convergence. We also use a very simple path planning policy from our coverage predictions, by evaluating camera poses sampled in the surroundings at each iteration. It would be very interesting to consider longer-term planning, to generate even more efficient trajectories.

Acknowledgements. This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013387 made by GENCI. We thank Elliot Vincent for inspiring discussions and valuable feedback.

References

[1] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised Scale-Consistent Depth and Ego-Motion Learning from Monocular Video. In *Advances in Neural Information Processing Systems*, 2019. 3

[2] Fredrik Bissmarck, Martin Svensson, and Gustav Tolt. Efficient Algorithms for Next Best View Evaluation. In *International Conference on Intelligent Robots and Systems*, 2015. 2

[3] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. CodeSLAM: Learning a Compact, Optimisable Representation for Dense Visual SLAM. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[4] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos. In *American Association for Artificial Intelligence Conference*, 2019. 3

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical report, Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015. 3

[6] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An Information-Rich 3D Model Repository. Technical report, Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 2, 7, 8

[7] Jia-Ren Chang and Yong-Sheng Chen. Pyramid Stereo Matching Network. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[8] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Kenneth Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping. In *Robotics: Science and Systems Conference*, 2015. 2

[9] S. Y. Chen and Youfu Li. Vision Sensor Planning for 3–D Model Acquisition. *IEEE Transactions on Systems, Man, and Cybernetics*, 2005. 2

[10] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-Image Depth Perception in the Wild. In *Advances in Neural Information Processing Systems*, 2016. 3

[11] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-Supervised Learning with Geometric Constraints in Monocular Video: Connecting Flow, Depth, and Camera. In *International Conference on Computer Vision*, 2019. 3

[12] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning Depth with Convolutional Spatial Propagation Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 3

[13] Titus Cieslewski, Elia Kaufmann, and Davide Scaramuzza. Rapid Exploration with Multi-Rotors: A Frontier Selection Method for High Speed Flight. In *International Conference on Intelligent Robots and Systems*, 2017. 2

[14] C. Ian Connolly. The Determination of Next Best Views. In *International Conference on Robotics and Automation*, 1985. 1

[15] Jonathan Daudelin and Mark Campbell. An Adaptable, Probabilistic, Next Best View Algorithm for Reconstruction of Unknown 3D Objects. *RAL*, 2017. 2

[16] Jeffrey Delmerico, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza. A Comparison of Volumetric Information Gain Metrics for Active 3D Object Reconstruction. *Autonomous Robots*, 2018. 8

[17] David Eigen and Rob Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In *International Conference on Computer Vision*, 2015. 3

[18] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems*, 2014. 3

[19] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision*, 2014. 3

[20] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[21] Ravi Garg, Vijay Kumar Bg, and Ian Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In *European Conference on Computer Vision*, 2016. 3

[22] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *Conference on Computer Vision and Pattern Recognition*, 2017. 3

[23] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *Conference on Computer Vision and Pattern Recognition*, 2017. 4, 6

[24] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into Self-Supervised Monocular Depth Estimation. In *International Conference on Computer Vision*, 2019. 3

[25] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into Self-Supervised Monocular Depth Estimation. In *International Conference on Computer Vision*, 2019. 3, 6

[26] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from Videos in the Wild: Unsupervised Monocular Depth Learning from Unknown Cameras. In *International Conference on Computer Vision*, 2019. 3

[27] Antoine Guédon, Pascal Monasse, and Vincent Lepetit. SCONE: Surface Coverage Optimization in Unknown Environments by Volumetric Integration. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[28] Vitor Guizilini, Rares Ambrus, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-Frame Self-Supervised Depth with Transformers. In *Conference on Computer Vision and Pattern Recognition*, 2022. 3

[29] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 4

[30] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. PM-Huber: Patchmatch with Huber Regularization for Stereo Matching. In *International Conference on Computer Vision*, 2013. 4

[31] Benjamin Hepp, Debadeepta Dey, Sudipta N. Sinha, Ashish Kapoor, Neel Joshi, and Otmar Hilliges. Learn-to-Score: Efficient 3D Scene Exploration by Predicting View Utility. In *European Conference on Computer Vision*, 2018. 2

[32] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-View Stereo by Temporal Nonparametric Fusion. In *International Conference on Computer Vision*, 2019. 3

[33] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-To-End Learning of Geometry and Context for Deep Stereo Regression. In *International Conference on Computer Vision*, 2017. 3

[34] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In *European Conference on Computer Vision*, 2020. 3

[35] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A Surface-Based Next-Best-View Approach for Automated 3D Model Completion of Unknown Objects. In *International Conference on Robotics and Automation*, 2011. 2

[36] Simon Kriegel, Christian Rink, Tim Bodenmüller, Alexander Narr, Michael Suppa, and Gerd Hirzinger. Next-Best-Scan Planning for Autonomous 3D Modeling. In *International Conference on Intelligent Robots and Systems*, 2012. 2

[37] Arun CS Kumar, Suchendra M. Bhandarkar, and Mukta Prasad. DepthNet: A Recurrent Neural Network Architecture for Monocular Depth Prediction. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 3

[38] Kiriakos N. Kutulakos and Steven M. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 2000. 5

[39] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. CoMoDA: Continuous Monocular Depth Adaptation Using Past Experiences. In *IEEE Winter Conference on Applications of Computer Vision*, 2021. 3

[40] Tristan Laidlow, Jan Czarnowski, and Stefan Leutenegger. DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM Using Single-View Depth and Gradient Predictions. In *International Conference on Robotics and Automation*, 2019. 3

[41] Inhwan Dennis Lee, Ji Hyun Seo, Young Min Kim, Jonghyun Choi, Soonhung Han, and Byounghyun Yoo. Automatic Pose Generation for Robotic 3D Scanning of Mechanical Parts. *IEEE Transactions on Robotics and Automation*, 2020. 2

[42] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised Monocular Depth Learning in Dynamic Scenes. In *CoRL*, 2020. 3

[43] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the Depths of Moving People by Watching Frozen People. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3

[44] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G. Narasimhan, and Jan Kautz. Neural RGB→D Sensing: Depth and Uncertainty from a Video Camera. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3

[45] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient Deep Learning for Stereo Matching. In *Conference on Computer Vision and Pattern Recognition*, 2016. 3

[46] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent Video Depth Estimation. In *ACM SIGGRAPH*, 2020. 3

[47] Robert Mccraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Monocular Depth Estimation with Self-Supervised Instance Adaptation. In *arXiv Preprint*, 2020. 3

[48] Miguel Mendoza, Juan Irving Vasquez-Gomez, Hind Taud, Luis Enrique Sucar, and Carolina Reta. Supervised Learning of the Next-Best-View for 3D Object Reconstruction. *Pattern Recognition Letters*, 2020. 2, 8

[49] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2

[50] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2013. 2

[51] Richard A. Newcombe and Andrew J. Davison. Live Dense Reconstruction with a Single Moving Camera. In *Conference on Computer Vision and Pattern Recognition*, 2010. 3

[52] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *International Conference on Computer Vision*, 2011. 3

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary Devito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NIPS*. Curran Associates Inc., 2019. 6

[54] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't Forget the Past: Recurrent Depth Estimation from Monocular Video. In *IEEE Robotics and Automation Letters*, 2020. 3

[55] Christian Potthast and Gaurav Sukhatme. A Probabilistic Framework for Next Best View Estimation in a Cluttered

Environment. *Journal of Visual Communication and Image Representation*, 2014. 2

[56] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[57] Anurag Ranjan, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3

[58] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. In *arXiv Preprint*, 2020. 6

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Conference on Medical Image Computing and Computer Assisted Intervention*, 2015. 4

[60] Amit Shaked and Lior Wolf. Improved Stereo Matching with Constant Highway Networks and Reflective Confidence Learning. In *Conference on Computer Vision and Pattern Recognition*, 2017. 3

[61] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-Metric Loss for Self-Supervised Learning of Depth and Egomotion. In *European Conference on Computer Vision*, 2020. 3

[62] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-Time Dense Monocular Slam with Learned Depth Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2017. 3

[63] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled Semantics for Comprehensive Scene Understanding from Videos. In *Conference on Computer Vision and Pattern Recognition*, 2020. 3

[64] Juan Vasquez-Gomez, Luis Sucar, and Rafael Murrieta-Cid. View/State Planning for Three-Dimensional Object Reconstruction Under Uncertainty. *Autonomous Robots*, 2017. 2

[65] Juan Vasquez-Gomez, Luis Sucar, Rafael Murrieta-Cid, and Efrain Lopez-Damian. Volumetric Next-Best-View Planning for 3D Object Reconstruction with Positioning Error. *IJARS*, 2014. 2, 8

[66] Juan Vasquez-Gomez, David Troncoso Romero, Israel Becerra, Luis Sucar, and Rafael Murrieta-Cid. Next-Best-View Regression Using a 3D Convolutional Neural Network. In *Machine Vision and Applications*, 2021. 2

[67] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 2017. 4

[68] Chaoqun Wang, Han Ma, Weinan Chen, Li Liu, and Max Meng. Efficient Autonomous Exploration with Incrementally Built Topological Map in 3D Environments. *TIM*, 2020. 2

[69] Jianrong Wang, Ge Zhang, Zhenyu Wu, XueWei Li, and Li Liu. Self-Supervised Joint Learning Framework of Depth Estimation via Implicit Cues. In *arXiv Preprint*, 2020. 3

[70] Rui Wang, Stephen M. Pizer, and Jan-Michael Frahm. Recurrent Neural Network for (un-)supervised Learning of Monocular Video Visual Odometry and Depth. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3

[71] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel J. Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Conference on Computer Vision and Pattern Recognition*, 2021. 3, 4, 6, 8

[72] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. MonoRec: Semi-Supervised Dense Reconstruction in Dynamic Environments from a Single Moving Camera. In *Conference on Computer Vision and Pattern Recognition*, 2021. 3

[73] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Spatial Correspondence with Generative Adversarial Network: Learning Depth from Monocular Videos. In *International Conference on Computer Vision*, 2019. 3

[74] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks. In *European Conference on Computer Vision*, 2016. 3

[75] Jiaxin Xie, Chenyang Lei, Zhuwen Li, Li Erran Li, and Qifeng Chen. Video Depth Estimation by Fusing Flow-to-Depth Proposals. In *International Conference on Intelligent Robots and Systems*, 2020. 3

[76] Luwei Yang, Feitong Tan, Ao Li, Zhaopeng Cui, Yasutaka Furukawa, and Ping Tan. Polarimetric Dense Monocular SLAM. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[77] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to Recover 3D Scene Shape from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2

[78] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[79] Jure ŽBontar and Yann LeCun. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *Journal of Machine Learning Research*, 2016. 3

[80] Rui Zeng, Wang Zhao, and Yong-Jin Liu. PC-NBV: A Point Cloud Based Deep Network for Efficient Next Best View Planning. In *International Conference on Intelligent Robots and Systems*, 2020. 2, 8

[81] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided Aggregation Net for End-To-End Stereo Matching. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3

[82] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-Invariant Stereo Matching Networks. In *European Conference on Computer Vision*, 2020. 3

[83] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting Temporal Consistency for Real-Time Video Depth Estimation. In *International Conference on Computer Vision*, 2019. 3

[84] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. MonoViT: Self-Supervised Monocular Depth Estimation with a Vision Transformer. In *International Conference on 3D Vision*, 2022. 3

[85] Tinghui Zhou, Matthew Brown, Noah Snavely, and David Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In *Conference on Computer Vision and Pattern Recognition*, 2017. 3