Semantically-enhanced Deep Collision Prediction for Autonomous Navigation using Aerial Robots

Mihir Kulkarni, Huan Nguyen, and Kostas Alexis

***Abstract*— This paper contributes a novel and modularized learning-based method for aerial robots navigating cluttered environments containing hard-to-perceive thin obstacles without assuming access to a map or the full pose estimation of the robot. The proposed solution builds upon a semantically-enhanced Variational Autoencoder that is trained with both real-world and simulated depth images to compress the input data, while preserving semantically-labeled thin obstacles and handling invalid pixels in the depth sensor's output. This compressed representation, in addition to the robot's partial state involving its linear/angular velocities and its attitude are then utilized to train an uncertainty-aware 3D Collision Prediction Network in simulation to predict collision scores for candidate action sequences in a predefined motion primitives library. A set of simulation and experimental studies in cluttered environments with various sizes and types of obstacles, including multiple hard-to-perceive thin objects, were conducted to evaluate the performance of the proposed method and compare against an end-to-end trained baseline. The results demonstrate the benefits of the proposed semantically-enhanced deep collision prediction for learning-based autonomous navigation.**

I. Introduction

The rapid progress in the field of aerial robotics has enabled their successful utilization in diverse applications including search-and-rescue [1,2], inspection [3,4], and forest monitoring [5,6]. Keys to this success have been techniques to plan paths, often via sampling-based methods and motion primitives [7]–[9], within dense 3D maps [10]–[13]. Yet, despite the progress, most approaches are limited by a) the need for consistent localization and mapping, b) the latency such systems introduce, as well as c) the maximum resolution of the map used for collision checking and the computational burden of increasing such resolution in order for narrow cross-section (thin) obstacles (e.g., wires, railings, rods, tree branches) to be represented. Recently, we have seen great advances coming from the field of deep learning for collision-free flight tailored to a host of demanding tasks including drone racing [14]. Yet, most such methods also assume consistent pose estimation [15], and often rely fully on simulated data for training [16] which, however, may fail to capture significant imperfections of range sensing such as holes in depth maps [17,18], while there is a lack of focus on methods that explicitly tackle avoiding thin, hard-to-perceive, obstacles.

Motivated by the above, this work contributes a modularized learning-based navigation method that a) is capable of navigating cluttered environments involving hard-to-perceive thin obstacles, while b) not requiring access to a (online or offline) map of the environment or the estimation of the full pose of the robot but instead relies solely on real-time depth observations and a partial estimate of the robot's linear/angular velocities and attitude. While the community has considered the explicit detection of certain thin obstacles (e.g., wires are studied in [19,20] using RGB images), the challenge of safe navigation in cluttered scenes with complex topologies and morphologies of thin obstacles is not solved. Furthermore, for such structures, acquiring accurate depth maps using real sensors is particularly difficult due to their narrow cross-sections. To overcome this limitation and considering the use of depth cameras, we first create and aggregate both real and simulated datasets of depth frames with pixel-wise labels on thin obstacles. Then, we contribute a semantically-enhanced Variational Autoencoder (seVAE) that a) learns to compress, encode and reconstruct depth maps while preserving thin obstacle features, b) utilizes both real and simulated depth data for training along with pixel-wise labels reflecting thin obstacles (when available), and c) can work directly with raw depth frames while remaining robust to complex cases of sensor noise and errors including holes in the depth maps. Through the encoder of the seVAE, the method arrives –at inference time– to a low-dimensionality latent space that encodes the information for collision avoidance including against thin obstacles. This latent space is then combined

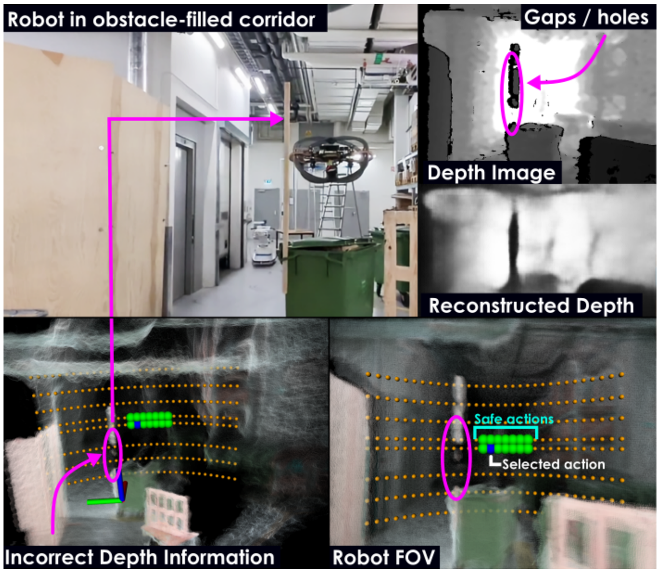

Fig. 1. Instance of an experiment demonstrating collision-free flight in a cluttered environment further involving thin obstacles (e.g., vertical wooden sticks) with the robot only exploiting real-time depth maps and estimates of its linear and angular velocities, as well as attitude.

This material was supported by a) the AFOSR Award No. FA8655-21-1-7033 and b) the Research Council of Norway Award NO-321435.

The authors are with the Autonomous Robots Lab, Norwegian University of Science and Technology (NTNU), O. S. Bragstads Plass 2D, 7034, Trondheim, Norway mihir.kulkarni@ntnu.no

with a Collision Prediction Network (CPN) building upon our previous work called "ORACLE" [16]. The CPN utilizes the compressed latent space of the encoder combined with the robot's partial state (considering its uncertainty) and provides collision scores for candidate 3D action sequences from a motion library thus enabling safe autonomous flight.

To validate the proposed approach, a set of evaluation studies, both in simulation and experimentally, were conducted. In simulation, the method was tested in environments with increasing density of obstacles and its results are compared against an end-to-end trained 3D extension of ORACLE (using simulated depth images only). Two real-world experiments were also performed to test the new method in densely cluttered environments with various sizes and types of obstacles. Similarly, the performance of the new method in these experiments is compared against that of the end-to-end trained ORACLE demonstrating the value added by the ability to incorporate real data in the training phase to enhance robustness against sensor imperfections.

In the remainder of this paper, Section II presents related work, followed by the proposed modularized learning-based navigation solution in Section III. Evaluation studies are detailed in Section IV, and conclusions in Section V.

II. Related Work

A set of contributions in learning-based navigation relate to this work, alongside efforts to enable avoidance of thin obstacles. Many works have employed deep learning techniques to tackle the issue of autonomous robot navigation, with a particular focus on using RGB/depth cameras. These sensors have garnered increased attention due to their low cost, low power consumption, and lightweight design. The authors in [15,21] use imitation learning to generate collision-free smooth trajectories from realistic simulated data. Though much effort has been devoted to obtaining more realistic simulation images [22,23], there persist effects that are difficult to model such as missing depth pixels when observing thin obstacles or low-texture, shiny surfaces. In [24], reactive navigation policies are learned through reinforcement learning and domain randomization using RGB image inputs. On the other hand, [25]–[27] predict probabilities of events such as collisions, going over bumpy terrain, or human disengagement from real-world RGB data. Our approach also leverages real depth data in the training pipeline, however, part of the depth frames and the data collection where the robot dynamics are rolled out and collision events happen is performed in simulation. The authors in [28] propose another method to mix real and simulated depth images for learning an environment representation for legged robot navigation. In this work, we focus on using both the real and simulated depth images augmented with the labeled semantic masks to ensure the reconstruction of hard-to-perceive obstacles such as thin objects in the environment representation.

Of further relevance to this work are also contributions proposing solutions to allow robots to detect and avoid thin obstacles in the environments. The authors in [29] present a thin-structure obstacle detection and 3D reconstruction method based on an edge detector and edge-based visual odometry techniques. On the other hand, a multi-view algorithm for wire reconstruction using a parametric catenary model is illustrated in [30]. The work in [31] proposes a planning approach using mixed integer programming to derive collision-free trajectories for a quadrotor to fly in environments populated with thin strings, however, the locations of the obstacles are given beforehand as convex hulls. The authors in [32] use a monocular wire detection method based on synthetic data and a dilated convolutional neural network (CNN) [19] to perceive thin-wire obstacles. The obstacles are then represented by a disparity-space representation for collision checking. A separate planning module evaluates a pre-computed trajectory library to choose and execute the best trajectory.

III. Proposed Approach

Our proposed modularized solution for learning-based navigation builds upon a Variational Autoencoder (VAE) [33,34] module that can be independently trained using both real and simulated data to produce a compressed latent representation of a raw depth image that is then exploited for collision prediction. Semantic labels can also be added to the images –when available– to ensure reconstruction of hard-to-perceive thin obstacles (e.g., skinny rods and wires [20]). This representation is able to reconstruct complex images without missing out on narrow cross-section (thin) obstacles, while being robust to sensor noise that arises from the sensor's inability to reconstruct depth for textureless or reflective regions and stereo shadows. This latent representation is then used to train a 3D Collision Prediction Network (CPN) in simulation to predict collision scores for candidate action sequences based on a library of motion primitives. Building upon our prior work called ORACLE [16], the method does not require the full robot state or an environment map but only real-time depth frames and access to an estimate of the partial robot state involving the system's attitude, as well as angular and linear velocities. Simultaneously, it takes into account the uncertainty associated with this partial state when predicting collision scores. Given the above and a goal vector provided by a high-level planner [35,36] or by an operator, the optimal collision-free action sequence is selected and executed by the robot in a receding horizon manner.

A. Semantically-enhanced Variational Autoencoder

VAEs are powerful tools that can allow encoding high-dimensional input data in a compressed representation. In this work, we show that a complex depth image with a typical resolution (of 270×480 pixels) can be sufficiently represented by a highly compressed latent representation (here 128 variables) while preserving features from hard-to-perceive thin obstacles. This latent representation can subsequently be used for predicting collisions given a set of candidate action sequences. We develop an approach to combine data from both simulated and real sensor observations, allowing our method to be a) robust to sensor noise, and at the same time b) able to utilize additional information such as instance semantic labels –primarily from simulators– thus being capable of reconstructing thin objects. In contrast, non-modularized methods that train a collision predictor end-to-end such as [16] may not allow to explicitly promote a focus on such semantically-driven thin obstacles or conveniently utilize real sensor data and their imperfections.

Representing a compressed latent space starting from high-dimensional depth data using hand-tuned parameters is difficult, given that the distribution generating the dataset of depth images is practically intractable to compute. Thus, we utilize a VAE to learn a latent representation that can be used for image reconstruction. We consider a dataset $\mathbf{X} = \{\mathbf{x}^{(i)}, \mathbf{x}^{(i)}_{val}, \mathbf{x}^{(i)}_{seg}\}_{i=1}^{N}$ consisting of samples of discrete images $\mathbf{x}^{(i)}$ of dimensions $H \times W$ (here 270×480 pixels), the mask of valid pixels from the input data $\mathbf{x}^{(i)}_{val}$, and the instance segmentation mask for each thin obstacle instance $\mathbf{x}^{(i)}_{seg}$ (when available). The pixels with defined depth information from the sensor are referred to as valid pixels and the pixels with missing depth values are referred to as invalid pixels. We assume that this dataset is generated by some random process involving an unobserved random variable $\mathbf{z}$ with J dimensions (here $J = 128$). We model this variable $\mathbf{z}$ as the latent code for our representation. We assume that a depth image $\mathbf{x}$ is generated from $\mathbf{z}$ by some generative model. A probabilistic decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ produces a distribution over the possible values of $\mathbf{x}$. Since the true value of the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is unknown, we assume an approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, to be a multivariate Gaussian with a diagonal covariance as:

$$\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I}), \qquad (1)$$

where the mean and the standard deviation of the approximate posterior, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, are outputs of the encoding neural network for the image $\mathbf{x}$ with parameters $\boldsymbol{\phi}$. In this realization, we use the reparameterization trick to obtain the values of $\boldsymbol{\mu}^{(i)}$ and $\log(\boldsymbol{\sigma}^{2(i)})$ from the network [33]. We then sample $\mathbf{z}^{(i)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$ using $\mathbf{z}^{(i)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}$, ($\odot$ is the element-wise multiplication operator), with an auxiliary noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A suitable encoder for this architecture is designed by modifying [37] to reduce the number of parameters in the neural network, while enhancing connectivity in the fully-connected layer. The decoder consists of deconvolutional layers stacked with activation functions without skip connections. The network architecture used in this work is depicted in Figure 2, where its fine tuning was guided by the observation that increasing the size of the fully connected hidden layers tended to enhance the quality of reconstruction from the network. The loss term is defined as:

$$\mathcal{L} = \mathcal{L}_{recon} + \beta_{norm}\mathcal{L}_{KL}, \qquad (2)$$

where $\mathcal{L}_{KL}$ denotes the KL-divergence loss defined as:

$$\mathcal{L}_{KL}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\boldsymbol{\sigma}_j)^2) - (\boldsymbol{\mu}_j)^2 - (\boldsymbol{\sigma}_j)^2\right). \qquad (3)$$

The constant $\beta_{norm} = \frac{\beta \cdot J}{H \cdot W}$ weighing the KL-divergence loss is normalized accounting for the latent space and image dimensions. The value of β (here $\beta = 1$) allows the mixing of the contributions from the KL-divergence loss and the reconstruction loss and is a tunable hyperparameter [38]. The depth cameras used to collect the datasets consist of a stereo pair and are unable to reconstruct depth from textureless features and miss the depth data around the edges of features resulting in a "stereo shadow" region. We design a semantic-weighted reconstruction loss function $\mathcal{L}_{recon}$ for training the neural network taking into consideration these errors from such sensors by ignoring the contribution of the invalid pixels in the loss function. The semantic-weighted reconstruction loss is defined as:

$$\mathcal{L}_{recon}(\mathbf{x}, \mathbf{x}_{recon}, \mathbf{x}_{val}, \mathbf{x}_{seg}) = \sum_{h=1}^{H}\sum_{w=1}^{W}\left((\mathbf{x} - \mathbf{x}_{recon})^2 \odot \mathbf{x}_{val} \odot \lambda(\mathbf{x}_{seg})\right)_{(h,w)}, \qquad (4)$$

where $\mathbf{x}_{recon}$ is the reconstructed image and $\{\mathbf{x}, \mathbf{x}_{val}, \mathbf{x}_{seg}\}$ represents one sample from the dataset $\mathbf{X}$. The use of $\mathbf{x}_{val}$ during seVAE training eliminates the contribution of the invalid input pixels from the loss function, allowing the network to represent distributions consisting of information in the valid pixels. The function $\lambda(\mathbf{x}_{seg})$ creates a pixel-wise weight mask to give higher weight to the pixels that correspond to disproportionately thin obstacles using the semantic label information $\mathbf{x}_{seg}$ (if available). The weight ν_ℓ of a pixel ℓ belonging to a semantic instance $S_k \in \mathbb{S}$, where $\mathbb{S}$ is the set of all instances of semantics in $\mathbf{x}_{seg}$, depends on the pixel count p_k of instance S_k. The weight of the pixels not belonging to a semantic instance is set to 1. Formally, the weight ν_ℓ is defined as:

$$\nu_\ell = \begin{cases} \max(W_{const}/p_k, \nu_{\min}), \ \ell \in S_k \text{ and } p_k > p_{\min} \\ 1, \text{ otherwise} \end{cases}, \qquad (5)$$

where the term W_{const} (here $W_{const} = 6000$) acts as a multiplicative constant to weigh the inverse count of pixels per semantic, while $\nu_{\min}$ (here $\nu_{\min} = 15$) limits the minimum per-pixel weight. This weighing term is applied to magnify the contribution of small-sized semantics while allowing larger-sized semantics to be proportionally weighed based on the number of pixels. We ignore semantics smaller than $p_{\min}$ (here $p_{\min} = 40$ pixels) to prevent the seVAE from trying to reconstruct extremely small regions, that may, in the real-world correspond to sensor noise and data imperfections. Given the loss $\mathcal{L}$ and the semantic information, the seVAE is trained to encode a depth image into the low-dimensional latent representation $\boldsymbol{\mu}$ to be used for collision prediction.

B. Collision Prediction Network and Action Planning

The latent space from the seVAE is then exploited by the CPN to facilitate safe navigation building upon our previous work ORACLE [16] and extending it to 3D navigation, while retaining the benefits of the modularized seVAE-based latent space including its ability to assimilate real sensor data. To this end, the CNN part of the original CPN in ORACLE is replaced by the seVAE which takes the current depth image $\mathbf{x}_t$ and outputs the compressed latent vector $\boldsymbol{\mu}_t$, that is then fed to the CPN as illustrated in Figure 3.

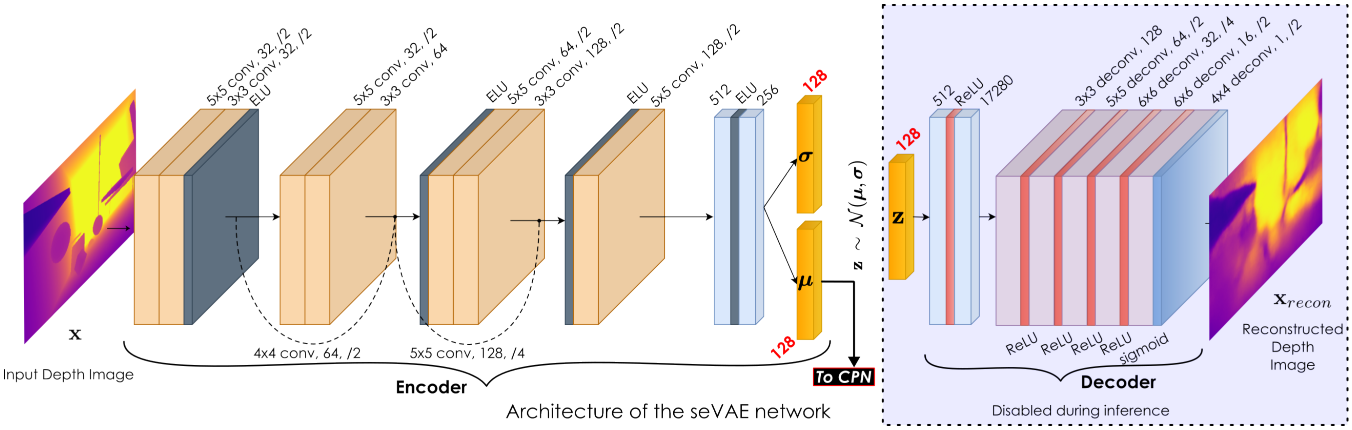

Fig. 2. Proposed architecture for the seVAE for collision prediction. The convolutional and deconvolutional layers' hyperparameters are represented in the format ($\mathfrak{a} \times \mathfrak{b}$ (de)conv, $\mathfrak{c}$, $/\mathfrak{d}$), where $\mathfrak{a} \times \mathfrak{b}$ refers to the kernel size, $\mathfrak{c}$ refers to the number of channels, and $\mathfrak{d}$ refers to the stride length. The dense layers only have the layer size mentioned alongside. A latent space $\mathbf{z}$ of 128 dimensions is sampled from the distribution with mean $\boldsymbol{\mu}$ and std. dev. $\boldsymbol{\sigma}$ and used to reconstruct the image by the decoder.

Specifically, let $\mathcal{B}, \mathcal{V}$ be the body frame and vehicle frame of the robot respectively, and $\mathbf{s}_t = [{\mathbf{v}_t}^T, \omega_t, \vartheta_t, \varphi_t]^T$ the estimated partial state of the robot at time t consisting of a) the 3D velocity in $\mathcal{V}$ ($\mathbf{v}_t = [v_{t,x}, v_{t,y}, v_{t,z}]^T \in \mathbb{R}^{3\times1}$), b) the angular velocity around the z-axis of the body-frame $\mathcal{B}$ (ω_t), as well as c) the roll (ϑ_t) and pitch angles (φ_t). Let $\boldsymbol{\Sigma}_t$ denote the covariance of the estimated robot's partial state, $\mathbf{n}_t^g$ the 3D unit goal vector, expressed in $\mathcal{V}$, that the robot has to follow, ψ_t the current yaw angle of the robot, and $\mathbf{a}_{t:t+T} = [\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+T-1}]$ an action sequence having length T where the action at time step $t+i$ ($i = 0, \ldots, T-1$) includes a) the reference 3D velocity expressed in the vehicle frame $\mathbf{v}_{t+i}^r$ and b) the steering angle (δ_{t+i}^r) from the current yaw angle of the robot (ψ_t), such that $\mathbf{a}_{t+i} = [{\mathbf{v}_{t+i}^r}^T, \delta_{t+i}^r]^T$. The method finds an optimized collision-free sequence of actions $\mathbf{a}_{t:t+T}$ given $(\mathbf{x}_t, \mathbf{s}_t, \boldsymbol{\Sigma}_t)$ (specifically using the latent vector $\boldsymbol{\mu}_t$ derived from $\mathbf{x}_t$ using the seVAE's encoder) enabling the robot to safely navigate to the goal vector $\mathbf{n}_t^g$.

The CPN processes $\boldsymbol{\mu}_t$, $\mathbf{s}_t$, and a Motion Primitives-based Library (MPL) of N_{MP} sequences of future velocity and steering angle references $\mathbf{a}_{t:t+T}$ to predict the collision scores $\hat{\mathbf{c}}_{t+1:t+T+1}^{col} = [\hat{c}_{t+1}^{col}, \hat{c}_{t+2}^{col}, \ldots, \hat{c}_{t+T}^{col}]$ of the robot at each time step from $t+1$ to $t+T$ in the future for each action sequence. To account for the uncertainty in the robot's partial state, we first calculate $N_\Sigma = 2\gamma + 1$ sigma points ($\mathbf{m}_1 = \mathbf{s}_t$, …, $\mathbf{m}_{N_\Sigma}$), where γ is the dimension of the robot's partial state (here $\gamma = 6$), based on the mean $\mathbf{s}_t$ and covariance $\boldsymbol{\Sigma}_t$ using the Unscented Transform (UT) [39]. Using the sigma points, the uncertainty-aware collision score $\hat{c}^{uac}$ is calculated as presented in [16]. A set of safe action sequences is derived by thresholding $\hat{c}^{uac}$ and the action sequence that leads to the end velocity of the robot best aligned with $\mathbf{n}_t^g$ is chosen. The first action in the sequence is executed, while the process is repeated in a receding horizon fashion. Notably, in ORACLE, the CPN is trained end-to-end entirely in simulation and requires a computationally-expensive pre-processing step in the depth image input to close the sim-to-real gap. In this work, the CPN is trained with the frozen weights of the VAE in simulation environments which have large obstacles with randomized shapes and thin obstacles with diameters less than 5 cm, as illustrated in Figure 4.3.

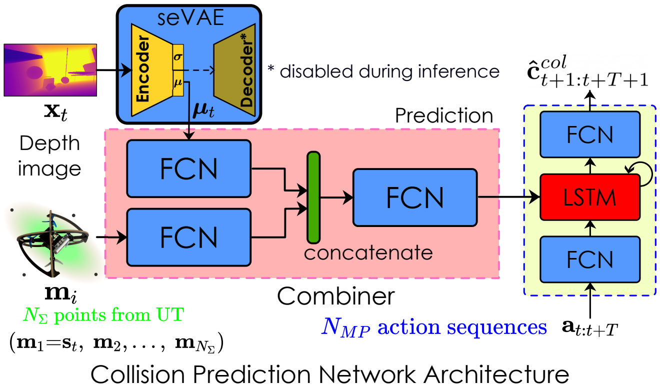
seVAE
Encoder
Decoder*
σ
μ
* disabled during inference
$\mathbf{x}_t$
Depth image
$\boldsymbol{\mu}_t$
$\hat{\mathbf{c}}_{t+1:t+T+1}^{col}$
Prediction
FCN
FCN
FCN
FCN
LSTM
FCN
concatenate
$\mathbf{m}_i$
N_Σ points from UT
($\mathbf{m}_1$=$\mathbf{s}_t$, $\mathbf{m}_2, \ldots, \mathbf{m}_{N_\Sigma}$)
Combiner
N_{MP} action sequences
$\mathbf{a}_{t:t+T}$
Collision Prediction Network Architecture

Fig. 3. Proposed architecture for the modularized approach involving the seVAE and the Collision Prediction Network (CPN). The seVAE encodes the input depth image $\mathbf{x}_t$ into the latent representation $\boldsymbol{\mu}_t$ which is used by the CPN to predict the collision scores $\hat{\mathbf{c}}_{t+1:t+T+1}^{col}$ for each action sequence $\mathbf{a}_{t:t+T}$ in the motion primitives library. Furthermore, the method utilizes the N_Σ sigma points calculated based on $\mathbf{s}_t$ and $\boldsymbol{\Sigma}_t$ to calculate the robot's partial state uncertainty-aware collision score $\hat{c}^{uac}$.

IV. Evaluation Studies

The proposed method, and its submodules, were extensively evaluated as presented below.

A. Training Methodology

A composite dataset of both real and simulated images was collected for training the VAE network. Real data was collected with an Intel RealSense D455 depth camera. This dataset consists of images in confined spaces, indoor rooms, long corridors, and outdoor environments with trees. Thin obstacles such as tree branches, rods, and poles are manually labeled in this dataset. We also utilize images from the NYU Depth Dataset v2 [40]. Simulation images are collected using both Gazebo Classic [41] and Isaac Gym Simulators [42]. Isaac Gym offers a segmentation camera providing instance segmentation masks for the corresponding depth images. Simulated meshes having cross sections below a size of 5 cm are assigned semantic instance IDs. The segmentation camera allows rendering images with pixel values equal to the ID of the semantic occupying that pixel. The aggregated dataset

$\mathbf{X}$ consists of $\sim 66,000$ images, of which $\sim 35,000$ are simulated. $21,000$ simulated images have semantic labels, and $\sim 1,000$ real images are labeled. The training and validation sets are split with a $80\% : 20\%$ ratio from this dataset. We train the network using the Adam Optimizer [43] with a learning rate of 10^{-4} for 40 epochs.

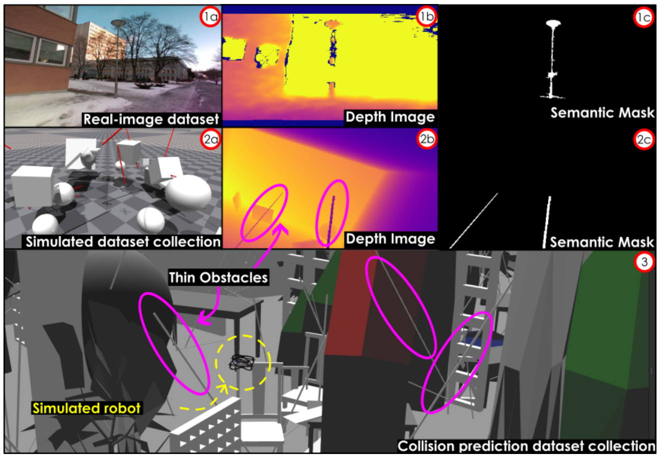

Fig. 4. Instances of depth images (1b, 2b) collected for the seVAE training from the real world (1a) and the IsaacGym (2a) simulator are shown. RGB images (1a) from a collected dataset (if available) are used to create semantic labels (1c) for the aligned depth images. The objects beyond the sensing range of the depth camera are not labeled. The IsaacGym simulator provides semantic labels for thin obstacles (2c). Simulation data for training the CPN is collected in Gazebo Classic (3) with a simulated drone in an obstacle-filled environment. Randomized action sequences are executed to collect a collision dataset.

Collision datasets for training the CPN are collected using the Gazebo-based RotorS simulator [44] with a simulated robot model. An environment consisting of obstacles having different shapes and sizes is constructed and thin obstacles with cross sections smaller than 5 cm are introduced as shown in Figure 4. Randomized action sequences $\mathbf{a}_{t:t+T}$ within the robot's FOV are generated and executed until a collision or a timeout occurs. Datapoints consisting of $d = (\mathbf{x}_t, \mathbf{s}_t, \mathbf{a}_{t:t+T}, \hat{\mathbf{c}}^{col}_{t+1:t+T+1})$ are collected. The data is also augmented by performing a horizontal flip and appending random actions at the end of the collision episodes, similar to [16]. The depth images $\mathbf{x}_t$ from this are passed through the learned seVAE model to obtain its latent representation $\boldsymbol{\mu}_t$. Finally, datapoints $d' = (\boldsymbol{\mu}_t, \mathbf{s}_t, \mathbf{a}_{t:t+T}, \hat{\mathbf{c}}^{col}_{t+1:t+T+1})$ are created replacing the depth image $\mathbf{x}_t$ with its corresponding latent representation $\boldsymbol{\mu}_t$ to train the CPN.

B. Comparison of reconstruction methods for thin obstacles

First, the proposed semantically-enhanced VAE is evaluated by comparing it against a baseline compression based on the Fast Fourier Transform (FFT) and a vanilla-VAE trained without the weighing function $\lambda(\mathbf{x}_{seg})$. To allow for a fair comparison with the 128 dimensional latent variable $\boldsymbol{\mu}$ of the VAEs, the FFT reconstructions are generated from the 64 complex frequency-domain representations having the largest magnitudes while zeroing out the others. Reconstructions from the compared methods are depicted in Figure 5. The FFT reconstruction is unable to preserve information at this resolution especially for the features corresponding to smaller-sized obstacles. Additionally, the presence of sensor noise in real images degrades the performance of the FFT even further. The vanilla-VAE is able to reconstruct larger parts of the images well but misses out on the regions with small cross-section (thin) obstacles for both real and simulated images. The seVAE is able to reconstruct regions with smaller cross sections better than the above approaches. The key role of semantically-augmented training is particularly visible in Figure 5. We statistically compare the performance of the three approaches to consider the MSE over the whole image and also specifically over only the semantically labeled pixels (thin obstacles) of an image and present the results in Table I. For comparison, we normalize the valid pixel values between 0 and 1. As shown, seVAE presents a significantly better performance in reconstructing the semantic regions, with a relatively small reduction in the performance over the whole image against the vanilla-VAE using the same number of parameters, while also outperforming FFT-based compression. It is noted that relatively inferior reconstruction quality is less significant for large objects, as long as collision-avoidance is concerned, but it is critical not to miss thin obstacles.

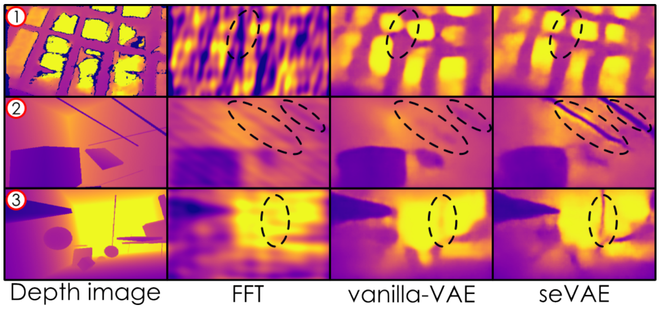

Fig. 5. The proposed method is compared with Fast Fourier Transform (FFT) and vanilla-VAE without semantic weighted loss on real (1) and simulated images (2,3). The FFT reconstruction is created from the largest 64 magnitudes in the frequency domain representation, while the vanilla-VAE and the semantically-enhanced VAE use a 128 dimensional latent vector. Ellipsoids are drawn around expected reconstructions of thin obstacles from the input depth images.

TABLE I

COMPARISON OF MSE FOR RECONSTRUCTED IMAGES WITH DIFFERENT METHODS (SEVAE, VANILLA-VAE, FFT-BASED).

Simulated Images (**Count:** 2239)			
MSE over:	**FFT**	**vanilla-VAE**	**seVAE**
Entire image	553.21	**276.64**	404.50
Semantic pixels	82.14	89.51	**22.13**
Real Images (**Count:** 365)			
MSE over:	**FFT**	**vanilla-VAE**	**seVAE**
Entire image	2804.28	**350.01**	420.35
Semantic pixels	156.95	69.16	**28.41**

C. Simulation Studies

Subsequently, the proposed method is evaluated in complex simulation studies involving diverse obstacle configurations including distributed thin objects such as rods. We design three simulation worlds consisting of three sections each, with different kinds of obstacles placed using Poisson disc sampling. The first section contains exclusively large-sized obstacles with Poisson disc sampling radius r_1, the second section contains both large sized, and thin obstacles with

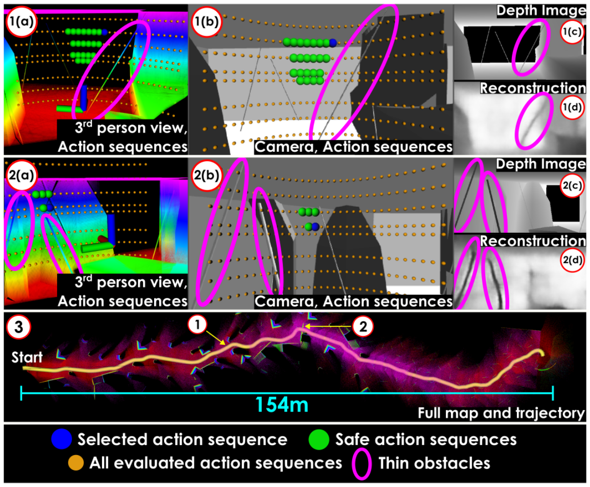

Fig. 6. Results from one of the simulation missions conducted in a Gazebo environment with diverse obstacle configurations including thin obstacles (cross section 4 cm). The sub-figures 1 and 2 show two instances during the mission detailing a) the point cloud, b) the onboard camera image both overlaid with markers representing estimated endpoints of collision-free action sequences, c) the raw depth image, and d) reconstructed depth image from seVAE. Sub-figure 3 shows the final map and the trajectory followed by the robot. The simulation shows the ability of the seVAE to reconstruct thin and regular objects well, and the overall method to avoid obstacles.

cross section 4 cm, sampled independently of each other with radii r_2 and r_3 respectively, and the third section consists only of thin rods with a similar cross section sampled with radius r_4. Each section is 50 m × 50 m and placed serially. The robot is commanded to travel 150 m along the course at a speed of 1.0 m/s. To derive the UT sigma points, we use $\mathbf{\Sigma}_t = \mathrm{diag}({\sigma_v}^2, {\sigma_v}^2, {\sigma_v}^2, 0, 0, 0)$, with $\sigma_v = 0.2$ m/s. We perform 20 runs of the experiment per environment for both the proposed method (trained with datapoints d') and a 3D extension of our end-to-end ORACLE method [16] (trained with datapoints d) with different initial positions and orientations. A few instances and the full path from one of the missions of the proposed method are shown in Figure 6 and the statistical results are logged in Table II. The environment sampling variables are listed beside each environment in the format (r_1, r_2, r_3, r_4). It is noted that the ensemble of neural networks is not used for ORACLE to have a fair comparison. The proposed method outperforms ORACLE with a higher success rate in terms of completing the course without collision, in denser environments.

TABLE II

COMPARISON BETWEEN SUCCESS RATES OF END-TO-END VS PROPOSED METHOD IN SIMULATION.

Environment (r_1, r_2, r_3, r_4)	**Method**	**Success %**
Sparse (6.5, 6.5, 4.5, 3.5)	Proposed method End-to-end ORACLE	95% 95%
Medium (6.25, 6.25, 3.5, 3.0)	Proposed method End-to-end ORACLE	**80%** 75%
Dense (6.0, 6.0, 3.0, 2.5)	Proposed method End-to-end ORACLE	**60%** 45%

D. Experimental Evaluation

Beyond simulations, the proposed method was evaluated in two real-world confined environments further involving thin obstacles using a collision-tolerant aerial robot design similar to [45] which integrates a sensor suite including an Intel Realsense D455 RGB-D sensor, an Intel Realsense T265 visual-inertial module, an mRo Pixracer flight controller, and an NVIDIA Xavier NX board. To derive the UT sigma points, we use $\mathbf{\Sigma}_t = \mathrm{diag}({\sigma_v}^2, {\sigma_v}^2, {\sigma_v}^2, 0, 0, 0)$, with $\sigma_v = 0.2$ m/s. One experiment is performed (with an average robot speed of 1.0 m/s) in a long cluttered corridor with thin obstacles (as small as 4.8 cm in width) obstructing the path of the robot. The goal point is defined 50 m from the starting point along the corridor and expressed as a goal direction for the navigation method exploiting only a partial state without position information. A second experiment is conducted (with an average robot speed of 0.75 m/s) in a smaller environment with challenging hard-to-perceive obstacles with small cross-sections (as small as 2.5 cm in width), and texture-less reflective surfaces. The size and the material of the obstacles in these experiments pose challenges for accurate depth reconstruction using the depth camera onboard the robot. The resolution of the RealSense D455 is set to 480 × 640 pixels and the depth image is downsampled to the input resolution of the encoder (270 × 480 pixels) before encoding. The inference times of seVAE and CPN on the platform are 11 ms and 29 ms respectively. The results of these experiments are shown in Figure 7.

Finally, the proposed approach is also experimentally compared against the end-to-end trained ORACLE as in Section IV-C. Specifically, the collision prediction step of ORACLE was ran for the depth frames of the above experiments. Notably, originally ORACLE also runs a filter that attempts to fill-in invalid pixels coming from the sensor [46], which is however computationally burdensome ($\sim$ 20 ms compute time on an NVIDIA Xavier NX) and not needed in the proposed method thus not executed for the purposes of fair comparison. As shown in Figure 8, the end-to-end ORACLE incorrectly predicts action sequences –that may cause the robot to collide with the environment– as collision-free, unlike the proposed approach which enables safe action sequences towards the goal direction. Despite the potential benefits of end-to-end training, ORACLE is unable to incorporate the perceptual focus on thin obstacles of the new exploit using the seVAE and is also unable to exploit real sensor observations in the training. As a result, learning to avoid hard-to-perceive thin obstacles is only trained from simulated data which present significant differences compared to real data (not merely white noise and other easy-to-simulate errors) [17,18,47] thus widening the sim-to-real gap. Note that, the ensemble of neural networks in [16] is not used to have a fair comparison.

V. CONCLUSIONS

This paper presented a modularized learning-based navigation method that can navigate cluttered environments involving hard-to-perceive obstacles, without access to a map or the full robot state. A semantically-enhanced VAE is designed that can encode raw sensor observations into a latent representation. This is trained with both simulated and real images and utilizes semantic labels to better encode the hard-to-perceive thin obstacles. A Collision Prediction Network is then trained completely in simulation with the

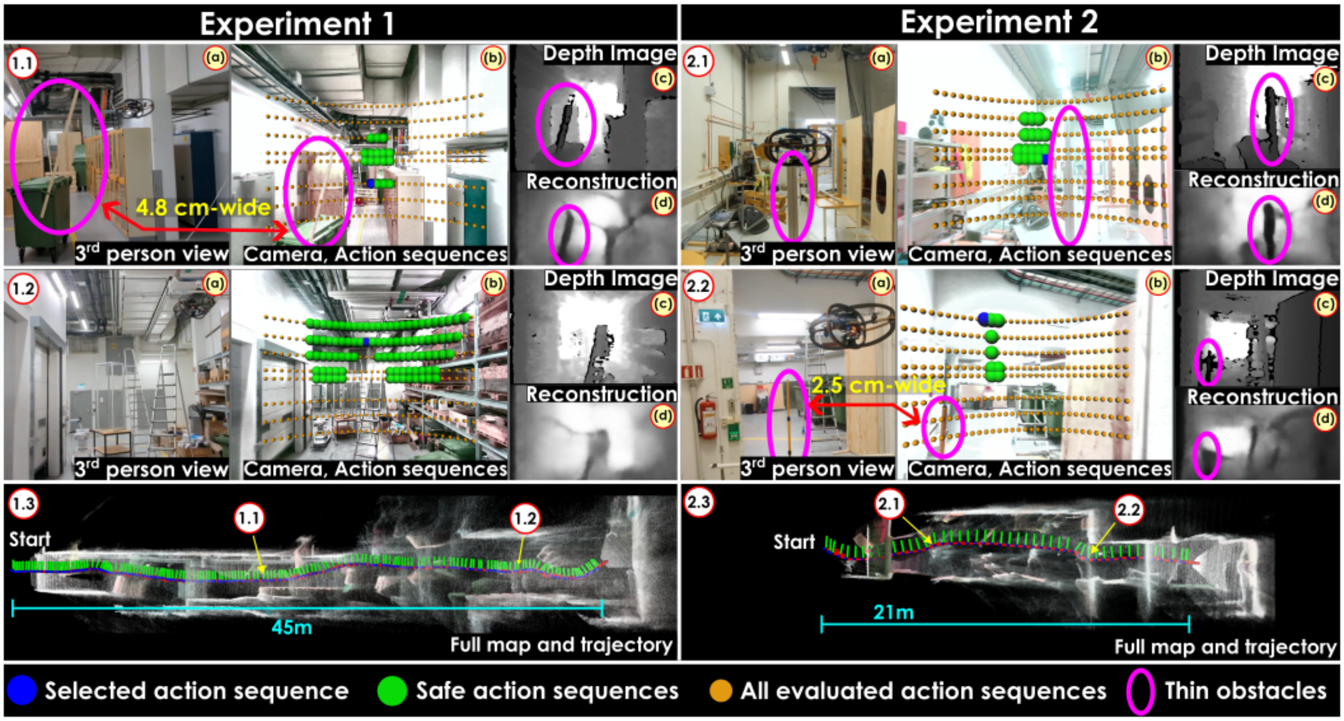

Fig. 7. Results from the two real-world experiments conducted to evaluate the proposed method. The first took place in a long cluttered corridor (~ 50 m) including thin obstacles as small as 4.8 cm in width. The second experiment was conducted in a relatively smaller corridor (~ 20 m) with predominantly thin obstacles, some as small as 2.5 cm in width, some with texture-less reflective surfaces. The sub-figures 1.1 and 1.2 show two instances during experiment 1 (similarly 2.1, 2.2 for experiment 2) detailing a) 3rd person view of the robot and the environment, b) onboard color camera image overlaid with markers representing estimated endpoints of collision-free action sequences, c) raw depth image, and d) reconstructed depth image from seVAE. 1.3 (similarly 2.3 for experiment 2) shows the final map and trajectory followed by the robot in experiment 1. The reconstructed maps are only for visualization purposes and are not computed onboard the robot during flights. It can be seen that the method is successfully able to respond to and avoid all the obstacles.

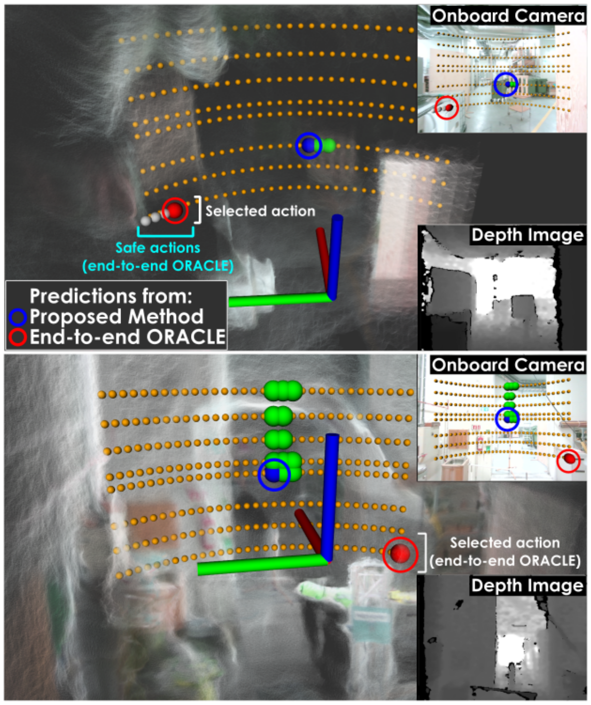

Fig. 8. Comparison of collision predictions of ORACLE end-to-end trained purely in simulation against the proposed method with un-filtered depth data in the real-world experiments. Our proposed method predicts collision-free action sequences and selects the safest one toward the goal whereas ORACLE incorrectly predicts the collision-free action sequences.

latent space from the seVAE to learn to predict collision scores for a set of action sequences. To evaluate our approach, first we compare the performance of the seVAE with other methods for compression. Then, simulation studies are performed to compare the collision avoidance performance of our method with an end-to-end trained (in simulation) approach. Furthermore, we conduct real experiments to show the performance of our method in cluttered environments containing hard-to-perceive thin obstacles, reflective and textureless surfaces. Finally we highlight the benefits of this method in handling real sensor data errors compared to methods trained only simulation. The method and datasets used in this work will be open-sourced to the community.

References

[1] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, L. M. Gambardella, R. Siegwart, and D. Scaramuzza, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, oct 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21887

[2] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[3] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, pp. 1–25, 2015.

[4] G. Caprari, A. Breitenmoser, W. Fischer, C. Hürzeler, F. Tâche, R. Siegwart, O. Nguyen, R. Moser, P. Schoeneich, and F. Mondada,

"Highly compact robots for inspection of power plants," *Journal of Field Robotics*, vol. 29, no. 1, pp. 47–68, 2012.

[5] J. Zhang, J. Hu, J. Lian, Z. Fan, X. Ouyang, and W. Ye, "Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring," *Biological Conservation*, vol. 198, 03 2016.

[6] E. Aucone, S. Kirchgeorg, A. Valentini, L. Pellissier, K. Deiner, and S. Mintchev, "Drone-assisted collection of environmental dna from tree branches for biodiversity monitoring," *Science Robotics*, vol. 8, no. 74, p. eadd5762, 2023. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.add5762

[7] S. LaValle and J. Kuffner, J.J., "Randomized kinodynamic planning," in *IEEE ICRA*, 1999, pp. 473–479 vol.1.

[8] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," 2011. [Online]. Available: https://arxiv.org/abs/1105.1186

[9] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson, "Motion primitives and 3d path planning for fast flight through a forest," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 357–377, 2015.

[10] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[11] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[12] K. Museth, "Vdb: High-resolution sparse volumes with dynamic topology," *ACM Trans. Graph.*, vol. 32, no. 3, jul 2013. [Online]. Available: https://doi.org/10.1145/2487228.2487235

[13] P. R. Florence, J. Carter, J. Ware, and R. Tedrake, "Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3d data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7631–7638.

[14] P. Foehn, D. Brescianini, E. Kaufmann, T. Cieslewski, M. Gehrig, M. Muglikar, and D. Scaramuzza, "Alphapilot: autonomous drone racing," *Autonomous Robots*, pp. 307–320, 2022.

[15] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," in *Science Robotics*, October 2021.

[16] H. Nguyen, S. H. Fyhn, P. De Petris, and K. Alexis, "Motion primitives-based navigation planning using deep collision prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9660–9667.

[17] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "Segstereo: Exploiting semantic information for disparity estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 636–651.

[18] M. S. Ahn, H. Chae, D. Noh, H. Nam, and D. Hong, "Analysis and noise modeling of the intel realsense d435 for mobile robots," in *2019 16th International Conference on Ubiquitous Robots (UR)*. IEEE, 2019, pp. 707–711.

[19] R. Madaan, D. Maturana, and S. A. Scherer, "Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3487–3494, 2017.

[20] M. Kulkarni, B. Moon, K. Alexis, and S. Scherer, *Aerial Field Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 1–15. [Online]. Available: https://doi.org/10.1007/978-3-642-41610-1_221-1

[21] V. Tolani, S. Bansal, A. Faust, and C. Tomlin, "Visual navigation among humans with optimal control as a supervisor," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2288–2295, 2021.

[22] S. Zakharov, R. Ambruș, V. Guizilini, W. Kehl, and A. Gaidon, "Photo-realistic neural domain randomization," in *ECCV*, 2022.

[23] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," 05 2018, pp. 4243–4250.

[24] F. Sadeghi and S. Levine, "CAD2RL: real single-image flight without a single real image," in *Robotics: Science and Systems XIII*, N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, Eds., 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p34.html

[25] D. Gandhi, L. Pinto, and A. Gupta, "Learning to fly by crashing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3948–3955.

[26] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[27] G. Kahn, P. Abbeel, and S. Levine, "Land: Learning to navigate from disengagements," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1872–1879, 2021.

[28] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[29] C. lei Zhou, J. Yang, C. Zhao, and G. Hua, "Fast, accurate thin-structure obstacle detection for autonomous mobile robots," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 318–327, 2017.

[30] R. Madaan, M. Kaess, and S. A. Scherer, "Multi-view reconstruction of wires using a catenary model," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5657–5664, 2019.

[31] B. Landry, R. Deits, P. R. Florence, and R. Tedrake, "Aggressive quadrotor flight through cluttered environments using mixed integer programming," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1469–1475.

[32] G. Dubey, R. Madaan, and S. Scherer, "Droan - disparity-space representation for obstacle avoidance: Enabling wire mapping & avoidance," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6311–6318.

[33] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, 2020.

[36] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion- and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.

[37] A. Loquercio, A. I. Maqueda, C. R. del Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[38] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl

[39] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., vol. 3068, International Society for Optics and Photonics. SPIE, 1997, pp. 182 – 193. [Online]. Available: https://doi.org/10.1117/12.280797

[40] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[41] O. R. Organization. Gazebo classic simulator. Open Robotics Organization. [Online]. Available: https://classic.gazebosim.org/

[42] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021. [Online]. Available: https://arxiv.org/abs/2108.10470

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[44] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors-a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[45] P. D. Petris, H. Nguyen, M. Dharmadhikari, M. Kulkarni, N. Khedekar, F. Mascarich, and K. Alexis, "Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 536–543.

[46] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the cpu," in *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, pp. 16–22.

[47] T. Mallick, P. P. Das, and A. K. Majumdar, "Characterizations of noise in kinect depth images: A review," *IEEE Sensors journal*, vol. 14, no. 6, pp. 1731–1740, 2014.