Task-driven Compression for Collision Encoding based on Depth Images*

Mihir Kulkarni[0000−0003−0895−5867] and Kostas Alexis[0000−0002−9989−298X]

Norwegian University of Science and Technology (NTNU), O. S. Bragstads Plass 2D, 7034, Trondheim, Norway
mihir.kulkarni@ntnu.no

Abstract. This paper contributes a novel learning-based method for aggressive task-driven compression of depth images and their encoding as images tailored to collision prediction for robotic systems. A novel 3D image processing methodology is proposed that accounts for the robot's size in order to appropriately "inflate" the obstacles represented in the depth image and thus obtain the distance that can be traversed by the robot in a collision-free manner along any given ray within the camera frustum. Such depth-and-collision image pairs are used to train a neural network that follows the architecture of Variational Autoencoders to compress-and-transform the information in the original depth image to derive a latent representation that encodes the collision information for the given depth image. We compare our proposed task-driven encoding method with classical task-agnostic methods and demonstrate superior performance for the task of collision image prediction from extremely low-dimensional latent spaces. A set of comparative studies show that the proposed approach is capable of encoding depth image-and-collision image tuples from complex scenes with thin obstacles at long distances better than the classical methods at compression ratios as high as 4050:1.

Keywords: Task-driven compression · Collision prediction · Robotics.

1 Introduction

Methods for autonomous collision-free navigation of aerial robots have traditionally relied on motion planning techniques that exploit a dense map representation of the environment [3,24,27,28]. Departing from such methods, the community has recently investigated the potential of deep learning to develop navigation methods that act directly on exteroceptive data such as depth images instead of reconstructed maps in order to plan the aerial vehicle's motions with minimal latency [12,15,16,22]. However, such methods face the challenge that exteroceptive data and especially depth images coming from stereo vision or other sensors are typically of very high dimensionality and the involved neural networks include layers that partially act as lossy information compression stages. This is reflected in the architectures of otherwise successful methods such as the works in [12,16,22]

* This work was supported by the AFOSR Award No. FA8655-21-1-7033.

that exploit depth images to evaluate which among a set of candidate robot trajectories would collide or not. In [16] the input depth image involves more than $300,000$ pixels (640×480 resolution) but through stages of a pre-trained MobileNetV3 architecture it gets processed to M feature vectors of size 32 each, where M is the number of candidate trajectories for which this method derives collision scores. Eventually by combining the 640×480 pixels depth image with robot pose information, the method attempts to predict which among M trajectories are safe, thus representing a process of information downsampling and targeted inference. In other words, despite the dimensionality reduction taking place through the neural network it is attempted that the method still ensures collision avoidance. However, it is known that such techniques do not provide 100% success ratio especially in complex and cluttered scenes.

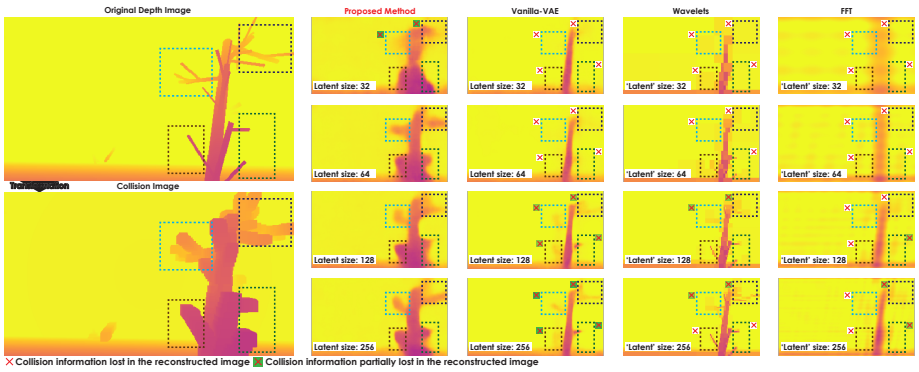

Fig. 1. Aggressive compression/encoding of depth images on aggresively low-dimensional latent spaces using conventional techniques is likely to lead to major loss of collision information. On the contrary, a task-driven compression paradigm is proposed that allows to retain most of the collision information even in exceptionally low latent spaces. This work serves as a modular step that delivers compressed latent spaces that retain collision information and can thus be utilized for further processing by methods that predict the possible collision of candidate trajectories of robots in complex scenes.

Responding to the above, this work contributes the concept of task-driven compression and encoding of depth images as visualized in Figure 1. Departing from the concept that methods aiming to predict the safety of candidate robot trajectories based on depth images should train collision prediction either a) directly in an end-to-end fashion through depth data [16,22] or through b) an explicit intermediate compression stage of the depth image itself [23], we propose the approach of using the depth image to encode a latent space presenting major dimensionality reduction that reflects not the depth image itself but instead a "collision image". The latter is a remapping of the depth image that has accounted about the robot's size and thus presents reduced overall complexity and greatly reduced presence of narrow/thin features that are hard-to-retain in an aggressive compression step. To achieve this goal, the method employs a probabilistic encoder-decoder architecture that is trained in a supervised manner such that given a depth image as input, it learns to encode and reconstruct

the collision image. To train this collision-predicting network –dubbed Depth image-based Collision Encoder (DCE)– the depth image is first processed such that the collision image is calculated given information for the robot's size. Focusing on aggressive dimensionality reduction, it is demonstrated that the scheme allows to get accurate reconstructions through a latent space that is more than 3 orders of magnitude smaller than the input image. The benefits of the approach are demonstrated through comparisons both with a conventional Variational Autoencoder (VAE) trained to encode the depth image and assessed regarding the extent to which the reconstructed image can serve as basis to derive a correct collision image, as well as traditional compression methods using the Fast Fourier Transform (FFT) and wavelets.

In the remaining paper Section 2 presents related work and Section 3 details the proposed method involving generation of training data, image augmentation and the training of the neural network. Section 4 compares our proposed method against traditional image compression methods and evaluates the performance of task-driven and task-agnostic compression methods at similar degrees of compression. Finally, conclusions are drawn in Section 5.

2 Related Work

This work draws its motivation from the set of deep learning methods that rely on directly processing sensor data (such as depth images) in order to predict if a candidate trajectory of a flying robot shall be in collision or not [12,15,16,22,23] and accordingly enable safe autonomous navigation. A subset of such methods instead of relying on direct end-to-end learning from exteroceptive data and robot pose information to predict if a certain candidate action/trajectory shall allow collision-free flight, they employ modularization and accordingly an explicit step of compression that pre-processes high-dimensional input image data arriving to a low-dimensional latent space [12,23].

Technically, the contribution relates to the body of work on image compression. In this large body of work, multiple methods are available including classical schemes that rely on FFT or wavelets [4,14]. Within the breadth of relevant techniques, of special interest is the utilization deep learning approaches [2, 19] and especially variational autoencoders [5,9,26] as means to achieve good reconstruction quality for high compression ratios [30,31]. Nevertheless, the majority of such methods follow the main paradigm of compression which implies that a uniform metric (e.g., mean squared loss) of over pixel-level reconstruction against the original image is employed. Even for works that exploit additional cues such as semantics [29], conventional compression remains the prime goal. Departing from this paradigm this work reflects the fact that in the line of works of collision prediction [12,15,16,22,23] it is the information over candidate collisions that matters and not the depth pixels themselves. In other words, it is the question if the robot - with the specific volume that it occupies - can fly along a path within the volume observed and captured by the depth image. This calls for a new concept that hereby is called purposeful task-driven depth

image compression/encoding for collision prediction utilizing minimal latent spaces. It is highlighted that the goal to arrive at a latent space that is multiple orders of magnitude smaller than the high dimensional depth images –offered by sensing solutions such as modern stereo vision– is driven from the need of robust performance and generalization in diverse natural environments. As established by seminal works such as ResNet [8], deeper models with more parameters require much more data to train. A low-dimensional compression latent space enables methods that shall then use it for collision prediction [12] to utilize smaller and simpler networks for the task, while they further combine with robot data which are also low-dimensional (e.g., pose states of a quadrotor aerial vehicle over the SE(3) special Euclidean group [13]).

3 Proposed Method

The proposed approach on task-driven compression and particularly depth image-based collision encoding is outlined below. First, the process to generate relevant training data is discussed, followed by the method to derive the collision image associated with each depth image. Subsequently, the depth images-based collision encoder motivated by the architecture of variational autoencoders is presented.

3.1 Dataset Generation

Deep learning techniques for data compression require large amounts of data for training. Moreover, the generalizability of the learned models depends on the quality of the training data and the variety of samples provided for learning. Available depth image datasets primarily focus on specific tasks to be performed using the depth images such as depth completion [21] or autonomous driving [7]. These datasets contain images from scenes that include urban structured indoor settings and open streets respectively with large-sized obstacles that are sparsely distributed in the environment. Consequently, such datasets - that are otherwise common within both research and industry - do not contain images from highly cluttered complex environments that present challenges to aerial robot navigation. For the latter, it is important to note that environments with a) high clutter leading to uncertainty as to the safest flying direction, and b) obstacles with narrow cross section ("thin" obstacles) are particularly hard to fly through. In order to train our neural network models for such cluttered environments containing narrow/thin obstacles, while ensuring generalizability, we rely on two popular robot simulators - namely Gazebo Classic [25] and Isaac Gym [18] to generate diverse simulated depth image data. These simulators provide the necessary interfaces that allow us to rearrange different objects randomly in a simulated environment. Images from Gazebo Classic are collected using the onboard depth camera of a simulated aerial robot in an obstacle-rich environment using the RotorS Simulator [6]. Subsequently, we utilize the Isaac Gym-based Aerial Gym Simulator [11] in order to simulate environments with randomly

placed obstacles and collect depth images in a parallelized manner from multiple randomly generated environments simultaneously. $85,000$ depth images are collected in environments consisting of a variety of objects ranging from multi-branched tree-like objects with thin cross-sections to large obstacles with cavities in them. Depth images are collected and aggregated to be processed for computing a robot-specific collision image.

3.2 Collision Image Generation

While the collected depth images provide information about the projected distance to a surface along the central axis of the camera, it is difficult to infer the collision-free regions in the robot's field-of-view. Traditional approaches to compute collision-free regions involve representing the depth image in an intermediate volumetric map-based representation [10,20,24] that can be queried to derive collision-free regions. These representations are limited by their discretization capabilities and often require a large amount of memory to maintain a persistent map [24]. Generation of such representations is also a computationally expensive step [10]. Finally, such reconstructions rely on aggregating multiple depth image readings and thus necessitate consistent pose estimation. At the same time, methods that use depth images to directly predict if a candidate path is collision-free or not [16,22] implicitly have to learn that the depth image itself is not a map of collision-free space but instead this information can be acquired by further correlating the range to a point and the size of the robot. Contrary to the current techniques on that front that typically either a) resort on end-to-end learning of collisions via depth, state and action tuples [16,22] or b) compress the depth image and use this lossy latent space to then learn collision prediction [12,23], we here propose the re-mapped representation of depth images in a new form that directly provides the collision-free distance that can be traversed by a robot along any direction. A collision image is defined as an image representing the collision-free distance (projected along the central axis of the camera) traversable by a robot of known dimensions along the rays corresponding to each pixel in an image. This revised image representation that encodes all necessary collision information can then be utilized directly for robot navigation tasks.

To derive collision images from depth images, we propose a computationally efficient method illustrated in Figure 2. Motivated by the observation that the most significant change between the depth image and the collision image occurs at the edges of obstacles in the field-of-view of the camera, a rendering-based approach is utilized to appropriately inflate the objects in the camera's field-of-view about their edges. We cannot perform this inflation accurately using traditional 2D computer vision techniques since the modified area around each edge pixel is both dependent on the size of the robot and the distance to the point in 3D space making the computation intractable. We rely on parallelized rendering frameworks to visualize virtual robot-sized meshes around the regions corresponding to the edges of the obstacles in order to inflate them by the size of the robot. Projecting them back onto the camera plane captures the appropriately inflated regions of the environments that represent the regions of collision for

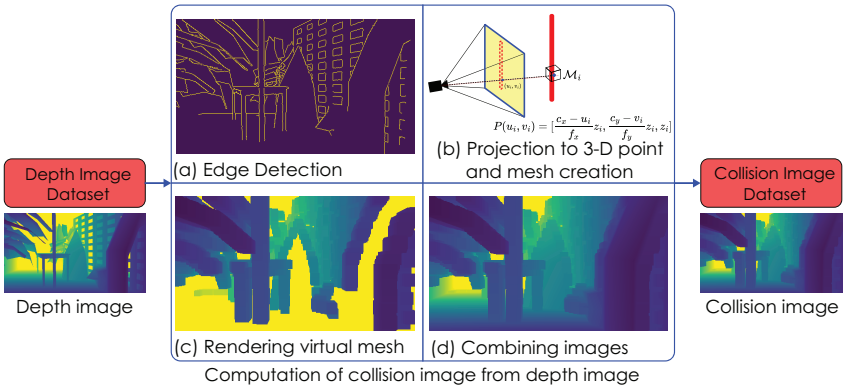

Fig. 2. The acquired dataset is processed for task-driven compression. Edge-detection is performed on the depth images and each edge pixel is projected to it's 3D coordinates to form a pointcloud and a virtual 3D mesh is rendered. The depth image of the virtual mesh is obtained and combined with an offset range image to form a collision image.

the robot. Edge detection is performed on the original depth image $\mathcal{D}$ using OpenCV [1] to obtain the set of pixels corresponding to the edges $\mathcal{E}$ as shown in Figure 2(a). A fraction of the edge pixels are randomly selected to render meshes. For each selected edge pixel i with coordinates $(u_i, v_i) \in \mathcal{E}$, the position of the corresponding point $P_i \in \mathbb{R}^3$ is calculated as:

$$P_i = (x_i, y_i, z_i), \tag{1}$$

where

$$x_i = \frac{c_x - u_i}{f_x} z_i, \tag{2}$$

$$y_i = \frac{c_y - v_i}{f_y} z_i, \tag{3}$$

$$z_i = \mathcal{D}(u_i, v_i). \tag{4}$$

A pinhole model of the camera is considered, with f_x and f_y as the focal lengths and c_x and c_y as the optical centers. The shape of the robot is considered to be cubical with edge length $2r$. For each projected point P_i, a robot-sized mesh $\mathcal{M}_i$ is centered at the coordinates (x_i, y_i, z_i) as shown in Figure 2(b). Meshes created around each point are merged into a single aggregated mesh $\mathcal{M}$. We use NVIDIA Warp [17], a high-performance graphics and simulation package that enables rendering simulated depth cameras in a virtual environment consisting of this aggregated mesh. A parallelized ray-casting operation is performed to project rays into this virtual mesh environment and obtain a depth image $\mathcal{D}_\mathcal{M}$ only containing these virtual meshes (Figure 2(c)). This depth image only contains the information regarding the distances to the virtual meshes corresponding to the edge pixels in the original depth image $\mathcal{D}$. Since rendering of virtual meshes is a computationally expensive step, it is reserved only for the edge pixels in the image. For pixels lying in the interior regions of the object in the depth image,

an offset depth image $\mathcal{D}_{\text{offset}}$ is created with all range values brought closer by the size of the robot r using the following operation:

$$\mathcal{D}_{\text{offset}} = \mathcal{R}^{-1}(\mathcal{R}(\mathcal{D}) - r), \tag{5}$$

where the transformation $\mathcal{R}$ converts the depth image to a range image, i.e., the value in each pixel of the image represents the Euclidean distance to the corresponding point on the object. The inverse function $\mathcal{R}^{-1}$ converts the range image back to a depth image. Finally, an approximate collision image $\mathcal{D}_{\text{coll}}$ is obtained by taking pixel-wise minimum values of the offset depth image $\mathcal{D}_{\text{offset}}$ and rendered image with inflated meshes $\mathcal{D}_{\mathcal{M}}$ as shown in Figure 2(d). This operation is given by:

$$\mathcal{D}_{\text{coll}} = min(\mathcal{D}_{\mathcal{M}}, \mathcal{D}_{\text{offset}}). \tag{6}$$

We use this to generate a collision image dataset given the depth image dataset with each image in the original dataset being processed in the above manner to produce a collision image. Both the original image and the collision image are aggregated into a common dataset to be used for training the probabilistic encoder-decoder network to derive and encode the collision information from the original depth images.

3.3 Depth Image Compression and Collision Encoding

The interpretation and representation of depth information to derive collision images requires spatial understanding of the environment. We utilize artificial neural networks to perform this task by learning a compressed representation that compresses and encodes the depth image to its associated collision image. The overall architecture is motivated by the success of VAEs but with the important distinction that the involved learning includes training of the depth-to-collision image map transformation. We consider a dataset containing depth images $\mathbf{x} \in \mathbb{D}$, and its derived secondary dataset containing collision images $\mathbf{x}_{coll} \in \mathbb{D}_{coll}$. A surjective function $\mathcal{P} : \mathbb{D} \mapsto \mathbb{D}_{coll}$ maps each element from the depth image dataset to an image in the collision image dataset. This function is imitated in the collision image generation step (Section 3.2). Each $\mathbf{x}_{coll} \in \mathbb{D}_{coll}$ can be assumed to be generated by a process using a latent random variable $\mathbf{z}$.

We employ probabilistic encoders and decoders to perform dimensionality reduction of the input depth data and learn a highly compressed latent representation for predicting collision images. A probabilistic decoder $p_\theta(\mathbf{x}_{coll}|\mathbf{z})$, given $\mathbf{z}$ produces a distribution over the possible values of $\mathbf{x}_{coll}$, while a probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{x})$ learns to encode the input image $\mathbf{x}$ to a latent distribution with mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$. This distribution is sampled to obtain $\mathbf{z}$ such that $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma} \cdot \mathbf{I})$. The encoder and decoder networks are jointly trained to produce a highly compressed but well performing latent representation $\mathbf{z}$ given a depth image $\mathbf{x}$ and its $\mathbf{x}_{coll}$. The decoder can be used to derive a collision image that approximates $\mathbf{x}_{coll}$ and accurately predicts the distances for collision-free

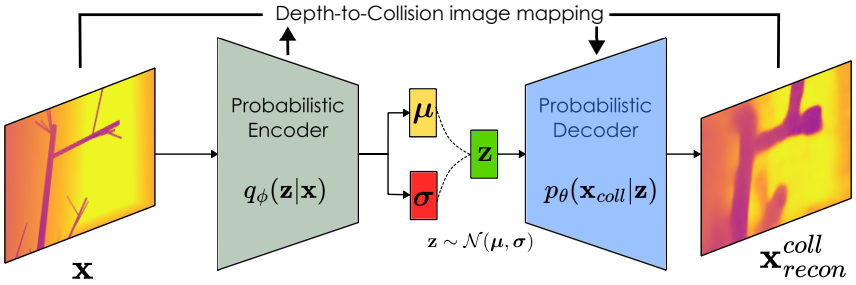

Fig. 3. The proposed neural network with an encoder-decoder architecture inspired by variational autoencoders and tailored to compress and re-map a depth image $\mathbf{x}$ to a latent representation $\mathbf{z}$ that can be used to produce the reconstructed image $\mathbf{x}_{recon}^{coll}$ that approximates the associated collision image $\mathbf{x}_{coll}$.

traversal using the given depth image. Figure 3 shows the structure of the DCE for task-driven compression. To train the DCE the loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{recon} + \beta_{norm}\mathcal{L}_{KL}, \tag{7}$$

where

$$\mathcal{L}_{recon}(\mathbf{x}_{coll}, \mathbf{x}_{recon}^{coll}) = \text{MSE}(\mathbf{x}_{coll}, \mathbf{x}_{recon}^{coll}), \tag{8}$$

$$\mathcal{L}_{KL}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log(\boldsymbol{\sigma}_j^2) - \boldsymbol{\mu}_j^2 - \boldsymbol{\sigma}_j^2\right). \tag{9}$$

Here, $\mathcal{L}$ denotes the overall loss term while $\mathcal{L}_{recon}$ and $\mathcal{L}_{KL}$ (scaled by a constant β_{norm} [9]) denote the reconstruction loss and the KL-divergence loss terms respectively in a manner motivated by autoencoder literature [5]. The Mean-Square Error (MSE) loss function is modified to ignore the errors of the pixels from the depth image that are invalid, i.e., the pixels that do not contain accurate depth information owing to the obstacles being too close to the camera in simulated images or also in case of the incorrect depth from stereo shadows for real-world depth images. The encoder is a residual neural network consisting of convolutional layers at each block and uses the ELU activation function. The final layers of the encoder network are fully connected layers that produce the mean and variance describing the latent distribution. The decoder consists of two fully connected layers followed by non-residual de-convolutional layers with ReLU activation functions. The last convolutional layer has a sigmoid activation to have bounded values for the collision image. The network is trained on a dataset consisting of $70,000$ depth and collision image pairs and tested on a dataset containing $15,000$ image pairs. Each image has a dimension of 270×480 pixels and contains the distance to the given obstacle projected along the central axis of the camera. As discussed in the next section, well performing latent spaces as low as 32 variables are achieved which represents more than 3 orders of magnitude compression, while simultaneously delivering and exploiting the described depth-to-collision image transformation.

4 Evaluation and Results

The main premise of the work is that the implicitly learned transformation of depth-to-collision image mapping, not only allows to learn directly the information pertinent to collision prediction, but also allows major compression while retaining the necessary information. To demonstrate this fact, we conduct a comprehensive set of evaluation studies comparing the performance of our proposed approach against traditional task-agnostic compression methods such as using the wavelet transform and the Fast Fourier Transform (FFT). We also compare our task-driven compression method against a conventionally trained task-agnostic VAE (vanilla-VAE) that shares the same neural network architecture as the DCE. We first show that neural network-based compression outperforms traditional compression methods such as FFT and wavelet transform-based compression for very high compression ratios for depth images. Furthermore, the reconstructed collision image obtained from the task-driven DCE accurately represents the calculated collision image as compared to the derived collision information from the image reconstructed from the vanilla-VAE. The performance of the proposed approach is evaluated for a set of different latent dimensions representing varying levels of extreme compression. Latent spaces of 32, 64, 128 and 256 latent dimensions corresponding to compression factors of 4050, 2025, 1012.5 and 506.25 respectively are considered. The proposed learning-based compression and image domain transformation method not only outperform the currently established approaches while achieving large compression ratios but also are capable of encoding spatial information from the depth image to represent collision information. This is made evident from the results where the depth image is accurately (and range- and robot size-dependent) "inflated" to obtain a collision image that occludes the obstacles in the background.

4.1 Comparison of vanilla-VAE with traditional compression methods

We compare a vanilla-VAE based compression with FFT and wavelet transform-based compression. The task-agnostic vanilla-VAE is trained using $70,000$ images to encode a depth image $\mathbf{x}$ into a latent distribution and also to reconstruct the input depth image $\mathbf{x}_{recon}^{vanilla}$. This is done to first ensure a fair comparison between task-agnostic methods. A separate network is trained on the dataset for each latent space size.

We obtain the image representation in the wavelet domain by decomposing the image with the Daubechies wavelet 'db1'. To obtain the compressed representation in this domain corresponding to a latent space size of n, the largest n magnitudes in the wavelet domain are retained, while all other values are set to 0. The resultant wavelet domain representation is reconstructed using the inverse wavelet transform to obtain $\mathbf{x}_{recon}^{wv}$. Similarly, to compress the image using FFT, the complex numbers in the frequency domain that correspond to the $n/2$ largest magnitudes are retained (with both their real and complex coefficients), while all others are set to 0. A reconstruction $\mathbf{x}_{recon}^{FFT}$ is obtained from this compressed

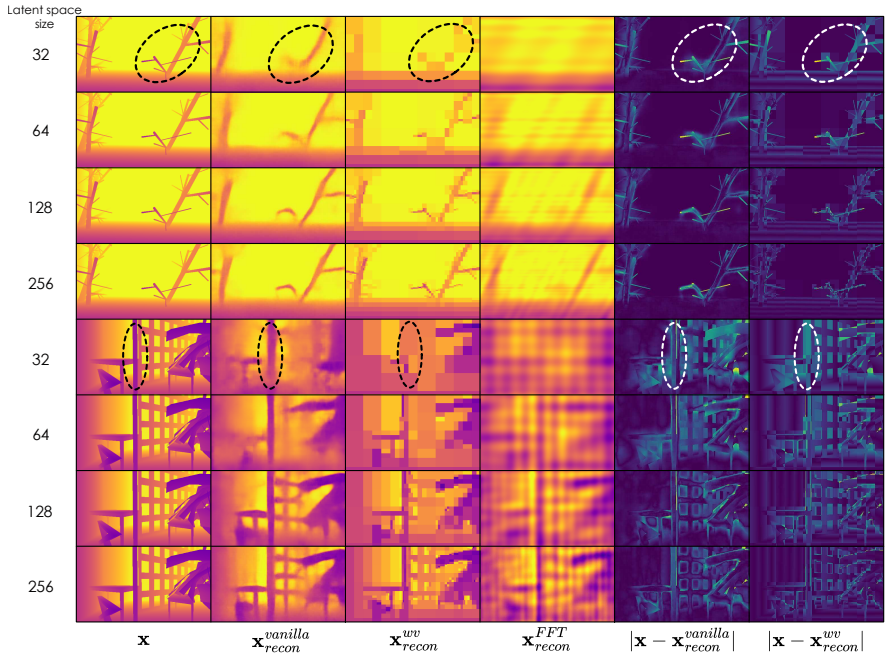

Fig. 4. Comparison between the reconstruction performance on depth images using traditional methods and the vanilla-VAE for different levels of compression. The images compressed and reconstructed using vanilla-VAE ($\mathbf{x}_{recon}^{vanilla}$), wavelet transform ($\mathbf{x}_{recon}^{wv}$) and FFT ($\mathbf{x}_{recon}^{FFT}$) are shown. The errors in the reconstruction are also highlighted.

representation by performing an inverse FFT. It must be noted that both these representations are computationally represented as ordered lists that contain the position-dependent coefficients for the decoder to reconstruct the image. While we retain only the top n coefficients, we do not remove their position information to allow the reconstruction software to work seamlessly. As a result, information retained using this scheme is *more* than just the n dimensional variable that we use in the case of the neural networks. Figure 4 compares the reconstructed images from the compressed representation for different latent space sizes using different compression methods. The vanilla-VAE preserves the features in the depth image for complex scenes for small latent sizes, while the wavelet transform-based compression performs well for larger latent space sizes. The difference between the reconstructed image using the vanilla-VAE and the wavelet transforms and the input image is shown to highlight the regions with a higher reconstruction error. A visual inspection of the reconstructed images from wavelets and frequency domain representations show that these methodologies are unable to encode the information in complex depth images for smaller latent space dimensions. The difference is especially highlighted in images that contain complex and cluttered settings, where the FFT reconstructions generate artificial patterns, while the wavelet reconstructions discretize regions of the image non-uniformly, losing out on the sharper details of the image.

The results are tabulated in Table 4.1 demonstrating that for high compression ratios corresponding to latent spaces of 32, 64 and 128 dimensions, the vanilla-

VAE based depth image compression method produces images with a lower MSE value with the input image. Interestingly, the wavelet transform-based method produces a lower MSE in the case where the information corresponding to the top 256 coefficients is retained. As shown in Figure 4, the wavelet reconstruction corresponding to this size produces sharper edges in the reconstructed image owing to the capability to encode more information regarding the smaller discretized regions in the image.

Table 1. Comparison of MSE for reconstructed images with vanilla-VAE, FFT and wavelet transform for different compressed latent dimensions.

MSE against input image x				
Latent dims:	**32**	**64**	**128**	**256**
$\mathbf{x}_{recon}^{vanilla}$	**1249.58**	**827.00**	**543.38**	477.88
$\mathbf{x}_{recon}^{wv}$	1481.36	952.58	612.31	**382.43**
$\mathbf{x}_{recon}^{FFT}$	2223.87	1634.52	1181.93	840.38

4.2 Task-driven compression for collision representation

While the task-agnostic vanilla-VAE demonstrates good compression capacity of complex depth images to a small latent code, it still faces limitations in producing reconstructions that can be used to derive an accurate collision representation especially in cluttered and complex scenes. As expected, aggressive compression leads to loss of important information. However, compared to the depth image, a collision image would typically contain less complex and more low-frequency information regarding the same scene owing to the "inflation" of the obstacles. Due to this process, pixels corresponding to thin features in a depth image end up being represented by a larger region of pixels showing collision-free distance values. It is noted, transforming the depth image to a collision image requires a spatial understanding of the scene as robot size-inflated regions in the collision image occlude the regions near the edges of obstacles represented in depth images. Nonetheless, once a network is trained to predict this, it also implies reduction in the information that has to be kept during compression.

We compare the performance of the proposed DCE against the task agnostic vanilla-VAE to compare the capability of these networks in retaining collision prediction information in the compressed latent space spanning from the depth image. The DCE is trained to directly reconstruct the collision image, while the vanilla-VAE is trained to reconstruct the input depth image and thus for the purposes of assessing its capacity to retain the information needed for collision prediction, a new collision image is derived (as in Section 3) from the images reconstructed from its latent space through the decoder. Essentially, to ensure a fair comparison, we use the mapping $\mathcal{P}(\mathbf{x}_{recon}^{vanilla})$ to obtain the derived collision image from the reconstructed input depth image. Figure 5 presents examples of images reconstructed using both the DCE and vanilla-VAE.

The reconstructed collision image $\mathbf{x}_{recon}^{coll}$ and the derived collision image from the vanilla-VAE reconstruction $\mathcal{P}(\mathbf{x}_{recon}^{vanilla})$ are compared against the true collision image. The areas of errors are highlighted in Figure 5. The collision

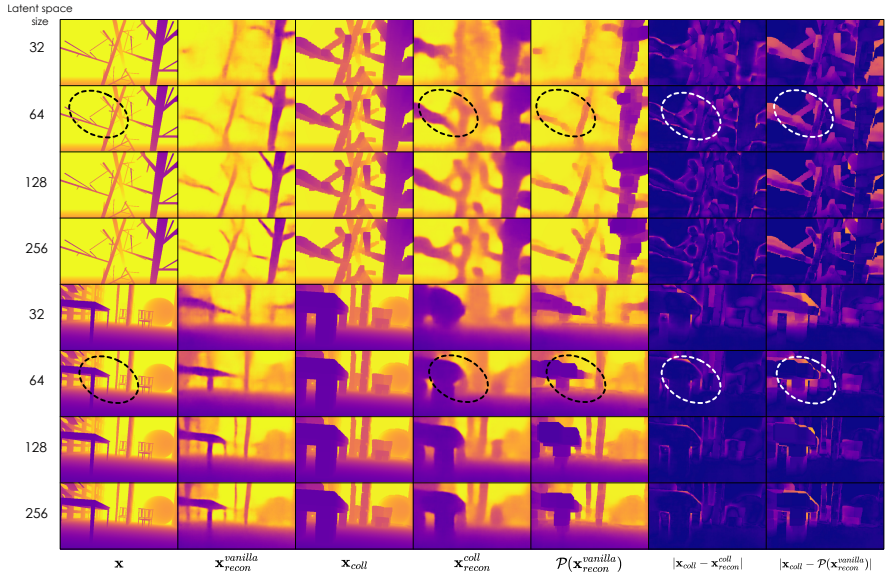

Fig. 5. Comparison between DCE and vanilla-VAE to derive the collision information from the input depth image for different levels of compression. The images compressed and reconstructed using DCE ($\mathbf{x}_{recon}^{coll}$), vanilla-VAE ($\mathbf{x}_{recon}^{vanilla}$) are shown. A collision image $\mathcal{P}(\mathbf{x}_{recon}^{vanilla})$ is derived from $\mathbf{x}_{recon}^{vanilla}$. The derived collision images are compared against the ground-truth collision image $\mathbf{x}_{coll}$ for errors.

image derived from the vanilla-VAE reconstruction shows a greater number of regions with erroneous collision information, while the image from the DCE $\mathbf{x}_{recon}^{coll}$ shows both smaller error magnitudes and only small regions of error when compared to the true collision image $\mathbf{x}_{coll}$. Moreover, the reconstructed collision image captures thin features such as branches in the environment and reconstruct the regions of collisions in the same. The results calculating the MSE of the reconstructed collision image and the derived collision image from the depth image reconstruction are presented in Table 4.2. As presented, the task-driven DCE outperforms the vanilla-VAE by a large margin.

Table 2. Comparison of MSE for reconstructed images with DCE and a transformed collision representation of the image reconstructed using vanilla-VAE.

MSE against Collision Image $\mathbf{x}_{coll}$				
Latent dims:	**32**	**64**	**128**	**256**
$\mathbf{x}_{recon}^{coll}$	**783.718**	**516.487**	**418.03**	**402.66**
$\mathcal{P}(\mathbf{x}_{recon}^{vanilla})$	4828.50	4339.76	2532.14	2539.89

5 Conclusions and Future Work

This paper presented a learning-based method for task-driven aggressive compression of depth images to a highly compressed latent representation tailored

to infer collision-free travel distances for a robot in the environment. A novel method was proposed to generate robot size-specific collision prediction data from given depth images using rendering frameworks. Such depth and collision prediction image tuples are then used to train a neural network performing the task-driven compression of encoding a latent space that captures collision information from depth images. We show that our proposed approach is able to encode depth images by a compression factor over 4000 : 1, while retaining the information necessary to predict collisions from depth images of complex cluttered scenes. Moreover, we show that such purposeful neural network-based compression techniques demonstrate superior performance against traditional methods using FFT and wavelets or even conventional variational autoencoders for image reconstruction from highly compressed latent spaces.

References

1. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
2. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Deep residual learning for image compression. In: CVPR Workshops. p. 0 (2019)
3. Dang, T., Tranzatto, M., Khattak, S., Mascarich, F., Alexis, K., Hutter, M.: Graph-based subterranean exploration path planning using aerial and legged robots. Journal of Field Robotics (2020)
4. Dhawan, S.: A review of image compression and comparison of its algorithms. International Journal of electronics & Communication technology **2**(1), 22–26 (2011)
5. Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016)
6. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: Rotors-a modular gazebo mav simulator framework. In: Robot Operating System (ROS), pp. 595–625. Springer (2016)
7. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (2017), `https://openreview.net/forum?id=Sy2fzU9gl`
10. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots (2013)
11. Kulkarni, M., Forgaard, T.J.L., Alexis, K.: Aerial gym - isaac gym simulator for aerial robots (2023)
12. Kulkarni, M., Nguyen, H., Alexis, K.: Semantically-enhanced deep collision prediction for autonomous navigation using aerial robots (2023)
13. Lee, T., Sreenath, K., Kumar, V.: Geometric control of cooperating multiple quadrotor uavs with a suspended payload. In: 52nd IEEE conference on decision and control. pp. 5510–5515. IEEE (2013)

14. Lewis, A.S., Knowles, G.: Image compression using the 2-d wavelet transform. IEEE Transactions on image Processing **1**(2), 244–250 (1992)
15. Loquercio, A.: Agile Autonomy: Learning High-Speed Vision-Based Flight, vol. 153. Springer Nature (2023)
16. Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., Scaramuzza, D.: Learning high-speed flight in the wild. In: Science Robotics (October 2021)
17. Macklin, M.: Warp: A high-performance python framework for gpu simulation and graphics. `https://github.com/nvidia/warp` (March 2022), nVIDIA GPU Technology Conference (GTC)
18. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., State, G.: Isaac gym: High performance gpu-based physics simulation for robot learning (2021). https://doi.org/10.48550/ARXIV.2108.10470, `https://arxiv.org/abs/2108.10470`
19. Mishra, D., Singh, S.K., Singh, R.K.: Deep architectures for image compression: a critical review. Signal Processing **191**, 108346 (2022)
20. Museth, K.: Vdb: High-resolution sparse volumes with dynamic topology. ACM Trans. Graph. **32**(3) (jul 2013). https://doi.org/10.1145/2487228.2487235, `https://doi.org/10.1145/2487228.2487235`
21. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
22. Nguyen, H., Fyhn, S.H., De Petris, P., Alexis, K.: Motion primitives-based navigation planning using deep collision prediction. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 9660–9667. IEEE (2022)
23. Niu, C., Newlands, C., Zauner, K.P., Tarapore, D.: An embarrassingly simple approach for visual navigation of forest environments. Frontiers in Robotics and AI **10** (2023)
24. Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J.: Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
25. Organization, O.R.: Gazebo classic simulator, `https://classic.gazebosim.org/`
26. Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L.: Variational autoencoder for deep learning of images, labels and captions. Advances in neural information processing systems **29** (2016)
27. Rocha, L., Saska, M., Vivaldini, K.: Overview of uav trajectory planning for high-speed flight. In: 2023 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 110–117. IEEE (2023)
28. Tordesillas, J., Lopez, B.T., How, J.P.: Faster: Fast and safe trajectory planner for flights in unknown environments. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 1934–1940. IEEE (2019)
29. Wang, C., Han, Y., Wang, W.: An end-to-end deep learning image compression framework based on semantic analysis. Applied Sciences **9**(17), 3580 (2019)
30. Wen, S., Zhou, J., Nakagawa, A., Kazui, K., Tan, Z.: Variational autoencoder based image compression with pyramidal features and context entropy model. In: CVPR Workshops. p. 0 (2019)
31. Zhou, L., Cai, C., Gao, Y., Su, S., Wu, J.: Variational autoencoder for low bit-rate image compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 2617–2620 (2018)