This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Visuo-Tactile Recognition of Partial Point Clouds Using PointNet and Curriculum Learning

Enabling Tactile Perception from Visual Data

xxxxx

By Christopher Parsons, Alessandro Albini, Daniele De Martini and Perla Maiolino

This article is about recognizing handheld objects from incomplete tactile observations with a classifier trained on only visual representations. Our method is based on the deep learning (DL) architecture PointNet and a curriculum learning (CL) technique for fostering the learning of descriptors robust to partial representations of objects. The learning procedure gradually decomposes the visual point clouds to synthesize sparser and sparser input data for the model. In this manner, we were able to employ one-shot learning, using the decomposed visual point clouds as augmentations, and reduce the data-collection requirement for training. The approach allows for a gradual improvement of prediction accuracy as more tactile data become available.

We evaluated the effectiveness of the curriculum strategy on our generated visual and tactile datasets, experimentally showing that the proposed method improved recognition accuracy by up to 23% on partial tactile data and boosted accuracy on full tactile data from 93 to 100%. The curriculum-trained network recognized objects with an accuracy of 80% using only 20% of the tactile data representing the objects, increasing to 100% accuracy on clouds containing at least 60% of the points.

Digital Object Identifier 10.1109/MRA.2022.3212316
Date of current version: 21 October 2022

Introduction

Sense of touch is crucial for humans in performing several tasks, from exploration to manipulation and object recognition. In literature, computer vision remains the dominant modality for object-recognition tasks; nevertheless, touch can provide crucial information in unstructured environments when vision is impaired due to poor lighting, unfavorable weather, translucent objects, or occlusion. This has motivated researchers in exploring not only tactile-based object recognition [1] but also combining vision and tactile information to support vision-based perception [2] and accurately reconstructing the shape of complex 3D objects, even when vision

1070-9932/22©2022IEEE

is subjected to occlusions [3], [4]. However, these methods require tactile-based models of the objects for training, which can only be obtained through time and resource expensive exploration tasks. In this respect, it would be convenient to 1) exploit the readiness and availability of visual data to train a system that can recognize objects through environment-robust touch and 2) recognize an object using partial data to mitigate the need to complete a long and expensive tactile exploration.

This article is concerned with the specific case of training an object-recognition system using only a priori visual data to recognize the same object from the tactile modality, albeit not previously sensed through the latter. Humans rely on this visuo-tactile cross modality and therefore are able to reconstruct a vision-based representation of objects and recognize them using sense of touch only. Similarly, exploiting this type of cross modality in robotics would allow learning an object representation in a controlled environment using vision and deploying the system in a more challenging scenario where the vision is not available (e.g., manipulation tasks in clutter, or where the target objects are not directly visible). The works proposed in [5] refer to this as *cross-modal recognition* (*CMR*) or visuo-tactile recognition (VTR), taking inspiration from its psychological definitions [6]. Beyond [5], other examples of a visuo-tactile cross modality can be found in the literature [7], [8]. The work presented in [7] explored a visuo-tactile cross modality to generate tactile images from visual images and vice versa; yet the training of the two systems required both tactile and visual data. The work in [8] used vision to estimate the pose of an object and proposed a Bayesian algorithm with linear Kalman filters to hone that prediction with each sequential touch.

To the best of our knowledge, only Falco et al. [5] have performed CMR by training a system with only the visual modality on a set of quasi-planar rigid objects. They found point clouds to be a suitable representation to encode visual and tactile data for this task. The approach enriches the ensemble of shape functions (ESF) with information from signature of histograms of orientations (SHOT) to form the cross-modal point cLoUd dEscriptor (CLUE) and subsequently use a geodesic flow kernel (GFK) transfer learning technique to increase cross-modal performance. The limitation of this work is that the proposed training pipeline, based on an ensemble of global handcrafted descriptors for point clouds, requires full tactile exploration of the object to perform the predictions. However, as tactile exploration is a time-expensive task, this article focuses on recognizing objects from partial observations and making predictions that can be iteratively improved as more data are gathered. The descriptors employed in [5] are global, requiring the full tactile model of the object, and therefore are not directly employable when attempting recognition from partial observations. Conversely, in this article, we investigate the use of data-driven techniques to learn more task-specific representations. Therefore, instead of exploiting existing descriptors, we define the task and allow the proposed learning procedure to compose the features.

Neural networks (NNs) have been used extensively for point cloud recognition in past years to statistically learn point cloud descriptors or shape embeddings [9]. These descriptors adapt based on the training dataset and formulation of the learning task, learning geometric relationships directly from the data. Rather than proposing a handcrafted descriptor that can capture local shapes, we utilize the established point-based architecture PointNet [10] to extract task-driven shape descriptors. The work presented in [11] noted the gap in the research of partial point cloud recognition and explored the ability of PointNet to recognize partial and noisy point clouds. The authors found that it was vital to expose the network to partial representations during training. In this article, we take it a step further by formulating a learning task with a training procedure based on CL [12] to foster learning of local descriptors from sparser and sparse tactile data, represented as point clouds.

PointNet, a point-based architecture, was chosen over projection- or volumetric-based methods such as a multiview convolutional NNs (MVCNNs) and VoxNet [13]. Projection-based methods rely on a meshing preprocessing that, besides being computationally expensive, assumes the emptiness of the unexplored regions; this is undesirable for recognizing partially explored shapes. Volumetric-based methods, which instead construct data structures to represent the occupancy of a 3D grid and enable the use of 3D convolutions, have been surpassed by point-based networks [9]. Over the last few years, PointNet has been influential in deep-shape recognition as several point-based networks incorporate it for shape-feature extraction [14] as well as for creating encodings for generative adversarial networks [15]. Although a dynamic graph CNN [13], an extension of PointNet, would also be a good choice, our choice fell on the latter as it is more computationally efficient because it does not require the computation of graph structures in latent space. In summary, current works mainly use tactile sensing alone and require slow-to-collect tactile datasets, without exploiting the readiness of visual data. Existing works tackle the problem by defining handcrafted descriptors. Paganoni et al. [11] instead studied partial point cloud recognition but derived samples from the ModelNet40 dataset, which is a high-quality and low-noise CAD dataset. They also analyzed the performance of PointNet under noise without attempting to improve it. However, as explained in the "Problem Definition" section, noise and uncertainties, due to representation differences, are core to the issue of VTR. Furthermore, in contrast to the partial point clouds that would be generated during a tactile exploration, the work in [11] used simulated laser scans and photogrammetry from single viewpoints. On the contrary, the tactile point clouds may be composed of sparse and unconnected clusters of points collected from any given surface of the object. This study explores decomposing whole point clouds into patchy partial samples to more closely resemble data gathered from a series of tactile interactions with an object.

The main contributions of this article are

- a data-driven pipeline capable of recognizing objects from partial tactile observations, and the experimental evaluation of this pipeline on our collected dataset.
- the CMR of objects represented by points distributed over a nonplanar manifold. This represents a challenge because, as explained in the "Problem Definition" section, it is not always possible to obtain a complete tactile model of nonplanar objects. Therefore, an architecture capable of performing predictions based on partial information is required in this scenario.
- a CL pipeline that encourages task-specific descriptors.

The remainder of this work is structured as follows. An overview of the problem statement and design requirements is given in the "Problem Definition" section. The "Methodology" section proposes the system architecture and presents a curriculum training procedure. The "Experiments" section presents details on the experimental setup, datasets, and data processing. Finally, the "Results and Discussion" and "Conclusion" sections conclude the article.

Problem Definition

Let $\mathcal{O}: \{O_l \mid l = 1, \ldots, L\}$ be a set of L known objects. The single object $O_l \in \mathcal{O}$ is represented using two distinct modalities: visual and tactile. In the specific case, visual information is acquired using a red, green, blue-depth (RGB-D) camera; conversely, tactile data are collected using a tactile-sensing array integrated on a robot's end effector. Both visual and tactile information can be represented as a point cloud, assuming the position of each tactile element is known with respect to a common reference frame. Therefore, when the robot's end effector gets in contact with the object, it is possible to associate a small point cloud (whose size is related to the number of sensors composing the tactile array) to a specific position in the space. This assumption is not specific for the tactile-sensing technology used in this article (see the "CySkin and Collection of Tactile Data" section); indeed, it holds for any tactile sensor composed of a set of independent transducers.

More formally, let us denote with $\mathbb{V}: \{V_i \mid i = 1 \ldots M, \mid V_i \mid = m_i\}$ the visual dataset containing the visual point clouds for all objects $O_l \in \mathcal{O}$ and, in the same fashion, $\mathbb{T}: \{T_j \mid j = 1 \ldots N, \mid T_j \mid = n_j\}$ containing the tactile point clouds. Here, $\mathbb{V}$ and $\mathbb{T}$ represent the objects $\mathcal{O}$ as sensed through a vision and a tactile sensor, respectively, where each point in the point clouds is expressed in Cartesian coordinates. Let us also define two functions, $c_V(\cdot)$ and $c_T(\cdot)$, which, taken visual and tactile point clouds, respectively, return the object instance O_l.

This article addresses the development of a data-driven model that can learn to recognize the specific object $O_l \in \mathcal{O}$ from a partial tactile point cloud $\overline{T}_j \subset T_j$, i.e., learn to approximate the mapping c_T and perform an estimate $\hat{O}_l$, from visual point clouds $\mathbb{V}$ and a known mapping c_V. Furthermore, we seek a solution that can operate under low amounts of training data $\mathbb{V}$, and specifically, one-shot learning.

To summarize, we design a system able to perform a tactile-based recognition under the following criteria:

- a partial representation of point clouds
- differences in sensor characteristics between visual and tactile point clouds
- few training samples per object.

Figure 1 presents examples of point clouds generated from visual and tactile data. The image shows the extent of the differences between the two representations: although the visual and tactile modalities both captured smooth faces and the overall shapes of objects well, the tactile point clouds (d)–(f) achieve lower surface coverage than vision (a)–(c). Furthermore, when considering nonplanar objects, a full tactile representation cannot be always reconstructed as some parts are not reachable (e.g., the inner surface of the tape reel in Figure 1(f). We remark that the differences shown in Figure 1 are not specific to the tactile-sensing technology described in the "CySkin and Collection of Tactile Data" section and used in this article. Indeed, as there are no standards at the hardware level [16], the spatial resolution and distribution of tactile-sensing elements change depending on the adopted technology. This article directly addresses the modality differences: although point clouds can represent both visual and tactile data, in general, their differences, as shown Figure 1, can affect the overall performances of the recognition system.

Furthermore, as the tactile-based exploration of the whole object is a time-consuming process, we aim at recognizing the object from partial information by defining a process where recognition accuracy improves as more parts of the objects are sensed. Figure 2 depicts an example of a partial tactile point cloud. Although beyond the scope of this article, this will enable the possibility of performing touch-based predictions online, thus avoiding the whole exploration of the object.

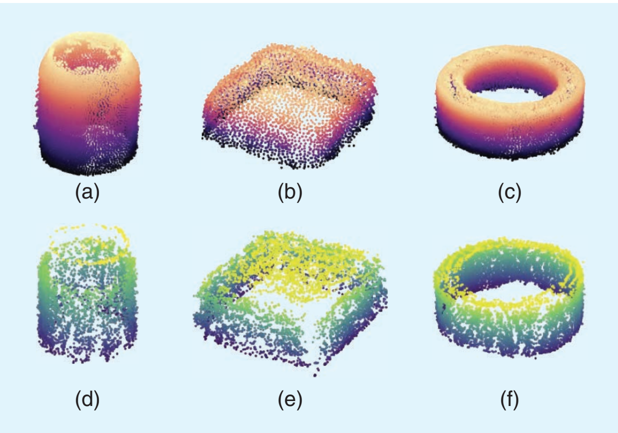

Figure 1. The top row shows (a) the raw visual point clouds of a beer can, (b) a Rubik's cube, and (c) tape. On the bottom row, (d)–(f) represent the corresponding tactile point clouds. Plots (b) and (e) highlight that the tactile point clouds failed to capture edges. Further, some surfaces captured in the visual clouds are not present in the tactile clouds, such as the inner surface of the tape reel, plotted in (c) and (f). The recognition system will need to be robust to such missing edges, manifolds, and surfaces in addition to varying densities and qualities of the point clouds.

Methodology

To address the problem of recognizing an object from its partial tactile representation, we form a custom CL procedure (see the "CL" section), which exposes the network to synthetic partial point clouds. We hypothesize that this curriculum will encourage learning partial point cloud embeddings and local descriptors, which will benefit online recognition. Furthermore, similar to [5], we apply point cloud filters to cope with the differences between tactile and visual representation.

Figure 3 depicts the whole procedure. The resolution of the difference in representations is tackled with preprocessing of point clouds to a unified representation, as proposed in [5], which used the same point cloud filters on both visual and tactile data. Different from [5], we incorporated a DL network, PointNet [10], to extract meaningful shape features and make predictions using the network's multilayer-perceptron output layers. We trained PointNet with the augmented vision dataset $\mathbb{V}$ to learn the model parameters then used in the prediction stage, where PointNet takes filtered tactile point clouds $\mathbb{T}$ to infer the corresponding objects from $\mathcal{O}$. This augmentation, discussed in the "CL" section, is a part of the broader CL strategy to recognize partial representations.

Preprocessing of Data

To reduce the differences in spatial resolution and noise and achieve a unified visuo-tactile point cloud representation, we considered the approach taken by the authors in [5]. Their approach unified the representations in two stages: 1) denoising and reconstructing the surfaces using a moving least-squares (MLS) filter and 2) equalizing local densities using a voxel filter. The results of these filtering operations are exemplified in Figure 4.

MLS Filtering

First, the MLS filter explored in [17] was used to promote local smoothness and reconstruct model surfaces. This process has the observed effect of removing noise perpendicular to the surface of the objects, hence sharpening planar surfaces and edges. We chose surface reconstruction using tangent estimation over polynomial estimation as it was less prone to introducing curvature into flat surfaces.

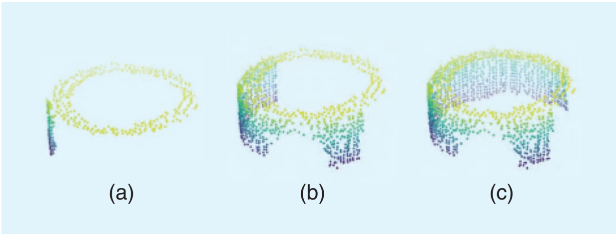

Figure 2. The system is required to recognize partial point clouds with increasing accuracy as the number of points in the point cloud increases. (a) 10% of points, (b) 40% of points, and (c) 70% of points.

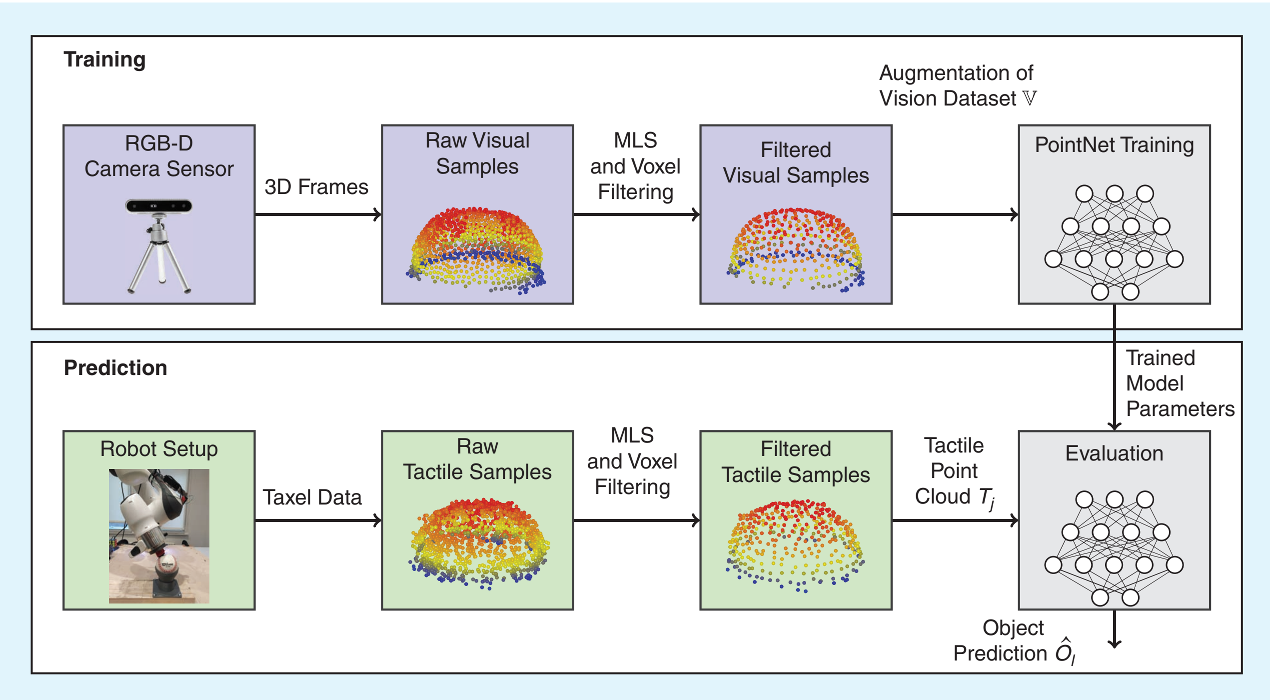

Figure 3. The system architecture. The top row depicts the training stages. A vision point cloud generated from an RGB-D camera is filtered using both moving least-squares (MLS) and voxel filters to reduce noise and equalize the spatial density of points. The vision dataset $\mathbb{V}$ is augmented by applying transforms to each point cloud V_i, as discussed in the "One-Shot Learning—Data Augmentation" section and used to train a PointNet. The bottom row depicts the prediction phase, where tactile samples are sensed and (optionally) filtered. PointNet learns to approximate the mapping c_T, forming from each tactile point cloud T_j a prediction of the object instance $\hat{O}_l$ it belongs to.

Voxel Filtering

Subsequently, we employed voxel filtering to address the homogenization of the local density of points. First, the voxel filter constructs a regular 3D grid around the point cloud. The resulting cubes are referred to as *voxels*. An efficient implementation of the algorithm loops through the point cloud, assigning to each point the index of the Voxel it is located within. The resulting cloud replaces all occupied voxels with the centroid of the voxel or the point closest to the centroid. Although more computationally expensive, we chose the latter option to avoid introducing spatial errors at the cost of slightly less uniformity. Both variants improve the spatial distribution of points by removing the local clustering.

CL

We argue that a curriculum, which trains with increasingly partial representations, can enhance the performance on representations typically observed during partial shape recognition. Global shape descriptors, such as unique signatures of histograms (USHs) [18], ESF [19], and CLUE [5], do not offer such routes to conditioning a network for partial shape recognition as the training of a single k-nearest neighbor or support vector machine classifier (used in [5]) is not an iterative process.

CL has been shown to improve NN performance and training efficiency [12] and takes inspiration from how humans learn. Initially, we train the network on easy examples before bringing the focus to more challenging tasks. In practice, this learning approach employs a scoring function, which assigns to each sample a score based on its difficulty, and a pacing function, which determines when more complex samples are introduced to learning. This study approaches the problem by synthesizing point clouds at difficulty stages, defined as the sparsity of any specific object's point cloud; we start with full coverage and swap them in the training pipeline according to a pacing function, i.e., in our case, at predetermined epochs.

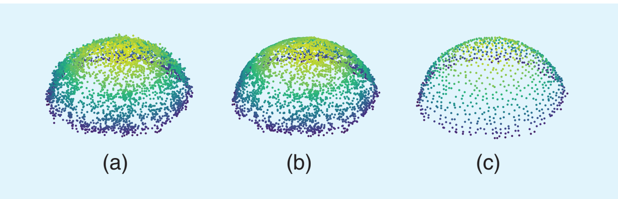

Figure 4. An MLS and a voxel filter applied to a tactile baseball point cloud. (a) Raw, (b) MLS, and (c) MLS and voxel.

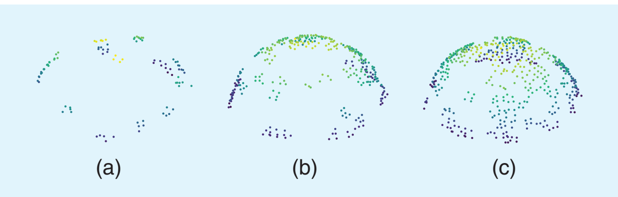

Figure 5. A k-means sampling: point clouds are divided into K disjoint partitions. A partial sample of proportion p is generated by the union of randomly-sampled P_k subsets. Here we depict the cases of $p = 0.1$, $p = 0.4$, and $p = 0.7$. (a) $p = 0.1$, (b) $p = 0.4$, and (c) $p = 0.7$.

To perform CL, we require a means to create samples of different difficulties. As the goal is to improve the accuracy of partial point cloud recognition, we hypothesize that exposing the network to partial representations will achieve this. Therefore, we score the full point clouds from our visual dataset as the "easiest" and synthesize various degrees of partial, more sparse point clouds from them to form our "harder" samples. Let's consider the generation of a partial point cloud from our full point cloud P. Here, we generalize for a given P as we apply this process to any $V_i \in \mathbb{V}$ or $T_j \in \mathbb{T}$ for training and evaluation purposes, respectively. We first apply a partitioning function $\gamma(\cdot, K)$ to generate K disjoint partitions of P, P_k (i.e., there are no common elements) such that $P = \bigcup_{k=1}^{K} P_k$. We implemented such a function γ using k-means clustering in Euclidean space. Given a desired cluster size λ, i.e., the average number of points belonging to a cluster—the number of clusters, K, can be computed as $\lfloor m_i/\lambda_V \rfloor$ or $\lfloor n_j/\lambda_T \rfloor$, for $V_i \in \mathbb{V}$ and $T_j \in \mathbb{T}$, respectively.

Finally, let $\tau(\cdot, p)$ be a partitioning function that can be applied to a generic point cloud P to subsample it into $\overline{P}_p \subset P$, where $p \in [0,1]$ is the proportion of sampled points. We can implement τ by randomly selecting p of the subsets P_k and combining them into the partial point cloud $\overline{P}$. Random selection was made by sampling a discrete uniform distribution without replacement; the results of this selection process can be seen in Figure 5 for the baseball point cloud in case of $p = 0.1$, $p = 0.4$, and $p = 0.7$. In this manner, we simulate many different pseudorandom partial representations of an object from a single point cloud. The number of possible different representations for each object we can generate with this method is given by the binomial coefficient $\binom{K}{\lfloor pK \rfloor}$. Furthermore, by lowering p, we can generate sparser and sparser, i.e., harder and harder, samples for the CL pipeline.

Experiments

The objects chosen were everyday household items inspired by the range of items used in previous Amazon picking challenges. A total of 10 objects were used, i.e., $|\mathcal{O}| = L = 10$. The objects can be seen in Figure 6. Each object resides in three approximate shape classes: cuboidal, cylindrical, and ellipsoidal. The selection contained some geometrically similar shapes, such as a golf ball and baseball, and different materials, such as metallic, plastic, or cardboard, which also have vastly different surface textures and reflective properties. The selection of objects was limited to rigid, nondeformable items.

CySkin and Collection of Tactile Data

The robot setup consisted of a seven degree-of-freedom Panda by Franka Emika industrial manipulator, with a CySkin module attached to the gripper. CySkin is a capacitive-based tactile-sensing technology. The sensor patch integrated into the end effector used in the experiment included seven tactile elements arranged as shown in Figure 7(b). Each tactile element (taxel) has a 3.5-mm diameter and provides a

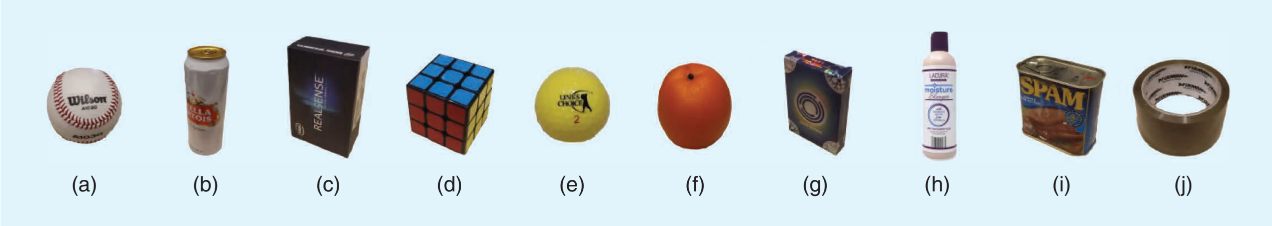

Figure 6. The 10 objects in our datasets. (a) A baseball, (b) beer, (c) a camera box, (d) Rubik's cube, (e) golf ball, (f) an orange; (g) a pack of cards, (h) shampoo, (i) Spam, and (j) tape.

16-bit measurement, which is related to the contact pressure. The pitch among adjacent taxels is 7.5 mm. A single contact between the end effector and the object activates one or more tactile elements when the response of the taxels exceeds a threshold value. The tactile point cloud is subsequently created by registering the 3D position of the active tactile elements with respect to the robot's reference frame.

The collection procedure began with fixing each of the 10 objects, in turn at a known position in front of the robot [see Figure 7(a)]. Next, an operator manually guided the robot to touch the object with the sensorized end effector. The robot was manually moved in small steps, incrementally exploring the surfaces until reaching full coverage of the object. As the goal of the procedure was to cover the whole surface of the objects thoroughly, the operator mainly applied small movements to the end effector. This approach biased the touches toward the local areas, in contrast to exploration strategies that make larger movements or seek to maximize the information gain per touch.

Some surfaces of the objects were not reachable due to the limits of the setup. These limitations are the following: 1) the planar tactile-sensing technology was not able to capture tight manifolds with an extreme concave curvature, such as the rim of the beer can; 2) it was not possible to explore the inside surface of the reel of tape as the dimensions of the end effector were larger than the confined space; 3) the lower surfaces of the spherical objects (baseball, golf ball, and orange) and the aspects covered by the jig fixing the objects in the environment were obstructed from exploration, resulting in hemispherical representations.

The explorations were repeated three times and resulted in 30 tactile point clouds, which compose the dataset $\mathbb{T}$. The point clouds ranged from 624 points for the golf ball up to 3,946 for the camera box (the dataset is provided as supplementary material; please see the "Acknowledgments" section).

In this work, the problem of performing an autonomous exploration of the objects is not considered. This would add additional challenges that are outside the scope of this work, which focuses on partial recognition. Indeed, an exploration procedure based on tactile sensing depends on the end-effector type, object type, and its relative position to the robot. Furthermore, nonplanar objects increase the complexity as some parts can be difficult or impossible to reach. Moreover, beyond the additional challenges related to control aspects, a proper exploration strategy is key to minimizing the number of contact interactions required to recognize the object with high confidence. These two aspects are currently beyond the consideration and scope of this work. Instead, this article focuses on analyzing the data to evaluate the possibility of recognizing the objects from partial representations. What is developed in this article will be used in a future extension of the method to design a proper exploration strategy driven by the confidence obtained when processing partial tactile point clouds.

Collection of Vision Data

An Intel RealSense D415 RGB-D sensor was used to capture successive images of each object. The algorithm used to stitch together frames was adapted from the Point Cloud Library in-hand scanner application. The algorithm has three distinct stages for adding a point to the cloud: 1) input preprocessing, including cropping of the input RGB-D points to a sufficiently small volume to include only the turntable and object to be scanned; 2) registration using an iterative closest point algorithm; and 3) integration, i.e., the decision-making module responsible for accepting or rejecting points into the point cloud. We rotated the objects during the registration process using a turntable to achieve 360° surface coverage.

The collection of the visual data, $\mathbb{V}$, was repeated twice to gather two sample point clouds for each object: one for training and another for validation. Subsequently, we manually removed the background from each visual point cloud and aligned its z-axis with the normal alignment of the tabletop. The total number of points in the vision point clouds ranged from 619 for the Rubik's cube, to 10,980 for the camera box.

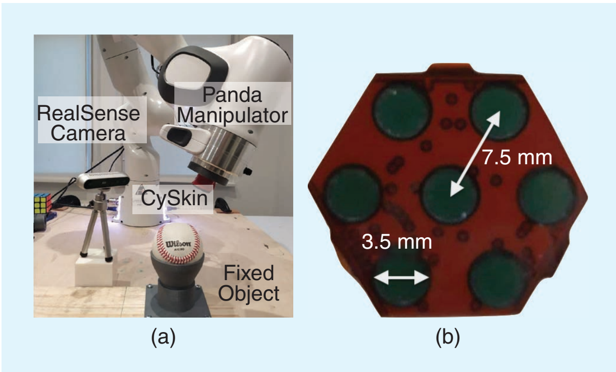

Figure 7. (a) The robot setup as seen with the Intel RealSense and the Panda Manipulator poised in front of the baseball, secured by a jig. (b) The CySkin patch of seven taxels fixed to the end effector. Annotated is both the pitch (the distance between taxel centroids) and the diameter of each taxel.

Curriculum Samples and Scheme
For the curriculum scheme we create a dataset of partial point clouds $\mathbb{V}_{2000}$, as described in the "CL" section, to create 2,000 samples for each proportion $p \in \{0.05, 0.1, \cdots, 1.0\}$. For the partitioning, we create a cluster size $\lambda_V = 10$. The value was elected empirically: a small value leads to clusters containing a small amount of information, and a high value reduces the number of possible partial point cloud we can generate. We decided to select it such that the vision clouds formed at least 40 clusters, considering that the smallest filtered vision point cloud is the golf ball, with 408 points. As the voxel filter equalizes the spatial density, the clusters roughly represent equal areas across models. The curriculum reduced the proportion p in later epochs; specifically, the network was trained with $p \in \{1.0, 0.7, 0.4, 0.2\}$ for epochs [1–40, 40–55, 56–70, 71–75], respectively.

Partial Tactile Point Clouds
Similar to the vision, the dataset $\mathbb{T}$ was decomposed to create a dataset of partial point clouds, $\mathbb{T}_{2000}$, containing partial tactile representations of the objects at different proportions p. A value of $\lambda_T = 7$ points was chosen for a cluster size similar to the end-effector taxel number. In this way, we considered each patch P_k to be generated by a single touch. Therefore, the resulting partial point clouds are generated randomly by composing small sets of clusters, each representing a small surface comparable to the size of the CySkin sensor.

One-Shot Learning—Data Augmentation
In addition to the training curriculum, we applied data augmentation techniques at runtime to both $\mathbb{V}$ and $\mathbb{V}_{2000}$. As the number of visual models was only one training point cloud and one validation point cloud per object, the challenge of training the network can be viewed as few-shot learning, or loosely as one-shot learning [20]. As is common in few-shot learning, the training datasets were augmented with additional samples derived from the original point clouds to reduce the risk of overfitting to the visual training set.

Data augmentation was performed according to the transforms in Table 1. For this specific implementation, the input space of PointNet was fixed to β = 1,024 as during training, samples need to be of equal length when grouped into batches. Then, the first step, which was also the first introduction of variation, was the uniform selection (without replacement) of β points from the clouds of shape $(\alpha, 3)$. For point clouds with fewer than 1,024 points, duplicate points were added to pad up to the correct shape. Subsequently, zero-mean Gaussian noise with a standard deviation of 0.5 mm, $\mathcal{N}(0, 0.5^2)$, was added to each point and the resulting point cloud rotated by a random 3D Euclidean transform. By exposing the network to random orientations, we encourage the network to learn pose-agnostic embeddings. Finally, a random scaling was applied from the uniform distribution $U(0.95, 1.05)$. Overall, this training scheme is designed to introduce variance through random scaling, additive noise, and random rotations to reduce the likelihood of overfitting or memorizing the arrangements of the points.

For clarity, the transforms used during validation and evaluation are shown in Table 2. Additive noise and scaling were never applied to the visual validation or tactile evaluation stages.

Table 1. Visual data augmentation during training.

Step	Transformation	Dimensions
0	Input point cloud	$\alpha \times 3$
1	Random selection of β points	$\beta \times 3$
2	Additive white Gaussian noise	$\beta \times 3$
3	Random 3D rotation	$\beta \times 3$
4	Random scaling by scale factor in the interval [0.95,1.05]	$\beta \times 3$

Table 2. Visual data preprocessing during validation.

Step	Transformation	Dimensions
0	Input point cloud	$\alpha \times 3$
1	Random selection of β points	$\beta \times 3$
2	Random 3D rotation	$\beta \times 3$

Training and Evaluation
We follow the training of PointNet as in [10], with cross-entropy loss and an Adam optimizer with a learning rate of 0.01. We trained the network for each experiment five times and present the run that performed best on the vision validation set. Our data pipeline, including the training and evaluating PointNet, took approximately 20 min using a GTX 1070 graphics card. We selected the filtering parameters with a grid search evaluated on visual monomodal accuracy. In the end, we selected an MLS radius of 5 mm and a leaf size of 3.5 mm for the voxel filter. To be clear, the tactile dataset always remained unseen during training, evaluation, and trained-model selection.

Results and Discussion
This section evaluates the ability of the pipeline to recognize degrees of partial tactile point clouds and improve using the point filters and proposed CL procedure. In this analysis, we also evaluate the performance of the system in terms of inference time. Furthermore, we simulated an online exploration of the objects performed as described in the "Experiments" section when collecting the data. This experiment is useful to 1) assess whether the proposed method is suitable for online recognition and 2) show how the exploration strategy affects performance. Finally, a comparative experiment with descriptor-based methods is proposed to benchmark our approach.

Partial Point Cloud Recognition
The first experiment focused on the effects of the filters and learning scheme on partial point cloud recognition. We

evaluated three configurations: 1) a vanilla training scheme without point filters, 2) a vanilla training scheme with point filters, and 3) CL with point filters. The vanilla and the curriculum training schemes used different datasets, $\mathbb{V}$ and $\mathbb{V}_{2000}$, respectively. The system performance was measured against the tactile partial point clouds in the $\mathbb{T}_{2000}$ dataset. It should be noted that although three tactile samples were collected for each object, when evaluating the network on partial point clouds, we applied the sampling scheme described in the "CL" section. Therefore, when $p \in [0.05, 0.9]$, the models are evaluated on $\mathbb{T}_{2000}$, which consists of 2,000 unique point clouds for each object and each proportion p. On the contrary, when $p = 1$, the models performed predictions on a test set of three samples for each object.

The results, as presented in Figure 8, show that the vanilla-trained network without filters performed the worst, and on full representations recognized objects with a baseline accuracy of 77%. The inclusion of filters in the vanilla scheme improved the performance on whole point clouds by 16%, reaching an accuracy of 93%. The training curriculum further boosted performance, reaching 100% accuracy on full point clouds and $p \geq 0.6$. Across the range of proportions, the vanilla scheme with filters outperformed the baseline by an average of 11%, and the curriculum bettered that by an additional 6%. For $p \geq 0.2$, the curriculum-trained network performed with an accuracy of at least 80%. For all proportions $(0.05 \geq p \geq 1)$, the curriculum scheme produced the best object-recognition performances from the tactile point clouds, especially for $0.6 \geq p \geq 1$.

It is also worth noting that the curriculum strategy is the only one capable of reaching 100% accuracy. Indeed, as previously discussed, when considering nonplanar shapes, some of the objects cannot be fully explored. Therefore, even when considering the full tactile point cloud (i.e., $p = 1$), the system has to make predictions using partial data as some surfaces are missing in the tactile representation compared with vision. As a result, the vanilla networks cannot reach 100% accuracy as they were not specifically trained on partial representations, further showing the effectiveness of the curriculum strategy applied to CMR.

Inference Time

To gauge whether the system has the potential to be used online, we measured the inference time of the network, which we defined as the time taken for a prediction to be computed for a single point cloud. We included in the measurement voxel filtering, MLS filtering, preprocessing operations, and forward pass of the NN. The experiment was performed on point clouds in the augmented dataset, which varied between 600 and 4,000 points. A batch size of one was used, as is desirable in an online recognition system, where the data are updated after each contact.

The pipeline was able to make a prediction in a mean time of 21.4 ms for a single point cloud, corresponding to a refresh rate of 47 Hz. We observed that point cloud filtering took almost half the time because the MLS and voxel filters were run on the CPU. An implementation on CUDA cores could likely speed up these operations significantly. The results are promising for running the pipeline on an online system.

Online Recognition

A separate experiment was performed to measure recognition accuracy of the system during an online exploration of an unknown object, using our 30 unpartitioned explorations $\mathbb{T}$. We wished to compute and improve the objects' predictions as the data became available. To this extent, we treated the tactile data, collected as described in the "Experiments" section, as an ordered sequence of samples, defining the exploration completion as follows: a 10% exploration contains the first 10% of points collected for that object as sequenced in time. The point filters were throughout and trained with both the vanilla and CL schemes, utilizing the $\mathbb{V}$ and $\mathbb{V}_{2000}$ datasets, respectively.

The accuracy results from testing on our tactile dataset of 30 samples are presented in Figure 9. The curriculum boosted performance by an average of 11% and outperformed the vanilla-trained network for all exploration completions excluding 5%. It is not unsurprising that at the extremely low exploration completion of 5%, the curriculum training did not make a difference; the data may not hold sufficient information for one object to be distinguished from another, no matter the recognition method used. For a 30% explored object, the curriculum improved recognition accuracy by 23%. Overall, the curriculum proved effective for online recognition.

Compared to the results shown in Figure 8, where, with only 20% of the samples the system was able to predict the classes with an accuracy of 80%, here, the network starts to provide acceptable results, i.e., an accuracy greater than 80%, with an exploration of 60%. This apparent drop in performance is an artefact of the

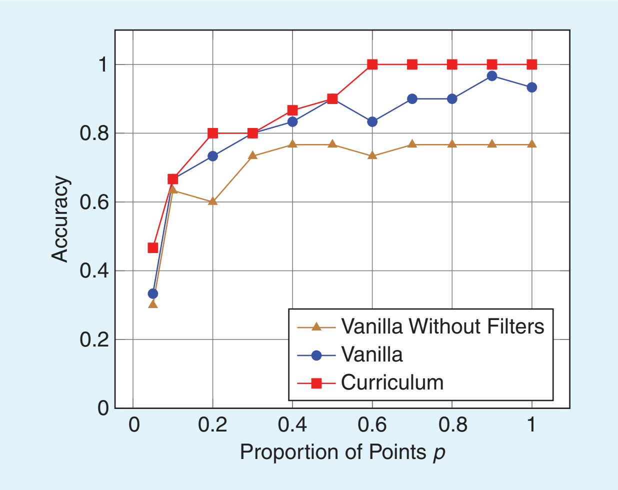

Figure 8. Performance of three configurations of the pipeline across a range of partial tactile representations, dictated by the proportion p. The three configurations are 1) a vanilla training scheme without filters, 2) vanilla scheme with filters, and 3) curriculum scheme with filters.

exploration strategy used to collect our data. As explained in the "Experiments" section, the data were collected by moving the robot's end effector in small steps, thus simulating an incremental exploration strategy. This meant that a low percentage of points representing the object could only capture local information. On the contrary, the random sampling procedure used in the "Partial Point Cloud Recognition" section, which resembles a random exploration strategy, obtains a broader geometric representation of the object shape using a low percentage of the points, easing the recognition process.

We remark that this is not a flaw of the proposed approach. These two experiments show that this method can be used effectively for cross-modal recognition of partial tactile point cloud. Furthermore, they make it clear that a proper exploration strategy is vital to efficient online recognition. This aspect will be investigated in a future extension of this work by considering methods based on information gain to work alongside the proposed DL and CL methods and minimize the number of contacts required to recognize the object.

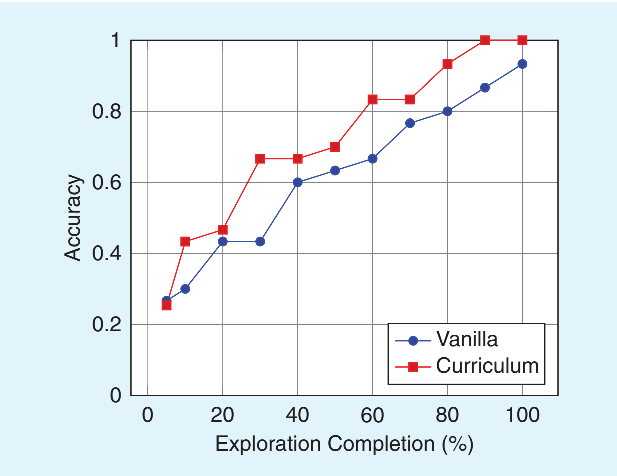

Figure 9. Performance of the classification pipeline when considering an ordered sequence of samples. The plot shows how the classification accuracy changes, as long as more tactile data become available.

Table 3. Cross-modal results using handcrafted descriptors computed on the full point cloud. (a) The results obtained with the filtering parameters used in [5] and (b) in this article.

	(a) MLS Radius=60 mm; Leaf Size=5 mm		(b) MLS Radius=5 mm; Leaf Size=3.5 mm	
	Three NN	SVM	Three NN	SVM
ESF	40%	53.34%	30%	73.34%
CLUE	20%	60%	23.34%	46.67%

SVM: support vector machine.

Comparison With Descriptor-Based Approaches

To compare the proposed approach with respect to the state of the art, we implemented two descriptor-based methods: an ESF and a CLUE [5]. As discussed previously, these descriptors are not suitable for partial point cloud recognition, and therefore, we evaluate them on the dataset of the full point cloud i.e., $p = 1$. The system was trained on the visual point clouds without data augmentation, as described in [5], and tested on the 30 tactile point clouds. We performed the classification with a three-NN classifier and an SVM (with a radial basis function kernel) in two different cases. In the first experiment, we applied the same values for the MLS and voxel filters used in [5]. In the second experiment, we applied the values we used in this article (see the "Training and Evaluation" section). As is visible from the results listed in Table 3, ESF and CLUE did not perform well in this scenario. We argue that there are two reasons for this. First, our dataset contains only one training sample for each object. Indeed, with respect to [5], we trained the network with a one-shot learning approach, using one visual sample for training and one for validation. The method in [5] does not consider this scenario, therefore, additional data could be required to train the model. Second, the authors in [5] consider only planar objects. As previously explained in the nonplanar case (as in general, the tactile point cloud cannot be fully explored), the system has to make a prediction on partial data even when the full tactile point cloud is considered. As discussed in the "CL" section, descriptor-based approaches are not suitable for partial recognition, therefore, they cannot be directly used for cross-modal recognition of nonplanar objects.

Conclusion

This study tackled VTR of the partial point clouds enabled by a CL procedure applied to only visual data and tested the network performance on our tactile dataset. We synthesized partial point clouds and proposed a curriculum that progressively introduced sparser samples to increase training difficulty at later epochs. We composed several partial tactile representations of the objects using our dataset and used these samples to benchmark the system.

The curriculum-trained network was able to perform recognition with an accuracy of 80% using only 20% of points generated from a random exploration strategy. The accuracy further increased to 100% on clouds using at least 60% of the data. In contrast, the vanilla-trained network averaged an accuracy of 90.7% for $0.6 \geq p \geq 1$. We benchmarked the inference time of the pipeline on our hardware, which refreshed predictions at a promising rate of 47 Hz.

We also treated the tactile data collected, as described in the "Experiments" section, as a time-ordered sequence of points and showed that CL improved the system's accuracy for online recognition, during exploration, by an average of 11%. Our experiments also highlighted the impact of the exploration strategy on overall system performance. A natural extension would study exploration strategies to minimize the number of contacts required to recognize an object with high confidence.

Acknowledgment

We gratefully acknowledge support from the Engineering and Physical Sciences Research Council program grant From Sensing to Collaboration (EP/V000748/1). This article has supplementary information downloadable at https://doi.org/10.1109MRA.2022.1003122204. Alessandro Albini is the corresponding author.

References

[1] H. Liu, Y. Wu, F. Sun, and D. Guo, "Recent progress on tactile object recognition," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 4, p. 1,729,881,417,717,056, 2017, doi: 10.1177/1729881417717056.

[2] H. Liu, F. Sun, B. Fang, and D. Guo, "Cross-modal zero-shot-learning for tactile object recognition," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 7, pp. 2466–2474, Jul. 2020, doi: 10.1109/TSMC.2018.2818184.

[3] S. Wang et al., "3D shape perception from monocular vision, touch, and shape priors," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2018, pp. 1606–1613, doi: 10.1109/IROS.2018.8593430.

[4] L. Rustler, J. Lundell, J. K. Behrens, V. Kyrki, and M. Hoffmann, "Active visuo-haptic object shape completion," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5254–5261, Apr. 2022, doi: 10.1109/LRA.2022.3152975.

[5] P. Falco, S. Lu, C. Natale, S. Pirozzi, and D. Lee, "A transfer learning approach to cross-modal object recognition: From visual observation to robotic haptic exploration," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 987–998, Aug. 2019, doi: 10.1109/TRO.2019.2914772.

[6] R. Blake, K. V. Sobel, and T. W. James, "Neural synergy between kinetic vision and touch," *Psychological Sci.*, vol. 15, no. 6, pp. 397–402, Jun. 2004, doi: 10.1111/j.0956-7976.2004.00691.x.

[7] J. T. Lee, D. Bollegala, and S. Luo, "'Touching to see' and 'seeing to feel': Robotic cross-modal sensory data generation for visual-tactile perception," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, vol. 2019, pp. 4276–4282, doi: 10.1109/ICRA.2019.8793763.

[8] P. K. Murali, M. Gentner, and M. Kaboli, "Active visuo-tactile point cloud registration for accurate pose estimation of objects in an unknown workspace," 2021. [Online]. Available: http://arxiv.org/abs/2108.04015

[9] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," 2019, *arXiv:1912.12033*.

[10] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.,* 2017, vol. 2017-Janua, pp. 652–660.

[11] S. Paganoni, E. Zappa, and S. Turrisi, "Classification reliability of 3D shapes using neural networks in case of partial and noisy models," in *Proc. I2MTC 2020 - Int. Instrum. Meas. Technol. Conf.*, pp. 1–6, doi: 10.1109/I2MTC43012.2020.9128469.

[12] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, "Automated curriculum learning for neural networks," in *Proc. 34th Int. Conf. Mach. Learn.,* 2017, vol. 3, pp. 2120–2129.

[13] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021, doi: 10.1109/TPAMI.2020.3005434.

[14] Y. Duan, Y. Zheng, J. Lu, J. Zhou, and Q. Tian, "Structural relational reasoning of point clouds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, vol. 2019-June, pp. 949–958, doi: 10.1109/CVPR.2019.00104.

[15] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. Mach. Learn.,* 2018, vol. 1, pp. 67–85.

[16] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing-from humans to humanoids," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1–20, Feb. 2010, doi: 10.1109/TRO.2009.2033627.

[17] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 1, pp. 3–15, Jan./Mar. 2003, doi: 10.1109/TVCG.2003.1175093.

[18] S. Salti, F. Tombari, and L. D. Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Comput. Vis. Image Understanding*, vol. 125, pp. 251–264, Aug. 2014, doi: 10.1016/j.cviu.2014.04.011.

[19] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *Proc. 2011 IEEE Int. Conf. Robot. Biomimetics*, pp. 2987–2992, doi: 10.1109/ROBIO.2011.6181760.

[20] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020, doi: 10.1145/3386252.

Christopher Parsons, Oxford Robotics Institute, University of Oxford, Oxford OX2 6NN, United Kingdom. E-mail: christopher.parsons@bnc.ox.ac.uk.

Alessandro Albini, Oxford Robotics Institute, University of Oxford, Oxford OX2 6NN, United Kingdom. E-mail: alessandro@robots.ox.ac.uk.

Daniele De Martini, Oxford Robotics Institute, University of Oxford, Oxford OX2 6NN, United Kingdom. E-mail: daniele.demartini@eng.ox.ac.uk.

Perla Maiolino, Oxford Robotics Institute, University of Oxford, Oxford OX2 6NN, United Kingdom. E-mail: perla.maiolino@eng.ox.ac.uk.