IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 3 October 2022, accepted 23 October 2022, date of publication 26 October 2022, date of current version 7 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217458

RESEARCH ARTICLE

Learning Physics Property Parameters of Fabrics and Garments With a Physics Similarity Neural Network

LI DUAN[1], LEWIS BOYD[2], AND GERARDO ARAGON-CAMARASA[1]
[1]School of Computing Science, University of Glasgow, G12 8RZ Glasgow, U.K.
[2]National Manufacturing Institute Scotland, University of Strathclyde, G1 1XQ Glasgow, U.K.

Corresponding authors: Li Duan (l.duan.1@research.gla.ac.uk) and Gerardo Aragon-Camarasa (gerardo.aragoncamarasa@glasgow.ac.uk)

The work of Gerardo Aragon-Camarasa was supported by the EPSRC under Grant EP/S019472/1. The authors would like to thank Ali AlQallaf and Ozan Bahadir for their constructive feedback and comments.

ABSTRACT Predicting the physics properties of deformable objects such as garments and fabrics is a challenge in robotic research. Directly measuring their physics properties in a real environment is difficult Bouman et al. (2010). Therefore, learning and predicting the physics property parameters of garments and fabrics can be conducted in simulated environments. However, garments have collars, sleeves, pockets and buttons that change how garments deform and simulating these is time-consuming. Therefore, in this paper, we propose to predict the physics parameters of real fabrics and garments by learning the physics similarities between simulated fabrics via a Physics Similarity Network (PhySNet). For this, we estimate wind speeds generated by an electric fan and area weights to predict the bending stiffness parameters of real fabrics and garments. We found that PhySNet coupled with a Bayesian optimiser can predict physics property parameters and improve state-of-art by 34.0% for fabrics and 68.1% for garments.

INDEX TERMS Physics similarity map, physics similarity distance, Bayesian optimization, deformable objects.

I. INTRODUCTION

Robot perception and manipulation of deformable objects remain a key challenge. Due to the object's complex geometric configurations and random deformations, a three-step process is usually adopted. The first step consists of modelling the objects in a simulated environment [2], [3] or using finite element methods (FEM) [4]. Then, the second step is about learning deformations of the object in the simulated environment while the object is manipulated [5], [6]. The final step comprises finding an optimised trajectory for manipulating the object [7], [8]. In these three steps, the challenge is to learn the stress-strain curve of these deformable objects [9] which depends on the physics property parameters of objects such as stiffness, area weight and damping factors. Therefore, learning the physics property parameters of deformable objects is key to enabling a robot to perform dexterous manipulation of deformable objects.

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar.

Previous approaches that estimate physics property parameters of materials consist of either learning physics property parameters by aligning simulated models with real objects [10], [11], [12], or learning physics property parameters from video frames [1], [13], [14]. The former approaches require high accuracy in aligning the objects using finite element methods, which are computationally expensive, while the latter approaches do not need simulated models that match the real objects. Therefore, learning from video sequences is computationally efficient and can be deployed in a robotic system where a robot can apply an external force on deformable objects.

For robotic deformable object manipulations, the physics property parameters of deformable objects are linked to the deformation patterns of manipulated garments. For example, stiffer garments tend to bend less than softer garments, and softer garments have more complex states than rigid garments. Suppose we assume that a robot has prior knowledge of the physics property parameters of garments. In that case, it can use these parameters to fine-tune a manipulation

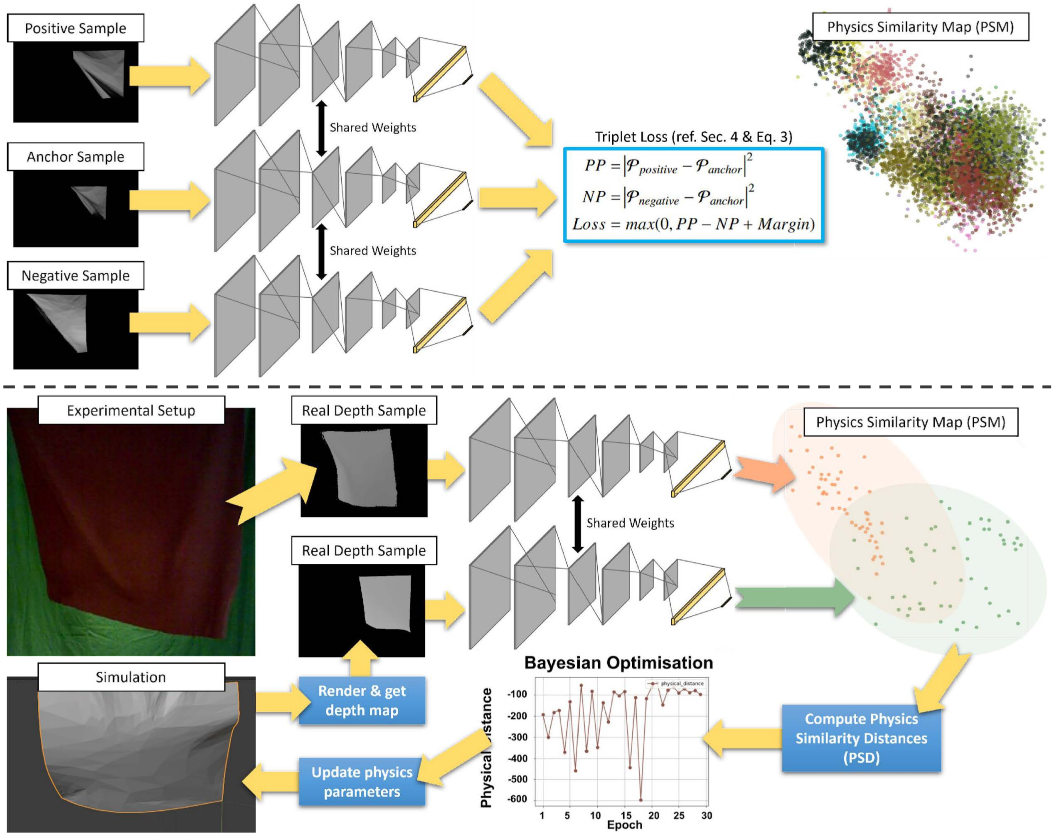


FIGURE 1. (*Top*) A triplet of images (an anchor, a positive and a negative) are input into PhySNet, and a Triplet loss function (Eq. 3) is used to learn whether fabrics in the images have the same or different physics property parameters. After training, PhySNet generates a Physics Similarity Map (PSM), where fabrics with similar physics property parameters have smaller Physics Similarity Distances (PSD, ref. Eq. 2). (*Bottom*) We use an Xtion camera to capture real fabrics, and we use an electric fan to exert a force onto the fabrics. Depth images of a simulated fabric with initial physics property parameters (defined in section V-B1) and the depth images of a real fabric are input into the trained PhySNet to have them mapped on the physics similarity map. Their PSDs are calculated from the map and are input into a Bayesian optimiser (Section IV-B). The Bayesian optimiser outputs optimal physics property parameters, which are used to generate a new simulated fabric. This loop iterates until the differences between optimal physics property parameters of the last three iterations are less than 10%.

plan, making the garments' manipulation effective. This paper proposes to learn the physics similarity between simulated and real fabrics. For this, we have implemented a Physics-Similarity Neural Network (PhySNet; inspired by [14]), as shown in Fig. 1, to predict real fabric physics parameters from simulated fabric physics parameters. The core idea here is that measurements of physics property parameters are difficult to obtain in a real environment without needing specialised equipment. For example, Bouman et al. [1] obtained fabric's stiffness parameters experimentally using specialised devices and designed a neural network architecture to regress stiffness parameters. Therefore, taking advantage of simulation software, where the physics property parameters can be obtained, can avoid the challenge of obtaining them in real environments. Hence, *is it possible to leverage a simulation engine to estimate physics parameters with a neural network when a wind force field is applied to the fabrics?*. To answer these questions, we have compiled a simulated fabric database to allow PhySNet to learn the physics similarity of simulated fabrics and to generate a Physics Similarity Map (PSM) for a fabric. After we train PhySNet, a piece of real fabric and a simulated fabric with initialised physics parameters are input into PhySNet to get their a Physics Similarity Distance (PSD). We then input the PSD into a Bayesian optimiser, which outputs updated physical parameters. We input the updated physics parameters into the simulator to generate a new simulated fabric. This procedure iterates until stable parameters (Section V-B1) are obtained from the Bayesian optimiser.

PhySNet is trained on simulated data to learn physical similarities between simulated fabrics. During testing, real garments are input into the PhySNet, and we match simulated fabrics with optimal physics distances. In this paper, the challenge in the target task is to predict the physics properties of real garments via learning physics similarities between simulated fabrics. Garments can not be easily simulated in simulation engines (ArcSim and Blender) because they have components such as collars, pockets and buttons. Simulating garments is time-consuming and computationally costly. This paper proposes that predictions of the physics properties of garments can be completed by simulating fabrics and learning physics similarities between simulated fabrics. This knowledge can then be used in a downstream task such as folding. The novelty of this paper is that *we test on the real garments that do not appear in the training database, indicating that our PhySNet is tested on unseen garments.* The contributions of this paper are threefold:

1) We propose that physics property parameters of real fabrics and garments can be predicted from learning physics similarities between simulated fabrics;
2) We propose that predicting the physics property parameters of real complicated objects, such as garments, can be achieved from learning physics similarities between simple simulated objects such as fabrics;
3) We demonstrate that learning physics property from depth images outperforms learning them from RGB images.

II. RELATED WORK

Previous research on the physics property parameters of deformable objects can be further divided into four categories: (i) using simulation models of objects to fit real models [12], [15]; (ii) learning model-free shape transformations given initial and goal object configurations [16], [17]; (iii) applying external forces and observing shape changes [11], [18]; and, (iv) learning dynamic characteristics from videos [1], [13], [19] by using knowledge learned from dynamic characteristics of simulation models on real models [14].

Tawbe et al. [15] proposed simulating sponges through a neural gas fitting method [20] rather than simulating meshes. They learnt and predicted the shapes of deformable objects without prior knowledge about the objects' material property parameters by applying the neural gas fitting on simplified 3D point-cloud models. These 3D point-cloud models focused on the parts of an object that had been deformed to improve learning. Their approach required a multi-step learning process to simplify the models and find the deformed parts. However, this approach was tested only on objects with simple geometries. Similarly, Arriola-Rios et al. [11] suggested learning materials of sponges by using a force sensor mounted on a finger in a robot gripper. The finger pressed a sponge to measure the applied force, which was then used to learn the material property parameters and to predict the sponge's deformation. Wang et al. [18] proposed learning external robot-exerted forces applied on objects. For this, they devised a Generative Adversarial Network (GAN) to predict their deformed shapes and combine the objects' visual shapes (depth images) and the force applied to the objects. Both [11] and [18] considered learning from both the deformations of objects and the exerted forces on objects because exerted forces are an essential indicator of the physics property parameters as defined by the slope of the strain-stress curve [9] of the deformable objects. Therefore, learning physic property parameters means learning the relationship between strain (deformations of objects) and stress (exerted forces).

Guler et al. [17] also aimed to learn the deformation of soft sponges, but they proposed a Mesh-less Shape Matching (MSM) approach, which comprises learning linear transformations between deformed objects. Similar to [17], Simeonov et al. [16] proposed that deformable objects can be manipulated by representing objects using cloud points rather than object models and calculating manipulating motion plans to estimate transformations between the object initial and goal configurations. Model-free physics property and deformation learning do not require learning actual object physics property parameters but *conceptualising* how objects can be deformed when an external force acts into the object. The above methods are, however, constrained to regular patterns of shape changes.

Bouman et al. [1] proposed to learn the physics property parameters of fabrics from videos. Bouman et al. focused on fabric stiffness, and their approach consisted of learning statistical features of the image's frequency domain of fabric videos and using a regression neural network to predict the stiffness parameters of fabrics. Similarly, Yang et al. [13] proposed predicting the physics property parameters of fabrics by learning the dynamics of fabrics from videos using a CNN-LSTM network architecture. However, these methods are constrained to fabrics with regular shapes, while our approach extends to garments with irregular and complex shapes.

Wang et al. [12] proposed reparameterising the stiffness of fabrics as a piecewise linear function of the fabrics' strain tensor. That is, they sampled the strain tensor with principle strains (maximum and minimum normal strains) and strain angulars, combined as a matrix of 24 parameters for stretching stiffness (i.e. resistance when fabrics are stretched) and 15 parameters for bending stiffness (i.e. resistance when fabrics are bent). To measure the stiffness of the fabrics, they opted for a FEM approach that aligned simulated meshes with the fabrics. They considered that stiffness is nonlinear, making simulations and stiffness measurements more accurate. However, the FEM method requires considerable time to compute accurately the deformation of objects which limits this approach's applicability to real-time robotic manipulation.

Learning from simulated objects to predict the physics property parameters of real objects has been proposed by Runia et al. [14]. They learnt physics similarity distances between simulated fabrics and predicted physics property parameters of real fabrics, where they decreased physics similarity distances between real and simulated fabrics by fine-tuning parameters of simulated fabrics via a Bayesian optimiser. Their approach paved the way for a novel alternative that frees a network from complicated simulation-reality approximations such as [12] and extends to regular shape fabrics, of which deformations are more complex, e.g. [11], [15], [18]. Our approach is similar to [14], but we propose to use depth information to learn the dynamics of fabrics and garments from their depth images and opt to use a triplet loss function instead of a pair-wise contrastive loss. Compared with [14], where they only used one material, our proposed pipeline can predict the physics property parameters of seven fabric materials and three garments.

Wu et al. [21] proposed learning newton properties (mass, density, etc.) of rigid objects (cubes) from unlabelled video sequences. They constructed a database that contains video sequences of objects in different scenarios: sliding down an inclined surface, attached to a spring and falling onto various surfaces. An unsupervised representation learning model has been introduced to learn the newton properties of the objects. Their work demonstrated the effectiveness of learning physics (newton) properties from videos, which is the focus of this paper. However, they only tested rigid objects, while in this paper, we focus on deformable objects of fabrics and garments.

Learning the physics properties of objects can also be achieved from learning 3D images. Gao et al. [22] introduced a TreeVes-Net to learn blood dynamics from CT angiography images (3D images). They proposed a tree-structured recurrent neural network (TreeVes-Net) that learns the bloody dynamics to diagnose myocardial ischemia. Their paper revealed that learning physics properties could be completed by learning object dynamics from 3D images.

In this paper, we have used the 'ArcSim' simulator [23] to simulate garments instead of using a finite element method for modelling fabrics and garments. We opted for ArcSim because it has been experimentally validated on ground truth obtained by mechanically modelling the stiffness-strain relationship of garments. Our experiments do not focus on investigating the mechanical aspects of fabrics and garments. That is, we do not measure the displacement between each pixel of the fabric/garment from one state to another. Instead, we focus on the physical similarity between simulated and real fabrics/garments and use this similarity to adjust the physics parameters of simulated fabrics. The information shared between simulated and real fabrics/garments is only the physics-property parameters being optimised as discussed in Section III and IV.

III. FABRIC PHYSICS PROPERTY PARAMETERS

The relationship in bending stiffness between strain and stress, as given by [12], is:

$$F = k_e sin(\theta/2)(h_1 + h_2)^{-1} |E| u \tag{1}$$

where F is the external force, and k_e is the material's bending stiffness. Figure 2 shows a visualisation of Eq. 1. In Figure 2, triangles 123 and 143 represent two faces of a piece of fabric where a force is applied to bend the fabric from triangle 123 to triangle 143. h_1 and h_2 are the normals of the two triangles, while E is an edge vector of the edge 13, which is shared by both the triangles 123 and 143. u is a bending model

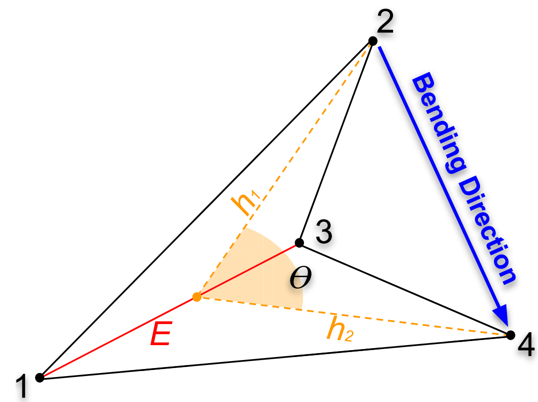


FIGURE 2. Bending stiffness: h_1 and h_2 are normals of the triangles 123 and 143, E is the edge vector of the edge 13.

described in [12]. In Eq. 1, Wang et al. [12] treated the bending stiffness, k_e, as a linear piecewise function of the reparametrisation $sin(\theta/2)(h_1 + h_2)^{-1}$. To estimate bending stiffness, the *bending angle*, θ (in Fig. 2), are set to 0°, 45° and 90°. For each value of θ, the bending stiffness is measured five times. These five measurement points represent bending behaviours of a piece of fabric in [12] experiments. Therefore, there are 15 points represented by a matrix of size 3×5 (angles $\times$ bending measurement points). We represent our predicted bending stiffness of the fabrics using this matrix representation (e.g. Figure 4).

Bending stiffness is difficult to be measured directly without specialised devices [1], but bending stiffness can be derived from the strain-stress curve of materials [9]. Therefore, if a neural network can learn the strain-stress relationship, it is possible to estimate the bending stiffness of fabrics and garments. That is, by observing deformations of fabrics and garments, if the predicted external forces (stresses) match measured external forces and deformations between simulated and real fabrics and garments, we can establish that the predicted bending stiffness can be approximated to the real values. We refer to the match between deformations of real and simulated objects as *Physics Similarity Distances* (PSD, Section IV-A).

In our experiments, we use an electric fan to wave real fabrics to exert an external force. We, therefore, predict wind speed, which is proportional to wind force, as $F_w = 1/2A\rho v$ where F_w is the wind force, ρ is the air density and A is the surface area of a deformable object. In our experiments, the fabrics used in our experiments have a surface area of 1 m^2.

IV. MATERIALS AND METHODS

A. PhySNet

In this paper, we propose a Physics Similarity Network (PhySNet), which is a Siamese network [14], [24] that clusters input data according to their labels. PhySNet comprises a convolutional neural network that extracts features from input data and a fully connected layer that maps the extracted features into a 2D *Physics Similarity Map* (PSM). We express our PhySNet as$\mathcal{P} = f_\theta(I)$, where f_θ denotes a neural network that contains convolutional layers and fully connected layers parameterised by the parameters θ, and I denotes an input video frame. We define $\mathcal{P}$ as a *physics similarity point*, which is a point on the PSM mapped from an input fabric image I. With these points in the PSM, we define a *Physics Similarity Distance* (PSD) as:

$$PSD_{i,j} = \left|\mathcal{P}_i - \mathcal{P}_j\right|^2 \tag{2}$$

where i and j are the *ith* and *jth* physics similarity points in the PSM of two different fabric images. Input fabric images can either be RGB or depth images of fabrics labelled according to their physics property parameters and external parameters.

The triplet loss compares positive and negative pairs. Positive pairs are an anchor sample and a positive sample within the same class (the physics property parameters in this paper). In contrast, negative pairs are an anchor sample and a negative sample of a different class from the anchor sample. The introduction of triplet loss ensures that samples of the same classes are clustered together while different classes are separated on the physics similarity map. Therefore, negative and positive samples are not directly compared, but negative and positive pairs are compared.

The triplet loss compares positive pairs and negative pairs. Positive pairs are an anchor sample and a positive sample within the same class (the physics property parameters in this research). In contrast, negative pairs are an anchor sample and a negative sample with a different class from the anchor sample. The introduction of triplet loss ensures that samples of the same classes are clustered together while different classes are separated on the physics similarity map. Therefore, negative and positive samples are not directly compared, but negative and positive pairs are compared, which performs better than other loss functions, such as contrastive loss. The triplet loss is less greedy than the contrastive loss. A triplet loss also ensures a margin between negative pairs and positive pairs, while contrastive loss only uses a margin for dissimilar pairs, irrespective of the positive pairs. This difference leads to the contrastive loss function reaching a local minimum while the triplet loss continues to optimise and perform better. That is, [25] shows that the contrastive loss underperforms with respect to the triplet loss by 14.1%

Images are triplet-classed, meaning that every input contains three images, one defined as an anchor and the other as positive and negative samples of the anchor. The input triplets are mapped onto the PSM through PhySNet as physics similarity points. Thus, our loss function is defined as:

$$\begin{aligned} PP &= \left|\mathcal{P}_{positive} - \mathcal{P}_{anchor}\right|^2 \\ NP &= \left|\mathcal{P}_{negative} - \mathcal{P}_{anchor}\right|^2 \\ Loss &= max(0, PP - NP + Margin) \end{aligned} \tag{3}$$

where $\mathcal{P}_{positive}$, $\mathcal{P}_{negative}$ and $\mathcal{P}_{anchor}$ are the positive, negative and anchor points, respectively. An anchor point is a point output from the PhySNet with an input of an image of a piece of fabric. A positive point is a point output from the PhySNet with an input of an image of a piece of fabric of the same physics property parameters as the anchor one. A negative point is a point output from the PhySNet with an input of an image of a piece of fabric with different physics property parameters to the anchor one. *PP* and *NP* are the positive pair and negative pair, respectively. The loss function aims to shorten the Physics Similarity Distances (PSDs) between the positive pairs and increase the PSDs between the negative pairs. The implementation of *Margin* ensures that the triplet loss does not concentrate on "simple pairs" (the positive and negative pairs that have a large difference, meaning they are easy to be clustered) but on "hard pairs" (the positive and negative pairs that have a small difference, meaning they are difficult to be clustered), facilitating the learning of physics similarities between simulated fabrics.

B. BAYESIAN OPTIMISATION

Initialised physics parameters are input into the simulation engine, and output simulated fabrics. The simulated and real fabrics are fed into a trained PhySNet, which outputs their physics distances. In this paper, we aim to close the gap between simulation and reality by finding physics parameters for the simulation that resemble those observed in reality. For this, we use Bayesian optimisation to find these physics parameters for the simulation, and the objective is to minimise physics distances between simulated and real fabrics.

Bayesian optimisation is used to find the optimal value of a black-box function, where the black box means that the structure and parameters of the function are unknown. The black-box function is "expensive to evaluate", which means evaluating the function is computationally costly. In this paper, Gaussian Processes are used to estimate the prior and posterior distribution.

This optimisation comprises a black-box function $f(X)$ that takes X as input (where the dimension of X is usually less than 20). Firstly, a random function (also called a "prior") is used when several initial values of X (termed as querying points) are used to evaluate $f(X)$, and some values of $f(X)$ are obtained. These values of $f(X)$ are used to update the "prior" to form a posterior distribution, which is then used to construct an acquisition function. The acquisition decides the next querying point of X, which is the next value of X to be input into $f(X)$ to evaluate the value of $f(X)$. The Bayesian optimisation terminates when an optimal value of $f(X)$ is found.

To minimise physics distances, we thus convert physics distances into negative values (for example, convert a physics distance of 100 to -100) and maximise negative values (optimal values are 0). We have used *Botorch* [26] to implement our Bayesian Optimisation. Figure 1 shows our pipeline.

According to [27], Bayesian optimisation is used for expensive-to-evaluate functions. In our experiments, evaluating the physics similarity distance between a simulated fabric and a real fabric/garment requires updating the simulated fabric's physics parameters, modelling a new fabric with ArcSim and rendering the new fabric with Blender, which requires time. If traditional optimisation algorithms such as stochastic gradient descent are applied, generating sufficient samples within a short time is impractical for this experiment. Therefore, only Bayesian optimisation was tested in the experiments.

V. EXPERIMENTS

A. FABRICS AND GARMENTS DATASET

For our experiments, we collected both simulated and real fabric samples. To simulate fabrics, we use ArcSim [28], which is a deformable object simulator that uses triangle meshes and linear piecewise functions (Section III). Inputs to ArcSim are the physics parameters of fabrics, including stretching stiffness, bending stiffness and area weights, and external environmental parameters, including gravity, wind speed and wind direction. In this experiment, our search space for the Bayesian Optimisation (as defined in Section IV-B) includes bending stiffness, wind speed and area weight; thus, we keep other parameter settings in their default values.The external parameters are (as we set them in ArcSim): (i) wind speed (from 1 to 6 m/s), (ii) fabric's area weight (see Table 1), and (iii) Bending stiffness (from 0.1 to 10 times of standard bending stiffness parameters, [12], [14]). We defined this search space based on the experimental settings described in [14].

We have tested seven different materials; *tablecloth*, *interlock*, *denim*, *sparkle fleece*, *nylon*, *ponte roma* and *jet set (red-violet)*. We choose these materials because they are common in the textile industry. Table 1 shows the search space for the different materials in terms of their area weight, and the area weight is determined by finding the manufacturer's information. We set the search space for wind speeds to 1-6 m/s.

ArcSim outputs a sequence of 60 3D models. The length of each video is 3 seconds with a sampling frequency of 20Hz. We input these 60 3D models into Blender [2] to render them into a video sequence of depth images, where each 3D model corresponds to one frame. Because depth images are sensitive to cameras' relative positions with respect to the captured object, randomising cameras' positions in the simulation environment can enhance PhySNet to recognise real fabrics and garments. Therefore, we randomised the camera locations in Blender and captured a fabric from six different locations. That is, we translate in ArcSim in the x (from 1 to 6) and z (from -0.5 to 0.3) axes while leaving fixed the y axis to 0.5. Similarly, we rotate the camera in ArcSim for z (from -260° to 280°) while we set the rotation in x to 90° and y to 0°. Bending stiffness settings are referenced in [12], where they provided measured values of the materials used in our experiments. Therefore, our search space for bending stiffness is from 0.1 to 10 of measured values in [12].

For each simulated material, we randomise 30 combinations of physics property parameters and external environmental parameters constrained within the search space

TABLE 1. Area weight search space for different materials.

Material	Area Weight Search Space
White Tablecloth	0.1-0.17 m/s^2
Gray Interlock	0.15-0.22 m/s^2
Black Denim	0.30-0.37 m/s^2
Sparkle Fleece	0.23-0.30 m/s^2
Pink Nylon	0.16-0.23 m/s^2
Ponte Roma	0.23-0.3 m/s^2
Red Violet	0.1-0.17 m/s^2

defined above. Combinations are uniformly distributed, and each combination comprises a sequence of 60 3D models. We input these 60 models into the Blender engine and render the models with 6 rendering camera positions. Therefore, we captured 10,800 images for each material, which are labelled with their combination number. ArcSim [28] and Blender [2] are used for generating images of simulated fabrics in this experiment. However, images not containing entire fabrics due to the camera positions are not included and removed from the dataset.

We use an Asus Xtion camera to collect real fabric and garment samples. An electric fan waves fabrics with wind speeds varying from 2.4-3.1 m/s. The varying wind speeds can test whether our approach can detect fabrics and garments' physics property parameters under different wind speeds. For each real sample, a video of 60 frames in length is recorded at a sampling frequency of 24 fps (2.5 s in real-time for each video). Wind speeds are measured by an electronic anemometer (model AOPUTTRIVER AP-816B), and area weights are measured using an electric scale. All fabrics are cut into a square of 1 $m \times 1$ m such that their weights scaled by the electric scale are unit area weights. Our testing points for wind speeds are located near the fabric. A list of the equipment used for these experiments and the simulated and real images can be found at https://liduanatglasgow.github.io/PhySNet-BayOptim/.

B. EXPERIMENTAL METHODOLOGY

We have implemented PhySNet in Pytorch. PhySNet consists of 2D convolutional layers with a PReLU layer and a MaxPool2D layer between adjacent convolutional layers. The convolutional layers are followed by a fully connected layer with three linear layers and a PReLU between adjacent linear layers. Input images are 1-channel depth with an image resolution of 256×256. We have used an Adam optimiser with a batch size of 32 and a learning rate of 1×10^{-2}. A learning scheduler with a step size of 8 and a decay factor of 1×10^{-1} has been used for the optimiser. We train our PhySNet for 30 epochs with a batch size of 32. Our PhySNet is trained on simulated fabrics images but tested on real, unseen fabrics and garments. Our code is available at https://liduanatglasgow.github.io/PhySNet-BayOptim/.

We compare the performance of our PhySNet network with the Spectrum Decomposition Network (SDN) proposed in [14]. This research was inspired by SDN, where the authors proposed learning wind speeds and fabric area weights. Other research on the physics property parameters of fabrics includes [1], [13]. However, they all trained and tested their proposed approaches directly on real fabrics rather than learning from simulated fabrics. SDN is the first research on the physics property parameters of fabrics from learning physics similarities between simulated fabrics when fabrics are waved under a wind field. Runia et al. [14] showed that the difficulties in measuring the physics property parameters of real fabrics using specific and sophisticated instruments could be solved by learning from physics similarities between simulated fabrics.

The SDN is a network that uses a Fourier transformation to convert time-domain RGB images into frequency-domain maps and extracts the top K maximum-frequency parts of the maps as features. For our baseline, we compare the performance of four networks; two networks are PhySNet trained on depth and RGB images, and the other two are SDN trained on depth and RGB images.

1) ESTIMATING PHYSICS PARAMETERS OF FABRICS AND GARMENTS

This experiment aims to find real fabrics' physics and external environmental parameters. Therefore, we adjust parameter settings in the simulation engine to generate a simulated fabric and calculate its PSD to the real fabric on the PSM. We halt the optimisation once a stable PSD is found between a simulated and real fabric (ref. Section IV-B). As discussed in Section III, we only compare predicted results of wind speeds and area weights because we do not have ground truth for the bending stiffness of the real fabrics. Still, wind speeds serve as indicators of the bending stiffness of the real fabrics and act as our ground truth to validate our proposed approach.

The Bayesian Optimiser described in section IV-B is used to find physics and external environmental parameters for simulated fabrics that can minimise the physic similarity distance between the simulated and real fabrics. Parameters optimised in this experiment are bending stiffness, wind speeds and area weights, which are normalised to $[-1, 1]$. Values for the parameters are initially set as 0. The search space for these parameters is the same as the search space set for simulated data as in Section V-A. We halt the Bayesian Optimisation when updated parameters become stable. That is, parameter updates do not change by more than 10% over the last three epochs. Wind speed and area weight estimations are compared with the measured ground truths, i.e. from the anemometer and electric scale.

Simulating fabrics is easier than simulating garments because fabrics have simple geometric shapes, whereas garments have complex shapes. If PhySNet can recognise real garments while being trained on simulated fabrics, we can bypass simulating complex garments. We hypothesise that *dynamics and physics property parameters are constant between garments and fabrics made of similar materials and can enable PhySNet to predict garment physics property parameters by training on simulated fabrics*. Therefore, we test this hypothesis by allowing PhySNet to predict the physics and external environmental parameters for real garments from the simulated fabrics. We selected three garments: a T-shirt, a shirt and jeans. To measure the physics parameters of these garments, we use our PhySNets trained on the grey interlock (for the T-shirt), a white tablecloth (for the shirt) and the black denim (for the jeans) because these garments are made of these fabrics and have similar physics parameters.

The electric fan waves garments and the wind speeds are recorded using the anemometer. Likewise, we follow the same methodology for fabrics to capture garments as video sequences. Garment images are input directly into PhySNet, and the Bayesian optimiser is used to find the garments' physics parameters. A garment is compared with a simulated fabric of the same material rendered with parameters set to 0. Updated parameters from the Bayesian optimiser are input into the simulator to output an updated fabric, and it is then compared to the real garments until stable parameters are obtained. We halt the Bayesian Optimisation when updated parameters become stable, as in the fabrics experiment.

TABLE 2. Clustering accuracy for PhySNet and the SDN networks [14] trained on depth and RGB images.

Name	PhySNet (Depth)	SDN (RGB)	PhySNet (RGB)	**SDN (Depth)**
White Tablecloth	80%	**91%**	89%	90%
Black Denim	88%	89%	86%	**97%**
Gray Interlock	83%	86%	84%	**92%**
Sparkle Fleece	77%	82%	78%	**92%**
Ponte Roma	79%	84%	80%	**93%**
Pink Nylon	80%	83%	77%	**93%**
Red Violet	80%	87%	78%	**92%**

VI. EXPERIMENTAL RESULTS

A. CLUSTERING ACCURACY OF PhySNet AND SDN

From Table 2, we observe that the best performance for clustering accuracy is on the SDN-trained network while using depth images. Whereas the network with the lowest accuracy is PhySNet trained on depth images. Overall, SDN has a better performance than PhySNet. This is because a Fourier transform outputs a frequency map for the transformed images, and on this frequency map, areas of the fabrics that deform fast from the waving wind are amplified while static areas are attenuated. The SDN benefits from these frequency maps while ignoring 'less deformed' areas, but this causes an information loss and overfitting of the training data. This loss of information can potentially reduce the network's ability to recognise real fabrics, as shown in Section VI-B.

From Table 2, PhySNet trained on RGB images performs better than PhySNet trained on depth images. For depth images, changes in physics parameters do not have the same levels of influence on spatial characteristics as texture characteristics. Depth information remains relatively constant between simulated and real fabrics, which means that depth

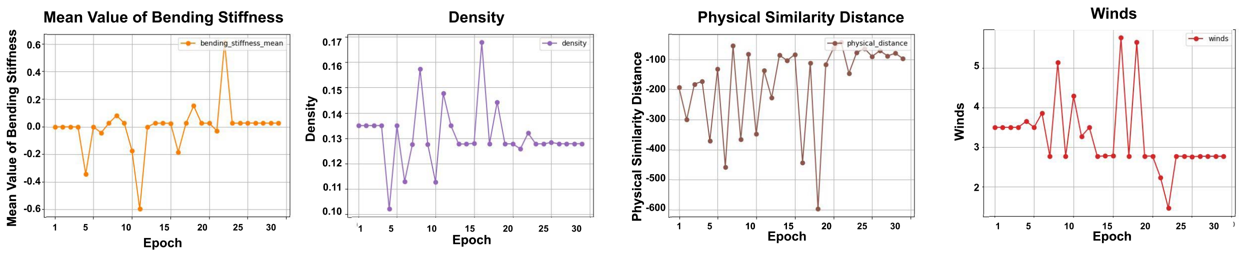


FIGURE 3. An example of a successful Bayesian Optimisation: PhySNet estimating the physics parameters of the white tablecloth.

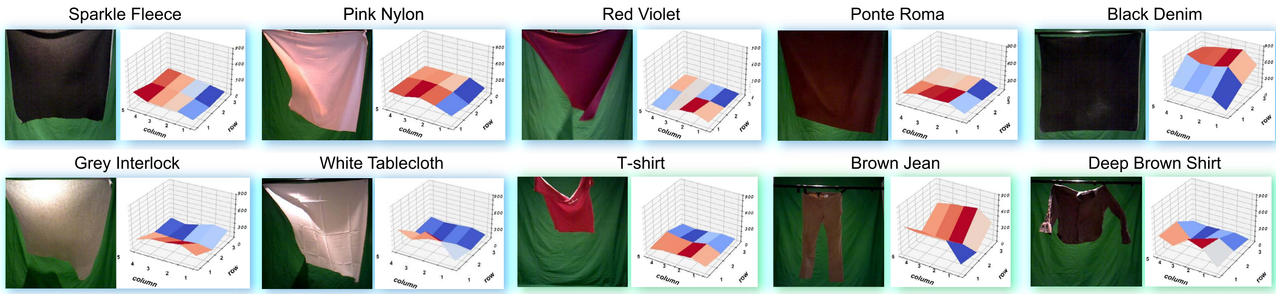


FIGURE 4. *Predicted bending stiffness of real fabrics and garments.* We use surface plots [29] to visualise predicted bending stiffness parameters of real fabrics and garments. The x and y axes in the surface plots represent the row and column indexes of the parameters, and the z axes are the bending stiffness parameters.

is suitable for finding the physics parameters of real fabrics and generalising better across domains.

B. PREDICTING FABRICS' AND GARMENTS' PHYSICS PARAMETERS

Table 3 shows that the best performance is obtained using the PhySNet trained on depth images. Our approach improves the state-of-art (SDN trained on RGB images) by 34.0%. Both the SDNs (trained on depth and RGB images) experience failures in finding the physics parameters of real fabrics (denoted as 'F'). The reason for the failures is that the SDN failed to map real fabric images onto the physics similarity map; hence, the Bayesian optimiser cannot find optimal values for the physics parameters of real fabrics. As discussed in Section VI, the SDN has the disadvantage of information loss that affects the network's ability to predict the physics property parameters of real fabrics. From Table 3, we also observe that PhySNet trained on depth images outperforms PhySNet trained on the RGB images. Depth images directly capture deformations, while RGB images capture changes in the texture and colour manifolds that are not descriptive of deformations and structural changes.

Figure 4 shows the predicted bending stiffness of real fabrics. Bending stiffness parameters are represented as matrices (as defined in Section III). Therefore, we use surface plots to display predicted values. From Figure 4, we can observe that *black denim* is the stiffest material, while the *sparkle fleece* is the softest material because *black denim* has the highest predicted bending stiffness while *sparkle fleece* has the lowest predicted value. These measurement results align with human intuitions, where denim (i.e. jeans) is stiffer than sparkle fleece (i.e. sweaters).

TABLE 3. *Fabric Physics Parameter Estimation.* Percentage errors are w.r.t group truths; wind (unit: m/s) and area weight (unit: kg/m^2).

Materials	PhySNet (depth)	PhySNet (RGB)	SDN (depth)	SDN (RGB)
White Tablecloth	**6.5%, 8.6%**	9.2%, 10%	119.6%, 15.0%	75.4%, 11.11%
Gray Interlock	9.6%, 19.6%	40.7%, 10.9%	5.4%, 15.8%	**5.4%, 3.3%**
Black Denim	**5.4%, 5.5%**	37.3%, 1.2%	F, F	90.0%, 8.2%
Ponte Roma	35%, 0.4%	34.6%, 0.4%	**22.7%, 0.8%**	33.8%, 0%
Sparkle Fleece	**34.6%, 0%**	**34.6%, 0%**	48.8%, 2.6%	F, F
Red Violet	**16.7%, 6.3%**	**16.7%, 6.3%**	F, F	F, F
Pink Nylon	**12.6%, 2.6%**	57.1%, 2.6%	F, F	F, F

TABLE 4. *Garment Physics Parameter Estimation.* Percentage errors are w.r.t group truths; wind (unit: m/s) and area weight (unit: kg/m^2).

Materials	**PhySNet (depth) Ours**	PhySNet (RGB)	SDN (depth)	SDN (RGB)
T-shirt	3.1%, 17.1%	**1.92%, 15.5%8**	F, F	27.3%, 0.5%
Deep Brown Shirt	**34.2%, 1.5%**	59.2%, 6.7%	60.4%, 18.7%	F, F
Brown Jeans	**21.7%, 0.6%**	97.9%, 2.5%	F, F	148.3%, 8.3%

Table 4 shows the Bayesian Optimisation results for garments. We can observe that PhySNet trained on depth images performs best while predicting garments' physics properties and external environmental parameters. However, from Table 4, we also observe that predictions for garments are not as accurate as the predictions for fabrics due to the different shapes between the garments and fabrics. SDN RGB and depth and the PhySNet RGB failed to optimise correctly and converged to incorrect values for each of the three garments. The results, similar to section VI-B, indicate the disadvantages of using RGB images and frequency maps for finding real-garment physics parameters. Predicted stiffness parameters are shown in Figure 4. We can observe that jeans are stiffer than T-shirts and shirts, which aligns with human intuition. These results suggest that it is possible to estimate the physics property parameters of garments by training PhySNet on simple fabrics with a mean average error of 17.2% for wind speeds and 6.5% for area weight parameters. Overall, we obtained a performance improvement between our approach (PhySNet on depth images) and SDN on RGB images (state of the art) is 68.1%

VII. CONCLUSION

In this paper, we proposed that predicting the physics property parameters of real fabrics and garments can be achieved by learning physics property similarities between simulated fabrics. Our PhySNet outperforms the state-of-art by 34.0% for fabrics and 68.1% for garments. However, there are limitations to our proposed approach. That is, only bending stiffness is considered, and physic's property parameters that determine strains (deformations) consist of stretching stiffness, bending stiffness and damping. The reason to limit the physics parameters is to reduce the search space for the Bayesian Optimisation and guarantee convergence. Further research involves developing a better optimisation method to optimise all physics property parameters. We also show that PhySNet is more effective while training on one rather than multiple materials. Our future research focuses on devising a methodology to enable a neural network to be trained on different materials and predict the physics property parameters of different fabric materials. Indeed, we envisage that a general purpose of using PhySNet for predicting the physics property parameters of fabrics and garments is to facilitate robotic fabric and garment manipulation.

In our experiments, we used an electric fan to exert an external force (waving) on fabrics and garments. We have shown that a robot can interact with garments [30], [31], and we envisage that robots can exert these forces on fabrics and garments while the robot interacts with the objects. A robot can stretch objects to measure stretching stiffness and facilitate manipulating objects by grasping and dropping them to observe their deformations. From these interactions, the network can effectively learn the physics parameters of deformable objects. Also, future work can focus on an ablation study of using data (video sequences) from multiple perspectives to verify the proposed approach's effectiveness and applicability.

ACKNOWLEDGMENT

The work of Gerardo Aragon-Camarasa was supported by the EPSRC under Grant EP/S019472/1. The authors would like to thank Ali AlQallaf and Ozan Bahadir for their constructive feedback and comments.

REFERENCES

[1] K. L. Bouman, B. Xiao, P. Battaglia, and W. T. Freeman, "Estimating the material properties of fabric from video," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1984–1991.

[2] R. Hess, *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Waltham, MA, USA: Focal Press, 2010.

[3] E. Coumans and Y. Bai. (2021). *PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning*. [Online]. Available: http://pybullet.org

[4] J.-I. Ueda and Y. Sadamoto, "A measurement of the effective mass of coil springs," *J. Phys. Soc. Jpn.*, vol. 66, no. 2, pp. 367–368, Feb. 1997.

[5] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 249–267, 2012.

[6] H. Lin, F. Guo, F. Wang, and Y.-B. Jia, "Picking up a soft 3D object by 'feeling' the grip," *Int. J. Robot. Res.*, vol. 34, no. 11, pp. 1361–1384, 2015.

[7] L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, and L. Sabourin, "Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand," *Robot. Auto. Syst.*, vol. 95, pp. 196–206, Sep. 2017.

[8] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Simple model and deformation control of a flexible rope using constant, high-speed motion of a robot arm," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2249–2254.

[9] T. H. Courtney, *Mechanical Behavior of Materials*. Long Grove, IL, USA: Waveland Press, 2005.

[10] A. Sengupta, R. Lagneau, A. Krupa, E. Marchand, and M. Marchal, "Simultaneous tracking and elasticity parameter estimation of deformable objects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10038–10044.

[11] V. E. Arriola-Rios and J. L. Wyatt, "A multimodal model of object deformation under robotic pushing," *IEEE Trans. Cognit. Develop. Syst.*, vol. 9, no. 2, pp. 153–169, Jun. 2017.

[12] H. Wang, J. F. O'Brien, and R. Ramamoorthi, "Data-driven elastic models for cloth: Modeling and measurement," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–12, Jul. 2011.

[13] S. Yang, J. Liang, and M. C. Lin, "Learning-based cloth material recovery from video," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4383–4393.

[14] T. F. H. Runia, K. Gavrilyuk, C. G. M. Snoek, and A. W. M. Smeulders, "Cloth in the wind: A case study of physical measurement through simulation," 2020, *arXiv:2003.05065*.

[15] B. Tawbe and A.-M. Cretu, "Acquisition and neural network prediction of 3D deformable object shape using a Kinect and a force-torque sensor," *Sensors*, vol. 17, no. 5, p. 1083, May 2017.

[16] A. Simeonov, Y. Du, B. Kim, F. R. Hogan, J. Tenenbaum, P. Agrawal, and A. Rodriguez, "A long horizon planning framework for manipulating rigid pointcloud objects," in *Proc. Conf. Robot Learn.*, 2020, pp. 1582–1601.

[17] P. Guler, A. Pieropan, M. Ishikawa, and D. Kragic, "Estimating deformability of objects using meshless shape matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5941–5948.

[18] Z. Wang, S. Rosa, B. Yang, S. Wang, N. Trigoni, and A. Markham, "3D-PhysNet: Learning the intuitive physics of non-rigid object deformations," 2018, *arXiv:1805.00328*.

[19] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popović, and S. M. Seitz, "Estimating cloth simulation parameters from video," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2003, pp. 37–51.

[20] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 558–569, Jul. 1993.

[21] J. Wu, J. Lim, H. Zhang, J. Tenenbaum, and W. Freeman, "Physics 101: Learning physical object properties from unlabeled videos," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, no. 6, 2016, p. 7.

[22] Z. Gao, X. Wang, S. Sun, D. Wu, J. Bai, Y. Yin, X. Liu, H. Zhang, and V. H. C. De Albuquerque, "Learning physical properties in complex visual scenes: An intelligent machine for perceiving blood flow dynamics from static CT angiography imaging," *Neural Netw.*, vol. 123, pp. 82–93, Mar. 2020.

[23] R. Narain, A. Samii, T. Pfaff, and J. O'Brien, "ARCSim: Adaptive refining and coarsening simulator," Dept. Elect. Eng. Comput. Sci., Univ. California-Berkley, Berkley, CA, USA, 2016. Accessed: Oct. 1, 2014.

[24] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, 2015, pp. 1–30.

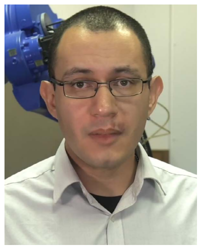
[25] R. Kuma, E. Weill, F. Aghdasi, and P. Sriram, "Vehicle re-identification: An efficient baseline using triplet embedding," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–9.

[26] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BoTorch: A framework for efficient Monte–Carlo Bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–15.

[27] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2015.
[28] R. Narain, A. Samii, and J. F. O'Brien, "Adaptive anisotropic remeshing for cloth simulation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–10, Nov. 2012.
[29] W. Schwanghart and N. J. Kuhn, "TopoToolbox: A set of MATLAB functions for topographic analysis," *Environ. Model. Softw.*, vol. 25, no. 6, pp. 770–781, Jun. 2010.
[30] L. Sun, G. Aragon-Camarasa, S. Rogers, and J. P. Siebert, "Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 185–192.
[31] L. Martínez, J. Ruiz-Del-Solar, L. Sun, J. P. Siebert, and G. Aragon-Camarasa, "Continuous perception for deformable objects understanding," *Robot. Auto. Syst.*, vol. 118, pp. 220–230, Aug. 2019.

LI DUAN is currently pursuing the Ph.D. degree with the School of Computing Science, University of Glasgow. He has authored three peer-reviewed scholarly articles and is seeking collaborations with other researchers on robotic deformable object manipulation. His interests include continuous perception for robots and effective garment flattening pipelines.

LEWIS BOYD is currently pursuing the Ph.D. degree in design, manufacturing and engineering management with the University of Strathclyde with interests in robotics, computer vision, and deep learning. His current research interests include training robotic behaviors, such as grasping and pose reaching, using deep reinforcement learning.



GERARDO ARAGON-CAMARASA is a Lecturer of autonomous and socially intelligent robotics with the School of Computing Science, University of Glasgow, with interests in robotics, computer vision, and machine learning. He has authored over 40 refereed articles in his research areas. His research interests include multidisciplinary and focuses on robotic manipulation and grasping, chemical robotics, and machine perception/vision.