# Defo-Net: Learning Body Deformation using Generative Adversarial Networks

Zhihua Wang*[1], Stefano Rosa*[1], Linhai Xie[1], Bo Yang[1], Sen Wang[2],
Niki Trigoni[1] and Andrew Markham[1]

***Abstract*— Modelling the physical properties of everyday objects is a fundamental prerequisite for autonomous robots. We present a novel generative adversarial network (DEFO-NET), able to predict body deformations under external forces from a single RGB-D image. The network is based on an invertible conditional Generative Adversarial Network (IcGAN) and is trained on a collection of different objects of interest generated by a physical finite element model simulator. DEFO-NET inherits the generalisation properties of GANs. This means that the network is able to reconstruct the whole 3-D appearance of the object given a single depth view of the object and to generalise to unseen object configurations. Contrary to traditional finite element methods, our approach is fast enough to be used in real-time applications. We apply the network to the problem of safe and fast navigation of mobile robots carrying payloads over different obstacles and floor materials. Experimental results in real scenarios show how a robot equipped with an RGB-D camera can use the network to predict terrain deformations under different payload configurations and use this to avoid unsafe areas.**

## I. INTRODUCTION

A key requirement for autonomous mobile robots is the ability to perceive their surroundings and model the environment in order to tackle high-level tasks, such as compliant manipulation and safe navigation. For instance, many everyday objects that a robot would need to interact with are deformable or non-rigid. In particular, traditional path planning approaches make the assumption that the environment contains only rigid components and obstacles. In reality, not all objects or paths are rigid. Without an understanding of the potential deformation of the terrain, a wheeled robot could get stuck in a soft material or unsafely overload a weak object. To tackle this problem, the robot needs to be able to predict the traversability of the terrain, a process called *terrain assessment*. The problem of estimating the deformation of traversable spaces is particularly important for mobile robots carrying payloads in partially or totally unconstrained environments, for search and rescue applications, outdoor or planetary robotics in general, and industrial applications such as Automated Guided Vehicles (AGVs).

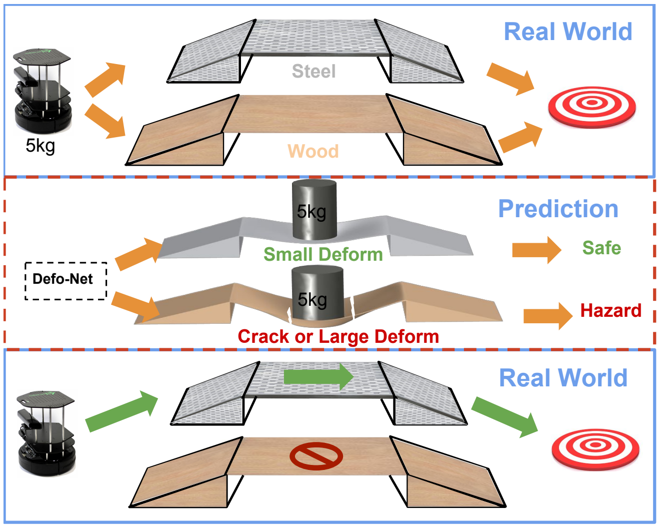

Fig. 1. An invertible conditional GAN is trained for predicting body deformations given an external force acting on it. This is used to estimate the 3-D deformation of potential routes. Top: the robot scans two possible routes it can take to reach the target. Middle: Using the inferred material, the RGB-D depth image, and the load, a prediction is made of the expected deformation. Bottom: Based on the results of the prediction, the path planner will take the steel bridge to reach the target.

Explicit modelling of material deformation usually requires extensive configuration and computational effort. In particular key material properties such as elasticity (or its inverse, rigidity) and the Young's Modulus need to be specified. Deformations of non-rigid objects have been modelled in the past using mass-spring systems. While mass-spring systems can model large deformations with relatively little computational effort, they are non-intuitive and do not accurately model material properties. They are also difficult to expand to 3-D modelling. Finite Element Method techniques (FEMs) on the other hand are highly accurate, but consequently computationally expensive, due to the large number of mesh nodes required to accurately model deformations. Co-rotational finite element approaches are a faster approximation of FEMs. Haptic sensors have been used to learn physical properties of materials, such as elasticity [1], [2]. Other methods for estimating properties of materials include a combination of induced vibrations and computer vision [3]. More recently, the use of low-cost vision sensors and deep convolutional networks has been investigated. In [4] a convolutional autoencoder learns to deform a voxelized representation of input objects given an "intention" such as "make it sportier". In [5] the authors recently proposed a method for pre-computation of the

*These two authors contributed equally

[1]Wang, Rosa, Xie, Yang, Trigoni and Markham are with Department of Computer Science, University of Oxford, Oxford OX1 3QD, United Kingdom {firstname.lastname} @cs.ox.ac.uk

[2]Wang is with School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom s.wang@hw.ac.uk

dynamics of fluid spaces using implicit surfaces. Material recognition is another problem that has been investigated using Convolutional Neural Networks [6] [7].

In this paper, we address the problem of estimating deformation of non-rigid structures, an open and under-explored topic in robotics.

We propose a novel deep network that combines an autoencoder and a conditional GAN, tightly coupled with an FEM-based physics simulator. Given a single RGB-D depth image of the deformable object (e.g. from a side scan of the object to be traversed) and conditioning input which includes the type of material (e.g. aluminium), the size of the force (e.g. 50 N), and the location of the force (e.g. 10 cm), the network is able to output a predicted 3-D deformation of the solid. This prediction can then be used by a path-planner to determine which is the fastest or safest path to take, given terrain information and robot payload.

We evaluate our approach in three real case scenarios involving a mobile robot travelling over different bridge-like obstacles and soft ground, under different payload conditions. We also show the generalisation capabilities of the network to unseen objects configurations. Real world results agree closely with GAN predictions, showing its predictive power for deformable objects such as bridge-like structures and soft ground. A single network prediction is order of magnitudes faster that an equivalent FEM simulation, at the cost of lower resolution; this makes the approach useful for online evaluation during navigation.

In particular, the contributions of this paper are as follows:

- To the best of our knowledge, this is the first application of an invertible conditional GAN to the problem of learning body deformations in 3-D.
- DEFO-NET provides a fast and accurate approximation of FEM, which makes it suitable for predictive terrain assessment for autonomous robots.
- We demonstrate through real-world experiments that we can generalize to different materials and structural configurations.

The remainder of this paper is as follows: in Section II we discuss some related work; in Section III we describe the proposed DEFO-NET, the physics engine used for training and the training procedure; in Section IV we validate the approach in three experimental tests; finally, in Section V we draw conclusions and discuss future work.

## II. RELATED WORK

### A. Deformable materials

Traditional estimation techniques have been applied to the estimation of material deformations. In [8] the authors addressed the problem of autonomous navigation in presence of deformable obstacles, such as curtains, and manipulation of soft objects. The deformation model is learnt by the robot via physical interaction, first by running FEM simulations using the learnt deformations, then by approximating deformation cost functions for specific objects using Gaussian process regression. In [9] RGB-D images are used to learn the elasticity parameters of soft objects. In [10], Lyapunov theory is used for estimating the deformation Jacobian matrix of compliant objects under elastic deformations. Predicting shape deformations using computer vision has also been investigated in surgery applications, using an Unscented Kalman Filter for modelling the deformation of flexible needles inside soft tissues [11]; RGB images of the needle are used in the filter updates. In [12] periodic stimuli are applied to grasped objects using a gripper and the deformations are learned from RGB by magnifying the optical flow. RGB-D images have been used in the past for estimating deformations using a variant of expectation maximization [13].

### B. Intuitive physics

In a seminal work [14] the authors proposed a generative model for learning physical scene understanding from video images, such as the effect of gravity and friction on objects. This is done by inverting a physics engine to obtain physical properties from observations. Recently, deep networks have been shown to be able to learn basic intuitive physics, such as predicting the stability of tower blocks [15] [16], object dynamics [17], interacting with humans [18] and with other objects [19], predicting the long-term effect of external forces [17], and correlating actions with effects [20].

Predicting how actions affect the world is an open challenge. In [21] a deep model was trained in an unsupervised way to predict action-conditioned future video images of moving objects, using Convolutional Dynamic Neural Advection (CDNA) and action-conditioned LSTMs. Generative networks have been able to predict future video snippets given conditions [22]. Recently, SE3-NETS [23] were proposed for learning to segment a scene into rigid objects and predict the motion of these segmented objects under the effect of applied forces. The network takes as inputs a point cloud and a force vector applied to it; the network is able to segment rigid objects in the image and predict the effect of the applied force on object motion, using a layer that encodes per-pixel roto-translations.

### C. Generative Adversarial Networks

Deep generative models such as Generative Adversarial Networks (GANs) [24] and Variational Autoencoders (VAEs) [25] have recently shown outstanding results in modelling high-dimensional representations and generalization abilities. GANs have been successfully applied in different domains such as generation of text [26], learning of latent spaces [27], 3-D reconstruction [28].

In the original GAN formulation the discriminator network is trained to classify real and fake examples. However, the loss function can be difficult to converge and training is often unstable. Recently, WGAN [29] made some progress towards stable learning of GANs by using Wasserstein distance with weight clipping. A recent work [30] proposed to penalize the norm of gradient of the discriminator with respect to its input, improving training stability. Conditional GANs (cGANs) use external conditional information to determine specific

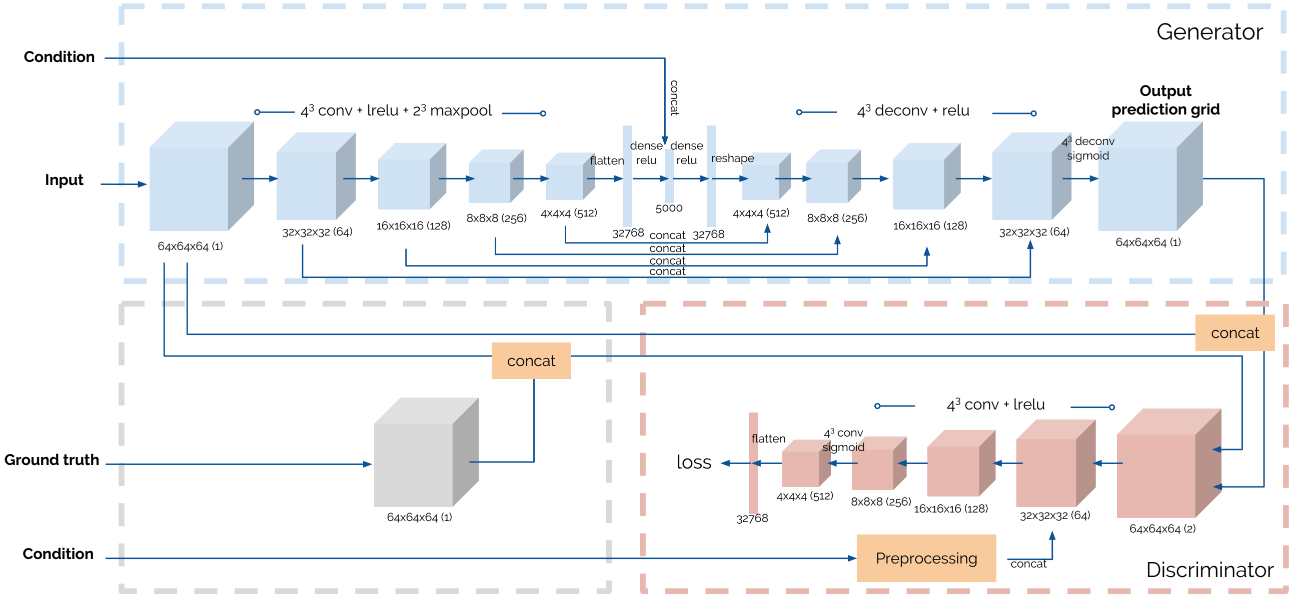

Fig. 2. The DEFO-NET architecture.

representations of the output. Invertible cGANs (IcGANs) [31] combine an autoencoder with a cGAN and have been proved to be able to learn a good latent representation of the inputs. cGANs have been recently used to learn mappings between input and output images with both paired [32] and unpaired [33] images.

## III. NETWORK

Figure 2 shows the architecture of the proposed DEFO-NET. It is composed of two main networks: a generator network $G$ and a discriminator network $D$, that are competing against each other. The architecture builds up on the one recently proposed in [28], with conditional input extensions that allow parameters such as force, material and application position to be specified. Broadly, the generator maps the undistorted 3-D model into a deformed 3-D model, conditioned on the supplied parameters. The discriminator is used during training only and is a classifier that determines whether its input is drawn from the ground-truth or the output of the generator. The generator and discriminator are adversarial i.e. they each get better over time. We now describe each network in detail.

### A. Generator

The network takes as input a voxel grid of size 64×64×64, representing a 3-D point cloud, which is obtained by subsampling the input depth image.

Since a traditional generator from a GAN does not have the ability to map a 3-D point cloud to its latent representation, the generator is implemented as an autoencoder network $E$, that is able to learn a latent representation from input voxel grids. The encoder enables the network to explore the latent space by interpolating or making variations on it. By concatenating a *condition* to the latent representation, explicitly controlled variations can be made as conditional information, such as the size of the force.

In order to facilitate the propagation of local structures in the input voxel grids, the autoencoder has skip connections between the encoder and the decoder. The encoder has five 3-D convolutional layers with a bank of 4×4×4 filters with strides of 1×1×1, followed by a leaky ReLU activation function and a max pooling layer with 2×2×2 filters and 2×2×2 strides. The encoder is followed by two fully-connected layers which flatten the 3-D representation into a 1-dimensional vector representing the latent encoding. The condition is also encoded as a 1-dimensional vector. The condition encapsulates three properties: the magnitude of the force, the location of the force, and the material. Each of these properties is discretized into a range of values and represented as a one-hot vector. The condition vector is of the form $(f_2, a_2, m_2)$, where $f_2, a_2, m_2$ are the binary representations of the discretized conditions. In our application scenario, we use 2 bits for the force, 7 for the point of application and 2 for the material. The decoder largely follows the inverse of the encoder, composed of five up-convolutional layers which are followed by ReLU activations except for the last layer which is followed by a sigmoid function.

### B. Discriminator

The discriminator classifies the voxel grids produced by the generator, trying to distinguish whether the predicted outputs are realistic. It is composed of five 3-D convolutional layers, each of which has a bank of 4×4×4 filters with strides of 2×2×2, followed by a ReLU activation function except for the last layer which is followed by a sigmoid activation function. The discriminator takes as input pairs of 'real' ground truth point clouds and 'fake' generated clouds, as well as the condition vector. The condition vector for the discriminator is encoded differently to the generator as 32×32×32 one-hot block masks. This is performed by the Preprocessing kernel, simply replicating the 1-D condition vector on the three axes. Including the condition at an early stage of the discriminator makes it possible to model input variations. We experimentally found that inserting the condition at the second layer gives optimal results.

### C. Physics engine

The physics engine is used to generate ground truth pairs of input point clouds and condition vectors for use in training. In this work we used the COMSOL Multiphysics software in order to generate the training voxel grids, but the network is agnostic to the simulator. The 1-dimensional condition vector is also obtained from the simulator, and it is formed by concatenating the object material and a force vector (point of application and magnitude). Note that a 3-D FEM simulation takes several minutes to several hours depending on the mesh resolution, running on a workstation (CPU - Intel Xeon(R) CPU E5-1603 v3 2.80GHz × 4). The simulation takes less time if the mesh is coarser, but the risk of not converging becomes higher.

In our FEM simulations each mesh is composed by around 3000 triangles.

### D. Training

Figure 3 shows the training and testing configurations. In the training phase, input and output pairs of voxel grids are generated by the physics simulator.

The adversarial loss $\mathscr{L}^g_{gan}$ is the WP-GAN loss function from [30], with $\lambda = 10$. The reconstruction loss $\mathscr{L}_{AE}$ for the autoencoder $E$ is a specialized form of Binary Cross-Entropy (BCE), as in [34], and is given by:

$$\mathscr{L}_{AE} = -\alpha t \log(o) - (1-\alpha)(1-t)\log(1-o) \qquad (1)$$

where $t$ is the true occupancy value for each voxel (0,1), $o$ is the predicted occupancy value in the range (0,1), $\alpha$ is a weight that balances false positives and false negatives.

The total loss is:

$$\mathscr{L}^g = \beta \mathscr{L}_{AE} + (1-\beta)\mathscr{L}^g_{gan}, \qquad (2)$$

where $\beta$ is a constant that balances the autoencoder loss and the GAN loss.

The network was trained using the Adam optimizer, with a batch size of 8. The learning rate is 0.0005 in the first epoch, and decays to 0.0001. The network was implemented in Tensorflow and trained on a single Nvidia Titan X GPU.

### E. Prediction (testing)

In the testing configuration, input is taken from an RGB-D image. To perform a prediction from real-world data, the 2.5D image can be segmented in order to extract the structure to be deformed. This can be performed using existing approaches, such as [35]. The segmented body can then be upsampled to form a 3-D voxel representation of the object to be traversed. In addition, the material of the body can also be supplied as one of the conditional parameters and obtained from different methods, i.e., the recent Differential Angular Imaging for Material Recognition (DAIN) network [7], trained on the GTOS (Ground Terrain in Outdoor Scenes) material reflectance database composed of 40 surface classes. A prediction on a single Titan X GPU takes under a second, orders of magnitude faster than an FEM simulator.

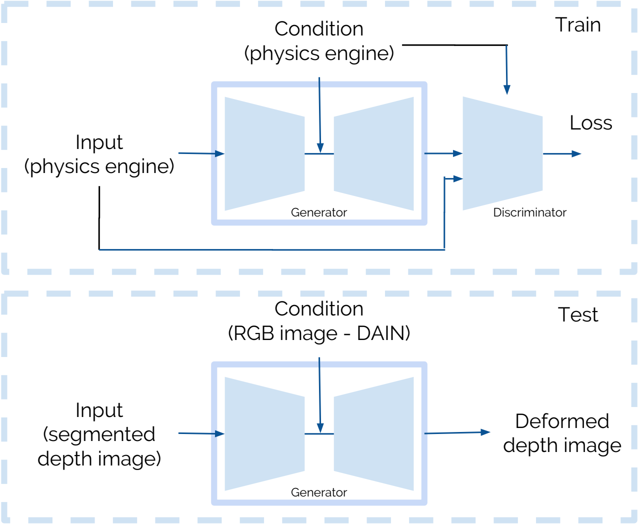

Fig. 3. Training and testing configurations.

## IV. Experimental evaluation

We test an application of Defo-Net to the problem of predicting deformation of traversable bridges and non-rigid terrain. In our experiments we use a Turtlebot 2 platform equipped with a Microsoft Kinect camera. The Turtlebot 2 robot weighs 6.3 kg itself and can carry a maximum payload of 5 kg. The mobile platform has two side wheels and two small castor wheels in the front and back. In our scenario we assume that the robot is able to segment different materials using DAIN [7]. For the navigation part we use the algorithms available in the Robot Operating System (ROS).

### A. Scenario 1: Safety assessment

In this experiment we consider a scenario in which a robot has to choose whether to cross a bridge and assess whether it is safe to do so by predicting the maximum deformation of the bridge under a known load (payload plus weight of robot). If the predicted deformation is too large compared with the ground clearance of the robot, the path is considered unsafe. We show four different cases: the robot without and with a payload crossing a bridge of length 0.6 m made of either plywood or aluminium. In our experiment, as shown in Figure 4, the thickness of the bridge is similar for the two materials. Note however that it would be possible to add the material thickness as another conditioning variable.

The robot first acquires a depth image of the bridge by facing it from the side and extracting it from the depth image.

We compare the predicted deformations with the ground truth from a Kinect camera placed transversally to the bridge, and we let the robot drive over the bridges with and without a payload to obtain the ground-truth. Figure 5 shows the results for the four different cases. We can see how DEFO-NET is able to predict the material deformation under different loads applied to different parts of the bridge. Moreover, it can be seen how the network is able to reconstruct the full 3-D object from the single input depth image.

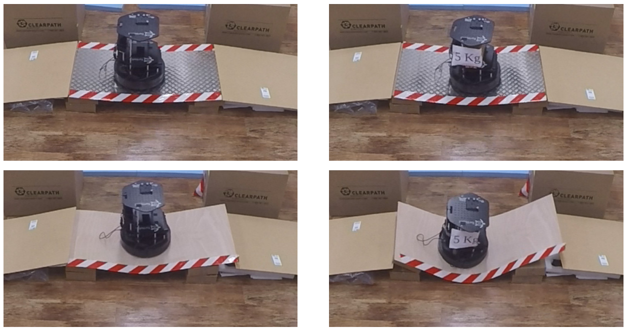

Fig. 4. The bridge-like scenario.

In this experiment we want to determine if crossing a particular bridge under a particular payload is safe. We define it as safe if the local curvature of the board is below the maximum ground clearance of the robot (0.015 m). From the ground truth in Figure 5 we can see that the first three cases are safe, while in the last one (wooden bridge with a payload) the deformation is too large and the robot would become stuck. For comparison, the simulation time for a single deformation is 12m 45s on an Intel Xeon 2.8 GHz CPU while the prediction time is 1 s on a Nvidia Titan X GPU. The path planner can then take this information into account to decide which is the optimal and safest path to a target destination. This experiment demonstrates that by being able to predict the extent of the deformation, a potentially unsafe trajectory can be avoided. However, when unloaded, the robot can safely travel over a wooden bridge - that is to say, the wooden bridge should not always be avoided, only when the robot is fully laden. In Table I we report the Root-Mean-Square error (RMSE) error for the maximum deformation at different point of the bridge with respect to the ground-truth for each case. In the case of plywood with payload, the real deformation is larger because the bridge collapsed under the weight slipping out of the supports.

| | 10cm | 20cm | 30cm |
|---|---|---|---|
| Wood - no payload | 0.1 | 0.4 | 1.5 |
| Aluminium - no payload | 1.2 | 0.7 | 1.3 |
| Wood - payload | 2.2 | 0.8 | 9.0 |
| Aluminium - payload | 0.8 | 1.4 | 2.7 |

TABLE I

RMSE ERROR (CM) BETWEEN THE PREDICTED MAXIMUM DEFORMATION OF THE BRIDGE AND THE GROUND-TRUTH AT DIFFERENT LOCATIONS.

### B. Scenario 2: Finding the fastest route

We show the performance of the network applied to soft materials like foam, in order to show how DEFO-NET is able to learn localized deformations from distributed forces. The experimental setup is shown in Figure 6. In this scenario a robot has to travel from its position to a predefined goal. Two different paths to the goal are available: we let the robot decide between a path containing a soft floor represented in our experiment by a foam board and another path containing only a hard floor but with a longer travel time. The foam board is easily segmented using RGB-D images and can be identified by DAIN as such.

We can predict how the soft floor material is deformed under different payloads for the driven and castor wheels, and based on this prediction make a decision on the maximum speed that the robot can achieve on the soft ground without getting stuck. Based on the maximum speed the robot is able to chose the best path, again based on the ground clearance of the robot, by simply marking the unsafe areas as obstacles for the sake of this experiment. In order to achieve a sufficient resolution, we predict the deformation around the point of contact of each wheel separately.

We show the predicted deformations for the front and side wheels in Figure 7, and compare them with the ground truth from the FEM simulation. We can see that the network is able to predict accurate deformations in the presence of different contact areas. More, importantly, it is able to correctly predict an excessive deformation of the foam for the castor wheels with the full payload. In this case, the robot would be unable to steer and could get stuck. Using this predicted information, the robot can safely avoid the foam when carrying a full payload. Note however, that when unloaded, the robot can choose the shortest path, i.e. over the foam. Thus, it can be seen how knowledge of the deformation of the terrain can greatly assist in path planning.

### C. Scenario 3: Generalisation ability

Finally, in order to test the generalisation abilities of our network, we further train the network on the same bridge-like structure on seven different lengths ranging from 0.8m to 1.3m. Then we test the predicted deformation on two lengths that were chosen at random and are not part of the training set (namely 0.9m and 1.2m), with two different forces acting on the middle of the bridge (robot without payload and with payload). Figure 8 shows the predicted deformations along with the ground-truth from the FEM simulator for reference. The results show how the generative network is able to reconstruct unseen voxel grids from partial depth images of the test set. The absolute error on the point of maximum deformation of the predictions with respect to the ground-truth is one voxel (2.2cm) for the bridge of length 0.9m and 2 voxels (4.4cm) for the 1.2m bridge.

This experiment demonstrates that DEFO-NET is able to generalize to unseen, yet related, scenarios. This is of key importance as it is impossible to show the network every possible structure that could be encountered in the real-

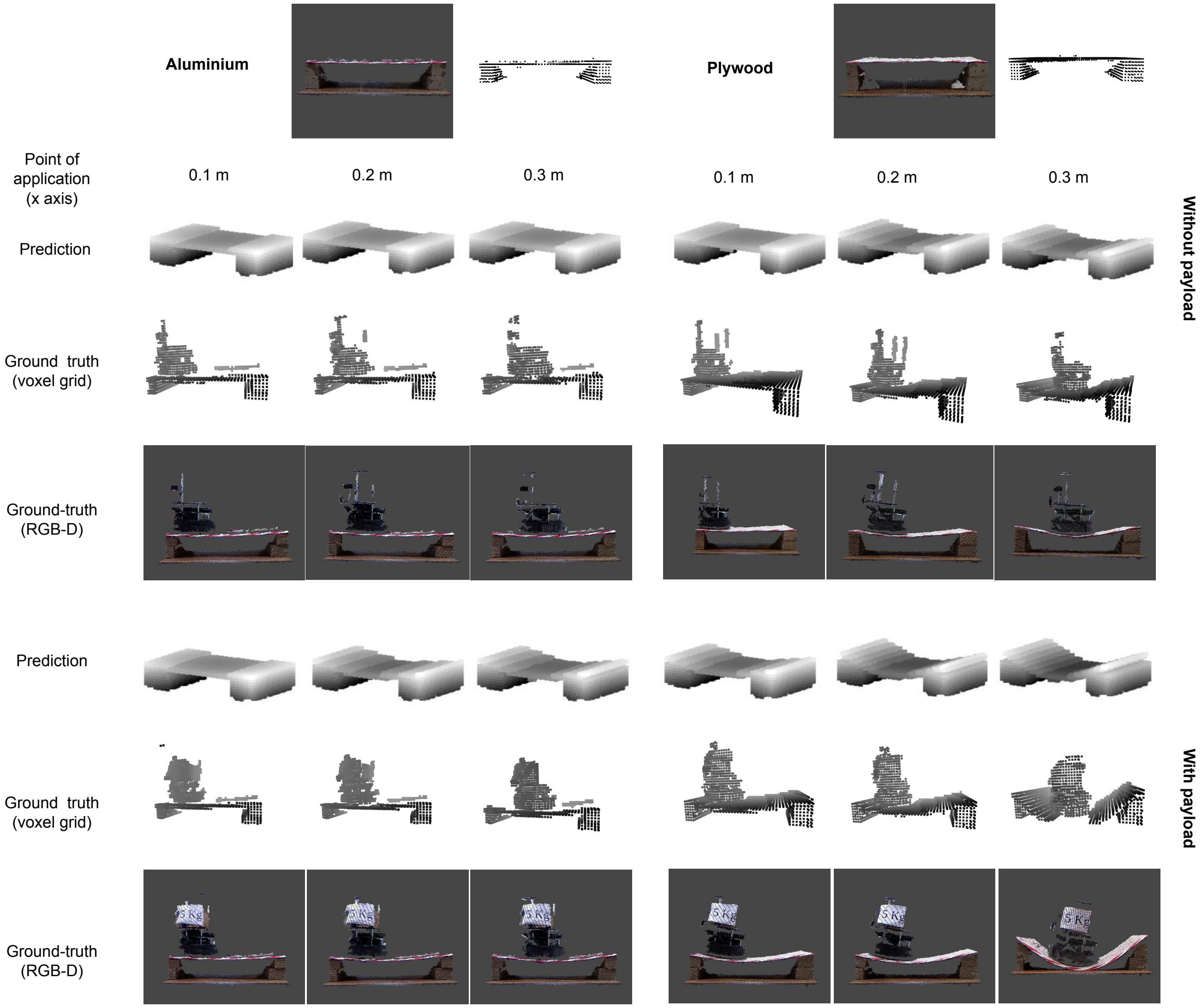

Fig. 5. Results for the first experiment, where we predict the deformation of a simply supported bridge. We examine two different materials: aluminium (left side of the image) and plywood (right side of the image), with no payload (top half of the image) and with a payload (bottom half of the image). The two input voxel grids for the two materials, along with the 2.5D image, are shown in the top row.

world. However, the network has learned the relationship between structure, material and force to predict deformations.

## V. Conclusions

We presented Defo-Net, a generative network for predicting 3-D deformations of bodies extracted from single RGB-D images using invertible conditional GANs. We applied the network to the problem of safe and optimal navigation of robots carrying payloads over different obstacles and ground floor materials. Experimental results in a real robotic scenario showed the generalisation potential of the approach to previously unseen body configurations. More importantly, the prediction can be up to 2-3 orders of magnitude faster than an FEM simulation, making it suitable for real-time navigation. Although this work has set out a new approach towards tackling an active research area, a number of extensions could be considered.

The first is to build models of more realistic and complex structures. Moreover, we have only considered a small subset of materials (e.g. wood, foam, aluminium), but it would be interesting to see how to treat say plywood differently from solid wood. It would also be interesting to see if the proposed approach could accurately model non-homogeneous media such as pebbles and sand and approximate deformations due to grain interactions.

The second is to consider non-rigid (e.g. soft-body) robots interacting with non-rigid objects e.g. a soft robot folding a blanket or closing curtains.

Another direction would be to investigate dynamic de-

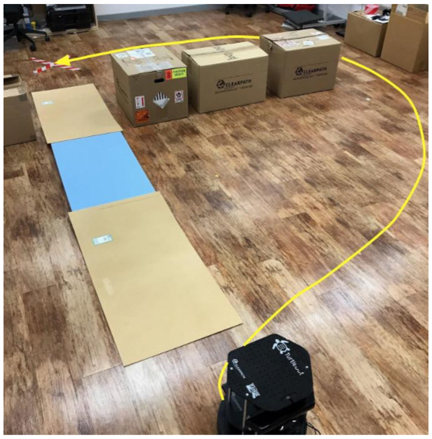

Fig. 6. The setup for the second scenario. The goal is shown as a red cross in the image. Two paths are available to the goal; the shorter one contains soft ground, the longer one is entirely hard floor. When carrying a payload, the robot avoids travelling over the foam.

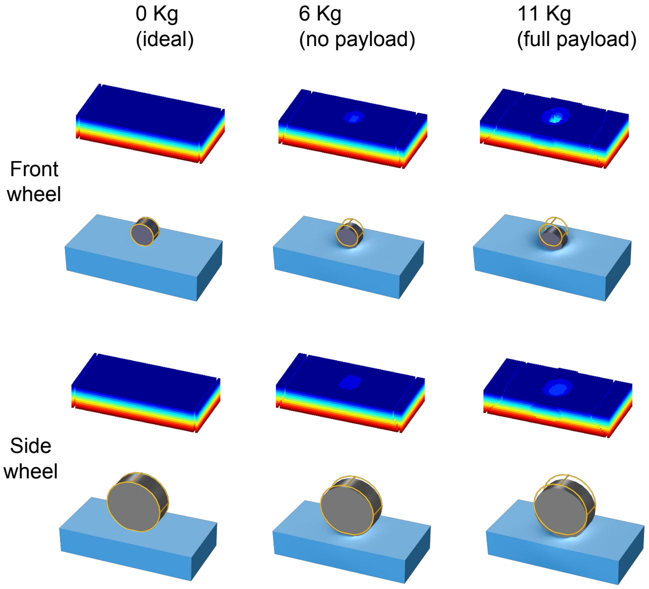

Fig. 7. Results for the second scenario. First two rows: front castor wheel; last two rows: side wheel. For each payload we show the predicted deformations and the ground truth deformations from the simulator for reference. For visibility, the depth is shown in false colors in the predictions.

formations such as those that would be caused by a robot applying a time-varying force to a non-rigid object, rather than the static loads considered here.

A further area of research would be to consider alternative world representations. Although voxels map neatly to RGB-D images, they are not a natural representation of deformable solids. Non-Euclidean approaches based on graph geometry and manifolds would be a better fit to the polygonal meshes used in FEM simulation.

## VI. Acknowledgements

This work has been partly supported by the EPSRC Program Grant "Mobile Robotics: Enabling a Pervasive Technology of the Future (GoW EP/M019918/1)".

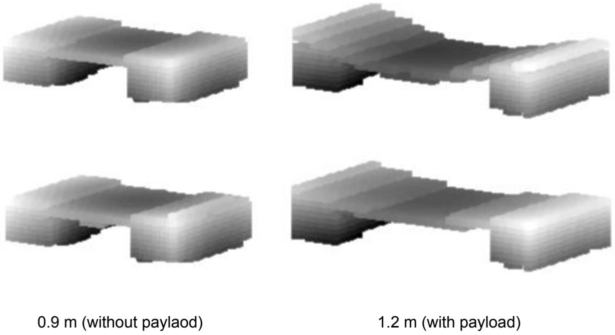

Fig. 8. Results for the third scenario. The top row shows the predicted 3-D shapes, while the bottom row shows the ground-truth.

## References

[1] M. C. Gemici and A. Saxena, "Learning haptic representation for manipulating deformable food objects," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 638–645.

[2] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 177–184.

[3] A. Davis, K. L. Bouman, J. G. Chen, M. Rubinstein, F. Durand, and W. T. Freeman, "Visual vibrometry: Estimating material properties from small motion in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5335–5343.

[4] M. E. Yumer and N. J. Mitra, "Learning semantic deformation flows with 3d convolutional networks," in *European Conference on Computer Vision (ECCV 2016)*. Springer, 2016, pp. –.

[5] B. Bonev, L. Prantl, and N. Thuerey, "Pre-computed liquid spaces with generative neural networks and optical flow," 2017.

[6] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3479–3487.

[7] J. Xue, H. Zhang, K. Dana, and K. Nishino, "Differential angular imaging for material recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[8] B. Frank, C. Stachniss, R. Schmedding, M. Teschner, and W. Burgard, "Learning object deformation models for robot motion planning," *Robotics and Autonomous Systems*, vol. 62, no. 8, pp. 1153 – 1174, 2014.

[9] B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard, "Learning deformable object models for mobile robot navigation using depth cameras and a manipulation robot."

[10] D. Navarro-Alarcon and Y.-h. Liu, "Uncalibrated vision-based deformation control of compliant objects with online estimation of the jacobian matrix," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 4977–4982.

[11] J. Chevrie, A. Krupa, and M. Babel, "Online prediction of needle shape deformation in moving soft tissues from visual feedback," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2375–2380.

[12] D. G. Dansereau, S. P. N. Singh, and J. Leitner, "Interactive computational imaging for deformable object analysis," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4914–4921.

[13] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1130–1137.

[14] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in neural information processing systems*, 2015, pp. 127–135.

[15] A. Lerer, S. Gross, and R. Fergus, "Learning physical intuition of block towers by example," *arXiv preprint arXiv:1603.01312*, 2016.

[16] W. Li, A. Leonardis, and M. Fritz, “Visual stability prediction and its application to manipulation,” *arXiv preprint arXiv:1609.04861*, 2016.

[17] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi, “Newtonian scene understanding: Unfolding the dynamics of objects in static images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3521–3529.

[18] B. Zheng, Y. Zhao, C. Y. Joey, K. Ikeuchi, and S.-C. Zhu, “Detecting potential falling objects by inferring human action and natural disturbance,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3417–3424.

[19] J. Wu, J. J. Lim, H. Zhang, J. B. Tenenbaum, and W. T. Freeman, “Physics 101: Learning physical object properties from unlabeled videos,” in *BMVC*, 2016.

[20] X. Wang, A. Farhadi, and A. Gupta, “Actions˜ transformations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2658–2667.

[21] C. Finn, I. Goodfellow, and S. Levine, “Unsupervised learning for physical interaction through video prediction,” in *Advances in Neural Information Processing Systems*, 2016, pp. 64–72.

[22] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in *NIPS*, 2016.

[23] A. Byravan and D. Fox, “Se3-nets: Learning rigid body motion using deep neural networks,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 173–180.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

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

[26] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, “Toward controlled generation of text,” 2017.

[27] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[28] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, *3D Object Reconstruction from a Single Depth View with Adversarial Learning*. United States: IEEE, 8 2017.

[29] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *CoRR*, vol. abs/1701.07875, 2017.

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein gans,” 2017.

[31] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, “Invertible conditional gans for image editing,” *arXiv preprint arXiv:1611.06355*, 2016.

[32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *arxiv*, 2016.

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv preprint arXiv:1703.10593*, 2017.

[34] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Generative and discriminative voxel modeling with convolutional neural networks,” *arXiv preprint arXiv:1608.04236*, 2016.

[35] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” *arXiv preprint arXiv:1612.00593*, 2016.