2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Measuring Bending Angle and Hallucinating Shape of Elongated Deformable Objects

Piotr Kicki[1], Michał Bednarek[1], Krzysztof Walas[1],

Abstract— **Many objects in a human-made environment have elongated shapes for easy manipulation and grasping. As humanoid robots are working in this environment, they require proper sensing and perception of such objects. Current approaches are providing mainly the perception of rigid objects, but many everyday items are non-rigid and more challenging to track due to their substantial shape variability. We want the robots to be able to grasp and manipulate thin, elongated, deformable objects. We propose a system based on the Deep Neural Network that can predict the bend angle of such objects using the single RGB image only. In our paper, we present the proposed neural network architecture used for prediction of the bending angle and finding the elongated shape in images with a cluttered background together with the dataset used for training. We observed that the proposed system even though it was trained on synthetic data was able to perform well on real data. The proposed architecture also provide us with the ability to hallucinate how the deformable pipe with any initial bend would look like when subjected to the arbitrary bend angle. Our findings have more profound consequences than the above mentioned. We were able to show that the proposed Encoder-Decoder neural network architecture has the *interpretable latent vector element* for describing a *measurable physical bend angle*. Moreover, we allow bending arrows to be situated out of the image plane. In the future work, we are planning to extend the current approach with the prediction of the full 3d shape of the elongated object from a single RGB image.**

I. INTRODUCTION

Many objects in human-made environments have elongated shapes as this allows easy grasping and manipulation of such objects. There is a biological evidence [1] that the elongated visual stimuli can lower ambiguity during grasp preparation as it provides a coarse cue to hand shaping and orientation that is sufficient to support an action planning. As humanoid robots are performing grasping and manipulation in human-made spaces, they require proper sensing and perception of such objects. In our case, we are providing this ability to the robotic system through the use of Deep Neural Networks, which are state of the art AI techniques used in the machine vision.

Object tracking and measuring are common research themes in the computer vision [2], [3]. The results of such research can be directly applied to humanoid robots performing the manipulation or grasping. However, in most cases, the problem is only solved for rigid objects. Whereas, many everyday objects are non-rigid and are more challenging to track due to their substantial shape variability. It is typical to tackle the non-rigid object tracking with the use of RGB-D cameras [4] that gives explicit information about the position of the object's elements. However, there is a question if we could perform tracking and measuring of such objects with a single camera for example when the robot is equipped with a stereo pair and the object is texture-less (no correspondences to establish disparity map)?

In our work, we provide a system, which allows for the tracking and measuring the bending angle of elongated, thin and deformable objects from a single RGB camera. The goal of the presented system is to measure the bend angle of the hose (flexible pipe), based on two images (before and after deformation), in the real time. To implement that, we proposed the Encoder-Decoder neural network architecture with an *interpretable latent vector element* for describing a *measurable physical bend angle*. Latent space can be associated with the feature space, but only one element is forced to express the feature understandable to humans. This approach sheds the new light on the *understanding of the representation transformations* taking place within artificial neural networks and allows one to *generate (hallucinate) images* with the control over non-trivial image features, i.e. bend angle. We also contribute through providing the new dataset for an assessment of robot vision systems dealing with elongated objects that will be soon publicly available. An essential feature of the proposed solution is the fact that although the entire training procedure was carried out on synthetic data, the test results on real data are still of high quality. Moreover, we allow bending arrows to be situated out of the image plane.

Presented solution can be used in a plethora of tasks in the robotic manipulation. Besides monitoring the bent level, estimation of the bending angle together with measurement of forces applied to the deformable object can be used for predicting material properties. From the other hand, generative abilities of the presented model can be used to validate the measurements of the bent angle in the presence of occlusions which is important in single camera vision systems. Moreover, hallucinating the desired object shape can be used in the visual servoing.

In the remainder of the paper, we will first present related work. Then, we will describe our dataset consisting mostly of synthetic images (training stage) and real data (testing stage). Next, we will focus on the deep neural network architecture with an interpretable transformation of the image representation. Then, we will describe our results and finally concluding remarks will be given.

[1]Piotr Kicki, Michał Bednarek and Krzysztof Walas are with the Institute of Control, Robotics and Information Engineering, Poznan University of Technology, Poznan, Poland piotr.kicki@student.put.poznan.pl, michal.gr.bednarek@doctorate.put.poznan.pl, krzysztof.walas@put.poznan.pl

978-1-5386-7282-2/18/$31.00 ©2018 IEEE

II. RELATED WORK

In the recent robotics literature, one can find a couple of examples of robotics systems dealing with deformable, thin and elongated objects. There is a publication on the knot tying [5] with two arm PR-2 robot. Additionally, one can find works on this topic, where the generic physics engines [6] or specific ones [4] are used for tracking deformable objects (including ropes) using point clouds. A separate topic is a robot control when manipulating an elastic rod [7]. There a theoretical background to the execution of such a task was provided. Such a control system requires an input from the vision system like the one that we present in this paper.

Additionally, thin objects have also gathered some attention and special treatment in computer graphics. Thin structures are investigated in image-based rendering [8] or are handled with much care in the point-based 3D reconstruction [9]. Latest paper on the reconstruction of thin structures of manifold surfaces is focused on the use of spatial curves [10]. The main problem with thin objects is that they are composed of a couple of pixels (points), which in many cases are missing (no object continuity) due to the missing data from 3D sensing devices such as passive and active stereo cameras [11]. Therefore, in our approach, we focused our attention on the single RGB camera setup.

Also, there are other application areas, besides robotics, where is a need of segmenting elongated objects, e.g. medical research – cell segmentation [12]. Additionally, the detection of thin elements is of particular interest in a visual inspection of infrastructure as it is used for detecting cracks in concrete or steel structures [13]. Especially in checking the road condition [14] or assessing the state of nuclear power plant reactors [15].

The problem described in this paper is highly related to the computer vision, where fundamental object transformations, like rotations or translations, have a solid foundation in the mathematics [16]. Similarly, the deformation of the surface of elastic objects is well described using the thin plate spline transformation proposed in [17]. However, that method requires two sets of points from a base and deformed image, which are in general relatively hard to obtain. Our approach is far more specific and less formal than the thin plate spline transformation, but much easier to apply. Concerning Deep Neural Networks and rigid body transformations representation, especially rotations, an interesting approach was proposed by Worrall et al. [18]. They encoded the image of a rigid object (no shape change) at the specific rotation, along with a selected rotation axis, into a latent vector, which can be rotated the same way as vectors in the Euclidean space, allowing to rotate the object in the image. Based on this assumption, in our work, we tried to describe the bend angle of the deformable hose (shape change) in a similar manner.

The work focused on a similar application, a problem of bending elastic objects, but using a different approach, 2D contours and grids drawn on the objects was described in [19]. The authors used classical segmentation algorithms, no learning part, to distinguish an object on the scene, but they focused on the deformations where the bending arrows of the photographed objects lay only in the plane perpendicular to the optical axis of the camera.

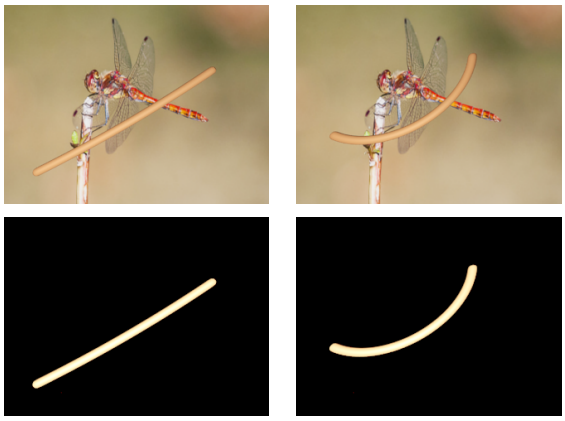


Fig. 1. Single synthetic dataset element: (top left) base pipe $\zeta_B = 1\,\mathrm{rad}$; (top right) deformed pipe $\zeta_D = 2.3\,\mathrm{rad}$; (bottom left) base pipe ground truth mask, (bottom right) deformed pipe ground truth mask.

III. DATASETS

In this paper, we present three datasets: Synthetic I (S1), Synthetic II (S2), both created in Blender, and Real (R) created manually with a phone camera. The S1 set is our basic dataset, which is meant to be used as a training set. S2 is a more complex set, which is meant to be used for the validation of generalisation abilities. Finally, R is a small testing dataset, which contains only real data samples. We provide these datasets, as to best of our knowledge there is no similar set of data providing a possibility of testing robotics systems working with thin, deformable, elastic objects. The use of synthetic data is indispensable in training data hungry Deep Neural Networks, and it is currently widespread practice in the Machine Learning [20]. Synthetic data generation gives us ease of providing ground truth data and full control of the environment. This is of course at the cost of having the reality gap, which might not be compensated by the trained system.

A. Synthetic Images Datasets

Set S1 contains 10000 different pipes in 26 bend angles each $(0, 0.1, ..., 2.5\,\mathrm{rad})$. The diversity in that dataset is achieved due to the randomly chosen background from ILSVRC2015 [21] dataset, pipe colour (each component is selected from the range 26-230 in a 256-level scale), position, orientation (selected from a range from -45 to 45 degrees for out of the plane rotations and -180 to 180 degrees for a planar rotation), diameter, length, glossiness and the light source position.

Set S2 contains 13600 different pipes in 26 bend angles each $(0, 0.1, ..., 2.5\,\mathrm{rad})$, it shares the backgrounds with S1, but it is more complex due to the second light source and broader ranges of pipes widths and lengths.

For training purposes, every dataset element contains four images: 2 images of pipes (with different bend angles) on

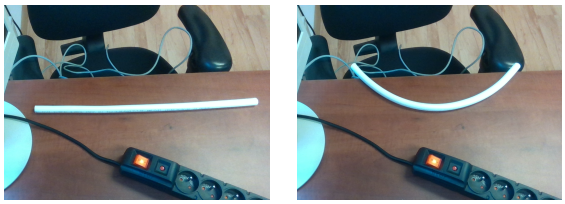


Fig. 2. Single real dataset element: (left) base pipe $\zeta_B = 0.25\,\mathrm{rad}$; (right) deformed pipe $\zeta_D = 1.76\,\mathrm{rad}$.

backgrounds gathered from ILSVRC2015 images and two ground truth masks, with ground truth bend angles. Examples of the generated dataset elements are shown in the Figure 1.

Generated sets were split into three subsets train, validation and testing. These subsets were disjoint both in background images and bend angles. Two different divisions of the datasets were proposed:

I Tr Va Te Tr Va Te ...

II Tr Va ... Tr Va Te ... Te Tr Va ... Tr Va

where: Tr stands for the training set element, Va – validation and Te for the test and each subsequent elements had $0.1\,\mathrm{rad}$ difference. Through that splits authors wanted to verify the network ability to generalise over bend angles. In split I the concern was "how at least $0.3\,\mathrm{rad}$ difference in the training dataset will affect the quality of transformation of pipes, which lies in between", whereas in split II "if it is possible to generalise from angles in range $[0.0; 0.6] \cup [1.7; 2.5]$ to angles between those two intervals. Each subset, in both configurations, contains about 40000 labelled pairs of images.

B. Real Images Dataset

The collection of images of real pipes R is a key element in verifying the performance of the proposed system. It is organised in 8 groups, in which in the same conditions, the pipe with different bend angles was pictured. It consists of 47 images in total, for which it is possible to generate 164 different pairs of base and bent pipes (due to combinations of pipes in the same group), with pipes bent at an angle from 0–$1.76\,\mathrm{rad}$. As the entire dataset is used for testing purposes, this set does not contain the ground truth masks for input images. Therefore, a single dataset element consists of only two images (e.g. see Figure 2) and information about the bend angle in radians.

IV. INTERPRETABLE NEURAL TRANSFORM

In general, it is desirable to understand transformations introduced by any system, which we are about to use. This understanding is crucial in robotics applications, where interaction with the physical world is performed. Understanding becomes more difficult, the more complex the system is. Moreover, it is tough for humans to understand the meaning of vectors in some arbitrarily chosen vector space. A great example of such complexity is the deep neural network, where transformations change every weights update, and created spaces often do not have intuitive and visible interpretations. There was much work done, especially in image processing field to understand how the neural network process the picture, which regions are critical and what is the meaning of particular filters in hidden layers [22]. A noticeable step in that field was made by Worrall et al. [18]. They introduced the neural network, in which the output can be transformed with the use of the well-known mathematics apparatus. The image of a rigid object (no shape change) at the specific rotation is encoded into a latent vector, which can be rotated the same way as vectors in Euclidean space allowing to rotate the object in the image in a controlled way. In our work, we decided to go even further and to train the neural network to produce at the output the representation of the object in a multidimensional space, in which one axis is imposed by the authors and have clear physical interpretation. Unfortunately, that interpretable feature is biased, thus the base image is required as the reference.

A. Network architecture

To tackle the deformation of the elastic object, particularly the bend estimation, the Encoder-Decoder architecture extended by the additional interpretation layer was proposed (bender network). The overview of the network architecture is shown in the Figure 3.

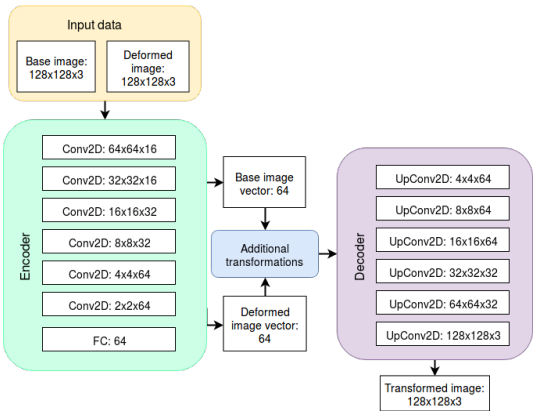


Fig. 3. Bender network architecture. Conv2D stands for 3x3 convolution layer with ELU(exponential linear unit) activation function and max-pooling layer. UpConv2D stands for transposed convolution layer with ELU activation function, strides (2,2) and 3x3 kernel.

The network has two inputs for the reference image and the deformed image. Two images are passed through the Encoder to obtain the latent vector representation. Additional transformations block is the key point in our neural network architecture, because it allows enforcing a desired latent vector configuration. In a basic scenario that block is used for the computation of the difference between the base image and deformed image vectors along the selected feature, which is the estimate of the bend angle difference. Moreover, it allows to transform the base image latent vector to obtain an image of the same pipe, but with changed bend angle. That generative behaviour is obtained by adding an angle (expressed in radians) to the selected element of the latent vector. We have a direct, measurable control over the bending angle.

Unfortunately, it is hard for the bender network to simultaneously learn how to represent the angle and the image background in the latent space. To mitigate that difficulty, we proposed splitting the challenge into two stages: extraction of the pipe from nontrivial background then processing the pipe image with the neutral background. To perform the mentioned extraction, UNet [23] like architecture was proposed. That neural network called *Mask Generation Network* processes two RGB images stacked together along with the third dimension and produces two binary masks. The architecture is shown in the Figure 4.

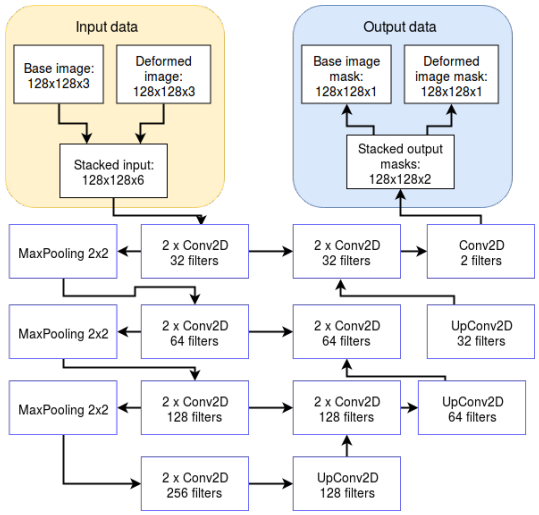


Fig. 4. Mask generation network architecture. Conv2D stands for convolution layer with leaky ReLU activation function 3x3 kernel (last convolution: 1x1 kernel). UpConv2D stands for transposed convolution layer, strides (2,2) and 3x3 kernel. Two arrows pointing one block means that block operates on stacked input tensors.

B. Deformation Estimation and Image Generation

Tasks specified in the title of this section are computed in the different parts of the network, but they are closely related to each other. That relation is visible during the training phase when gradients returned by the Decoder (which is responsible for image generation) accelerate the training of imposed image representation created in the Encoder.

The following formula gives the estimation of the pipe bend angle:

$$\hat{\zeta} = \mathrm{En}(\Upsilon_B)_1 - \mathrm{En}(\Upsilon_D)_1, \tag{1}$$

where: Υ_B and Υ_D are the base and deformed image consecutively. En is a transformation function introduced by the Encoder. In that particular setup, first elements of both vectors were chosen, but in general, it can be any element (the same for both vectors) due to the usage of fully connected layer at the end of the encoder. The second part of the proposed system is responsible for the image generation from the interpretable latent vector. The further insight into these to parts of the system is given in the next two sections.

C. Training procedure

Training procedure consists of two stages:

1) mask generation and bender training,
2) training of end-to-end network (pre-trained mask generation stacked at the top of pre-trained bender).

The first stage consists of two independent training procedures of mask generation and bender nets. It can be performed in parallel. In the mask generation network, the logarithmic loss function was used to produce a belief map. It has to decide whether the pixels belong to the pipe or not. For the bender, the training loss is a sum of two factors: image decoding and latent vector loss. The image decoding loss is responsible for achieving the accurate image generation, while the latent vector loss penalises the relative pipe bend estimation error. It is defined as a sum of squares and absolute difference between the estimated $\hat{\zeta}$ and the ground truth value of the bend angle ζ.

Image decoding loss was defined as the Euclidean norm of the element-wise difference between target I and the generated image $\hat{I}$. $Loss_1$ is given in the Equation 2. It was investigated that the L2 norm gives better results than L1 [24]. The likely reason for that is the fact that the decoder was learning faster. Because of that, it managed to learn a certain invariance along the selected dimension, so it was not able to propagate sufficiently large gradients back to the Encoder.

$$Loss_1 = \|\hat{I} - I\|^2 \tag{2}$$

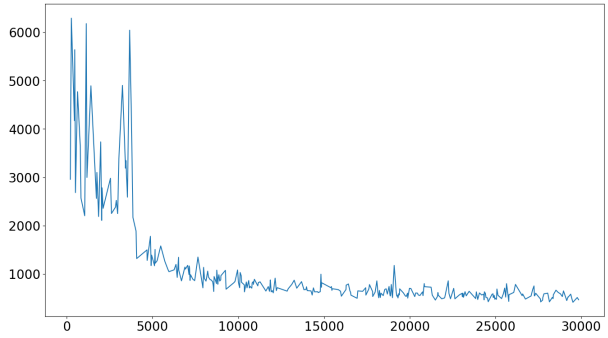
In the bender training phase, the selected element of the latent vector of the base image ζ_B is translated along a specified dimension by the $-\hat{\zeta}$, which should be relatively close to ζ_D and then it is decoded. The result of the decoding operation should generate a pipe similar to the deformed pipe. Latent vector loss function is defined as a $Loss_2$ in the Equation 3. Total loss function for the bender architecture is defined as a sum of $Loss_1$ and $Loss_2$. Figure 5 presents the image decoding and latent vector losses for the validation dataset during the bender training. Both charts illustrate that after approximately one epoch there was a sudden drop in the loss values. It can be interpreted as the proper identification of a physical trait, which we have imposed as the one that should be represented by the specified dimension of the latent vector.

$$Loss_2 = (\zeta_B - \hat{\zeta} - \zeta_D)^2 + |\zeta_B - \hat{\zeta} - \zeta_D| \tag{3}$$

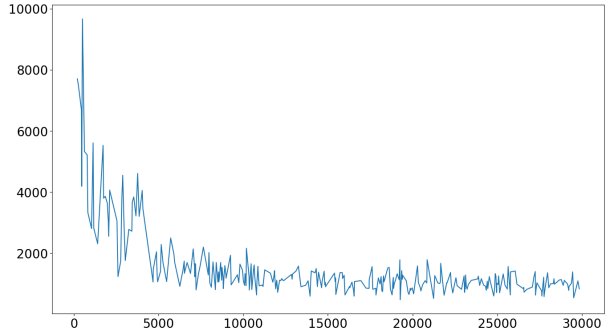
The second stage of the system learning is focused on the training of the end-to-end system using only bender losses to subordinate the mask generation performance to the bender. That stage was the longest one, whereas separate bender and mask generation training lasted about 1-2 hours, the collaborative training could take up to 8 hours.

V. EXPERIMENTAL RESULTS

Experimental verification was taken in four phases:



(a) image decoding loss



(b) latent vector loss

Fig. 5. Bender loss for validation dataset in number of training steps.

Fig. 6. Bender network test: (left) reference pipe, (middle) bent pipe, (right) reference pipe bent to the same angle as bent pipe.

1) verification of bender network with the use of pipes on black (neutral) background,
2) tests of the whole network (bender+mask generation) on synthetic and real datasets,
3) tests of the whole network (bender+mask generation) on a real dataset,
4) tests of the generative abilities of the network.

A. Bending pipes on the black background

That stage was a preliminary procedure to prove that the idea of the interpretable neural transformation can provide a correct bend angle difference between pipes. Achieved bending abilities are depicted in the Figure 6. Mean of absolute errors are below $0.02\,\mathrm{rad}$. The results of this test proved that it is possible to estimate the bend angle and manipulate the pipe bend angle.

B. Bending pipes on the ILSVRC background - synthetic datasets

In this part of the experiments, the quality of the merged mask generation and bender model was tested. The results of

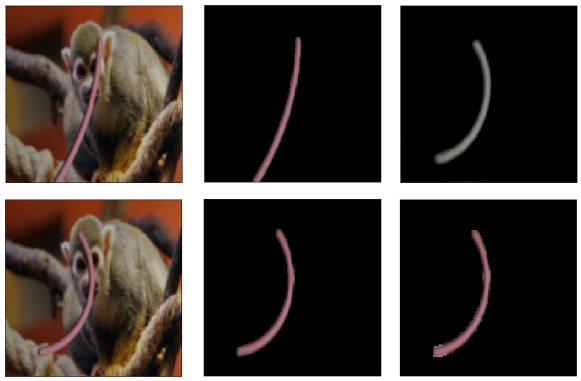


Fig. 7. Test case from S1 dataset: (left top) reference pipe, (top middle) reference pipe with masked out background, (top right) reference pipe bent to the same angle as bent pipe, (bottom left) bent pipe, (bottom middle) bent pipe with masked out background, (bottom right) ground truth bent pipe without the background.

that tests are shown in the Table I with data splits presented in the Section III-A. Our solution exhibits an ability to infer the correct behaviour on previously unseen data based on similar examples, which differ from each other in the pipe bend angle, colour, glossiness, light source position and background picture (S1). Mean Absolute Error (MAE) level reached on S1 suggests, that trained mask generation introduces almost no overhead in the final results. It is visible that changes in the data (S2), such as a second light source and more diverse pipes resulted in worse scores. The reason for such a significant difference is, that those changes affect the procedure twice, both the mask generation and bender. Exceptionally high level of centile 95 and standard deviation, combined with a small median of errors suggest that, if the pipe differs significantly from the pipes from the S1 dataset, the generalisation abilities of our solution are limited. That data gap shows how important is to expose the network during the training on synthetic data, which covers a diversity of the testing dataset. That gap simulates the expected behaviour, generalisation abilities of our solution on datasets, which differs substantially from the training set. Example of the test case is shown in the Figure 7.

C. Bending real pipes

In last, but not least, part of angle estimation experiments the real pipes bend angle estimation was considered. MAE values achieved in that setup are slightly higher than those obtained on the S2 dataset, but standard deviation and centile 95 is twice smaller (see Table I). Furthermore, MAE obtained for S1, and S2 datasets differ significantly between the two splits, what is not observed for the R dataset. Those phenomena combined with the very high median suggest the constant bias in the whole R dataset. It can be assumed that some not modelled features of natural environment had the significant impact on the system accuracy. One of those features can be the light colour, which was fixed in the training dataset and varying in the real dataset. Other differences are in the strong assumptions made in synthetic

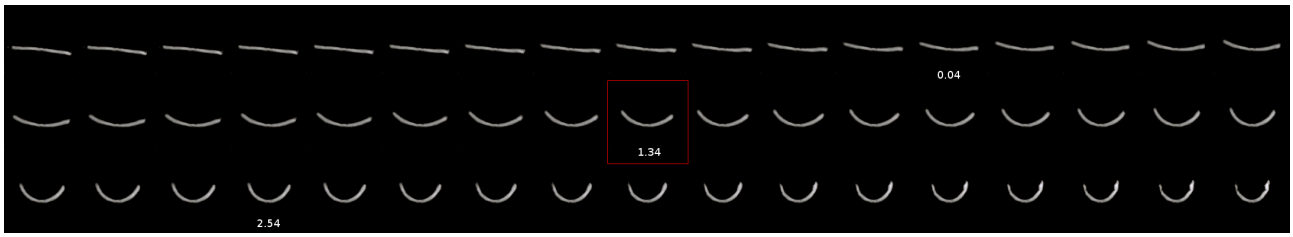


Fig. 8. Sequence of generated images of bent pipe (base pipe in red frame, consecutive pipes differs by $0.1\,\mathrm{rad}$)

TABLE I
ERROR STATISTICS OF THE BEST MODELS ON THE TEST DATA FOR I AND II DATASET SPLIT

Dataset, split	MAE [rad]	σ(MAE) [rad]	Median [rad]	95th %tile [rad]	Max. error [rad]
S1, I	0.038	0.060	0.020	0.129	1.354
S2, I	0.257	0.506	0.045	1.443	4.428
R, I	0.184	0.163	0.165	0.491	0.665
S1, II	0.019	0.025	0.012	0.061	0.850
S2, II	0.151	0.338	0.021	0.908	3.923
R, II	0.182	0.151	0.155	0.444	0.738

sets, where bent pipes were the torus segment, what was not ideally kept in the real one. Moreover, in a real dataset one can observe small changes in the environment as well as in the pipe position. All of those effects explain the worse results, but also give an insight into what traits should be covered in the training set to obtain better results. Maximum errors provided in the Table I are showing that there are some outliers, but they do not affect the MAE score significantly.

D. Bent pipes generation – hallucinating the deformation

In that part of the test, the generation abilities of the neural network were tested. Figure 8 depicts that it is possible to generate the same pipe but bent differently from a base image of a real pipe. An important insight is that bending works consistently even for the bend angles from the outside of the training set, but it is not perfectly accurate. The major constraint is that the zero level of the generation is significantly biased.

E. Environment and performance

Experiments were conducted using the NVIDIA GeForce 950M GPU with 2GB RAM and 768 CUDA cores. Processed images were re-scaled to fit the 128x128 window at the very beginning. In that setup, the inference process with the single image generation takes $0.023\,\mathrm{s}$ per images pair in the average, and it is never greater than $0.025\,\mathrm{s}$. Without image generation (the pure bend angle estimation) the average time is $0.016\,\mathrm{s}$ and no more than $0.018\,\mathrm{s}$ per images pair.

VI. CONCLUSIONS

In our work, we proposed an end-to-end system that can generate the supposed view of an elongated object subjected to the arbitrary angle of bending. First of all, to develop the machine learning algorithm, the dataset of bent elongated objects was prepared. After that, we proposed a deep neural network and performed extensive tests of it. We proved that our system can perform the localisation of the elongated object (even on the cluttered background) and then generate a view of its bent version. Moreover, the localisation and angle estimation can be run simultaneously, achieving the real-time performance. Also, in our work, we showed that through the change of one, arbitrarily chosen variable from the latent vector of the autoencoder we can influence the bending angle of such objects in the resulting image providing measurable physical quantity (in radians).

For further work, we find it very interesting to investigate more possibilities in such manipulation of the latent representation to know how the deformations are encoded in this space and how we can influence them to achieve desired results. We are also planning to extend the current approach with the prediction of the full 3D shape of the elongated object from a single RGB image and estimating a physical parameter – bending stiffness.

REFERENCES

[1] J. Almeida, B. Z. Mahon, V. Zapater-Raberov, A. Dziuba, T. Cabaço, J. F. Marques, and A. Caramazza, "Grasping with the eyes: The role of elongation in visual recognition of manipulable objects," *Cognitive, Affective and Behavioral Neuroscience*, vol. 14, no. 1, pp. 319–335, 2014.

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*. Ieee, 2013, pp. 2411–2418.

[3] D. Xu and Q. Wang, "A new vision measurement method based on active object gazing," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, p. 1729881417715984, 2017. [Online]. Available: https://doi.org/10.1177/1729881417715984

[4] A. Petit, V. Lippiello, G. Fontanelli, and B. Siciliano, "Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot," vol. 88, 09 2016.

[5] A. Lee, S. Huang, D. Hadfield-Menell, E. Tzeng, and P. Abbeel, "Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 4402–4407.

[6] 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*, May 2013, pp. 1130–1137.

[7] M. Mukadam, A. Borum, and T. Bretl, "Quasi-static manipulation of a planar elastic rod using multiple robotic grippers," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 55–60.

[8] T. Thonat, A. Djelouah, F. Durand, G. Drettakis, T. Thonat, A. Djelouah, F. Durand, G. Drettakis, T. Structures, T. Thonat, A. Djelouah, F. Durand, and G. Drettakis, "Thin Structures in Image Based Rendering To cite this version : HAL Id : hal-01817948," 2018.

[9] B. Ummenhofer and T. Brox, "Point-based 3D reconstruction of thin objects," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 969–976, 2013.

[10] S. Li, Y. Yao, T. Fang, and L. Quan, "Reconstructing thin structures of manifold surfaces by integrating spatial curves," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] T. Tang, Y. Fan, H. Lin, and M. Tomizuka, "State estimation for deformable objects by point registration and dynamic simulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 2427–2433.

[12] A. Serna, B. Marcotegui, E. Decencière, T. Baldeweck, A. M. Pena, and S. Brizion, "Segmentation of elongated objects using attribute profiles and area stability: Application to melanocyte segmentation in engineered skin," *Pattern Recognition Letters*, vol. 47, pp. 172–182, 2014.

[13] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017. [Online]. Available: http://doi.wiley.com/10.1111/mice.12263

[14] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3708–3712. [Online]. Available: http://ieeexplore.ieee.org/document/7533052/

[15] F.-C. Chen and R. M. R. Jahanshahi, "NB-CNN: Deep Learning-based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion," *IEEE Transactions on Industrial Electronics*, vol. 0046, no. 0278, pp. 1–1, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8074762/

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[17] J. Duchon, "Splines minimizing rotation-invariant semi-norms in sobolev spaces," in *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, pp. 85–100.

[18] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 4, 2017.

[19] A.-M. Cretu, P. Payeur, and E. M. Petriu, "Soft object deformation monitoring and learning for model-based robotic hand manipulation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 740–753, 2012.

[20] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *CoRR*, vol. abs/1703.06907, 2017. [Online]. Available: http://arxiv.org/abs/1703.06907

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[22] A. Mahendran and A. Vedaldi, "Salient deconvolutional networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[23] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a

[24] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, March 2017.