An Approach to Realistic Physical Simulation of Digitally Captured Deformable Linear Objects

Nahum Alvarez, Kimitoshi Yamazaki and Takamitsu Matsubara

Abstract— **Deformable Linear Object manipulation stands as an important aspect of robot operation, with wide applications in industrial and daily life. However, it is also a difficult task to perform; the problems we encounter in Deformable Linear Object manipulation are related to the object's intrinsic physical characteristics that require complex physical prediction in order to properly interact with them, like its flexibility or elasticity. One of the approaches available to researchers for obtaining manipulation plans consists in recreating the object in a virtual environment and simulating the interactions between the robot and the object, predicting its behavior using physics models. However, in order to accurately simulate the object, we need to capture the object's physical properties, something often done manually by an expert. In this paper, we present a system that automatically calculates the physical parametrization of Deformable Linear Objects for subsequent uses in simulation processes. The system uses photographs of the object behavior after certain interactions as input reference and generates a virtual counterpart of said object ready to be used in a virtual environment. We also performed an experiment with the goal of verifying the accuracy of our system and tested it in conjunction of a manipulation planer in order to assess its usefulness.**

I. Introduction

Manipulation of Deformable Linear Objects (henceforth DLO) is an important task for autonomous robots with many applications in the industry. It has been a subject of researchers from long ago and recently it has shown promising advances. Automatic generation of manipulation plans involving this type of deformable objects is difficult to perform because their physical behavior: for example, not only is necessary to calculate and avoid the potential collisions of the object with the robot's manipulator, with the environment or even with itself, but also this calculations have to be updated constantly due to its changing configuration, generating new potential collisions continuously over time. To solve these issues, researchers have at hand a number of techniques, mainly the use of machine learning methods or using physics simulation engines, and are aware of the requirement of interpreting the physical attributes of objects for manipulation models in order to obtain accurate results [1], needing a significant amount of computational power due to the complexity of predicting the object's behavior [2].

However, even with an advanced prediction engine, additional information is needed to get correct results: in the real world, having different materials or physical characteristics yield very different outcomes for the same interaction. For example, a cable and a rope have a very different movement and response to interactions. As we see, realistic simulation requires a realistic parametrization [3], so it is necessary to capture accurately the physical parameters in order to realistically recreate the behavior of a DLO. But this is a difficult task, as those parameters are difficult to obtain due to be "invisible" and it would be necessary to test the DLO extensively in different ways to get each one of them. If we think about how humans obtain a rough idea of the physical quality of an object in the real world, usually it is done by performing an initial quick interaction or observation; we know somehow that a cable is more or less flexible from seeing it moving. Thus, our question is: can we replicate this cognitive process? This question brings out our motivation: if we can develop a similar method to do that task, DLO manipulation will become more accurate, as we can provide those mechanical parameters to the simulation engine that calculates the DLO manipulation plan.

Our contribution in this paper is a system with a novel method for initial calibration that estimates the physical parameters of a DLO from a small number of photographs depicting initial interactions with the object. This system would generate a set of mechanical parameters related to the DLO needed for the performance of realistic simulation. The method we present works by simulating repeatedly a virtual DLO with the same configuration of the input photographs, each time with different physical parameters, and performs the interactions depicted in the photos, selecting the set of parameters whose instance of the object with a resulting configuration most similar to the one pictured. Once these parameters are obtained, they can be used for later simulations in order to generate manipulation plans, behavior databases or any other goal the researcher has.

This document is structured as follows: section 2 surveys the literature on deformable linear object manipulation and how to obtain its mechanical properties, describing different approaches and studies. Section 3 describes in detail our method and architecture, and section 4 shows the experiment we carried out in order to validate it. Finally, we discuss the results obtained in the experiment in section 5 and we present

*Research supported by New Energy and Industrial Technology Development Organization (NEDO).

Nahum Alvarez is with the Shinshu University, Nagano, Japan (e-mail: nahum@shinshu-u.ac.jp).

Kimitoshi Yamazaki is with the Shinshu University, Nagano, Japan (e-mail: kyamazaki@shinshu-u.ac.jp).

Takamitsu Matsubara is with Nara Institute of Science and Technology (NAIST), Nara, Japan (e-mail: takam-m@is.naist.jp)

our conclusions and our plans for future research in section 6.

II. Related work

Manipulation of flexible objects using physics simulators has been subject of extensive research in the field of robotics, from the recognition of the object to its manipulation, and aiming to generate a valid plan of actions over the object in order to put it in some goal state. A light and powerful method for defining a DLO is the Point Chain Model [4] [5], which consists on generating a DLO representation composed of an ordered list of connected points. This model simplifies the processing without losing too much information: the DLO will be able to bend over these points, and enough of them will allow an acceptable realistic simulation, minimizing the computational time required to do it. Other models uses in addition of the Point Chain Model a list of crossing points, or even uses instead the complete definition of a segmented and curved cylinder [5][6], or make use of deep learning techniques in order to label the key features of the object [7]. However, in order to allow an accurate physical simulation of the object, the point configuration of the DLO is not enough, and it is necessary to provide additional attributes like its thickness, friction, or elasticity, among others.

Obtaining the most appropriate attributes would benefit all the other tasks related to DLO manipulation. For example, the selection of the initial gripping point and how the DLO is picked determines its subsequent manipulation [8], and just with different values of one physical attribute, like for example its flexibility, a DLO would behave very different, making this parametrization an important factor for this activity, and other more complex tasks related with DLO manipulation and behavior prediction. Few models capture those parameters: even when they incorporate mechanical attributes to the simulation, they leave them as default values or as an input from an expert [9], and the works that incorporate methods for capturing them require performing a number of physical tests on the DLO beforehand [10]. The methods for capturing those attributes are varied but usually are very intrusive, being difficult to be performed "on the fly"; for example, using force-feedback devices and microphones to constructing a texture model that include mechanical parameters such as hardiness or contact sound [11], which needs to place the object in a "test station" and a specific model for the tested material, or a haptic device to determine the elasticity and deformation on soft objects [3].

However, the literature is sparse about non-intrusive methods for capturing this information: for example one technique allows to model a deformable object through analyzing video and depth data [12], but it is aimed to digitalize this data in a 3D environment on real time, avoiding subsequent dynamic simulations. Another work uses an aspiration methods to determinate soft tissue properties in surgery applications, which is designed to be performed moments before the operation [13].

On the technical side, we use a 3D simulator engine for the physics calculations and the DLO behavior's prediction. Previous research on automatic robot manipulation has taken advantage of using third party simulators, saving time and complexity [14]. Using such engines for simulating object behavior is not widely extended, being used instead for other purposes, with different degrees of physics simulation: building a sensor simulator and taking care of ray casting physics [15], using only the graphic simulation for manipulator arms but using a different physics engine, due to the specific physics calculations needed for motion planning [16], or simulating robot models and their sensors [17].

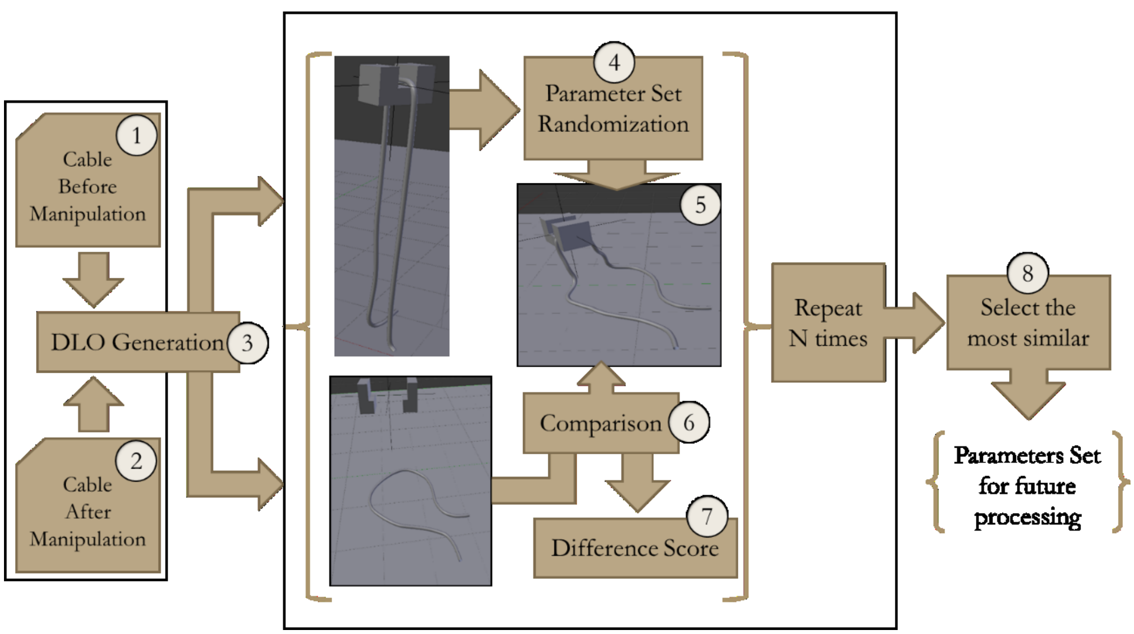

Fig. 1: Process chart for the system. The method we used is based on generating a high number of set of parameters and testing the behavior of the DLO after applying them to it. The most similar behavior will correspond to the best set of parameters, which will be selected. The numbers in the diagram reference the corresponding number in the text.

III. PHYSIC PARAMETERS PREDICTION

In this paper we present a system that uses a novel method to capture the physical attributes of a DLO in order to simulate it realistically. We developed this system for working with a virtual environment simulator of DLOs we already had created. In this environment, it is possible to recreate a DLO by providing its Point Chain Model definition formatted in an xml file, in order to perform manipulation tasks on it. The simulator uses a license free physics and rendering simulator engine that simulates DLO using a soft body physics model internally based on mass-spring system that works as a black box module, taking care of all the calculations needed to generate the behavior of the objects based in the current environment. We added also a built-in interface to define actions to manipulate the DLO, and a planner able to generate a list of steps to bring a DLO from a configuration to another.

In order to obtain the desired mechanical parameters of an input DLO, the system uses a number of pictures of the object before and after of a certain manipulation, also defined in the system. The system subsequently simulates the configuration of the object with a randomized set of parameters a number of times and selects the set that better fits the DLO's behavior. Figure 1 shows a diagram of this process, which follows the next steps:

1. First, a number of pictures representing the initial configurations of the object is inserted in the system, which creates a virtual version of the object using the Point Chain Model. Figure 2 depicts this generation. We will call each one of those objects "initial DLO".
2. Next, the system is provided with another set of pictures related to the first one, each picture representing the resulting configuration of the DLO. This configuration results after performing a certain action on the initial DLO. With each picture, the action that was performed is defined using a generic interface describing the type of action, the point of the DLO where it's performed, and other parameters like the coordinates and direction of the action, gripper hands involved, etc.
3. The system generates an object for each one of the resulting configuration pictures, in order to have a goal reference for the Initial DLOs. We call each one of those objects "goal DLO". Initial and Goal DLOs will be composed by the same number of points, a feature that allows us to compare the objects point to point.
4. Then, the system prepares a set of physical parameters for the DLOs. Currently it works with four attributes: Flexibility, elasticity, friction and self-collision coefficient. The last parameter is a measure that refers to the physical volume of each one of the points that compose the Point Chain Model of the DLO, and defines how the object behaves when self-colliding, and on very high values has also impact in the flexibility of the object. The parameters are specific to the simulator, whose values have no explicit equivalence in physic units. They are randomized within the range of values from 0 to 100, and assigned to all the Initial DLOs.
5. Next, the system starts the process by simulating the provided action on its related the Initial DLO and after the action finishes and the DLO reach a stable configuration, the resulting new configuration is compared with its related Goal DLO. We consider that a DLO is in a stable configuration when its points did not move more than a threshold distance from their position until a determined time has passed.
6. The comparison function consists of a geometrical distance accumulation method with added features. More advanced shape comparison techniques exist in the literature [18] [19], but as curve comparison is not the focus of this work and in preliminary testing we found that geometrical distance was enough to generate a viable parametrization, we opted for the simpler and faster method. The comparison method used by the system is conducted by adding all the vector distances between each point of the Initial DLO and the point correspondent to the focal point in the Goal DLO. This focal point is the point nearest to the gripper hand that is grasping the goal DLO if exist, or the first point if no gripping hand is present. The vector distances corresponding to points that are "geometric features" of the DLO, like grip points and local maxima and minima, are weighted in order to be more relevant in the result. This comparison method takes in account the orientation and facing of the objects, allowing the planner to calculate actions that have the objective of re-orient the initial object in order to reach more similar configurations to the goal object. The comparison function is summarized by the formula in (1).

$$\sum_{i=1}^{n}\left(\left|p_{ref}^{goal}-p_{i}^{goal}\right|-\left|p_{ref}^{curr}-p_{i}^{curr}\right|\right)+\sum_{j=\ell}^{m}\left|p_{j}^{goal}-p_{j}^{curr}\right| \quad (1)$$

$$p=\begin{pmatrix}x\\y\\z\end{pmatrix} \qquad \mathbb{P}=\{p_1,\ldots,p_n\} \quad \mathbb{M}=\{p_\ell,\ldots,p_m\}\subset\mathbb{P}^{goal}$$

$$1\le\ell<m\le n$$

Where $\mathbb{P}$ is the set of points composing a DLO, $\mathbb{M}$ is the subset of $\mathbb{P}$ containing the goal DLO's points that are local maxima in the z-axis. In the formula, p_{ref}^{goal} is the focal point in the goal DLO and p_{i}^{goal} the *i*-th point in the goal DLO, and conversely, p_{ref}^{curr} and p_{i}^{curr} are the equivalent points in the starting DLO on its current configuration. Finally, p_{j}^{goal} and p_{j}^{curr} are respectively the *j*-th points in the goal DLO and the starting DLO,

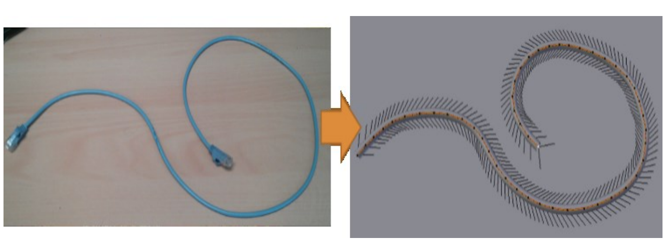

Fig. 2: A virtual DLO generated from a picture. We use a model that considers the DLO as a line of interconnected points and the simulator recreates the object using the line as a Bezier curve.

taken from the subset of local maxima, minima and grip points, composed of m elements. The final result of the comparison is a vector that can be converted in a numeric value by calculating its module; the smaller the value, the more similar the DLO configurations are. We call this value the "difference score". From our tests of the system in the virtual environment, and after checking how are distances measured by the simulator and observing the results of trial action performance in it, we decided that similar objects (i.e. whose general shape and features like the gripping points and local maxima are similar) have an score lower than 100, while very similar objects (i.e. with very few differences) have less than 50.

7. At this step, the system will have a difference score related to each action and set of parameters. These values are stored and the system return all the Initial DLOs to their starting configuration. Next, it generates another set of random parameters and assign them to the Initial DLO, repeats again the process from Step 5. This repetition will continue a determined number of rounds. In the end, the system will have a collection of lists of difference scores associated to a set of parameters each one.
8. Finally, the system selects the set of parameters related to the list of difference scores with lowest average value which will be the output of the system.

This set of parameters can be used in other applications focused in DLO interaction, as the number and types of physical parameters used in modelling a DLO is common among this kind of application, and even if new parameters were needed, it is not difficult to add them to the system, provided that the simulation engine admits it.

IV. Experiment

We designed an experiment in order to confirm the consistency and validity of our method for capturing DLO's mechanical attributes. We used a set of pictures of 5 different real objects as the input of the system, having all different characteristics, like different thickness, flexibility, weight, etc. Figure 3 shows the objects used in the experiment and their characteristics. The pictures used as input depict the objects' state before and after we took them by a determined point and move them between 5 and 10 centimeters, placing it in that point. For the DLO generation process from the input pictures we opted for the development of a human assisted conversion tool specific for this task. This tool works by allowing the user to mark the meaningful points of a DLO in a given photography, and generates the Point Chain Model out of it. The user also can indicate if a gripper should be present and where is located its grip point.

The goal of the experiment was to verify two aspects of our method: consistency and validity. With consistency we mean that the system generates similar values for the mechanical attributes of similar objects (same materials, similar length, etc.), on the other hand, validity refers to the correctness of the parameters, and if after applying them to a virtual DLO it would behave like a real object with similar attributes.

Object 1	**Medium thickness High flexibility Light**	
Object 2	**Thick Medium flexibility**	
Object 3	**Very thin High flexibility Light**	
Object 4	**Thin Medium flexibility Light Very elastic**	
Object 5	**Very thick Low flexibility Heavy**	

Fig. 3: The objects used in the experiments. Each object has a different set of visible mechanical attributes.

For the first aspect, in order to check if our method generates a coherent set of parameters, we configured the system to do 150 rounds per run and ran it 20 times for each one of the selected objects, comparing the resulting set of parameters and their difference score. Our hypothesis is that if the values returned by system are consistent, then the resulting 20 sets of parameters for each object should be similar. However, as we are dealing with a set of values with different physics significance, we expected that some of the parameters could have more impact in the outcome, and some others being less important. Also, depending of the object, some parameter may be more weight in the resulting behavior than the others. In the next section we will return to this issue to discuss it. In order to check the validity of our method, the second aspect in the experiment's goal, we compared the behavior of each virtual object with the set of parameters obtained in each round of the previous stage and compared with the behavior of their real counterpart. We did this by performing an action that was not input in the first stage of the experiment to both objects (in the virtual environment and in the real world), then, we captured the resulting configuration of the real world's object, recreated it in the virtual world, and compared its configuration with the resulting configuration of the original virtual object. If the objects are similar, the difference score should be less than 50.

V. Results and discussion

After performing the experiments, the results we obtained confirm relatively our first hypothesis, as the values of each parameter are similar for each object, but not as much as we expected. The results are depicted in Table 1.

Table 1: Average and Standard deviation values for the resulting mechanical parameters of each object. The possible values for those parameters are in the 0-100 range.

	Elasticity	Flexibility	Friction	Self-Collide
Object 1				
Average	49.55	68.75	33.85	52.35
Std. Dev.	13.88	17.38	20.02	20.01
Object 2				
Average	20.45	65.15	53.65	38.3
Std. Dev.	13.94	19.02	27.48	16.41
Object 3				
Average	31.25	75.45	37.75	57.8
Std. Dev.	10.82	18.65	27.91	22.87
Object 4				
Average	88.55	43.8	48.05	37.35
Std. Dev.	1.5	18.65	26.82	23.06
Object 5				
Average	28.85	48.35	52.15	88.35
Std. Dev.	20.09	17.89	22.82	14.98

We observe that there are parameters that vary much more than others whatever the object is, with a high deviation, like the friction. This can mean that friction was not a very crucial factor when simulating the DLO behavior, at least for small scale manipulation, like moving some object or holding it. On the other hand, others, like elasticity and flexibility are more important and allow a smaller degree of variation, but even so, we observe some disparity in the values depending of the object. For example, in the case of the elasticity, the fifth object has more variation. This can be explained because it was a heavy object, mostly rigid except at some points, with almost no deformation against small actions, so elasticity has not much effect. On the other hand, Object 4 was a very elastic one, so this parameter became crucial in simulating its behavior. In the case of flexibility we observe as well that the values are coherent with the real objects they represent, with moderate deviation rate. Finally, the self-collision coefficient allows less variability in the more rigid objects (objects 2 and 5), and the most variability in the thinnest ones (objects 3 and 4), being a parameter with variable importance depending of other factors of the object. Aside of that, we can observe that the average results for each parameter are coherent with what intuitively could be inferred from the apparent characteristics of the objects: flexible objects were conferred high flexibility, or objects with elasticity received a higher values of this parameter. Still we observed two issues that do not work properly: First, as we stated earlier, the parameter for friction does not appear to affect much to the object's behavior, therefore not having uniform or indicative values (and certainly is difficult to deduct accurate values for friction from images of light objects), so we may opt to discard this parameter in future installments of the system due to the small impact it has.

Secondly, objects very heavy or rigid seem to receive not very correct values for elasticity or flexibility, but compensate those incorrect values with their self-collision coefficient. By seeing these results, we plan to perform as a next step a larger experiment in order to test each parameter independently, and see how it affects the DLO's behavior. If we understand their effects and how they influence each other, then we will be able to link a parameter with concrete behaviors of the DLO, and obtain a range of candidate parameters from the pictures for using them in the comparison function, instead of only compare the object configuration, but also the randomly generated parameters.

The second aspect we evaluated in the experiment was if we could consider the method as valid, i.e. that the resulting parameters would generate a behavior on a virtual DLO similar to its real counterpart. The average difference scores obtained in the second part of the experiment for each one of the objects after applying the resulting parameter set is contained in Table2, and an example with the result obtained for Object 1 is shown in Figure 4.

Table 2: Average and Standard deviation values for the difference scores obtained in the second stage of the experiment. A value lower than 50 means the resulting object was similar to the goal.

Difference Score		
	Average	Std. Dev.
Object 1	24.96	1.33
Object 2	23.9	2.73
Object 3	24.34	0.62
Object 4	32.25	3.46
Object 5	53.39	1.12

As we can see in the table, after being manipulated every object obtained a difference score below 50, with the exception of the object number 5. This reinforces the idea that our method work well generally except with not very flexible objects or objects with parts more rigid than others, as we observed from the parameter results. We think this issue could be solved by improving the representation we use for the DLOs by adding additional mechanical parameters assigned to individual parts of the DLO, and at the same time, to improve the method for obtaining each parameter aside of the others in order to avoid having some parameters compensating for others. Even so, the results were very similar for each object's run, so this fact also supports the idea of the consistency of the values given to the parameters: for the same object the resulting parameters generated a similar behavior every time we manipulated it.

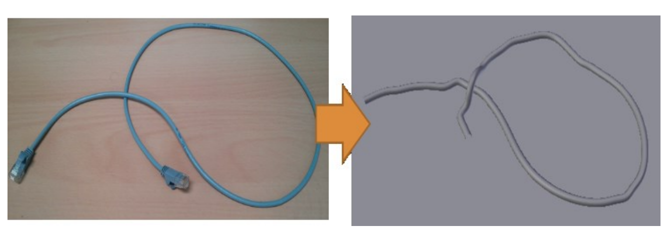

Fig. 4: Resulting configuration for Object 1. After the manipulation with the obtained parameters set, the resulting configuration is very similar to the goal DLO.

VI. CONCLUSION

In this paper, we presented a novel method for estimating through visual analysis the values of a DLO's mechanical parameters that condition its behavior. With this purpose, we developed a system that virtually recreates the configuration of the object before and after a certain manipulation and simulate such manipulation giving a randomly generated set of parameters to it. After repeating the process a determined number of times, the system selects the parameter's set from the object most similar to the picture after the manipulation. As is a novel approach, is difficult to compare it with other similar techniques, most of them requiring complex testing protocols, being the most similar the ones used in medical applications, but those methods are based on very specific rules of a concrete domain, hardly applicable to other situations.

We conducted an experiment in order to determine if our method generates consistently similar values for the parameters of similar objects, and if those parameters generate a realistic behavior when applied to the virtual simulation. The results of the experiment were positive in the sense that the parameters we found to be more relevant in the behavior of the DLO were consistently receiving similar values. The behavior of the DLOs after setting their parameters with the values was considered very similar to the real objects obtaining generally good results.

Some of the results were not so positive, and as a consequence we identified a number of issues that we could improve in our method. First, as we said, some of the parameters were more relevant than others. Concretely the friction proved to be not very determining of the DLO behavior in manipulation of small and light objects. The system also has a lower performance when predicting the parameters of moderately rigid and heavy DLOs: the results contained some values that intuitively should not be attributed to them, with other parameters compensating for them. As our next steps, we are planning to refine the obtaining of the parameters, as purely randomizing and testing them as a whole set leave us to rely too much on having by chance the correct combination of parameters, discarding combinations that may have the correct value for one parameter but incorrect for the rest, wasting time and needing a high number of testing rounds. Our plan is to isolate the generation and testing of each parameter in order and bounding the range of values by a geometric estimation, analyzing the behavioral patterns related to each parameter.

In conclusion, the method we presented shows promising and positive results as an initial approach that opens a number of potential ways to research. This method could be very useful for performing initial object calibration by robots that has to interact with new objects and only has initial visual information. We think that this style of approximating the attributes of an object by vision will be very relevant for such autonomous operation tasks.

REFERENCES

[1] Hovland, G. E., Sikka, P., & McCarragher, B. J. (1996, April). Skill acquisition from human demonstration using a hidden markov model. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on (Vol. 3, pp. 2706-2711). Ieee.

[2] Wakamatsu, H., & Hirai, S. (2004). Static modeling of linear object deformation based on differential geometry. The International Journal of Robotics Research, 23(3), 293-311.

[3] Schoner, J. L., Lang, J., & Seidel, H. P. (2004, September). Measurement-Based Interactive Simulation of Viscoelastic Solids. In Computer Graphics Forum (Vol. 23, No. 3, pp. 547-556). Blackwell Publishing, Inc.

[4] Takamatsu, J., Morita, T., Ogawara, K., Kimura, H., & Ikeuchi, K. (2005). Representation of knot tying tasks for robot execution. JOURNAL-ROBOTICS SOCIETY OF JAPAN, 23(5), 66.

[5] Shirakawa, Tomoya; Matsuno, Takayuki; Yanou, Akira; Minami, Mamoru, "String shape recognition using enhanced matching method from 3D point cloud data," in System Integration (SII), 2015 IEEE/SICE International Symposium on , vol., no., pp.449-454, 11-13 Dec. 2015. doi: 10.1109/SII.2015.7405021

[6] Saha, M., Isto, P., & Latombe, J. C. (2008). Motion planning for robotic manipulation of deformable linear objects. In Experimental Robotics (pp. 23-32). Springer Berlin Heidelberg.

[7] Huang, S. H., Pan, J., Mulcaire, G., & Abbeel, P. (2015, September). Leveraging appearance priors in non-rigid registration, with application to manipulation of deformable objects. In Intelligent Robots and Systems (IROS), International Conference on (pp. 878-885). IEEE.

[8] Remde, A., & Henrich, D. (1999). Picking-up deformable linear objects with industrial robots.

[9] James, D. L., & Pai, D. K. (1999, July). ArtDefo: accurate real time deformable objects. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (pp. 65-72). ACM Press/Addison-Wesley Publishing Co.

[10] Bickel, B., Bächer, M., Otaduy, M. A., Matusik, W., Pfister, H., & Gross, M. (2009, July). Capture and modeling of non-linear heterogeneous soft tissue. In ACM Transactions on Graphics (TOG) (Vol. 28, No. 3, p. 89). ACM.

[11] Pai, D. K., Doel, K. V. D., James, D. L., Lang, J., Lloyd, J. E., Richmond, J. L., & Yau, S. H. (2001, August). Scanning physical interaction behavior of 3D objects. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 87-96). ACM.

[12] Gong, M., Liao, M., Wang, H., Yang, R., & Zhang, Q. (2009). Modeling deformable objects from a single depth camera. ICCV.

[13] Nava, A., Mazza, E., Kleinermann, F., Avis, N. J., & McClure, J. (2003, November). Determination of the mechanical properties of soft human tissues through aspiration experiments. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 222-229). Springer Berlin Heidelberg.

[14] León, B., Ulbrich, S., Diankov, R., Puche, G., Przybylski, M., Morales, A., Asfour, T., Moisio, S., Bohg, J., Kuffner, J. and Dillmann, R., 2010. Opengrasp: a toolkit for robot grasping simulation. In Simulation, Modeling, and Programming for Autonomous Robots (pp. 109-120). Springer Berlin Heidelberg.

[15] Gschwandtner, M., Kwitt, R., Uhl, A., & Pree, W. (2011). BlenSor: blender sensor simulation toolbox. In Advances in Visual Computing (pp. 199-208). Springer Berlin Heidelberg.

[16] Ferraguti, F., Golinelli, N., Secchi, C., Preda, N., & Bonfe, M. (2013, September). A component-based software architecture for control and simulation of robotic manipulators. In Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on (pp. 1-5). IEEE.

[17] Echeverria, G., Lassabe, N., Degroote, A., & Lemaignan, S. (2011). Modular open robots simulation engine: Morse. In Robotics and Automation (ICRA). International Conference on (pp. 46-51). IEEE.

[18] Wahl, E., Hillenbrand, U., & Hirzinger, G. (2003). Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on* (pp. 474-481). IEEE.

[19] Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2002). Shape distributions. *ACM Transactions on Graphics (TOG)*, *21*(4), 807-832