Motion Planning for High-DOF Deformable Objects using Nonlinear Dimension Reduction

Zherong Pan[1] and Dinesh Manocha[2]

Abstract— **Motion Planning for high-DOF deformable objects is a fundamental problem in robotics with countless applications. We present our recent work on motion planners for very high-DOF deformable objects, including fluid bodies and elasticallly deformable robots. Due to their high-DOF configuration space, direct applications of previous motion planning algorithm is computationally infeasible. We overcome this problem by using nonlinear dimension reduction to map the configuration space to a low-DOF feature space. We show that both sampling-based and optimization-based motion planning algorithms can benefit from these low-DOF spaces to achieve $1-2$ orders of magnitude speedup. The applications of this technique include robot liquid transfer and navigating high-DOF elastically deformable robots.**

I. Introduction

Motion planning for high-DOF deformable objects is a fundamental problem in robotics with countless applications. For example, an mechanical engineer can create a vivid deformable octopus by modelling muscles on its eight-arms and animate the octopus by controlling the muscle contractions. As another example, when a robot performs dangerous chemical experiments in a laboratory, the robot arm needs to move the liquid reagents and mix them together to trigger reactions. The physical objects in these examples, i.e., the octopus and the reagents, are non-rigid and have a high number of DOFs. With the growing computational power of desktop machines, algorithms to plan motion for these objects are becoming practical but still challenging for two reasons. First, many prior planning algorithms for low-DOF dynamic systems have quadratic or cubic complexity in the number of DOFs. Such complexity is acceptable for dynamic systems such as articulated humanoid robots with less than 100 DOFs [15]. However, they becomes impractical for deformable objects whose configuration space has more than 10000 DOFs [27], [25], which is typical in our examples. Second, deformable bodies exhibit a lot of non-smooth behaviors that are difficult to model in the planning algorithm, such as environmental contacts, topology changes, and self-intersections. For example, to transfer liquid reagents from one container to another, we require the robot to avoid fast movements that will result in spilling. However, no mathematical model is available to predict whether spilling will happen. In summary, our major challenge is to reduced the computational overhead and model the non-smooth behaviors of deformable objects in the planning algorithm.

In this extended abstract, we summarize our two recently work on motion planners for Newtonian fluid bodies and elastically deformable robots. A common point of our planning algorithms is that, in order to make a planning algorithm computationally feasible or efficient, a procedure of nonlinear dimension reduction (NDR) is introduced to reduce a high-DOF configuration space to a low-DOF feature space. Nonlinear dimension reduction [40] has been one of the main focuses of recent research on deep learning. General image classification [37] and image compression [35] can viewed as a form of NDR. Although these techniques are successful in computer vision, it is non-trivial to extend them to other domain such as robot motion planning. In this abstract, we show that, by using task-specific or dynamic-system-specific heuristics, efficient NDR techniques can be developed for these new problems. As a result, sampling-based planning algorithms become more efficient by requiring less samples, and optimization-based planning algorithms run faster due to a reduced problem size.

In Section III, we design a motion planner for a robot to perform the specific task of liquid transfer. By using task-specific heuristics, we identify a 1-D feature space. In Section IV, we design a motion planner to navigate high-DOF elastically deformable robots using trajectory optimization. The optimization is accelerated by two orders of magnitude by using a dynamic-model-specific NDR procedure that reduces the dimension of the underlying dynamic system.

II. Related Work

We review related work on high-DOF deformable object modelling and deformable object planning algorithms.

A. High-DOF Deformable Object Modelling

A large amount of work has been done on the continuous and computational modelling of both fluid bodies [2] and elastically deformable objects [12]. The continuous dynamics of fluid bodies are governed by the Navier-Stokes equation. Computational models of the Navier-Stokes equation can be derived using various discretization schemes such as finite difference [13], [10], finite element [34], [21], smoothed particle hydrodynamics [20], [22], and hybrid scheme [42]. In robotics, approximate models are developed for specific tasks such as liquid transfer [24], [29]. The continuous dynamics of elastically deformable objects are governed by the Euler-Lagrange equation. The prominent method to derive the corresponding computational model is by using the finite element method [12], although alternative methods [17] can also be used. When a elastically deformable object undergoes only small deformations, dimension reduction technique exists for the dynamic system [36]. Later, this technique of dimension reduction has been extend to large deformations [4], [1], [23].

B. Planning Algorithms for Fluid Bodies

In robotics, fluid body motion planning problem arises in various manipulation tasks such as liquid transfer [27], [26],

[1,2]Zherong and Dinesh are with Department of Computer Science, the University of North Carolina, {zherong,dm}@cs.unc.edu

[30], [31] and spray painting [28], [7]. Since only one specific task is considered, the planning algorithm can become more efficient by exploiting simplified fluid models such as [29], [24]. In computer graphics, the problems are keyframe-based, where the goal is to deform the shape of the fluid body into a target shape. Early work [39] formulated the keyframe-based planning problem as a trajectory optimization solved using general-purpose gradient-based optimization. Besides trajectory optimization, we can use the keyframe as a target shape and adopt a tracker-type controller [11], which is faster to compute. But tracker-type controller does not provide any optimality guarantee.

C. Planning Algorithms for Elastically Deformable Objects

In robotics, deformable objects are passive and manipulated by the robot. Deformable object planning finds a lot of applications such as knot tying [16], cloth folding [19], and human dressing [8]. Compared with fluid bodies, the number of DOFs in an elastically deformable object can be made small by using a low resolution computational model. Therefore, conventional techniques such as sampling-based methods [6] and trajectory optimization [19], [14] are still computationally feasible. In computer graphics, animators model virtual actuated characters that do not exist in real-life as elastically deformable objects. Early work [41] uses brute force trajectory optimization to animate the character, which has high computational overhead due to the high-DOF. More recently, the overhead of trajectory optimization is reduced by using dimension reduction [5], [18], [33], achieving interactive performance. However, all these algorithms do not take environmental interactions into account. More recent work [38], [9] uses model predictive control to optimize deformable body configurations as well as environmental contact forces.

III. Robot Liquid Transfer

In this section, we introduce the application of NDR to extract features from a high-DOF configuration space of a Newtonian fluid body, with the application to robot liquid transfer. Liquid transfer is a well-known task in robotics that finds a lot of applications such as spray painting, lubricant replacement, car washing, and chemical experiments. These tasks are all challenging due to the existence of the high-DOF fluid body. Our research focuses on the specific task of liquid transfer, where the goal is to move the fluid body from a source container to a target container as illustrated in Figure 1.

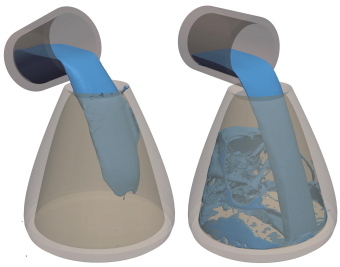

Fig. 1: An illustration of the liquid transfer task.

To search for the motion trajectory of the source container (the cylinder in Figure 1), we use optimization-based motion planning algorithm [32]. However, directly apply trajectory optimization is not computationally feasible for two reasons. First, each iteration of optimization, e.g. the computation of gradient vector when a gradient-based optimizer is used, requires the simulation of the 3D fluid bodies, for which even the state-of-the-art algorithm [3] takes hours. Therefore, the expected computational time for an entire trajectory optimization is on the level of weeks. Second, the shape changes of fluid body involve a lot of non-smooth procedures, such as contacts between the fluid body and the containers, as shown in Figure 1. These procedures cannot be handled by a gradient-based optimizer.

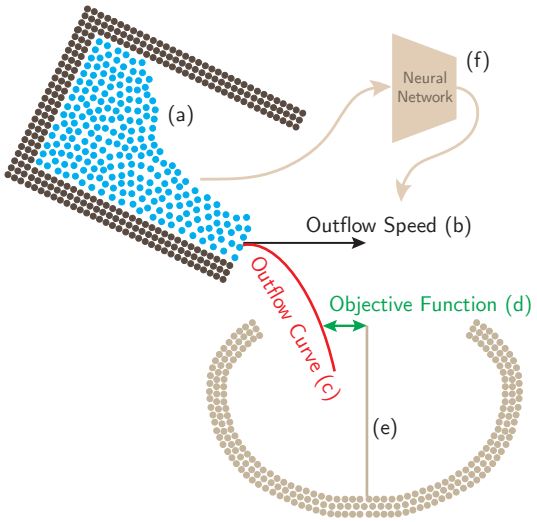

Fig. 2: Assuming that both the container and the fluid body in the simulated domain is modelled as a set of particles, if we model the fluid body in a brute force manner, we need to model the movement of all the fluid particles (blue points in (a)). Instead, for liquid transfer we only need to know the outflow speed (b). Based on the outflow speed, liquid outflow is approximated by a quadratic curve (c). Given the quadratic curve, the objective function (d) for trajectory optimization can be formulated as the distance between the quadratic curve and the center of target container opening (e). To perform NDR, we predict the outflow speed using a neural-network (f).

To solve this problem, we use two task-specific heuristics as illustrated in Figure 2. First, we observe that we can approximate the liquid outflow curve as a quadratic curve (Figure 2 (c)). Second, we observe that for the specific task of liquid transfer, we only need to know the liquid outflow speed (Figure 2 (b)), instead of modelling the entire shape of fluid (Figure 2 (a)). Based on this observation, we construct a neural-network (Figure 2 (f)) that takes the current fluid shape as input and predicts the outflow speed as a 1-D feature. In other words, the NDR procedure is accomplished using a neural-network. After the neural-network has computed the outflow speed, we can approximate the liquid outflow as a quadratic curve (Figure 2 (c)), and we can perform trajectory optimization to minimize the distance between the quadratic curve and the center of target container

opening (Figure 2 (e)). More importantly, the entire fluid simulation procedure is approximated using a smooth neural-network function that can be handled by a gradient-based trajectory optimizer.

When this neural-network is combined with a trajectory optimization, we can compute a successful trajectory within a couple of minutes. Some examples are shown in Figure 3. More information about this work can be found at http://gamma.cs.unc.edu/FluidMotion and http://gamma.cs.unc.edu/RLFluid.

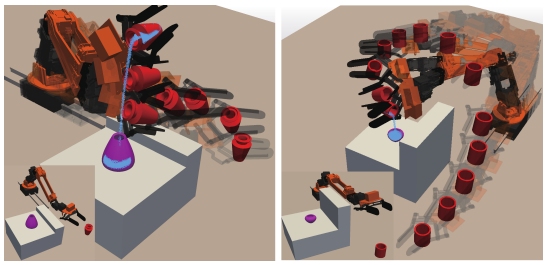

Fig. 3: Example of computed liquid transfer trajectories.

IV. Navigating Elastically Deformable Robots

In this section, we use NDR to accelerate both simulation and trajectory optimization for an elastically deformable object. This problem arises due to the recent trend to go beyond low-DOF articulated robots. For example, if we consider the 4-legged spider robot that is made of rubber, as illustrated in Figure 4 (a), it is computationally challenging to search for its walking gaits because there are infinitely many ways that the spider can deform.

To accelerate trajectory optimization for this application, we use a nonlinear dimension reduction method called rotation-strain (RS) transformation [23]. This method starts from conventional FEM-based discretization of the governing Euler-Lagrange equation denoted as $x_{i+1} = f(x_i, u_i, \Delta t)$ where x_i is the high-DOF configuration at timestep i, u_i is the control signal, $f(\bullet)$ is the function representing the FEM simulator, and Δt is the timestep size. RS transformation can be denoted as a function $\text{RS}(\bar{x}) = x$. Under an FEM discretization with moderate resolution, we usually have $|x| > 1000$. However, with $|\bar{x}| < 20$, we can model most dynamic behaviors via the RS transformation. Moreover, we can further reduce the FEM simulation function into the RS space by writing:

$$RS(\bar{x}_{i+1}) = f(RS(\bar{x}_i), u_i, \Delta t) \Longleftrightarrow \bar{x}_{i+1} = \bar{f}(\bar{x}_i, u_i, \Delta t),$$

where $\bar{f}$ is the RS-space FEM simulation function, which can be evaluated more than two orders of magnitude faster, compared with the original $f(\bullet)$. We refer readers to [23] for more details about this technique.

To apply RS-sapce FEM simulation to trajectory optimization, we propose a new framework [25] where we combine several objective functions including physical correctness, self-collision freedom, control objectives, etc. For the physical correctness, the objective function takes the following form:

$$E(\bar{x}_{i+1}, \bar{x}_i) = \|\bar{x}_{i+1} - \bar{f}(\bar{x}_i, u_i, \Delta t)\|^2.$$

With fast RS-space FEM simulation, our trajectory optimization framework can find a variety of movement gaits as illustrated in Figure 5, such as jumping, walking, and swimming, within a couple of hours on a desktop machine. These gaits are used by the deformable robot to move in a specific direction. After a set of gaits are found allowing the deformable robot to move in different directions, we can further train a neural-network-based motion planner that dynamically select moving directs to navigate the deformable robot to a distant target while avoiding obstacles, as illustrated in Figure 6.

V. Conclusion

In this abstract, we demonstrate two applications of motion planning for high-DOF deformable objects. In both applications, a key technique is to use NDR that maps the configuration space of a high-DOF deformable object to a low-DOF feature space.

In the first application, we use task-specific heuristics to reduce the configuration space of a fluid body to a low-DOF feature space. In this case, NDR is achieved by a neural-network. As a result, the fluid dynamics is approximated with a smooth neural-network function that can be efficiently integrated into trajectory optimization.

In the second application, we use a dynamic-system-specific technique to perform the NDR. In this case, NDR operator is the rotation-strain transformation, which maps the high-DOF FEM-discretized dynamic system to a low-DOF linearized space. As a result, trajectory optimization becomes two orders of magnitude fast and can be used to search for movement gaits.

References

[1] S. S. An, T. Kim, and D. L. James, "Optimizing cubature for efficient integration of subspace deformations," *ACM transactions on graphics (TOG)*, vol. 27, no. 5, p. 165, 2008.

[2] J. D. Anderson and J. Wendt, *Computational fluid dynamics*. Springer, 1995, vol. 206.

[3] R. Ando, N. Thürey, and C. Wojtan, "Highly adaptive liquid simulations on tetrahedral meshes," *ACM Trans. Graph. (Proc. SIGGRAPH 2013)*, July 2013.

[4] J. Barbič and D. L. James, "Real-time subspace integration for st. venant-kirchhoff deformable models," *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 982–990, 2005.

[5] J. Barbič, F. Sin, and E. Grinspun, "Interactive editing of deformable simulations," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 70, 2012.

[6] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 2126–2133.

[7] H. Chen, N. Xi, Z. Wei, Y. Chen, and J. Dahl, "Robot trajectory integration for painting automotive parts with multiple patches," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 3984–3989.

[8] A. Clegg, J. Tan, G. Turk, and C. K. Liu, "Animating human dressing," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 116:1–116:9, July 2015.

[9] S. Coros, S. Martin, B. Thomaszewski, C. Schumacher, R. Sumner, and M. Gross, "Deformable objects alive!" *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 69, 2012.

[10] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 736–744.

[11] R. Fattal and D. Lischinski, "Target-driven smoke animation," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 441–448.

[12] Y.-c. Fung, P. Tong, and X. Chen, *Classical and computational solid mechanics*. World Scientific Publishing Company, 2017, vol. 2.

[13] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *The physics of fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

[14] S. Javdani, S. Tandon, J. Tang, J. F. O'Brien, and P. Abbeel, "Modeling and perception of deformable one-dimensional objects," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1607–1614.

[15] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," in *Robotics Research. The Eleventh International Symposium*, P. Dario and R. Chatila, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 365–374.

[16] 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.

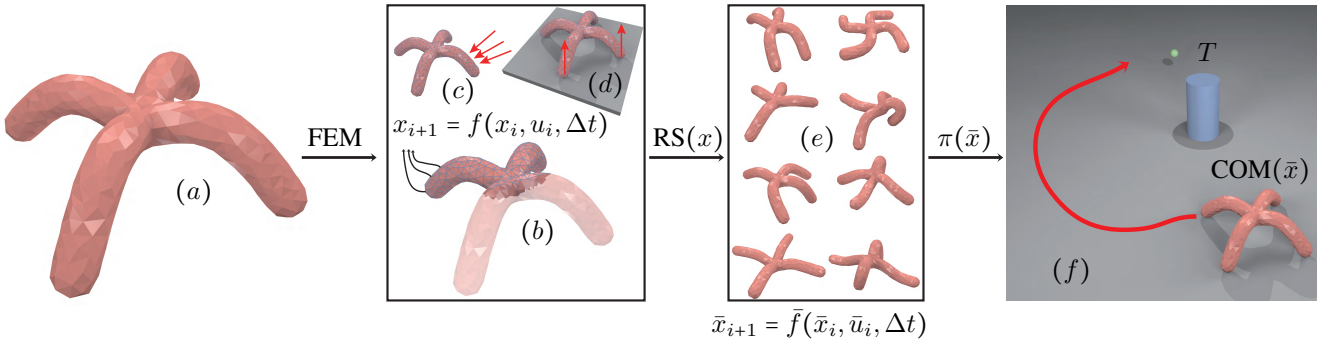

Fig. 4: Our overall idea of motion planning for an elastically deformable body: Given an arbitrarily-shaped deformable body (a), we use the FEM method to discretize the governing dynamics equation, which is denoted as $x_{i+1} = f(x_i, u_i, \Delta t)$ (b). This formulation also takes into account various external forces. These forces include fluid drag force for swimming (c) and frictional contact force for walking (d). Function f is costly to evaluate, so we apply the RS transformation [23] to derive a low-dimensional dynamic system that captures most of the deformations on a low-dimensional manifold. Some examples of the deformations on this manifold are given in (e). We highlight the path computed using our planner to navigate the deformable body to reach a target (green) while avoiding obstacles (blue) (f), where our method can handle different obstacle shapes and input environmental features.

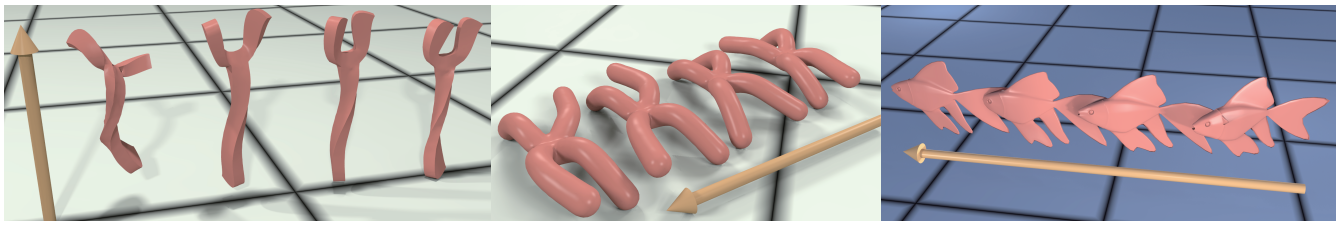

Fig. 5: Gaits found using our trajectory optimization framework based on RS-space FEM simulation.

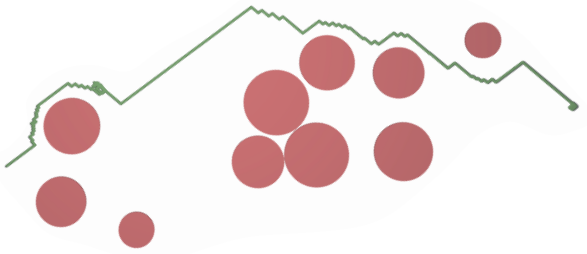

Fig. 6: The deformable robot is navigated along the green curve to reach a distant target, while avoiding the obstacles (red circles).

[17] D. I. Levin, J. Litven, G. L. Jones, S. Sueda, and D. K. Pai, "Eulerian solid simulation with contact," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 36.

[18] S. Li, J. Huang, F. de Goes, X. Jin, H. Bao, and M. Desbrun, "Space-time editing of elastic motion through material optimization and reduction," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 108, 2014.

[19] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 6000–6006.

[20] M. Liu and G. Liu, "Smoothed particle hydrodynamics (sph): an overview and recent developments," *Archives of computational methods in engineering*, vol. 17, no. 1, pp. 25–76, 2010.

[21] M. K. Misztal, K. Erleben, A. Bargteil, J. Fursund, B. Christensen, J. A. Bærentzen, and R. Bridson, "Multiphase flow of immiscible fluids on unstructured moving meshes," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 2012, pp. 97–106.

[22] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 154–159.

[23] Z. Pan, H. Bao, and J. Huang, "Subspace dynamic simulation using rotation-strain coordinates," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 242:1–242:12, Oct. 2015.

[24] Z. Pan and D. Manocha, "Motion planning for fluid manipulation using simplified dynamics," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4224–4231.

[25] ——, "Active animations of reduced deformable models with environment interactions," *arXiv:1708.08188*, 2017.

[26] ——, "Efficient solver for spacetime control of smoke," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 5, p. 162, 2017.

[27] Z. Pan, C. Park, and D. Manocha, "Robot motion planning for pouring liquids," in *ICAPS*, 2016, pp. 518–526.

[28] A. Pichler, M. Vincze, H. Andersen, O. Madsen, and K. Hausler, "A method for automatic spray painting of unknown parts," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 444–449.

[29] S. Rockel, . Konen, S. Stock, J. Hertzberg, F. Pecora, and J. Zhang, "Integrating physics-based prediction with semantic plan execution monitoring," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 2883–2888.

[30] C. Schenck and D. Fox, "Perceiving and reasoning about liquids using fully convolutional networks," *The International Journal of Robotics Research*, p. 0278364917734052, 2017.

[31] ——, "Visual closed-loop control for pouring liquids," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2629–2636.

[32] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[33] C. Schulz, C. von Tycowicz, H.-P. Seidel, and K. Hildebrandt, "Animating deformable objects using sparse spacetime constraints," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 109, 2014.

[34] C. Schwab, *p-and hp-finite element methods: Theory and applications in solid and fluid mechanics*. Oxford University Press, 1998.

[35] A. Sento, "Image compression with auto-encoder algorithm using deep neural network (dnn)," in *2016 Management and Innovation Technology International Conference (MITicon)*, Oct 2016, pp. MIT–99–MIT–103.

[36] A. A. Shabana, *Theory of vibration: Volume II: discrete and continuous systems*. Springer Science & Business Media, 2012.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] J. Tan, G. Turk, and C. K. Liu, "Soft body locomotion," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 26:1–26:11, 2012.

[39] A. Treuille, A. McNamara, Z. Popović, and J. Stam, "Keyframe control of smoke simulations," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 716–723, July 2003.

[40] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," 2009.

[41] C. Wojtan, P. J. Mucha, and G. Turk, "Keyframe control of complex particle systems using the adjoint method," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006, pp. 15–23.

[42] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 965–972, 2005.