Intelligent Learning for Deformable Object Manipulation

Ayanna M. Howard, George A. Bekey
Institute for Robotics and Intelligent Systems, School of Engineering
University of Southern California, Los Angeles, California 90089

Abstract— **The majority of manipulation systems are designed with the assumption that the objects being handled are rigid and do not deform when grasped. This paper addresses the problem of robotic grasping and manipulation of 3-D deformable objects, such as rubber balls or bags filled with sand. Specifically, we have developed a generalized learning algorithm for handling of 3-D deformable objects in which prior knowledge of object attributes is not required and thus it can be applied to a large class of object types. Our methodology relies on the implementation of two main tasks. Our first task is to calculate deformation characteristics for a non-rigid object represented by a physically-based model. Using nonlinear partial differential equations, we model the particle motion of the deformable object in order to calculate the deformation characteristics. For our second task, we must calculate the minimum force required to successfully lift the deformable object. This minimum lifting force can be learned using a technique called 'iterative lifting'. Once the deformation characteristics and the associated lifting force term are determined, they are used to train a neural network for extracting the minimum force required for subsequent deformable object manipulation tasks. Our developed algorithm is validated with two sets of experiments. The first experimental results are derived from the implementation of the algorithm in a simulated environment. The second set involves a physical implementation of the technique whose outcome is compared with the simulation results to test the real world validity of the developed methodology.**

Keywords— **Deformable Object Manipulation, Learning, Iterative Lifting.**

I. Introduction

Most robotic systems currently in existence have been built under the assumption that manipulation of rigid objects remains the primary task. Geometry of the object is usually static, with very little variance between one instance of the object and another. The robot must have knowledge pertaining to the exact structure and location of objects in the environment and the precise actions to be performed. In reality though, many objects are non-rigid. Most are unsymmetrical, compliant, and have alterable shapes. Even solid objects can deform when the object's dimensions become extensive. In general, deformable objects may have one-, two-, or three-flexible degrees of freedom. In the real world, these classes of objects include such items as rubber balls, beams, hoses, cloth, and wire.

In this paper, we discuss a generalized learning algorithm for handling of 3-D deformable objects. We define this as a 'generalized approach' since it does not assume prior knowledge of object characteristics and can thus be utilized for an extensive range of 3-D deformable object types. We define a '3-D deformable object' as an object whose three flexible degrees of freedom are characterized by viscoelastic interactions between molecules. Such an object, when an external force is applied, changes the volumetric space it occupies as well as its shape. This thus excludes such objects as cloth sheets, steel beams, and glass plates. Furthermore, we do not consider crushable objects that are permanently deformed under pressure, since they can be considered rigid after crushing.

In order to manipulate a 3-D deformable object in the presence of gravity, we must determine the force required to successfully lift the object. We define the minimum force necessary for such an operation as the minimum object 'lifting force'. Based on this classification, we define the basic problem as: Given a 3-D deformable object, calculate the object lifting force. Based on our methodology, we show that the attributes which the system needs to know for calculating the object lifting force can be learned off-line for a wide range of three dimensional deformable objects. The attributes learned can then be mapped such that, during run-time, enough relevant attributes can be retrieved to grasp any three dimensional deformable object presented to the system. The research described in this paper addresses this issue.

II. Related Past Work

Manipulation of 3-D deformable objects is one of the least addressed research areas in robotics. Classical robotic systems assume only rigid objects will be manipulated by the system. Such systems assume that object geometry maintains a single, stable, configuration during the manipulation task. Many robotic systems which address manipulation of deformable objects focus on force control and stability in the presence of environments with compliant properties [5] [4]. With this focus, in-depth knowledge of the object deformation characteristics is usually incorporated directly into force calculation and force feedback is utilized to ensure grasp stability.

Some systems focus on deformation control versus force control. In these systems, the robot manipulator is designed to control the deformation of the object [6] [7]. Those systems which do not assume knowledge of object characteristics are limited in that they assume manipulation of an object with no more than two flexible degrees of freedom. Deformable objects also appear in the field of robotics in terms of grasping with soft fingers [12] [14]. This type of research suffers from the same limitations as does rigid manipulation of compliant objects.

In computer graphics, deformation is seen as a tool for producing realistic looking animations. Currently there are some efforts which focus on virtual control prototyping in which a user interacts with a virtual deformable object in the exact same way they would interact with the physical object [9] [1]. The main limitation with these techniques are the same which face the robotic world. These systems assume in-depth knowledge of object characteristics is available for inclusion into the simulated environment.

The problems which arise in grasping of non-rigid objects have been inadequately addressed in previous research attempts. These limitations have motivated the development of a methodology which does not require prior in depth knowledge of object attributes for manipulating 3-D deformable objects.

III. Methodology

Our main focus is to learn an adequate grasp for a deformable object. We choose to represent grasping as the act of pushing against an object from two opposite ends [15]. Our system, therefore, utilizes two cooperative manipulators, each possessing an end-effector constructed as a flat surface palm and possessing a force sensor able to detect and record any force applied to the palm's surface area.

For determination of an adequate grasp we must learn the minimum forces a multiple robotic mechanism must exert in order to lift a common deformable object cooperatively. If F_w is the force required to lift a rigid object of weight W with frictional coefficient μ, we define the minimum deformable object lifting force L_f as $F_w + F_d$ where F_d is the minimum additional force term required to compensate for the deformation of the object.

We shall begin the process of determining L_f by focusing on the physical changes of the deformable body. We show that once a representation for both the external and internal positional movements of the deformable object is retrieved, the object lifting force can be determined. In effect, we show that a relationship between object deformation and force can be learned such that an adequate grasp with minimal force can be achieved. Once this relationship is learned, we can utilize these factors to maintain a firm grasp on any deformable body by comparing the current run-time object deformation with the learned relationship. When they are equivalent, we can retrieve the necessary object lifting force required for manipulation.

A. Deformable Object Model

At the submicroscopic scale, all solid material is composed of atoms. Adjacent atoms in a solid exhibit both attractive and repulsive forces which keep the atoms at an equilibrium distance from each other. When the spacing between atoms is increased, a pulling force is created which tends to pull the atoms toward each other. In an equivalent manner, if the spacing is decreased, a pushing force is created which pushes the atoms away from each other. The tendency to maintain this equilibrium spacing leads to an 'elastic' manifestation.

The grouping of atoms leads to the formation of solids. A solid can be classified as crystalline, amorphous, or a combination of both. A crystalline material is made up of crystals, an orderly array of atoms arranged in three dimensional rows such that each atom is at an equilibrium distance from each other. This structural arrangement is called a space lattice [2] (Fig 1).

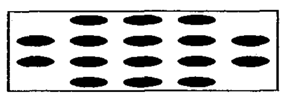

Fig. 1. Space Lattice Arrangement

A crystalline material has an orderly arrangement and thus we shall utilize its atomic structural characteristics to approximate the modeling of an elastic solid. We shall utilize the space lattice characteristic to represent an object as a particle based system constructed from a discretized sampling of its volume. Let $\alpha > 0$ represent the object discretization constant. The particle based representation of the object is thus given by a set Ψ of particles where each particle $\psi_n \in \Psi$[1], has an associated Cartesian position vector

$$\bar{p}_n = (x_n, y_n, z_n) = x_n\hat{i} + y_n\hat{j} + z_n\hat{k}$$

[1] where $n \leq \alpha^3$

Deformation of a material is caused at the microscopic level by visco-elastic interactions between atoms. This visco-elastic interaction can be modeled by the Kelvin Model which is characterized by a spring and damper in parallel. We shall utilize this model and characterize a deformable object as a set of particles locally interconnected by damped springs (Fig 2).

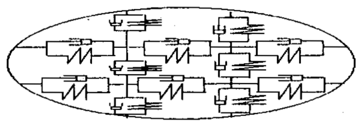

Fig. 2. Network of Interconnected Particles and Springs

B. *Calculating Deformation Characteristics*

Let us now extract a piece of the interconnected particle system as shown in Figure 2. Let $\bar{f}_n$ represent the external force

$$(f_x, f_y, f_z) = f_x\hat{i} + f_y\hat{j} + f_z\hat{k}$$

applied to particle ψ_n . Let m_n represent the mass of particle ψ_n. Let S_n represent the number of damped springs connected to particle ψ_n. Based on the mathematical equations defining the Kelvin Model, the forces acting on the n^{th} particle are accumulation of external force, inertial force, damping force and spring force. Using Newton's law of motion, the partial differential equation of motion for the n^{th} particle can be written as:

$$\sum_{i=1}^{S_n} m_n \frac{\partial^2 \Delta\bar{p}_{n,i}}{\partial t^2}(\bar{f}_n, t) + \sum_{i=1}^{S_n} \lambda_{n,i} \frac{\partial \Delta\bar{p}_{n,i}}{\partial t}(\bar{f}_n, t) + \sum_{i=1}^{S_n} D_{n,i}\Delta\bar{p}_{n,i}(\bar{f}_n, t) = \bar{f}_n \quad (1)$$

where D_n, the deformability coefficient, is a function of the force and the change in spring length, λ_n, the damping coefficient, is a function of the force and the instantaneous change in spring length, $\Delta\bar{p}_{n,i}(\bar{f}_n, t)$ represents the change in spring length in each Cartesian direction, and $\frac{\partial \Delta\bar{p}_{n,i}}{\partial t}(\bar{f}_n, t)$ represents the instantaneous change in spring length. We shall designate the summation $\sum_{i=1}^{S_n} \Delta\bar{p}_{n,i}(\bar{f}_n, t)$ as the particle displacement vector $\bar{d}(\bar{f}_n, t)$ and the summation $\sum_{i=1}^{S_n} \frac{\partial \Delta\bar{p}_{n,i}}{\partial t}(\bar{f}_n, t)$ as the particle displacement velocity vector $\bar{d}'(\bar{f}_n, t)$.

From Equation 1, we see that the only parameters which are not directly defined as a function of time are the mass, deformability, and damping coefficients. If we assume that the mass of the object is given, then we can classify the overall deformation of the object in terms of the deformability function D and the damping function λ. The algorithm we implement for determining the deformability function is as follows:

•1. Using both manipulators, apply a force against the object's surface. At time $t_m = t_0, t_1, \cdots, t_n$, record the force $\bar{f}(t_m)$ felt by the manipulator.
2. Calculate the particle displacement vector $\bar{d}(\bar{f}(t_m), t_m)$ for all $t_m \in t$
3. The deformability function is $D(\bar{f}, \bar{d}) = \frac{\bar{f}(t_m)}{\bar{d}(\bar{f}(t_m), t_m)}$ for all $t_m \in t$

The technique we use for determining the damping function is as follows:

•1. Using both manipulators, apply a force against the object surface. At time $t_m = t_0, t_1, \cdots, t_n$, record the force $\bar{f}(t_m)$ felt by the manipulator.
2. Calculate the particle displacement velocity vector $\bar{d}'(\bar{f}_a(t_m), t_m)$ for all $t_m \in t$ where $\bar{f}_a(t_m) = \bar{f}(t_m) - D\bar{d}(\bar{f}(t_m), t_m)$ is the damping force.
3. The damping function is $\lambda(\bar{f}_a, \bar{d}') = \frac{\bar{f}_a(t_m)}{\bar{d}'(\bar{f}_a(t_m), t_m)}$

C. *Learning An Adequate Grasp*

"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time" [10]. In this research, we wish to learn how to efficiently and effectively grasp a deformable object. To learn the characteristics of an adequate grasp, we must determine the relationship between mass, deformation, and force. Once this relationship is learned, we can utilize these factors to maintain a firm grasp on any deformable body by comparing the current run-time mass/deformation of the object with the learned relationship. When they are equivalent, we can retrieve the minimum force required for grasping a deformable object. The steps required to handle manipulation of 3-D deformable objects are as follows:

- Learn what forces a robotic system must exert in order to successfully grasp a deformable 3-D object
  1. Record dimensions of a known object
  2. Calculate the deformability and damping functions
  3. Determine the minimum force necessary to grasp known object by iteratively lifting object
  4. Link object attributes and grasping force into a index table

We assume that an object first presented to the system is initially in an equilibrium state. Thus the summation of all forces inherent to the object converges to zero. To begin the iterative lifting process we apply a force F_w to the object and lift the object until the object particles have stabilized their positions. If, at this time, the object has not slipped, the object is in equilibrium and a firm grasp on the object has been achieved. This then is the object lifting force L_f. We record its value at this time and terminate the lifting process. If the object has slipped from the manipulator grasp, we increase the applied manipulator force and repeat the lifting process. Eventually a minimum force will be reached which, when applied, will be able to lift the deformable object. The force required to lift a rigid object of weight W is F_w. Therefore we know that the object lifting force must be greater or equal to this force[2]: $L_f = F_w + F_d \geq F_w$.

After learning values for the lifting force, we store this information into an index table where given the mass, deformability, and damping functions we can extract the object lifting force. The table constructed is only valid for the objects explicitly learned. However, we wish to determine the general relationship between mass, deformability, damping, and lifting force. To accomplish this, we utilize a neural network characterized by three inputs, one hidden layer, and one output node to learn the desired relationship (Fig 3).

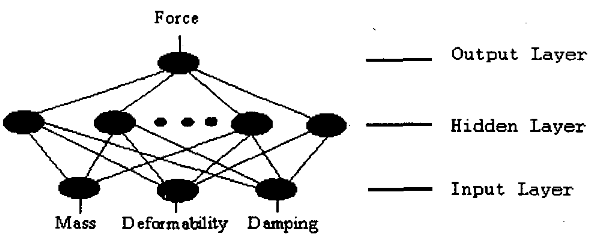

Fig. 3. Neural Network

A simple backpropagation algorithm [11] is used for the training process.

IV. Experimental Setup

We have validated our methodology for the successful manipulation of 3-D deformable objects using two experimental setups. The first setup consists of the implementation of the algorithm in a simulated environment. The second involves a physical implementation of the algorithm whose outcome is compared with the simulation results in order to test the real world validity of the developed methodology.

[2]we define the deformable object lifting force L_f as $F_w + F_d$ where F_d is the minimum additional force required to compensate for the deformation of the object.

A. Simulation

Reznik and Laugier's simulated the deformation of a "virtual" deformable finger as it was pressed against a rigid surface [9]. We shall utilize Reznik and Laugier's algorithm as a basis for simulating the deformation of a "virtual" deformable object as the rigid surface of the manipulators is pressed against it. Their modified algorithm will provide us with the necessary force and deformation feedback parameters required to model a 3-D deformable object in a simulated environment. Using this model, we have implemented the 'iterative lifting' simulation program which calculates the deformation characteristics of 3-D deformable objects and learns the desired minimum object lifting force.

B. Automation of Manipulator System

The physical test setup involves running our algorithm on an automated stereo-vision dual manipulator system (Figure 4). The dual robotic devices are used to perform the deformable object grasping routines. The vision system provides object dimension and position to the robotic manipulators for task initialization. In addition, the vision system provides feedback for determining object slippage in order to successfully accomplish the recursive learning process.

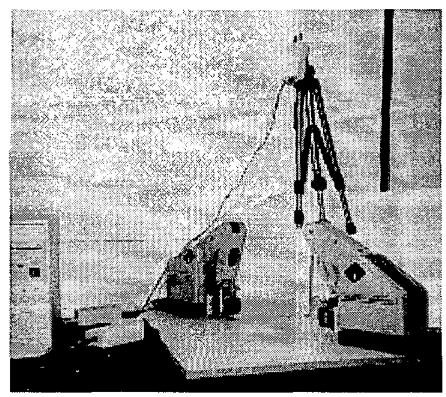

Fig. 4. Dual Manipulator System

The two manipulators are commanded using inverse kinematics. Thus, given a location x,y,z in 3D world coordinates, the system determines the necessary joint angles required to position the manipulator end-effector at the desired Cartesian location. The only difference between the joint angles required to control the two manipulators is that both the base and wrist joints must be equal and opposite in magnitude. This allows us to achieve the appearance of approximate parallelism. All other joint parameters will maintain equivalent values with respect to each manipulator.

For automation, the camera system determines object dimension and passes the center side position to the Microbots. Internally, we discretize the object using the calculated dimensions and a discretization constant equal to 1.0. The Microbots then move to each side of the object ensuring that the flat-plate end-effector is positioned normal to the object side. The Microbots then perform X squeezing operations[3]. After each such operation, the outputs from the Microbot wrist force sensors are read. These force values, along with the amount the object was squeezed are used for calculating the deformability and damping coefficients. The iterative lifting process is then begun. For the first iteration attempt, a force equivalent to the Weight Force is applied. The Microbots then attempt to lift the object until the manipulators have moved a distance of 0.5*object height. After "lifting", the cameras are used to determine whether the object has been successfully grasped. If the object is no longer in the grasp of the manipulator, the Microbots reposition themselves against the object's side, increment the lifting force by 20%, and begin the lifting process again. At some point, the object will be successfully grasped and the object lifting force is recorded.

Thus, our system autonomously determines the required lifting force for an elastic deformable object.

Object	Interior	Weight(N)	x,y,z Dimensions(cm)
A	Sponge	0.7	17.8 x 19.1 x 9.2
B	Cotton	1.4	19.1 x 19.7 x 19.1
C	Sand	2.9	11.4 x 11.4 x 9.5
D	Water	4.9	10.8 x 11.4 x 8.9

TABLE I
PHYSICAL IMPLEMENTATION: OBJECT CHARACTERISTICS

	Calculated			Simulation
	Deformability ($\frac{N}{cm}$)	Damping ($\frac{Ns}{cm}$)	Lifting Force (N)	Lifting Force (N)
Sponge	49.18	6.47	0.84	0.7
Cotton	13.03	0.33	1.72	2.23
Sand	30.02	0.47	5.97	4.86
Water	10.52	0.00	10.23	11.13

TABLE II
PHYSICAL IMPLEMENTATION RESULTS

V. RESULTS

For the experimental implementation, we utilize four deformable objects, each with different deformation characteristics and weights[4]. The objects relative properties are given in Table I and our physical implementation results are tabulated in Table II. The calculated results in Table II are derived from the physical experimentation whereas the simulation results are derived from the simulated environment. Based on the results, we conclude that we can achieve an error level of 14% with respect to the simulation lifting force.

A. Neural Network Results

To test the learning capability of the system, we train the neural network on simulation results retrieved from a multitude of deformable objects possessing different deformation characteristics. Our network possesses 3 input nodes representing the deformability coefficient, damping coefficient, and object weight, two hidden nodes, and 1 output node representing the minimum object lifting force. For our simulation, we incorporate 40 corresponding input-output pairs into the training set and 33 corresponding input-output pairs into the testing set. The minimum error derived from the testing set occurred when we trained the network for 1000 epochs. A sample of the neural network testing set output results are shown in Table III. The maximum error retrieved from the network is $< 14.8\%$. Based on this error calculation, we conclude that there is a derivable relationship between weight, deformability, damping and the object lifting force which can be learned.

VI. SUMMARY

In this paper, we discussed a generalized algorithm capable of successfully manipulating 3-D deformable objects without assuming prior in-depth knowledge of object attributes. Our methodology was able to incorporate a wide variety of deformable object types. We validated our developed algorithm with two sets of experiments. The first experimental results were derived from the implementation of the algorithm in a simulated environment. The second set involved a physical implementation of the technique whose outcome was compared with the simulation results to test the real

[3]Each Microbot applies a force against the object, thus "squeezing" the object. We perform 3 such operations.

[4]Each object is composed of a cloth bag filled with the specified interior material.

NN Input			Expected	NN Output
Weight (N)	Deformability ($\frac{N}{cm}$)	Damping ($\frac{Ns}{cm}$)	Lifting Force (N)	Lifting Force (N)
40.81	41.66	0.15	74.57	74.41
56.33	45.99	12.32	73.73	73.95
47.22	46.08	0.08	86.99	87.92
27.00	42.91	0.29	47.99	48.96
59.92	33.03	3.14	96.51	99.86
21.15	50.71	6.60	28.72	30.12
23.32	43.19	2.61	36.28	38.47
21.38	49.99	4.07	31.45	34.48
28.94	45.56	11.43	36.20	30.87

TABLE III

SIMULATION RESULTS: NEURAL NETWORK OUTPUT

world validity of the developed methodology. Based on the results, we are able to show that even using a number of simplfying assumptions for a simulation model, we can achieve both a physical and simulation lifting force for the same deformable object which only differ from each other by a maximum of 14%. In fact, with more accurate equipment, we could use the results extracted from the trained neural network to get an approximate measurement of the lifting force without having to physically perform the iterative lifting process. This force could be directly input into the robotic device for lifting. The iterative lifting process would then only need to be performed to incrementally compensate for the approximation error with a given deformable material.

REFERENCES

[1] B. Bradau. Deformability and force return for virtual objects grasping. In *Second ECPD International Conference on Advanced Robotics*, pages 510–515, Vienna, 1996.

[2] E.F. Byars and R.D. Snyder. *Engineering Mechanics of Deformable Bodies.* International Textbook Company, Pennsylvania, 1969.

[3] S. Chaudhuri and S. Chatterjee. Motion analysis of a homogeneously deformable object using subset correspondences. *Pattern Recognition*, 241(8):739–745, 1991.

[4] S. Chiaverini, B. Siciliano, and L. Villani. Force/position regulation of compliant robot manipulators. *IEEE Transactions on Automatic Control*, 39(3):647–652, 1994.

[5] N. Mandal and S. Payandeh. Control strategies for robotic contact tasks: An experimental study. *Journal of Robotic Systems*, 12(1):67–92, 1995.

[6] D.W. Meer and S.M. Rock. Experiments in object impedance control for flexible objects. In *Proceedings International Conference on Robotics and Automation*, pages 1222–1227, 1994.

[7] R. Patton, F. Swern, S. Tricamo, and A. van der Veen. Automated cloth handling using adaptive force feedback. *Journal of Dynamic Systems Measurements and Control*, 114:731–733, 1992.

[8] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface '95*, pages 147–154, 1995.

[9] D. Reznik and C. Laugier. Dynamic simulation and virtual control of a deformable fingertip. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 669–674, 1996.

[10] E. Rich and K. Knight, editors. *Artificial Intelligence.* McGraw-Hill, New York, 1991.

[11] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In *Readings in Machine Learning*, pages 115–137. Morgan Kaufmann, 1990.

[12] K.B. Shimoga and A.A. Goldenberg. Soft robotic fingertips part i: A comparison of construction material. *International Journal of Robotics Research*, 15(4):320–334, 1996.

[13] N.M. Thalmann and D. Thalmann. Finite elements in task-level animation. *Finite Elements in Analysis and Design*, 19:115–137, 1995.

[14] M.R. Tremblay and M.R. Cutkosky. Estimating friction using incipient slip sensing during a manipulation task. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pages 363–368, Altanta, GA, 1993.

[15] X. Yun. Object handling using two arms without grasping. *International Journal of Robotics Research*, 12(1):99–106, 1993.