Estimating material properties of deformable objects by considering global object behavior in video streams

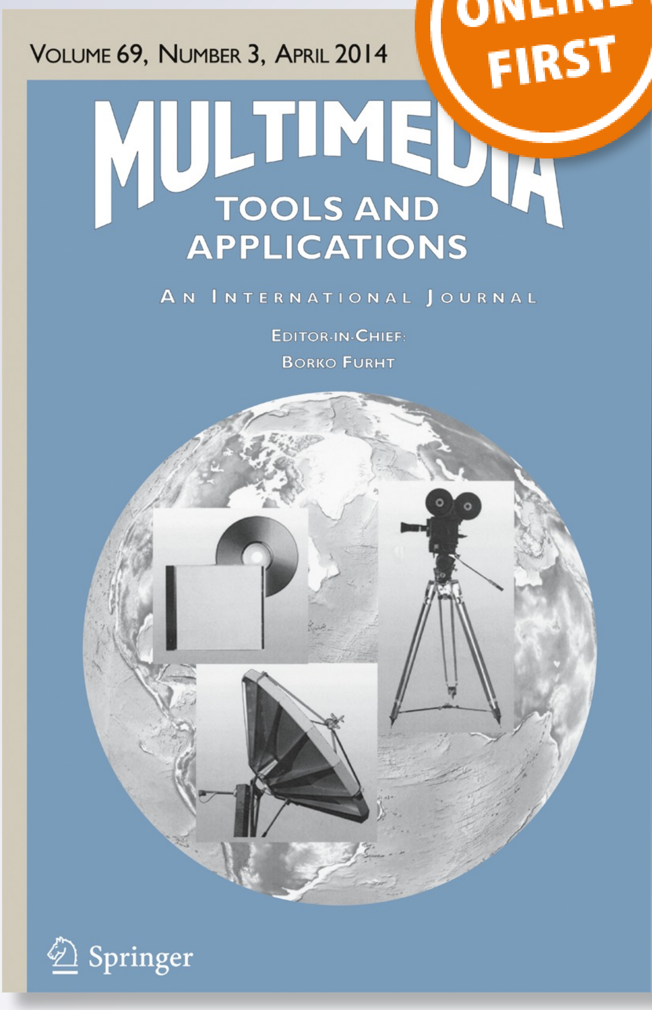
Min-Hyung Choi, Steven C. Wilber & Min Hong

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-014-1995-1

ONLINE FIRST



Multimed Tools Appl
DOI 10.1007/s11042-014-1995-1

Estimating material properties of deformable objects by considering global object behavior in video streams

Min-Hyung Choi · Steven C. Wilber · Min Hong

Abstract One of the crucial components in improving simulation quality in physics-based animation of deformable object is finding proper material properties that define the movement upon external excitation. Most work in the estimation of material properties for highly deformable objects involves applying localized force to a point on the object's surface with mechanical devices and measuring the displacement of the surface at the contact point and surrounding points. While understanding this localized behavior provides a step towards accurately simulating objects with known material properties, an understanding of the global behavior of the object undergoing deformation is more important for many practical applications. This paper describes both the computer vision based techniques for tracking global position information of moving deformable objects from a video stream and the optimization routine for estimating the elasticity parameters of a mass-spring simulation. The collected data is the object's surface node position of object over time which is used to a data-driven simulation of that object to match the behavior of a virtual object to the corresponding real one. This paper demonstrates that estimating material properties of highly elastic objects by matching the global behavior of the object in a video is possible with the proposed method and the experimental results show that the captured and simulated motions are well matched each other.

Keywords Global deformation · Computer vision · Material property estimation · Data-driven animation · Physically-based simulation

M.-H. Choi · S. C. Wilber
Department of Computer Science, University of Colorado Denver, Denver, CO, USA

M.-H. Choi
e-mail: min.choi@ucdenver.edu

S. C. Wilber
e-mail: steven.wilber@ucdnever.edu

M. Hong (✉)
Department of Computer Software Engineering, Soonchunhyang University, Asan, Korea
e-mail: mhong@sch.ac.kr

Published online: 29 April 2014

1 Introduction

Building realistic computer 3D models is an essential component for computer animation, game, 3D movie, computer simulation, virtual reality, augmented reality, and so on. To create virtual 3D objects for featured animations or movies, animators usually utilize their keen eye to estimate various parameters to obtain plausible object motion, instead of measuring the material properties and their associated effects to the movement of objects. Generally, animators set up essential key frames and then fulfill the rest of in-between animation using interpolation technique. However, these processes are not easy to represent realistic movement of objects and they often involve time consuming repetitive works before achieving target motion. In addition, mostly these results are not physically correct.

Alternatively, another popular choice is a physically-based approach. Under that scheme, objects are given some physical properties such as weight and elasticity coefficients which are then used by a physics engine to determine the position of the object at a given time-interval taking into account internal and external forces acting upon the object. These physically-based simulations have been well studied, and much time has been devoted to increasing the accuracy of the physics equations for many different types of simulated matter, increasing the stability and speed of the time-integration techniques and finding steps to simulate more complex models faster. However, since the exact material properties of real objects are not easy to set up for the initial conditions of dynamic simulation, desired and realistic behavior of simulated objects require much trials and errors. Some researchers have been tried mechanical methods for measuring material properties. These include stretching cloth in different directions to measure stretching and shearing [21], using a robotic arm to drop a glass ball and measuring the displacement and shatter pattern, using a spring to push an object across a diffuse surface and measuring the friction by the displaced spring, etc. All of these methods have merits of their own, but they all suffer from loss of information due to limits of human measurement.

In addition, most researches for deformable objects have been only focused on the local deformation of objects [21, 14, 1, 20, 9, 11]. Typically, these researches involve a force gauge to apply some localized force to the surface of an object and measure the displacement of the surface of objects. They use these data to determine the coefficients of governing equation to achieve the same behaviors of simulated objects when arbitrary force is applied to the surface of the simulated object. While localized analysis is useful for understanding of deformation contour changes at specific points, the global behavior of an object over a relatively longer period of time can be more valuable for producing an animation using physics based simulation. For instance, applying localized force to the surface of a rubber ball will show the elastic behavior of the ball under a very specific type of test, but it is not easy to determine the appropriate trajectories and velocities when the rubber ball is thrown against a wall. A more convincing example is a highly inflated balloon. Whatever the balloon was filled with air or water, the localized deformation test will show similar behaviors. Another good case is how the balloon responds when the force is removed from the object. For instance, an air-filled balloon will bounce differently if dropped from the same height as a water-filled balloon. Generally speaking, most real world objects include multi-layered materials with intricate structural configuration. However, although we may know the material property of all individual components, the global behavior of that object is not easy to simulate due to complex interaction between those layers. For example, a dynamic behavior of air-compressed rubber ball is dictated by the air pressure associated with the rubber's material property. A baseball which consisted of a solid core, wound thread, rubber mantle, and stitched leather is extremely complicated to simulate even if we have a complete understanding of individual material.

Instead of trusting an animator's trained eye which can be subjective to judge how a simulated object should behave and of trying to simulate the behavior of an object based on multi-layered material properties, this paper proposes the data-driven based approach for matching the global behavior of a virtual deformable object to a real one with an optimization technique to automatically estimate the material properties of the real object from the captured video data. From the recorded movement of real 3D objects, computer vision based object movement tracking algorithm extract the trajectory of the object over time, and then material properties of the virtual object are adjusted to achieve similar behaviors until it converges to an optimal range. In this paper, our main focus is to construct a virtual object that shows relatively similar global behavior as real one, and not to obtain a complete understanding of individual material properties of constituent components with structural configuration. Therefore, whatever we are using a real base ball with multi-material solid core or an real air pressured hollow rubber ball, the virtual ball is constructed with a 3D tetrahedral mesh with different material properties. The accuracy of extracting an object's behavior from the recorded images is limited by several factors. These factors include the frame rate of video, resolution of the recorded image, influence of illumination and shadow, color depth, and image compression algorithm. However, despite of these shortcomings, our results show that the proposed method successfully extracted the movement of deformable object over time from the video stream, effective enough to construct a virtual object that mimics global behavior of the real one.

2 Related work

Recently, the estimation of material properties from the captured behavioral information has been gained some attention in a dynamic simulation. There are well known equations in Mechanical Engineering that elegantly defines the relationship between strain and stress for material properties. When finite elements are employed, the mesh is divided up into equally sized tetrahedrons and these two material properties are applied for simulation. When mass-spring systems are employed, which is used often for cloth or fast 3D object simulations, the spring coefficients and the spring damping coefficients are required for simulation.

Wang et al. [21] manually measured the stretching and bending properties of cloth by performing a series of controlled mechanical tests. They attached a measured swatch of cloth to an apparatus with a grid backdrop. Using a weight and pulley system, they stretched the cloth in multiple directions and manually measured the cloth displacement using a grid backdrop. They then measured the bending property of the cloth as different lengths of cloth were draped over their own weight. Jochen Lang [14] described the ACME system for the deformable object using a robotic manipulator with an attached force gauge. Along with a trionocular stereo cameras and a point-cloud scanning device, the ACME system was able to build a 3D mesh of the object under various stages of the deformation. Using the known locally applied force to the object, as well as the displacements of the vertices of the object, Lang was able to estimate the Green's strain functions which can then be used in Finite Element or Boundary Element simulations to build a simulated object. Similar to this work, Becker et al. [1] developed a quadratic programming optimization routine for estimating the Young's modulus and Poisson's ratio of a homogenous isotropic object simulated using the linear finite element method, when a force/displacement measurement is known beforehand. Syllebranque and Boivin [20] developed a system of finding the Young's modulus and Poisson's ratio of a homogenous isotropic object by simulating the object in an iterative fashion. An error value at each iteration is calculated and if this value is above some threshold, the Young's modules and Poisson's ratio parameters are updated appropriately using simulated annealing method.

Barbara Frank et al. [9] used a robotic arm attached force gauge to apply localized pressure to a single point on the surface of the object. With a depth camera, they were able to determine the displacement of the object's surface during all phases of the experiment. They also developed an iterative method of simulation to find parameters the Young's modulus and Poisson's ratio which are adjusted each time. They applied an error function which uses the iterative closest point (ICP) algorithm to align the simulated deformed mesh with the captured deformed mesh and then returned the differences between simulated and deformed points.

With respect to data-driven cloth simulation, Bhat [2] found the location of cloth folds using by computing the gradient vector field of the image. Then, it defined an optimization routine which caused a simulated cloth swatch have the same behavior using simulated annealing. Similar work with video capture of cloth material properties was done by Kunitomo [13]. Wang [21] developed a method to mechanically measure the stretching and bending of cloth by performing a series of controlled tests.

Most of the previous works deal with deformable objects in some localized manner and directly measuring the strain/stress relationship to gather data points for generating the Green's function to be used in a FEM or BEM simulation. They define a global deformation experiment, thus the goal is to see how the whole object responds to under some strain/ stress activity which happens within the object and how the object responds when the force is removed. Gilles et al. [10] developed a novel method for modeling complex deformable meshless objects by using a material property map (stiffness map) to define how the control points (mesh nodes) are distributed across the object.

Bickel et al. [4] devised a new mechanism for data-driving simulated facial expressions using a hybrid method which both computes the large scale motion using linear shell deformation [3] but also uses sample poses to achieve highly detailed facial features. This work is done by marking the face with feature points, then marking the target face with the same feature points. Finally, these feature points from the source are mapped to the target. Miguel et al. [15] defined a data-driven process for estimating the non-linear material coefficients of cloth using a cloth swatch printed with a unique pattern, they used two types of clips which grab the cloth and a set of actuators to pull the clips with a certain amount of force. A camera was used to record the experiment and then computer vision algorithm was processed to output of cloth and clip configuration at each frame. The 3D geometry of the object at each frame was reconstructed with the binocular camera techniques by Bradley [5] and Otaduy et al. [16] provided a survey of recent work in the area of data-driven computer animation.

3 Capturing the material behavior from video stream

The experiment for capturing the global material behavior of deformable object is performed to find a trajectory of object. Initially, an object is propelled by an external force and then it changes its velocity upon collisions with other objects. During each collision and contact, the material properties play a deterministic role in exerting bounce off force that establishes the forthcoming behavior. We set some initial conditions and then let time, space, and the forces of nature to demonstrate how the object behaves. This is an important type of experiment in capturing the behavior of an object as it removes the limitations with how the object can behave. For instance, if we were to capture the behavior of cloth by stretching it in different directions using different amounts of stress, the cloth would exhibit a certain amount of strain, but that only gets us halfway to understanding how the cloth flaps in the wind or how it floats to the ground when dropped. These types of tests can consider both the environment and behavior of deformable objects as a whole.

In our experiments, a ball is dropped from several heights and recorded until the ball stops bouncing and it satisfies the global deformation requirement because no external force beyond gravity is introduced directly by the experimenter. The ball is initially set into the specific position, and then gravity, air, ball, and ground surface are the only factors for affecting to the deformable object.

When the ball is dropped from the predefined height to the ground, the movement of deformable ball and the width and height of the ball also is tracked for the duration of experiments. From these tests, we can understand how energy is lost when contact is made with the ground. When there is a large deformation of objects during the collision, more internal energy should be transferred to movement as more will be used by the internally moving particles, typically modeled with variance of elastic properties. To show this, we picked several types of balls (soft ball, waffle ball, ping pong ball, and air filled children's ball) with different material and they deform differently under the same external load. The goal of this test is to pick spheres from all different ranges of the elasticity spectrum, while all other aspects of the experiment are same.

3.1 Computer vision based input video analysis

In order to extract the height and width of ball over time, we developed a computer vision based tracking algorithm using openCV [6] to analyze each video frame, find the ball, and calculate its position and dimensions. The proposed method recorded the movement of object and shape that determined by two-dimensional information over time at each frame. The primary information for this process is how high the ball travels after each bounce. To reduce the effect of uneven illumination in an image frame, we applied the YCbCr color space (Y: luminance information, Cb: difference between the blue component, Cr: difference between the blue component) which separates the brightness information from the chrominance and chromaticity for robust tracking [17]. In addition, we painted the balls with red color and the background wall with blue color for our experimental tests, since these colors have high color contrast in the YCbCr color space.

Finding the ball in the image frame was conducted in four step processes. First, the image is blurred using a Gaussian filter [18]. Second, the RGB color image is converted to YCbCr color space. Third, since input image may include unwanted background, the image is cropped by a user selection of ROI (Region of Interest) [7]. Finally, a color threshold based scheme is used to find the min and max of x and y pixels for the ball. To improve the tracking of ball, user can select the red ball region to set up the threshold of pixel values. These user interventions are only necessary for the first frame of input video, and our color segmentation algorithm automatically finds the relevant threshold for Cb and Cr value based on the user-selected guideline for entire video stream to effectively extract ball position. Figure 1 shows the snapshot of our proposed method displaying the extracted ball region with bright green line. The histogram and binary image for Cb, Cr are shown in this figure as well. The experiments were conducted in front of a Casio Exilim EX-ZR200 camera which records a video at 120 frames per second. The balls were dropped from 1 to 1.6 m.

3.2 Capturing data points over time

Since the goal of the global deformation experiment is to find the trajectories of object from video stream, our program is focused to extract the position of ball and ratio of width and height over time which is shown in Fig. 2. The width/height ratio is fluctuated according to the pre-collision stretch and post-collision contraction, as an elastic ball is slightly squeezed and

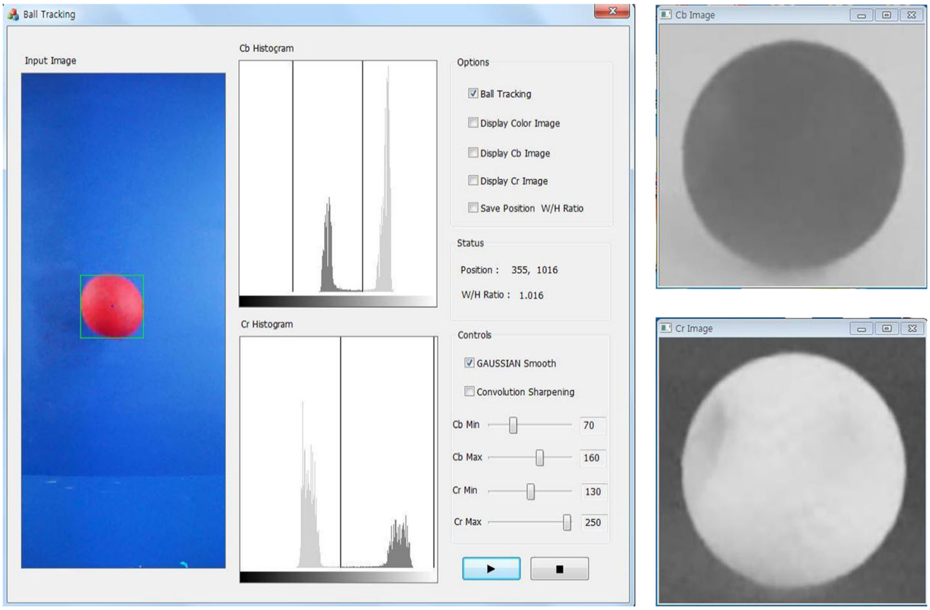


Fig. 1 Ball drop experiment with color segmentation scheme to find the ball

stretched as it moves from a compressed form. In this test, due to the frame rate of the camera being 60–120 fps, the exact collision moments were not always captured and the air-pressured ball's deformation is not that significant, and therefore the width/height ratio fluctuation is merely noticeable. In addition, due to the lighting condition, the lower area of ball is displayed darker when it rests on the floor, and therefore the height of the ball appears shorter and the width/height ratio stays less than 1 as the ball reaches to a resting status. Figure 3 shows a series of extracted ball position images over time to find the trajectory of the object.

4 Data-driving simulation coefficient

One of the main thrusts of this work is obtaining the elasticity coefficients through a data-driven simulation based on a given geometric structure, while the governing equation is

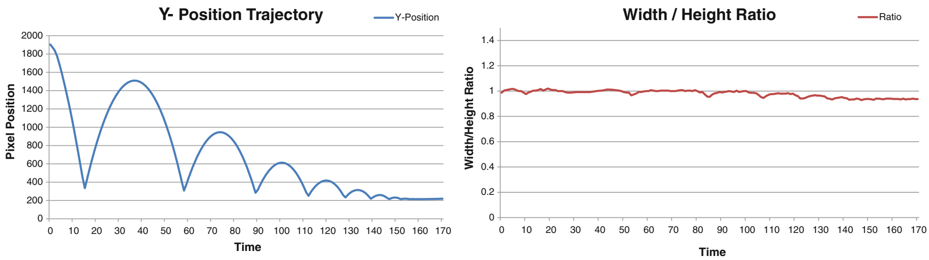


Fig. 2 Graphs of the ball's height over time and the width/height ratio over time

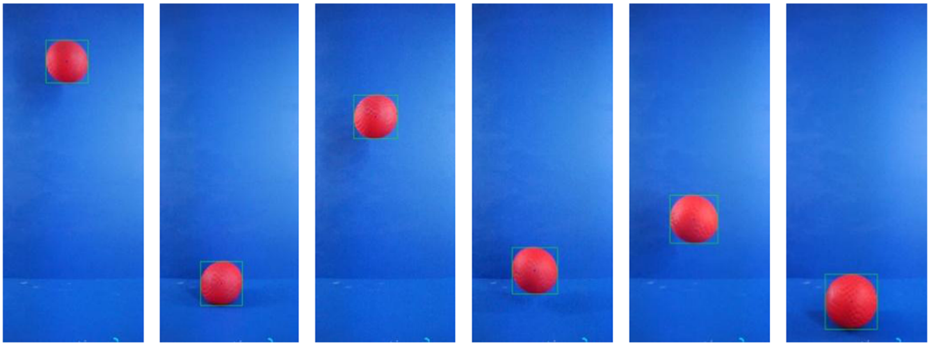


Fig. 3 Series of extracted ball position images over time

unchanged. Finding proper material properties are casted as a multivariate optimization problem where the objective function is to minimize the difference between the recorded and simulated ball's position over time and the status variables are material properties. In principle, deformation is influenced by intricate interaction among surface and internal mesh structure, material properties, and external forces, and therefore the way a virtual object is meshed is particularly important in deformation analysis. To isolate the mutual influence between applied force and elastic deformation within a predictable range, we use a pre-determined mesh structure and resolution for a given virtual object geometry. By using the fixed resolution mesh, the interplay between material properties and external excitation become apparent and they are directly associated each other, resulting in effective material property analysis. For instance, when the ball hits the ground, the global material properties are primary source of varying degrees of deformation. The proposed algorithms track the sphere's behavior over time as global material properties encompass both the moment contact forces are applied and the object's response to the energy potential created during contact. The search process starts with the initial guess of material properties and iteratively compares current virtual ball position with captured ball position, until they converge to a reasonable threshold. At every iteration, material properties' derivatives are evaluated to drive the function evaluation toward local minima. The threshold is directly related to the camera resolution and the recorded frame rate. Higher resolutions and frame rates return more precise position information, thus the matching threshold can be much smaller.

4.1 Simulation coefficients

The initial modeling operations are nicely divided into two phases; geometry and physics. First, a virtual model is generated based on the geometric characteristics of the real counterpart, such as dimensions and shape, as well as with reasonable granularity of internal mesh configuration. Secondly physical properties are either measured and assigned (i.e. total weight and mass distribution) or estimated with initial guess for elasticity, volume stiffness, and damping, etc. Although we may apply the trivariate B-splines approach [12] to efficiently create 3D deformable objects, we utilized TetGen [19] to generate internal mesh of 3D volumetric models in this research. The main geometric properties include the resolution, the radius/edge ratio constraint, and the volume constraint of tetrahedral elements. The resolution defines the density of the mesh at the surface of the object. The radius/edge ratio constraint defines the shape requirement of the tetrahedrons. A high radius/edge ratio would allow for long/thin tetrahedrons, while a low threshold would force all tetrahedrons to have similar sized faces. The volume constraint defines a maximum for the size of tetrahedron during mesh generation. Figure 4 shows the spheres with different setup:

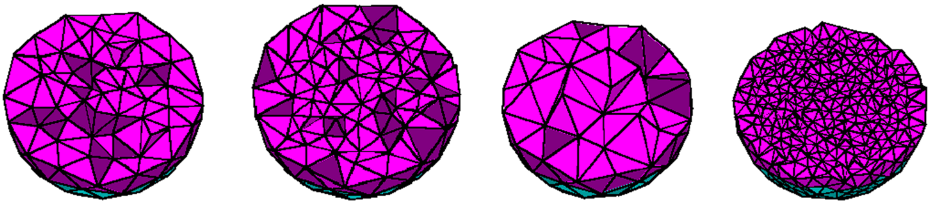


Fig. 4 Spheres with differing radius/edge length and volume constraints

the first sphere has a high radius/edge constraint and the second sphere a low radius/edge constraint. The third sphere has a high volume constraint and the forth sphere has a low volume constraint (from left to right). These geometric characteristics are determined prior to a data-driven simulation based on the characteristic requirements of a virtual object.

For physics modeling, the bullet engine [8] was used to simulate the bouncing balls deformation and overall behavior. This engine primarily uses mass-spring system and the coefficients of the governing equation can change the object's material properties by modifying the linear stiffness, angular stiffness, volume stiffness, pressure, damping, and impulse force distribution. The first four coefficients describe how the object deforms under stress, while the latter two coefficients are artificial forces that applied to the object to change their global behavior. While changing the elasticity coefficients has a substantial impact on the deformation of object, the modeling of internal sub-structure of the object also plays a key role for simulation. Initially basic coefficients are estimated by users and the proposed optimization routine explained below finds proper sets of material properties for the given geometric structure.

4.2 Estimating coefficients

Finding the optimal coefficients to make the simulated ball behave like the recorded ball is an optimization problem where the objective function should be minimized the bounce height difference between the simulated ball and the real ball. The bounding constraints are the numeric limits of the coefficients and most coefficients have been normalized to values between 0 and 1 for practical reasons, while boundless properties such as damping or lift are given with larger integer values (in our experiments 5–10). Our work is primarily focused on the linear elastic coefficients which are defined by the API of the bullet physics engine. The important coefficients for these experiments are the linear elasticity, angular elasticity, volumetric elasticity, and pressure. These coefficients had the highest impact on the object deformation and the residual bounce height. The secondary importance coefficients are drag, lift, damping, and definition of impulse after collision. These factors define how much of the energy is absorbed by the sphere and the ground when they are collided.

Another set of crucial coefficients for dynamic simulation is the geometric properties of the sphere. For the experimental tests, we defined the several coefficients for the resolution of the object and for the shape and size of the tetrahedrons that are used to build the spherical mesh. These are discrete data sets which are pre-determined in the geometric modeling stage and therefore the possible variances are limited by the number of available data sets. Note that these conditions may not be a factor when the captured motions can be obtained in all geometric resolutions sets. However, it can be a critical condition, when a specific motion is not obtainable using a specific set of geometry such as too lower resolution mesh that could not reach to the same behavior. The constraints are defined by the simulation coefficients which go into constructing the sphere and giving the sphere material properties. Before the optimization routine begins, the user may select the initial coefficients to be modified but the proposed algorithm will adjust them. This first step is similar to a weight factor in optimization and it is useful as we desire the most influential coefficients to be adjusted arbitrarily. Based on our experiments and priorities in our motion analysis, the coefficient order which produced the best results was: Linear Elasticity, Angular Elasticity, Volumetric Elasticity, Pressure, Damping, Drag, Lift, Soft/Rigid Impulse Split, Soft/Kinetic Impulse Split, Soft/Soft Impulse Split, Geometric Resolution, Tetrahedral Radius/Edge Length, and Tetrahedral Volume Constraint.

$$
\begin{aligned}
&\textit{minimize}\ f(r,s) = \sum_{i}^{bounce} (r_i - s_i)^2 \\
&\textit{subject to} \quad 0 \le elasticity_{linear} \le 1 \\
&\qquad 0 \le elasticit\ y_{angular} \le 1 \\
&\qquad 0 \le elasticit\ y_{volume} \le 1 \\
&\qquad -\infty \le pressure \le \infty \\
&\qquad 0 \le damping \le 1 \\
&\qquad 0 \le drag \le \infty \\
&\qquad 0 \le lift \le \infty \\
&\qquad 0 \le implus\, e_{\frac{soft}{rigid}} \le 1 \\
&\qquad 0 \le implus\ e_{\frac{soft}{kinetic}} \le 1 \\
&\qquad 0 \le implus\ e_{\frac{soft}{soft}} \le 1 \\
&\qquad 8 \le geometry_{resolution} \le 1024 \\
&\qquad 1.0 \le geometr\, y_{\frac{radius}{edgelenght}} \le 3.0 \\
&\qquad 0.01 \le geometry_{volume} \le 0.1
\end{aligned}
$$

Using the first derivatives of the object's discretized height over time, we found the direction change of ball, and the second derivative provides when the data was trending downwards. From this we found when and how high the ball bounced from the start of the experiment, until the ball came to the rest state. For equation described above, the optimization routine provides the estimation of proper coefficients for global behavior of deformable objects, where r is the array of recorded bounce heights in order and s is the array of simulated bounce heights in order and we had r_i for all i which is defined as the i^{th} bounce height for $1 <= i <= n$.

The proposed algorithm involves the simulating a bouncing ball over and over, adjusting the simulation coefficients each time, and stopping once the simulated object's bounce heights cannot get closer to the real object's bounce heights by adjusting the coefficients further. If we treat this optimization routine as a black box without priorities on preferred material properties, the convergency is substantially diminished and it takes much longer time. Since we already acknowledge some material properties such as linear elasticity have much greater impact on the bounce height, we treat each material property serially according to the pre-determined orders. Firstly, the algorithm starts with some default user defined values for the coefficient. After the first iteration, the Jacobian of each parameter is evaluated to determine whether the coefficient brings the simulation result closer to the real object bounce heights or further. Since the above process is to obtain a local minima starting from the initial guess, the initial guess is further altered in binary search way to make sure that the near optimal values can be acquired. This process is serially repeated for other parameters according to the pre-defined orders of material parameters. Setting up the coefficient priorities in-priori allow us to perform the optimization much quicker in practical sense without the complication of combinatorial multivariate optimization. The underlying assumption is that the order of coefficient priorities are correct, which was confirmed with our exhaustive trials. Thus our method initially runs the simulation with default coefficients, calculates the objective function for optimization, and then compares the result value of objective function value with the previous iteration. Depending on the outcome of this calculation, the proposed algorithm then knows how the coefficient affects for the bounce heights, and how to adjust the coefficient going forward. The coefficient is adjusted between each simulation run and the value of the objective function is tracked. Using the first derivative of this data at each iteration, our method finds the minimum of difference between simulated and real ball, and save the coefficient's optimal value before moving on to the next step and performing the same set of operations iteratively.

5 Experiment test and analysis

5.1 Global behavior capture and material property estimation applications

Our global behavior capture application utilizes an audio visual interleave video file (.avi) as an input file to capture the global behavior of object and allows selecting the ball's color by mouse clicks. Then the proposed algorithm finds the ball region with color components and returns the extracted ball region which is represented with bright green line. In addition, the histogram and binary image for Cb, Cr are also displayed for extra information as shown in Fig. 1. Our material property estimation application takes the data captured by the global behavior capture application and attempts to find the coefficients of physics simulation which make the behavior of simulated ball to match with the behavior of real ball. Our application provides the options for using a mass-spring system or generated tetrahedral meshes. Figure 5 shows the snapshot of our

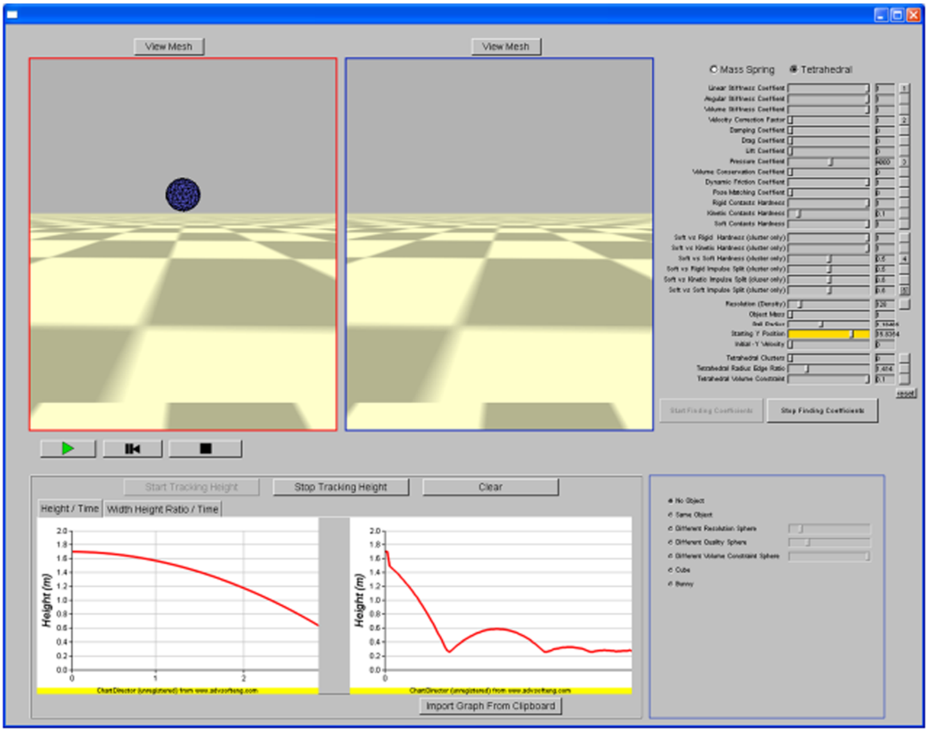


Fig. 5 Screen shots of the material property estimation application

implemented application and sliders along the right side allow for manual adjustment of the simulation coefficients by users to provide a platform for what-if questions.

5.2 Material property extraction

The first step of proposed method is taking a recorded experiment of ball that is dropped from some height and then this video is analyzed to extract and track the y position of object and

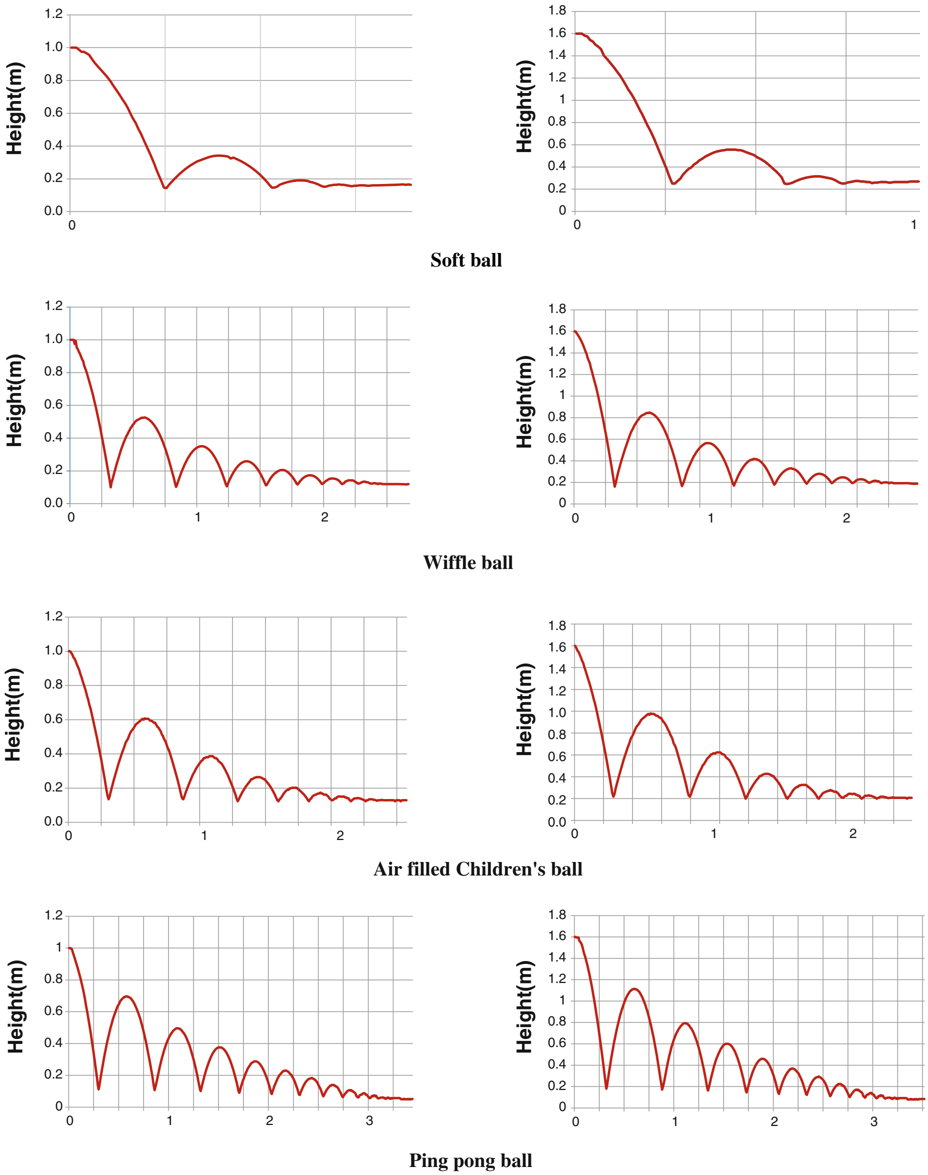


Fig. 6 Graphs of the several types of ball positions over time when dropped from 1.0 m (*left*) and 1.6 m (*right*)

dimensions. To verify our computer vision, data analysis, and tracking logic for the material property extraction, we used four different types of ball with different materials thus they have different behaviors when they are bounced. These balls include a ping-pong ball, wiffle ball, softball, and highly deformable air-filled children's ball. Figure 6 shows the result graphs which represent the y position of the balls in the test which drops several balls from 1.0 to1.6 m.

Since softball has low elasticity, the final resting position is quicker than the other three balls. The hollow wiffle ball bounces higher than the softball, but not as high as the air-filled ball, and finally the ping-pong ball bounces the highest and longest of all tested balls. The ping-pong ball has highly elastic plastic material, thus it is bounce well comparing with other balls. Our experimental result returns a smooth graph for the y-position of ball and it shows that the used frame rate is enough to precisely capture the peak and apex of the curve for ball movement. In all four cases, the proper material property combinations are obtained that well follow the global behavior of real objects in video streams.

5.3 Global deformation behavior simulation

The goal of global deformation simulation is to estimate the coefficients of the governing equation which cause the simulated ball to have the same bounce heights as the recorded ball. It is not always the case that the behavior of simulated ball is exactly matched with the behavior of recorded ball, but the proposed algorithm tries to find the coefficients such that the simulation solution is closer to the optimal solution when the simulation is started with default values. Although simulation time steps can be altered for various animation purposes, the time synchronization between real and virtual object is not concerned in this paper.

Sometimes when coarse tetrahedral meshes are used to the interior of objects, their internal structure diffuses much of the kinetic energy attained by the ball at the moment of collision. Therefore, the energy doesn't well convert into the upward momentum enough to propel the object to highly bounce for the wiffle ball, air-filled ball, or ping-pong balls. In such cases, material properties search alone fails to produce acceptable motion and an artificial extra force is needed to match the height and thus the match the global behavior. Stiffer material, such as a softball that bounces to much lower height, is lenient on this condition and was able to find elasticity coefficients with broader meshing options which make the simulated ball behave like the softball. Figure 7 shows that the proposed algorithm successfully finding the coefficients which made the simulated ball has the same bounce heights as the captured ball during the simulation.

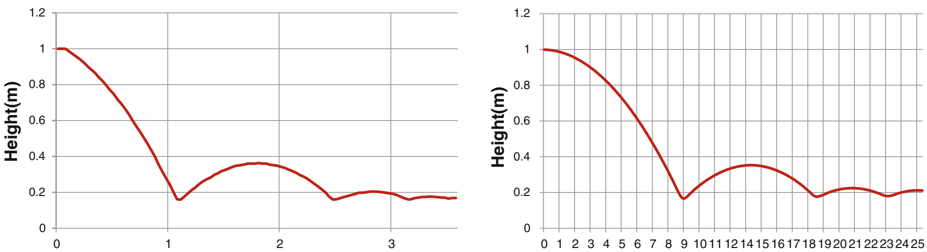


Fig. 7 Graphs of the y-position for captured real ball (*left*) and simulated ball (*right*)

6 Conclusion

Usually, animators want to create realistic 3D objects that are plausibly interacted with other 3D models to be used in movies, games, and research projects. This paper introduces the framework which allows animators to accomplish these goals quicker and more accurately than the traditional guess-and-simulate cyclic method that is currently employed. Typical data-driven approaches mainly focus on the localized behavior of object at the surface and not how the object globally reacts when the deformable object receives external forces. In this paper, we have successfully simulated an object which has the same globally elastic behavior as a real life object. To accomplish this, the proposed method extracts the material properties and uses that to data-driven based simulation while minimizing human interaction in the process. Furthermore, this research has shown that material property extraction is possible with recording devices and computer vision algorithm.

As a future study, although it requires a robust and complex image matching algorithm, multiple cameras including a high speed camera could be applied to achieve much more three dimensional information and clear images. Also, we can apply the shadow removal algorithm or can upgrade the illumination environments to achieve high quality video input for experiments.

Acknowledgments This work was partially supported by the Soonchunhyang University Research Fund (No. 20130575).

References

1. Becker M (2007) Robust and efficient estimation of elasticity parameters using the linear finite element method. Proc Simul Vis 15–28
2. Bhat K, Twigg C, Hodgins J (2003) Estimating cloth simulation parameters from video. Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, pp 37–51
3. Bickel B, Botsch M, Angst R, Matusik W, Otaduy M, Pfister H, Gross M (2007) Multi-scale capture of facial geometry and motion. ACM Trans Graph (TOG) 26(3):33
4. Bickel B, Lang M, Botsch M (2008) Pose-space animation and transfer of facial details. Proceedings of the 2008 ACM SIGGRAPHEurographics Symposium on Computer Animation, Eurographics Association, pp 57–66
5. Bradley D, Boubekeur T, Heidrich W (2008) Accurate multi-view reconstruction using robust binocular stereo and surface meshing. IEEE Conf Comput Vis Pattern Recognit 1–8:2008
6. Bradski G (2000) The OpenCV Library. Dr. Dobb's Journal of Software Tools
7. Chaudhary A, Vatwani K, Agrawal T, Raheja JL (2012) A vision-based method to find fingertips in a closed hand. J Inf Process Syst 8(3):399–408
8. Erwin C. Bullet Physics Library. [Online]. https://code.google.com/p/bullet/
9. Frank B, Schmedding R (2010) Learning the elasticity parameters of deformable objects with a manipulation robot. Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, pp 1877–1883
10. Gilles B, Bousquet G, Faure F, Pai DK (2011) Frame-based elastic models. ACM Trans Graph (TOG) 30(2):15
11. Jeon JH, Choi M-H, Hong M (2012) Enhanced FFD-AABB collision algorithm for deformable objects. J Inf Process Syst 8(4):713–720
12. Kim J, Yoon S, Lee Y (2014) Trivariate B-spline Approximation of Spherical Solid Objects. J Inf Process Syst 10(1):23–35
13. Kunitomo S, Nakamura S, Morishima S (2010) Optimization of cloth simulation parameters by considering static and dynamic features. ACM SIGGRAPH 2010 Posters 15:2010
14. Lang J (2001) Deformable model acquisition and validation. PhD Thesis, University of British Columbia
15. Miguel E, Bradley D, Thomaszewski B, Bickel B, Matusik W, Otaduy MA, Marschner S (2012) Data-driven estimation of cloth simulation models. Comput Graph Forum 31(2):519–528

16. Otaduy M, Bickel B, Bradley D, Wang H (2012) Data-driven simulation methods in computer graphics: cloth, tissue and faces. ACM SIGGRAPH 2012 Courses
17. Sebastian P, Voon YV, Comley R (2010) Colour space effect on tracking in video surveillance. Int J Electr Eng Inf 2(4):298–312
18. Shah M (1997) Fundamentals of computer vision. University of Central Florida, Orlando
19. Si H (2004) TetGen: a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, version 1.3, software. Weierstrass Inst. for Appl. Anal. and Stochastics
20. Syllebranque C, Boivin S (2008) Estimation of mechanical parameters of deformable solids from videos. Vis Comput 4(11):963–972
21. Wang H, O'Brien J, Ramamoorthi R (2011) Data-driven elastic models for cloth: modeling and measurement. ACM Trans Graph (TOG) 30(4):1–11

Min-Hyung Choi is an associate professor at the Department of Computer Science and Engineering at University of Colorado Denver. He received his M.S. and Ph.D. from University of Iowa in 1996 and 1999 respectively. His research interests are in Computer Graphics, Scientific Visualization and Human Computer Interaction with an emphasis on physically-based modeling and simulation for Neuroscience and Bioinformatics applications. Currently he is the Director of Computer Graphics and Virtual Environments Laboratory. In 2003, he received the NSF CAREER Award from National Science Foundation.

Steven C. Wilber is received his B.S. from the Metropolitan State University of Denver in Computer Science. He is currently working towards his Ph.D at the Department of Computer Science and Engineering at University of Colorado Denver. His research interests are in Computer Graphics, and Physically-based modeling and simulation.

Min Hong is an associate professor at the Department of Computer Software and Engineering, Soonchunhyang University in Asan, Korea. He received B.S. in Computer Science from Soonchunhyang University in 1995. He also received M.S. in Computer Science and Ph.D. in Bioinformatics from the University of Colorado in 2001 and 2005, respectively. His research interests are in Computer Graphics, Mobile Computing, Physically-based Modeling and Simulation, Bioinformatics Applications, and u-Healthcare Applications. In present, he is a director of Computer Graphics Laboratory at Soonchunhyang University.