2011 12th International Conference on Computer-Aided Design and Computer Graphics

Anisotropic Kernels for Meshless Elastic Solids

Ning Liu, Fei Zhu, Sheng Li, Guoping Wang
Graphics and Interaction Lab
Peking University
Beijing, China
Email: {liuning,zhuf}@graphics.pku.edu.cn, {lisheng, wgp}@pku.edu.cn

***Abstract*—We propose a meshless method to simulate elastic solids. Explicit integration methods are widely used in fluid/solid simulators for their efficiency, but these methods are not unconditionally stable: without sufficient small timesteps, simulated particles may move beyond range of each other, resulting in simulation breakdown or other unexpected errors. This problem which usually appears under large deformations is called numerical fracture. We use anisotropic kernels to reduce numerical fracture without resampling procedure. During each timestep, we update the anisotropic kernels from the analysis of the strain tensor to capture the directions of the deformation. Results illustrate that our method improves the stability of the simulation with minimum computation cost.**

***Keywords*-Anisotropic Kernels; Meshless; Solid Simulation;**

I. Introduction

Physically based solid simulation is an active topic in computer graphics research and applications. Solid behavior such as elasticity leads to many appealing visual effects in animated films, virtual reality and video games. Since the early work of [1], a large body of techniques has been developed to simulate elastic objects. Among all the methods, the Finite Element Method(FEM) is one of the most popular techniques used in research and engineering fields. However, the accuracy of FEM computation depends on maintaining a high quality mesh which is sometimes costly and difficult to maintain. Meshless methods were developed with the objective of eliminating part of the difficulties and have been applied to graphics successfully. The unstructured scheme of meshless methods is quite suitable for simulating topology changing phenomenon such as running fluid, fluid-fluid fluid-solid interaction. Recent years, it is also a appealing topic to simulate elastic solids with meshless methods [2].

In meshless methods, objects are treated as a set of particles, each of which has local influence only on nearby neighbors(shown in figure 1), quantitively the influence depends on kernel functions. To animate the object, implicit methods and explicit methods are used. Implicit methods are unconditionally stable, but are also complex to implement and computational expensive. Considering efficiency and simplicity, explicit methods are used intensively in fluid/solid simulators. However, explicit methods are not unconditionally stable. Under large deformations, particles may lose neighboring informations which are important for computing elasticity, this leads to simulation breakdown or other unexpected errors. This problem is referred to as numerical fracture.

In this work, our main **contribution** is a meshless framework using well-designed anisotropic kernels to reduce numerical fracture. The simple but effective idea is that we expand the kernels along the tensile directions, thus maintain the neighboring information as much as possible. The experimental results illustrate that our anisotropic kernels improve the stability of elastic simulation. Moreover, the anisotropic kernels we propose can be treated as a general technique, which can be integrate into different meshless methods to simulate more potential physical behaviors.

II. Related Work

Our work is closely related to physically based solid simulation and meshless methods in computer graphics.

To simulate elastic solids, a large variety of approaches have been proposed by researchers. Some representative work includes: Finite Difference Method[1], the Boundary Element Method (BEM) [3], the Finite Element Method (FEM) [4], the Finite Volume Method(FVM) [5], mass-spring systems [6]. For details about physically based simulation techniques, we refer the interested readers to two surveys [7], [8]. Here we pay more attention to meshless techniques in computer graphics.

Meshless methods such as smooth particle hydrodynamics is quite popular in fluid simulation [9], fluid-solid interaction[10], [11] and granular material simulation [12], [13]. The unstructured nature of meshless methods makes them good choices for simulating frequently topology-changing objects such as fluid. In recent years, researchers also try to simulate solid with meshless methods. [14] models soft inelastic material using particle systems. [15] introduces SPH method to model deformable solids and later extended it to handle space and time adaptive sampling [16]. By extending fluid simulation with viscoelasticity and viscoplasticity, toothpaste-like solid can be simulated in [17], [18], [19]. More realistic animation is achieved using computational mechanics[20], [21], [22]. [21] uses a Moving Least Square method to compute deformation gradient. Their method is capable of simulating a wide range of elastically

978-0-7695-4497-7/11 $26.00 © 2011 IEEE
DOI 10.1109/CAD/Graphics.2011.33

and plastically deformable objects, however, the method demands that each particle had at least three neighbors in non-degenerate locations. [23] couples meshless finite element formulation and keyframe targeting, motion control of the simulated deformable objects is achieved at interactive rates. [24] replaces [21]'s MLS method by SPH formation, which has the advantage of handling coarsely sampled or even coplanar particle configuration. However, their formation is not rotationally invariant, which leads to erroneous rotation. [25] solves the incorrect rotation problem in [24] from a co-rotational point of view, that is the similar idea which has been proposed in [26]. [27] introduces an interactive technique called shape matching which is later improved by [28] and [29]. [30] also propose a novel method to compute deformation gradient and no rest configuration is required. As they diagonalize the elastic deformation using singular value decomposition, the forces computed are rotationally invariant. More recently, [2] developes an accurate unified treatment of elastica. They derive a new quadrature rule for volumetric deformation fields which offers unified treatment when simulating spanning rods, shells, and solids. [31] presents a new method to reconstruct smooth surface from particles, their method is able to reduce surface blobbing using anisotropic kernels. For more general techniques about meshless methods, we refer the readers to [32].

Technically, our method is most related to [33], [34]. [34] defines the kernels from geometric features, but their kernels are still isotropic. [33] derives the anisotropic kernels in rate form, but their method depends on accurate time integration which is computational expensive. Our method directly updates the anisotropic kernels according to analysis of the strain tensors, no accurate kernel integration is needed.

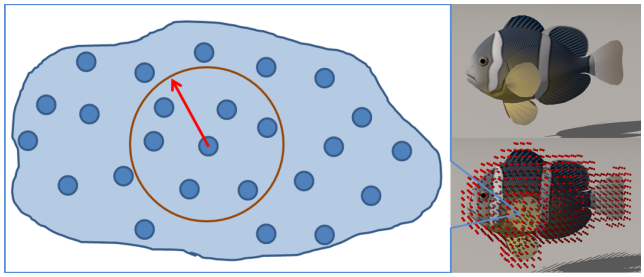

Figure 1. The elasticity for a specific particle is computed by nearby neighboring particles in the range of the kernel(shown in read circle).

III. Overview

Figure 2 shows the pipeline of our meshless simulation framework. In each timestep, particles are moved to new positions by explicit time integration. Unlike previous method, we add 'Update Kernel' step to compute anisotropic kernels according to current strain tensor. Then stress and external forces(gravity, collision force) are computed to accelerate the particles. We show the implementation details and results in section V.

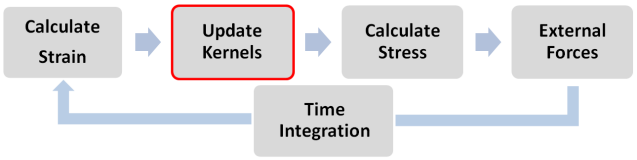

Figure 2. The overview of our simulation pipeline. We dynamically update the anisotropic kernels for elastic computation.

IV. Meshless Elasticity Using Anisotropic Kernels

In computer graphics, meshless methods receive a lot of attention due to its potential in eliminating the costly effort of mesh generation, which is a common operation in finite element analysis. After success application in fluid simulation, recent research is active in exploring ways to simulate solids with meshless methods. Comparing to traditional finite element methods, elastic solids simulated with meshless methods can be more easily to extended to simulate phenomenon such as elastoplasticity, viscoelasticity, fracture and fluid-solid interaction.

To validate our anisotropic kernels, we implemented an elastic simulator based on Corotated SPH[25]. Other simulator frameworks such as MLS([21]) can use our anisotropic kernels in a similar way.

A. Anisotropic Kernels

The Smooth Particle Hydrodynamics takes the object as discrete particles p_i with position $\mathbf{x_i}$, mass m_i, volume V_i and material density $\rho_i = m_i/V_i$. Continuous property A (such as force, pressure) and its corresponding gradient is computed from the discrete particles using SPH formation:

$$A(\mathbf{x}) = \sum_j \frac{m_j}{\rho_j} A_j W(\mathbf{r}, h) \tag{1}$$

where j is the index of the neighboring particle p_j at distance below h from $\mathbf{x}$, $\mathbf{r} = \mathbf{x} - \mathbf{x_j}$ is the distance vector. Choice can be made from many different kernel functions according to specific application, for more detail on $W(\mathbf{x} - \mathbf{x_j}, h)$, see [35], [36]. Although it is free to choose from different kernels, the computation of elastic forces need more consideration. When particles move closer to each other, a larger elastic force is needed to repel the particles from each other. So the kernel function must have non-zero gradient value when distance between particles fall to zero. For this reason, we choose the Spiking kernel used in [15] for elastic force computation.

$$W(\mathbf{r}, h) = \frac{15}{\pi h^3}(1 - \frac{\|\mathbf{r}\|}{h})^3 \tag{2}$$

To construct the anisotropic kernels, we replace the scalar smoothing length h in equation (1) by a 3 by 3 tensor $\mathbf{G}$.

$$A(\mathbf{x}) = \sum_j \frac{m_j}{\rho_j} A_j W(\mathbf{r}, \mathbf{G}), \tag{3}$$

with anisotropic version of equation (2) as

$$W(\mathbf{r}, \mathbf{G}) = \frac{15}{\pi} \|\mathbf{G}\| (1 - \|\mathbf{G}\mathbf{r}\|)^3 \tag{4}$$

Obviously the isotropic kernel can be treated as a special case of the anisotropic kernel by $\mathbf{G} = h^{-1}\mathbf{I}$ where $\mathbf{I}$ is Identity matrix. The anisotropic tensor $\mathbf{G}$ can be thought of as a transform matrix that transform distance vector $\mathbf{r}$ to $\mathbf{G}\mathbf{r}$. We also normalize $A(\mathbf{x})$ to satisfy the unity property for a specific $\mathbf{G}$. We show how $\mathbf{G}$ is defined in section IV-C.

B. Continuum Equations for Elastic Solid

In this section, we introduce the basic equations for elastic force computation.

When a solid object deforms, each point in the material space moves from the original position $\mathbf{x^0}$ to a new position $\mathbf{x}$. The displacement field can be derived as $\mathbf{u} = \mathbf{x} - \mathbf{x^0} = [\mathbf{u}, \mathbf{v}, \mathbf{w}]^\mathbf{T}$. The Jacobian of the transform is given by $\mathbf{J} = \mathbf{u}^\mathbf{T} + \mathbf{I}$ with $\mathbf{I}$ standing for the identity matrix. The strain can be computed via the non-linear Green-Staint-Venant strain tensor

$$\varepsilon = \frac{1}{2}(\mathbf{J}^\mathbf{T}\mathbf{J} - \mathbf{I}) \tag{5}$$

or its linearized Cauchy-Green tensor

$$\varepsilon = \frac{1}{2}(\mathbf{J}^\mathbf{T} + \mathbf{J} - \mathbf{I}) \tag{6}$$

for isotropic materials, the stress is computed through the Hooke constitutive law $\sigma = \mathbf{C}\varepsilon$ which indicates the stress and strain is linearly related. C is a 6 by 6 matrix which only depends on Young's Modulus and Poisson Ratio.

For each particle i with its strain and stress, we can calculate its strain energy U_i as

$$U_i = V_i \frac{1}{2}(\varepsilon_i \cdot \ \sigma_i) \tag{7}$$

where V_i represent the volume of particle i such that $V_i = m_i/\rho_i$. The elastic force that particle i exerts on its neighboring particle j can then be defined as the negative of the strain energy with respect to displacement.

$$\mathbf{F_{ji}} = -\nabla_{\mathbf{u_j}} U_i = -V_i(\mathbf{I} + \nabla \mathbf{u_i}^T)\sigma_i \mathbf{d_{ij}} \tag{8}$$

and $\mathbf{d_{ij}}$ is defined by

$$\mathbf{d_{ij}} = V_j \nabla W(\mathbf{x_{ij}}, \mathbf{G}) \tag{9}$$

To compute the $\nabla \mathbf{u}^\mathbf{T}$ so as to compute elastic force, several meshless methods exist. [21], [37] use a MLS mothod to get the deformation gradient, but the method will fail when the particles are scatterd in a coplanar or collinear configuration. [24] instead use a SPH formation, but they did not take rigid rotation into consideration when computing deformation gradient, so erroneous forces arise and prevent the solid from correct rotation. [25] derive the method from [24] and solve the rotation problem in a corotational way. We will follow the method of [25], but with our anisotropic kernel $\mathbf{G}$.

$$\nabla \mathbf{u_i} = \sum_j V_j \mathbf{u_{ji}} \nabla W(\mathbf{x_{ij}}, \mathbf{G}) \tag{10}$$

where $\mathbf{u_{ji}}$ is the vector between neighboring particles i and j

$$\mathbf{u_{ji}} = \mathbf{u_j} - \mathbf{u_i} = \mathbf{R^{-1}}(\mathbf{x_j} - \mathbf{x_i}) - (\mathbf{x_j^0} - \mathbf{x_i^0}) \tag{11}$$

The rotation matrix $\mathbf{R}$ which indicates the rigid rotation between the particle configuration and the rest configuration. $\mathbf{R}$ is computed as

$$\mathbf{A_{pq_i}} = \sum_j (\mathbf{x_j} - \mathbf{x_i})(\mathbf{x_j^0} - \mathbf{x_i^0})^\mathbf{T} \tag{12}$$

$\mathbf{A_{pq_i}}$ is computed according to the local neighboring particles of particle i. $\mathbf{A_{pq}}$ has one rotation part $\mathbf{R}$ and one symmetric part $\mathbf{S}$ such that $\mathbf{A_{pq}} = \mathbf{RS}$ and $\mathbf{R}$ can be extracted by stable SVD[25], [30] or polar decomposition[27]. We compute $\mathbf{A_{pq_i}}$ using the initial neighboring particles by the a initial neighbor table which is stored when simulation starts.

Finally the elastic force F_i for each particle is computed in a symmetrized form

$$\mathbf{F_i} = \sum_j \frac{-\mathbf{R_i}\mathbf{f_{ji}} + \mathbf{R_j}\mathbf{f_{ij}}}{2} \tag{13}$$

C. Updating Anisotropic Kernels

The basic idea of constructing the anisotropic kernels is to expand the kernels along tensile directions. Under large stretch deformation, the new constructed kernel can still cover the original neighboring particles to some extent, this reduces the chance of numerical fracture.

Supposing we are computing some quantity for the red particle shown in Figure 3, under large stretch deformation along the axis, the red particle's neighbors(green particles) move away from itself. For isotropic kernels(figure in the middle), the contributions from the green particles drop to very small values which are not sufficient to compute accurate values of the red particle. While for anisotropic kernels(figure at the bottom), the kernel expand along the stretch direction and there are still sufficient contribution from green particles to the red particle.

To get the directions of the deformation for particle P_i, we pick the strain values $\{\lambda_1, \lambda_2, \lambda_3\}$ on the diagonal of the P_i's strain tensor. The values on the diagonal of strain reflect the deformation along each axis(x,y and z). If $\lambda_k > 0$

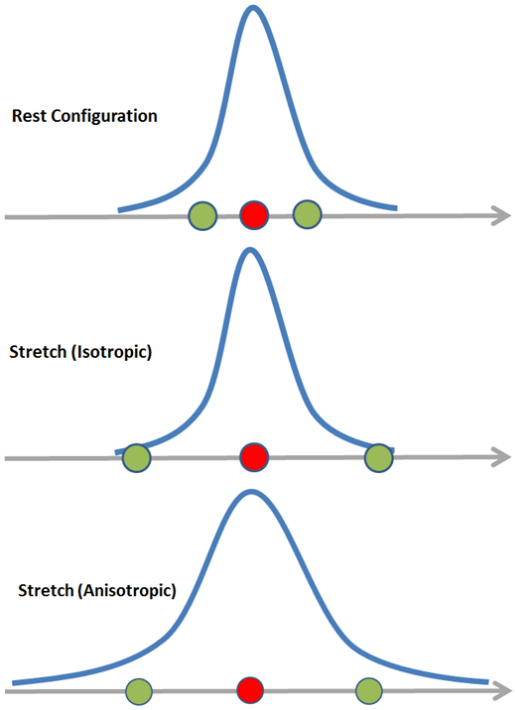

Figure 3. This 1D example shows the isotropic and anisotropic kernels under large stretch deformation. Particles(red and green circles) are set apart from each other under deformation. Our anisotropic kernels evolve according to the deformation direction.

$(k = 1, 2, 3)$, the object is stretching along axis k, we set entry G_{kk} of $\mathbf{G}$ as

$$G_{kk} = \frac{1}{(1 + \eta s)h} \tag{14}$$

where h is kernel length in rest state, s is a scale factor indicating how much the kernel expands, η is a constant parameter indicates the maximum or minimum scaling compared with h. We set η to be 1.0, meaning that the kernel can expand as much as $2h$ in length along axis k. The scale factor s is related to the physical fracture threshold ε_{max} of the material, which is constant in isotropic materials with

$$s = \begin{cases} 1 & \varepsilon_{kk} \geq \varepsilon_{max} \\ \frac{\varepsilon_{kk}}{\varepsilon_{max}} & \varepsilon_{kk} < \varepsilon_{max} \end{cases} \tag{15}$$

where ε_{kk} is the item from P_i's stain ε.

The reason why s is defined by physical fracture threshold ε_{max} is simple: under ideal conditions, the anisotropic kernels will keep the neighboring particles together until physical fracture happens. This setting prevents the situation when anisotropic kernels erroneously hold together the particles, which should have been torn apart by physical fracture.

V. Implementation and Results

Particles in our implementation are arranged into a spatial hash grid [9] to accelerate neighbor finding. At the beginning of each time step, we insert the particles into the spatial grid for later use. Anisotropy is handled when finding neighbors by transform the distance vector by the anisotropic kernel $\mathbf{G}$.

Collision detection is handled for particle-particle and particle-obstacle interaction. We use penalty forces as the collision response. The penalty force experts on a particle is determined by the relative velocity and distance to the particle. Collision forces between particles are computed only for non-neighboring particles, these forces prevent the object from penetrate itself under large deformations.

To move the particles to new positions, we use Leap-Frog integration which is computationally efficient as well as accurate. We choose the timestep smaller than the threshold $\delta t = \frac{h}{\|\mathbf{V_{max}}\|}$ to avoid self-penetration, where h stands for the minimum kernel dimension and $\mathbf{V_{max}}$ is the maximum particle velocity.

For rendering, the point based rendering technique [38], [39] can be applied to our framework. However, to take advantage of off-the-shelf high quality ray-tracing tools, we embed a detailed triangle surface into the particles using barycentric coordinates and the surface vertices move by interpolating its corresponding particles' positions. The surfaces are then rendered by POV-Ray(www.povray.org).

Figure 4 and Figure 5 show elastic bars under tensile and twisting forces. We use same models and parameters for isotropic(top) and anisotropic(bottom) kernels. Under small deformations, both bars are stable. While deformation accumulates, the bar with isotropic kernels begin to break down with particles falling apart, while at the same time, bars with anisotropic kernels can still hold the particles together to advance the simulation.

Figure 6 shows our test case of a elastic plant. At the beginning of the simulation, the top of the plant is dragged by a instant force, then the whole plant starts to shake. This model has detailed small features such as thin sheet and long rod. Result shows that our method can handle complex models like this one.

Figure 8 shows a elastic fish drops onto floor and bounces back. Figure 7 shows a swimming fish in sea. Notice the large deformation of the fishes' tail fins, our method is able to keep the simulation stable. Figure 9 shows another drop sequence(topleft,topright,downleft,downright) of a armadillo model.

Table I shows the geometry complexity and timing of our method. The geometry statistics shows that detailed triangle surfaces can be embedded in particles of much lower resolution. 'Building Grid' represents the cost of building the spatial hash. 'Integration&Collision' measures the time of time integration and collision handling. 'Elastic Calculation' is the most time consuming part of our method. Performance comparison between isotropic kernels([25]) and our anisotropic kernels implementation is shown in Figure 10. As our method add minimum computation cost(such as 3

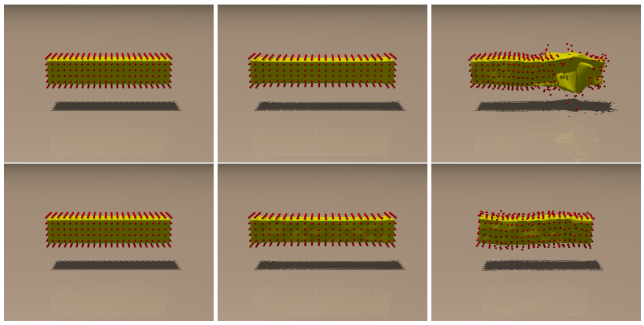

Figure 4. A Bar is tested under tensile force to show its stability with isotropic kernels(top) and anisotropic kernels(bottom). Anisotropic kernels prove to be more stable compared to isotropic kernels. Particles with isotropic kernels break down early than those with anisotropic kernels.

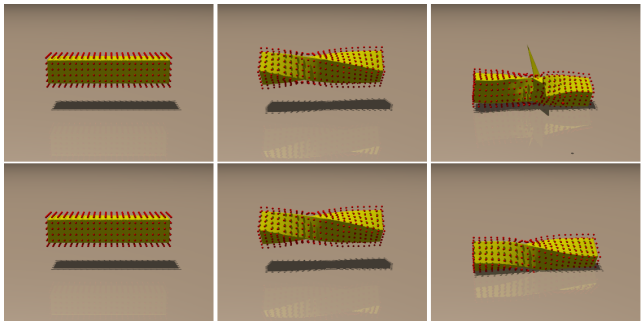

Figure 5. A Bar is twisted from its two ends to show its stability with isotropic kernels(top) and anisotropic kernels(bottom). Anisotropic kernels prove to be more stable compared to isotropic kernels. Particles with isotropic kernels break down early than those with anisotropic kernels.

by 3 transform) to the original isotropic implementation, our method is very fast with little performance dropoff. The efficiency of our method can be used in interactive applications. Please refer to our accompanied video for detailed results.

General physical parameters are shown in Table II. Our un-optimized C++ code runs on PC with 2.93GHz CPU, NVIDIA GT240 graphics card and 3G RAM memory.

VI. Further Discussion

Meshless Framework. Different meshless methods have diverse approaches to compute elasticity, but share the same concept that quantity for a position is computed by contributions from nearby particles through a kernel function.

Table II
General physical parameters.

Parameters	Value
Timestep	$(1 \rightarrow 3)E^{-4}$
Gravity	-9.8m/s^2
Kernel Range(h)	0.008m
Initial Spacing	0.0048m
Young's Modulus/Poisson Ratio	$9.0E^8$/0.33
particle mass	$4.0E^{-3}kg$

Figure 6. The top of the plant is dragged by a instant force. Leaves and flowers moves with it. This complex model include detailed features of thin sheet and rod.

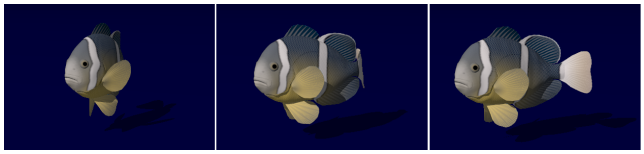

Figure 7. A fish swims in the water with its tail fin moves in large deformation. Please refer to our accompanied video for more details.

Although our implementation is build on Corotated-SPH [25], the anisotropic kernels we propose can be viewed as a general technique which is not limited by any specific meshless method, so they can be integrated into other meshless framework such as MLS [21].

Relation with Resampling. Another technique to reduce numerical fracture is resampling. In areas with large deformations, the distribution of particles become sparse, new particles can be inserted into these areas to maintain the accuracy of the simulation. We don't resample the particles in our method because in interactive applications, changing the number of simulated nodes will cause frame-rate jitter, which is an undesirable artifact. Luckily, our method focus on the local neighborhood adjustment for a particle instead of global distribution, and get no conflict with resampling. Under some circumstances, these two techniques can be combined to gain better stability.

VII. Conclusion and Future Work

We presente a meshless method to simulate elastic solids. Meshless simulation with isotropic kernels may cause numerical fracture under large deformation due to loss of neighboring particles' contributions. We capture the deformation direction using novel anisotropic kernels to better maintain neighboring particles' contributions. Results show that our method is applicable to large deformations in varied

Table I
GEOMETRY COMPLEXITY AND TIMING OF THE TEST MODELS.

Model	#Triangles	#Particles	Building Grid	Integration&Collision	Elastic Calculation	Rendering
Bar	11618	792	14ms	4ms	7ms	2s/frame
Swimming Fish	6628	1835	18ms	14ms	20ms	1s/frame
Plant	25656	2352	19ms	16ms	22ms	3s/frame
Dropping Fish	8874	3711	26ms	29ms	43ms	2s/frame
Armadillo	25273	4152	26ms	27ms	44ms	2s/frame

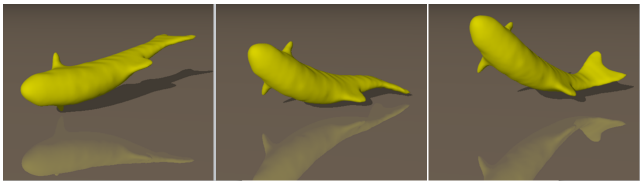

Figure 8. A dropping fish hits the floor and bounces back.

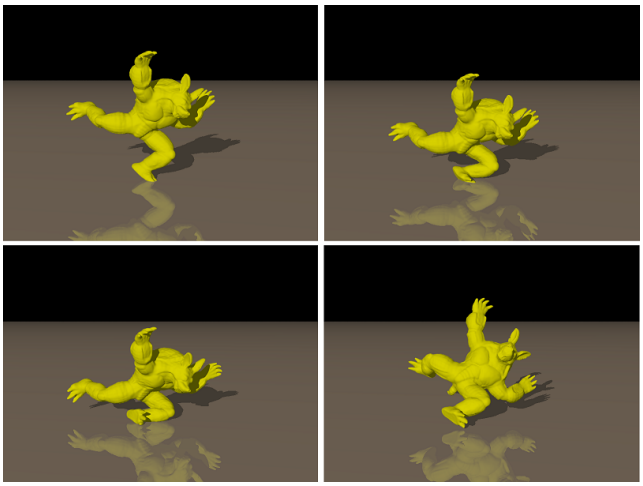

Figure 9. The figure sequence(topleft, topright, downleft, downright) show armadillo model drop to the floor with its leg under itself.

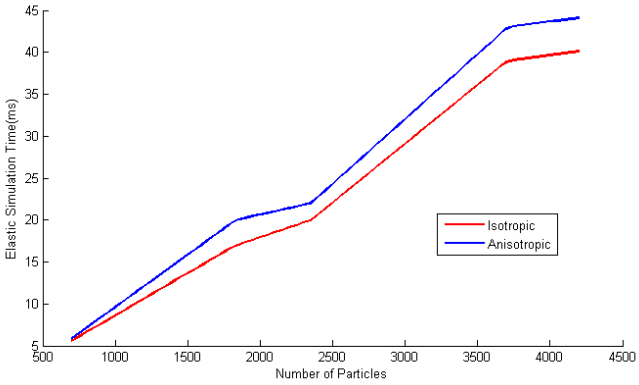

Figure 10. Performance comparison of Isotropic([25]) and Anisotropic Kernels. Our method improve the simulation stability with minor performance dropoff.

cases, with improved stability at the cost of acceptable performance dropoff.

In addition, our method can be coupled with resampling techniques to support more stable simulations. As our work focus on improving the kernel functions, it is not limited to specific meshless implementation and can be applied to different meshless frameworks.

Future work includes applying the anisotropic kernels to simulate more solid behaviors such as elastoplasticity, viscoelasticity and physically-based fracture. As our method is targeted at the numerical fracture, only stretch deformation is considered, it is also interesting to investigate the compression stability problem with anisotropic kernels.

ACKNOWLEDGMENT

This research was supported by grant No.2010CB328002 from the National Basic Research Program of China, and by Nos. 60833007, 60925007 and 90915010 from National Natural Science Foundation of China. It was also supported by grants No.2009AA01Z324 from the National High Technology Research and Development Program of China and research fund from the Doctoral Program of Higher Education (No.20070001024) .

REFERENCES

[1] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214. ACM, 1987.

[2] S. Martin, P. Kaufmann, M. Botsch, E. Grinspun, and M. Gross. Unified simulation of elastic rods, shells, and solids. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010.

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

[4] G. Debunne, M. Desbrun, M.P. Cani, and A.H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36. ACM, 2001.

[5] J. Teran, S. Blemker, V. Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on*

Computer animation, pages 68–74. Eurographics Association, 2003.

[6] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International, 2004. Proceedings*, pages 312–319. IEEE, 2005.

[7] S.F.F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. *MERL, TR-97*, 19, 1997.

[8] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In *Computer Graphics Forum*, volume 25, pages 809–836. Wiley Online Library, 2006.

[9] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In D. Breen and M. Lin, editors, *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 154–159, San Diego, California, 2003. Eurographics Association.

[10] M. Becker, H. Tessendorf, and M. Teschner. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics*, pages 493–503, 2008.

[11] T. Lenaerts, B. Adams, and P. Dutré. Porous flow in particle-based fluid simulations. *ACM Transactions on Graphics (TOG)*, 27(3):49, 2008.

[12] N. Bell, Y. Yu, and P.J. Mucha. Particle-based simulation of granular materials. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 77–86. ACM, 2005.

[13] Toon Lenaerts and Philip Dutré. Mixing fluids and granular materials. *Comput. Graph. Forum*, 28(2):213–218, 2009.

[14] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. In *SIGGRAPH*, pages 287–290, 1995.

[15] M. Desbrun and M.P. Gascuel. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation*, volume 96, pages 61–76. Citeseer, 1996.

[16] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *SIGGRAPH*, pages 31–36, 2001.

[17] S. Clavet, P. Beaudoin, and P. Poulin. Particle-based viscoelastic fluid simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, page 228. ACM, 2005.

[18] T.G. Goktekin, A.W. Bargteil, and J.F. O'Brien. A method for animating viscoelastic fluids. In *ACM SIGGRAPH 2004 Papers*, page 468. ACM, 2004.

[19] A. Paiva, F. Petronetto, T. Lewiner, and G. Tavares. Particle-based viscoplastic fluid/solid simulation. *Computer-Aided Design*, 41(4):306–314, 2009.

[20] T.P. Fries and H.G. Matthies. Classification and overview of meshfree methods. *Department of Mathematics and Computer Science, Technical University of Braunschweig*, 2003.

[21] Matthias Müller, Richard Keiser, Andrew Nealen, Mark Pauly, Markus Gross, and Marc Alexa. Point based animation of elastic, plastic and melting objects. In Dinesh K. Pai and Ronan Boulic, editors, *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 141–151, Grenoble, France, 2004. Eurographics Association.

[22] Xiaohu Guo and Hong Qin. Real-time meshless deformation. *Journal of Visualization and Computer Animation*, 16(3-4):189–200, 2005.

[23] B. Adams, M. Ovsjanikov, M. Wand, H.P. Seidel, and L.J. Guibas. Meshless modeling of deformable shapes and their motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 77–86. Eurographics Association, 2008.

[24] B. Solenthaler, J. Schläfli, and R. Pajarola. A unified particle model for fluid-solid interactions. *Computer Animation and Virtual Worlds*, 18(1):69–82, 2007.

[25] Markus Becker, Markus Ihmsen, and Matthias Teschner. Corotated sph for deformable solids. In *Proc. Eurographics Workshop on Natural Phenomena*, Munich, Germany, 2009.

[26] Michael Hauth and Wolfgang Strasser. Corotational simulation of deformable solids. In V. Skala, editor, *Journal of WSCG*, volume 12, Plzen, Czech Republic, February 2004. UNION Agency - Science Press.

[27] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Transactions on Graphics (TOG)*, 24(3):478, 2005.

[28] A.R. Rivers and D.L. James. FastLSM: fast lattice shape matching for robust real-time deformation. In *ACM SIGGRAPH 2007 papers*, page 82. ACM, 2007.

[29] Denis Steinemann, Miguel A. Otaduy, and Markus Gross. Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 87–94, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[30] Adam W. Bargteil Dan Gerszewski, Haimasree Bhattacharya. A point-based method for animating elastoplastic solids. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aug 2009.

[31] Jihun Yu and Greg Turk. Reconstructing surfaces of particle-based fluids using anisotropic kernels. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 217–225, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[32] B. Adams and M. Wicke. Meshless approximation methods and applications in physics based modeling and animation. In *Eurographics*, pages 213–239, 2009.

[33] J.M. Owen, J.V. Villumsen, P.R. Shapiro, and H. Martel. Adaptive Smoothed Particle Hydrodynamics: Methodology. II. *The Astrophysical Journal Supplement Series*, 116(2):155–209, 1998.

[34] Bart Adams, Mark Pauly, Richard Keiser, and Leonidas J. Guibas. Adaptively sampled particle fluids. *ACM Trans. Graph*, 26(3):48, 2007.

[35] J.J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703, 2005.

[36] G.R. Liu and MB Liu. *Smoothed particle hydrodynamics: a meshfree particle method*. World Scientific Pub Co Inc, 2003.

[37] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, and M. Gross. A unified lagrangian approach to solid-fluid animation. In *Proceedings of the Eurographics Symposium on Point-Based Graphics*, pages 125–134. Citeseer, 2005.

[38] Mark Pauly, Richard Keiser, Bart Adams, Philip Dutré, Markus Gross, and Leonidas J. Guibas. Meshless animation of fracturing solids. *ACM Transactions on Graphics*, 24(3):957–964, July 2005.

[39] M. Gross and H. Pfister. *Point-based graphics*. Morgan Kaufmann Pub, 2007.